SECTION NAME	ID	RULE TEXT	POS	NEG COMMENTARY
AADL Specifications	p41n1	(N1) An AADL specification has one global namespace. The package and property set identifiers	0	0 Checked when processing package and property set nodesBUT: predeclared packages and properties are not checked
AADL Specifications	p41n2	(N2) These package and property set identifiers qualify the names of individual elements contained	0	0
AADL Specifications	p41n3	(N3) Package declarations represent labeled namespaces for component type, component impler	0	0
AADL Specifications	p41n4	(N4) Property set declarations represent labeled namespaces for property type and property defin	0	0
AADL Specifications	p41n5	(N5) Packages and property sets may be separately stored. Those packages and property sets ar	0	0 Provided by parser
AADL Specifications	p41n6	(N6) Defining identifiers in AADL must not be one of the reserved words of the language (see Sec	0	0 Provided by parser
AADL Specifications	p41n7	(N7) The AADL identifiers and reserved words can be in upper or lower case (or a mixture of the t	0	0 Provided by realization of AADLIdentifer class
AADL Specifications	p41n8	(N8) The AADL does not require that an identifier be declared before it is referenced.	0	0
Packages	p42n1	(N1) A defining package name consists of a sequence of one or more package identifiers separate	0	0 Checked by counting private and public package declarations
Packages	p42n2	(N2) The public and private section of a package may be declared in separate package declaratio	0	0 Provided by parser
Packages	p42n3	(N3) Associated with every package is a package namespace that contains the names for all the	0	0
Packages	p42n4	(N4) The package namespace is divided into a public section and a private section. Items declare		0
Packages	p42n5	(N5) The reference to an item declared in another package must be an item name qualified with a	0	0 Can be checked after all possible references are known
Packages	p42n6	(N6) The reference to a property other than predeclared properties must be an property name gua	0	0 Can be checked after all possible references are known
Packages	p42n7	(N7) The package name in a import declaration must exist in the global name space.	0	
Packages	p42n8	(N8) The property set identifier in a import declaration must exist in the global name space.	0	
Packages	p42n9	(N9) Items declared in the private section of the package can only be referenced from within the p	0	Can be checked after all possible references are known
Packages	p42n10	(N10) If the qualifying package identifier of a qualified reference is missing, the referenced compor		
Packages	p42n10	(N11) The package name referenced in an alias declaration must exist in the global namespace a		
Packages	p421111	(N12) The classifier referenced by the alias declaration must exist in the name space of the public	0	Checked when processing package aliases Checked when processing classifier aliases
			0	
Packages	p42n13	(N13) The classifier referenced by the alias declaration must refer to a component type or a feature	0	0 Provided by parser
Packages	p42n14	(N14) The defining identifier of an alias_declaration must be unique in the namespace of the packa	0	Checked when processing package and classifier aliases
Packages	p42n15	(N15) The alias_declaration makes the publicly visible identifier of classifiers declared in another p	0	0 Checked when processing classifier aliases
Packages	p42n16	(N16) If the alias_declaration renames all publicly visible identifiers of component types and feature	0	0 Checked when processing all aliases by intersection of two namespaces
Packages	p42n17	(N17) The identifiers introduced by the alias_declaration are only accessible within the package. W	0	0 Can be checked after all poible references are known
Packages	p42n18	(N18) The alias declared for a component type can be used instead of a qualified component type		
Packages	p42l1	(L1) The defining package name following the reserved word end must be identical to the defining	0	of the state of th
Packages	p42l2	(L2) For each package there may be at most one public section declaration and one private section	0	0 Checked with p42n1
Packages	p42l3	(L3) A component implementation may be declared in both the public and private part of a packag	0	0 Checked when processing component implementations
Packages	p42l4	(L4) The component category in an alias declaration must match the category of the referenced co	0	0 Checked when processing classifier aliases
Component Types	p43n1	(N1) The defining identifier for a component type must be unique in the namespace of the package	0	O Checked when creating local namespaces of the packages BUT: currently no checking of possible intersection between public and private
Component Types	p43n2	(N2) Each component type has a local namespace for defining identifiers of prototypes, features, to	0	0 Checked when creating local namespaces of component types
Component Types	p43n3	(N3) The component type identifier of the ancestor in a component type extension, i.e., that appear		Checked when processing component type declarations
Component Types	p43n4	(N4) When a component type extends another component type, a component type namespace in	0	0
Component Types	p43n5	(N5) A component type that extends another component type does not include the identifiers of the	0	0
Component Types	p43n6	(N6) The defining identifier of a feature, flow specification, mode, mode transition, or prototype mu	0	0 Same as p43n2?
Component Types	p43n7	(N7) The refinement identifier of a feature, flow specification, or prototype refinement refers to the	0	0
Component Types	p43n8	(N8) The prototypes referenced by prototype binding declarations must exist in the local namespa	0	0
Component Types	p43n9	(N9) Mode transitions declared in the component type may not refer to event or event data ports of	0	0
Component Types	p43l1	(L1) The defining identifier following the reserved word end must be identical to the defining identifier	0	0 Provided by parser
Component Types	p43l2	(L2) The prototypes, features, flows, modes, and properties subclauses are optional. If a subclaus	0	0 Provided by parser(kinda, error is - "No viable alternative")
Component Types	p43l3	(L3) The category of the component type being extended must match the category of the extending	0	0 Checked when processing component type declarations
Component Types	p43l4	(L4) The classifier being extended in a component type extension may include prototype bindings.	0	0
Component Types	p43l5	(L5) A component type must not contain both a requires_modes_subclause and a modes_subclause	0	0 Provided by parser(kinda, error is - extraneous input 'requires' expecting {ANNEX', 'END', 'PROPERTIES', IDENTIFIER})
Component Types	p43l6	(L6) If the extended component type and an ancestor component type in the extends hierarchy cor	0	0 Checked when processing component type declarations
Component Types	???	The defining identifier for a component type cannot contain '.'	0	
		<u> </u>		
Component Implementations	p44n1	(N1) A component implementation name consists of a component type identifier and a component	0	1 - Provided by parser(kinda, error is - mismatched input 'end' expecting '.') 2 - checked when component implementation is created
Component Implementations	p44n2	(N2) The defining identifier of the component implementation must be unique within the local name		Checked when creating local namespaces of the packagesBUT: currently no checking of possible intersection between public and private
omponent implementations	p44112	(142) The defining definite of the component implementation must be unique within the local name	U	Oncored when dealing local namespaces of the packagespon, currently no electricity of possible illier section between public and private the province of the p

O and the second based on the first of	T 44T 0	
Component Implementations	p44n3	(N3) Every component implementation defines a local namespace for all defining identifiers of pro 0 Checked when creating local namespaces of component implementations (without types or ancestors)
Component Implementations	p44n4	(N4) This local namespace inherits the namespace of the associated component type, i.e., definin 0 Checked by intersection of local namespaces of type and impt. Problem: error marker marks the whole impl and not the intersecting iden
Component Implementations	p44n5	(NS) Refinement identifiers of features must exist in the namespace of the associated component 0 0
Component Implementations	p44n6	(N6) In a component implementation extension, the component type identifier of the component in 0 0
Component Implementations	p44n7	(N7) When a component implementation extends another component implementation, the local ne 0 0
Component Implementations	p44n8	(N8) Within the scope of the component implementation, subcomponent declarations, connections 0 0
Component Implementations	p44n9	(N9) The prototype referenced by the prototype binding declaration must exist in the local namesp 0 0
Component Implementations	p44l1	(L1) The pair of identifiers separated by a dot (вЪњ.вЪќ) following the reserved word end must be 0 0
Component Implementations	p44l2	(L2) The prototypes, subcomponents, connections, calls, flows, modes, and properties subclauses 0 0
Component Implementations	p44l3	(L3) The category of the component implementation must be identical to the category of the comp 0 0 Checked when processing classifier implementations
Component Implementations	p44l4	(L4) If the component implementation extends another component implementation, the category o 0 Checked when processing classifier implementations
Component Implementations	p44l5	(L5) The classifier being extended in a component implementation extension may include prototyp 0 0
Component Implementations	p44l6	(L6) If the component type of the component implementation contains a requires_modes_subclaus 0 Checked when processing classifier implementations, BUT: not watching at ancestors or aliases, so not complete
Component Implementations	p44I7	(L7) If modes are declared in the component type, then modes cannot be declared in component i 0 0 Checked when processing classifier implementations.
Component Implementations	p44l8	(L8) If modes or mode transitions are declared in the component type, then mode transitions can t 0 0
Component Implementations	p44I9	(L9) The category of a subcomponent being refined must match the category of the refining subco 0 0
Component Implementations	p44I10	(L10) For all other refinement declarations the categories must match (see the respective sections). 0 0
Component Implementations	p44l11	(L11) Component implementations and component implementation extensions must not refine prof 0 0
Subcomponents	p45n1	(N1) The defining identifier of a subcomponent declaration placed in a component implementation 0 0
Subcomponents	p45n2	(N2) The defining identifier of a subcomponent refinement must exist as a defining subcomponent 0 0
Subcomponents	p45n3	(N3) The component type identifier or the component implementation name of a component classi 0 0
Subcomponents	p45n4	(N4) The prototype identifier of a prototype reference must exist in the local name space of the coi 0 0
Subcomponents	p45n5	(N5) The prototype referenced by the prototype binding declarations must exist in the local names 0 0
Subcomponents	p45n6	(N6) The modes named in the in modes statement of a subcomponent must refer to modes in the 0 0
Subcomponents	p45l1	(L1) The category of the subcomponent declaration must match the category of its corresponding 0 0
Subcomponents	p45l2	(L2) The component classifier reference of a subcomponent declaration may include prototype bir 0 0
Subcomponents	p45l3	(L3) In a subcomponent refinement declaration the component category may be refined from abst 0 0
Subcomponents	p45l4	(L4) The Classifier Substitution Rule property specifies the rule to be applied when a refinement 0 0
Subcomponents	p45l5	(L5) In the case of a signature match, the component type of the subcomponent being refined mu: 0 0
Subcomponents	p45l6	(L6) The component category and optional component classifier or prototype reference can be foll 0 0
Subcomponents	p45l7	(L7) The array size specification for the dimensions is optional. In this case the array declaration is 0 0
Subcomponents	p45l8	(L8) When refining a subcomponent array the number of dimensions of the array cannot be chang 0 0
Subcomponents	p45l9	(L9) When the subcomponent is declared as an array with array dimension sizes then a list of corr 0 0
Subcomponents	p45l10	(L10) Selecting index ranges in one or more dimensions of an array is only possible if the size of the 0 0
Subcomponents	p45l11	(L11) An array element implementation list is valid only if (a) the subcomponent classifier is a comr 0 0
Subcomponents	p45c1	(C1) The classifier of a subcomponent cannot recursively contain subcomponents with the same or 0 0
Abstract Components	p46l1	(L1) An abstract component type declaration can contain feature declarations (including abstract fi 0 0
Abstract Components	p46l2	(L2) An abstract component implementation can contain subcomponent declarations of any categraph 0 0
Abstract Components	p46l3	(L3) An abstract component implementation can contain a modes subclause, a connections subclause a connection subclause.
Abstract Components	p46l4	(L4) An abstract subcomponent can be contained in the implementation of any component catego 0 0
Abstract Components	p46l5	(L5) If an abstract subcomponent is refined to a concrete category, the concrete category must be 0 0
Abstract Components	p46l6	(L6) An abstract subcomponent can be declared as an array of subcomponents. 0 0
Abstract Components	p46I7	(L7) If an abstract component type is refined to a concrete category, the features, modes, and flow 0 0
Abstract Components	p46l8	(L8) If an abstract component implementation is refined to a concrete category, the subcomponen 0 0
Aboutact Components	P-4010	(Let) if the deviation compensation in a commence category, and absolution in a
Prototypes	p47n1	(N1) The prototype identifier on the left-hand side of a prototype binding must exist in the local nar 0 0
	p47111	(N2) The prototype identifier on the right-hand side of a prototype binding, if present, must exist in 0 0
Prototypes Prototypes	p47n2 p47n3	(N3) Unique component classifier references must exist in the public section of the package being 0 0
Prototypes Prototypes	p47113	(N4) Unique feature group type references must exist in the public section of the package being id 0 0
Prototypes		
Prototypes	p47l1	(L1) The component category declared in the component prototype binding must match the compt 0 0

Prototypes	p47l2	(L2) The component category of the optional component classifier reference in the prototype declar	0	c	
Prototypes	p4712	(L3) If the component category of the optional component category, then any component type at	0		0
Prototypes	p4713	(L4) If the component prototype declaration includes a component classifier reference, then the cla	0		0
	p4714 p4715	(L5) The category of the component implementation that contains the prototype declaration places	0		0
Prototypes Prototypes	p4715		0		0
Prototypes Prototypes	p4716 p4717	(L6) If the direction is declared for feature prototypes, then the prototype actual satisfies the direction.(L7) In the case of feature group prototypes, the supplied feature group types must match the declared from the declared	0		0
Prototypes	p4717	(L8) A classifier supplied in a feature prototype binding must match the classifier of the prototype	0		0
Prototypes	p4718	(L9) Component prototypes declared with square brackets specify that they expect a list of compo	0		0
	p47I9	(L10) The component category of the classifier reference or prototype reference in a prototype bind	0		0
Prototypes Prototypes	p47110		0		0
Prototypes Prototypes	p47111	(L11) If a direction is specified for an abstract feature in a prototype declaration, then the direction	0		0
Prototypes		(L12) Component prototype bindings must only bind component prototypes, feature group prototype	0		
Prototypes	p47l13	(L13) Component prototype refinements must only refine component prototypes, feature group pro	U		0
Appey Cubolouses and Appey	Libraria n 40 n 4	(M4) The appear identifies must be the name of an approved appear or a partial asset identifies of	0	_	
Annex Subclauses and Annex I		(N1) The annex identifier must be the name of an approved annex or a project-specific identifier d	0		0
Annex Subclauses and Annex I		(N2) The mode identifiers in the in_modes statement must refer to modes in the component type (U		0
Annex Subclauses and Annex I		(L1) Annex subclauses can only be declared in component types, component implementations, ar	0		0
Annex Subclauses and Annex I		(L2) A component type, component implementation, or feature group type declaration may contain	0		0
Annex Subclauses and Annex I		(L3) Annex libraries must be declared in packages.	0		0
Annex Subclauses and Annex I	Librarie p48l4	(L4) A package declaration may contain at most one annex library declaration for each annex.	0	(0
Data	p51l1	(L1) A data type declaration can contain provides subprogram access declarations as well as program access declarations as a program access declaration access as a program access declaration access as a program access declaration access as a program access as a progra	0		0
Data	p51l2	(L2) A data type declaration must not contain a flow specification or modes subclause.	0		0
Data	p51l3	(L3) A data implementation can contain abstract, data and subprogram subcomponents, access of	0		0
Data	p51l4	(L4) A data implementation must not contain a flow implementation, an end-to-end flow specification	0	C	0
Subprograms and Subprogram	n Calls p52n1	(N1) The defining identifier of a subprogram call sequence declaration must be unique within the l	0		0
Subprograms and Subprogram	n Calls p52n2	(N2) The defining identifier of a subprogram call declaration must be unique within the local name	0	C	0
Subprograms and Subprogram	n Calls p52n3	(N3) If the called subprogram name is a subprogram classifier reference, its component type ident	0	C	0
Subprograms and Subprogram	n Calls p52n4	(N4) The subprogram classifier reference of a subprogram call may be a subprogram type reference	. 0	C	0
Subprograms and Subprogram	n Calls p52n5	(N5) If the called subprogram name is a subprogram subcomponent reference, the subprogram su	0	C	0
Subprograms and Subprogram	m Calls p52n6	(N6) If the called subprogram name is a requires subprogram access reference, the requires subp	0	C	0
Subprograms and Subprogram	n Calls p52l1	(L1) A subprogram type declaration can contain parameter, out event port, out event data port, an	0	C	0
Subprograms and Subprogram	n Calls p52l2	(L2) A subprogram implementation can contain abstract, subprogram, and data subcomponents, a	0	C	0
Subprograms and Subprogram		(L3) Only one subprogram call sequence can apply to a given mode.	0	C	0
Subprograms and Subprogram		(C1) The reference to a provides subprogram access of a processor in a subprogram call (process	0	C	0
Subprograms and Subprogram		(C2) A subprogram call may reference a subprogram classifier. A project may enforce a consistent	0	C	0
					П
Subprogram Groups and Subpr	program p53n1	(N1) The defining identifier of a subprogram group type must be unique within the package names	0	0	0
Subprogram Groups and		(N2) Each subprogram group provides a local namespace. The defining subprogram identifiers of	0		0
Subprogram Groups and		(N3) The local namespace of a subprogram group type extension includes the defining identifiers	0		0
Subprogram Groups and		(N4) The defining subprogram identifiers of subprogram access feature declarations in feature gro	0		0
Subprogram Groups and		(N5) The package name of the unique subprogram group type reference must refer to a package in	0		0
Subprogram Groups and		(L1) A subprogram group type can contain provides and requires subprogram access, and provide	0		0
			0		0
Subprogram Groups and		(L2) A subprogram group implementation can contain abstract, data, subprogram group, and subprogram group type or implementation may contain zero or more subcomponent declars.	0		0
Subprogram Groups and	program posis	(L3) A subprogram group type or implementation may contain zero or more subcomponent declar	U		U
Threads	n 5 414	(I.4) A thread time declaration are partial and feature group requires defining the last of		-	
Threads	p54l1	(L1) A thread type declaration can contain port, feature group, requires data access declarations,	0		0
Threads	p54l2	(L2) A thread component implementation can contain abstract, data, subprogram, and subprogram	0		0
Threads	p54l3	(L3) The Complete out event port, and Error out event data port are predeclared, i.e., are implicitly	0		0
Threads	p54c3	(C3) Either the Compute_Entrypoint, Compute_Entrypoint_Source_Text Compute_Entrypoint_Cal	0	(0
Threads	p54c4	(C4) The Period property must have a value if the Dispatch_Protocol property value is periodic, spi	0	(O

Thread Orange	-5514	(LAV A through group posture and there are contain are idea and continue data.	0	
Thread Groups	p55l1	(L1) A thread group component type can contain provides and requires data access, as well as po	0	0
Thread Groups	p55l2	(L2) A thread group component implementation can contain abstract, data, subprogram, subprogram	0	0
Thread Groups	p55l3	(L3) A thread group implementation can contain a connections subclause, a flows subclause, a month of the subclause of the subclause.	0	0
Thread Groups	p55l4	(L4) A thread group must not contain a subprogram calls subclause.	U	0
Drassass	- FOI4	(LAN A present company of the property of the	0	
Processes	p56l1	(L1) A process component type can contain port, feature group, provides and requires data acces	0	0
Processes	p56l2	(L2) A process component implementation can contain abstract, data, subprogram, subprogram g	0	0
Processes	p56l3	(L3) A process implementation can contain a connections subclause, a flows subclause, a modes	0	0
Processes	p56l4	(L4) A thread group must not contain a subprogram calls subclause.	0	0
Processes	p56c1	(C1) The complete source text associated with a process component must form a complete and le	0	0
Processors	p61l1	(L1) A processor component type can contain port, feature group, provides subprogram access, p	0	0
Processors	p61l2	(L2) A processor component implementation can contain declarations of memory, bus, virtual bus	0	0
Processors	p61l3	(L3) A processor implementation can contain a modes subclause, flows subclause, and a properti-	0	0
Processors	p61l4	(L4) A processor implementation can contain bus access, subprogram access, subprogram group	0	0
Processors	p61I5	(L5) A processor implementation must not contain a subprogram calls subclause.	0	0
Virtual Processors	p62l1	(L1) A virtual processor component type can contain port, feature group, provides subprogram acc	0	0
Virtual Processors	p62l2	(L2) A virtual processor component implementation can contain declarations of virtual bus, virtual	0	0
Virtual Processors	p62l3	(L3) A virtual processor implementation can contain a modes subclause, flows subclause, and a p	0	0
Virtual Processors	p62l4	(L4) A virtual processor implementation must not contain a subprogram calls subclause.	0	0
			0	0
Virtual Processors	p62l5	(C1) In a fully bound contain supprogram access, subprogram group acces	0	
Virtual Processors	p62c1	(C1) In a fully bound system every virtual processor must be directly or indirectly bound to, or directly to the control of th	0	0
Virtual Processors	p62c2	(C2) In a fully deployed system a requires virtual bus binding of a virtual processor specified by the	0	0
Memory	p63I1	(L1) A memory type can contain bus access declarations, feature groups, a modes subclause, and		0
Memory	p63l2	(L2) A memory implementation can contain abstract, memory, and bus subcomponent declaration	0	0
Memory	p63l3	(L3) A memory implementation can contain a modes subclause and property associations.	0	0
Memory	p63l4	(L4) A memory implementation can contain bus access connection declarations. Bus access conn	0	0
Memory	p63I5	(L5) A memory implementation must not contain flows subclause, or subprogram calls subclause.	0	0
Buses	p64I1	(L1) A bus type can have requires bus access declarations, a modes subclause, and property ass	0	0
Buses	p64l2	(L2) A bus type must not contain any flow specifications.	0	0
Buses	p64l3	(L3) A bus implementation can contain virtual bus and abstract subcomponent declarations.	0	0
			0	0
Buses	p64l4 p64l5	(L4) A bus implementation can contain a modes subclause and property associations.	0	
Buses	p6415	(L5) A bus implementation must not contain flows subclause, or subprogram calls subclause.	0	0
Virtual Buses	p65l1	(L1) A virtual bus type can have property associations.	0	
Virtual Buses	p65l2	(L2) A virtual bus type must not contain flow specifications.	0	0
Virtual Buses	p65l3	(L3) A virtual bus implementation can contain virtual bus subcomponent declarations.	0	0
Virtual Buses	p65l4	(L4) A virtual bus implementation can contain a modes subclause and property associations.	0	0
Virtual Buses	p65l5	(L5) A virtual bus implementation must not contain a connections subclause, flows subclause, or s	0	0
Virtual Buses	p65c1	(C1) In a fully deployed system virtual buses must be directly or indirectly bound to processors or t	0	0
Devices	p66l1	(L1) A device type can contain port, feature group, provides subprogram access, provides subprog	0	0
	p66l2		0	0
Devices		(L2) A device component implementation must not contain a subprogram calls subclause.	-	
Devices	p66l3	(L3) A device implementation can contain abstract, data, virtual bus, and bus subcomponents, but	0	0
				_
Systems	p71l1	(L1) A system component type can contain subprogram, subprogram group, data and bus access	0	0
Systems	p71l2	(L2) A system component implementation can contain abstract, data, subprogram, subprogram gr	0	0
Systems	p71l3	(L3) A system implementation can contain a modes subclause, a connections subclause, a flows	0	0
Systems	p71l4	(L4) A thread group must not contain a subprogram calls subclause.	0	0
-				

System						
Systems p71 10, 10, 10 Each informing statuse scent for that appears in a bitan method near concrete manufactor of a count of p1 10, 10 10 10 10 10 10 10	Systems	p71n1	(N1) The defining identifier of a feature must be unique within the namespace of the associated co	0	0	0
System	Systems	p71n2	(N2) Thread features may not be declared using the predeclared ports names Complete or Error.	0	0	0
Systems p.1710 [20]. The centred process is assessed for a feature must derive our element of a feature and feature must be received by the systems p.1712 [21]. A feature reference to decidation of a feature and the original feature must born be designed to a process of the p.1712 [22]. A feature reference to decidation of a feature and the original feature must born be designed to a process of the p.1712 [23]. A feature array must not be designed to the control of the p.1712 [24]. A feature array must not be decided to the reference to decidation of a feature and the original feature must be set of the p.1712 [25]. A feature array must not all any original feature and the p.1712 [25]. A feature array must not all any original feature and the p.1712 [25]. A feature array must not all any original feature and the p.1712 [25]. A contrained propose and a standy original feature and the p.1712 [25]. A contrained propose and a feature feature and the p.1712 [25]. A contrained propose and a feature array must not a feature and the p.1712 [25]. A contrained propose and a feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a feature and the p.1712 [25]. A feature array must not a featu	Systems	p71n3	(N3) Each refining feature identifier that appears in a feature refinement declaration must also app	0	0	0
Systems p711 (2.1) Such restance and servined a most action are not yet to education of a feature strong and the objective of the property	Systems	p71n4	(N4) A feature is referenced in one of two ways. Within the component implementations for a com-	0	0	0
Systems 9712 (2) A status reflement deteration of a feature and the crypt feature must both to decidence of systems 9713 (3) Feature array manufact does not not be feature being reflement and to 9 c	Systems	p71n5	(N5) The path of a contained property association for a feature must refer to an element of a featu	0	0	0
Systems	Systems	p71l1	(L1) Each feature can be refined at most once in the same type extension.	0	0	0
Systems 9714 (14) The betainer informant spools are array dismonstrate. Internal to the state of the	Systems	p71l2	(L2) A feature refinement declaration of a feature and the original feature must both be declared a	0	0	0
Systems p/16 (b). If the ordinance specifies in any characteristic size, the property association must perform a set status group. 9 c	Systems	p71l3	(L3) Feature arrays must only be declared for threads, devices, and processors.	0	0	0
Systems 9/19 (b) A continued propony association must only to used when he selecte is a feeting clear to endow 0 0 0 0 0 0 0 0 0	Systems	p71I4	(L4) If the feature refinement specifies an array dimension, then the feature being refined must ha	0	0	0
Systems p 7117 (17) in the case of a cleanure with a cleanure of reference, the cleanure of the centre of a cleanure with a cleanure of the centre of of the ce	Systems	p71I5	(L5) If the refinement specifies an array dimension size, then the feature being refined must not have	0	0	0
Abstract Features 8111 (1.1) The feature direction is a refined feature declaration must be infertious in the feature direction. 0 0 0 0 0 0 0 0 0	Systems	p71l6	(L6) A contained property association must only be used when the feature is a feature group.	0	0	0
Abstract Features 9,8112 Abstract Features 9,8115 Abstract Features 9,8115 Abstract Features (Corp. p. 1867) Features Groups and Features Group and Features Group and Features Groups and Features	Systems	p71I7	(L7) In the case of a feature with a classifier reference, the classifier of the refined feature declarate	0	0	0
Abstract Features 9,8112 Abstract Features 9,8115 Abstract Features 9,8115 Abstract Features (Corp. p. 1867) Features Groups and Features Group and Features Group and Features Groups and Features						
Abstract Features p\$10 (3.) An abstract feature within a feature problegies interflier and the protetype being referenced must 0 0 0 Feature Groups and Feature Group 1 p\$201 (N) The defining identifier of a feature group type must be unique within the package namespace of the feature feature Group 1 p\$201 (N) The defining identifier of a feature group type must be unique within the package namespace of the feature Group 1 p\$201 (N) The defining identifier of a feature group type must be unique within the package namespace of the feature Group 1 p\$201 (N) The defining identifier of a feature group type enterior of the feature Group 1 p\$201 (N) The defining identifier of a feature group declarations must be unique in the feature Group 1 p\$201 (N) The defining identifier of a feature group declaration must be unique in the feature Group 1 p\$201 (N) The defining identifier of a feature group declaration must be unique in the feature Group 1 p\$201 (N) The defining identifier of a feature group declaration must be unique in the feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) The defining identifier of a feature Group 1 p\$201 (N) T	Abstract Features	p81I1	(L1) The feature direction in a refined feature declaration must be identical to the feature direction	0	0	0
Abstract Features p814 (J.) An abstract feature refinement declaration of a feature group open must be unque within the package manager and a feature group per must be unque within the package manager and package and p	Abstract Features	p81I2	(L2) If the direction of an abstract feature is specified, then the direction must be satisfied by the re	0	0	0
Feature Groups and Feature Group 1; 923-11 (H1) The defining identifier of a feature group type must be unique within the package namespace of the package feature Group 1; 923-21 (Feature Groups and	Abstract Features	p81I3	(L3) An abstract feature with a feature prototype identifier and the prototype being referenced mus	0	0	0
Feature Groups and Feature (Corpo Tip 82074 Feature Groups and Feature Group Tip 82074 Feature Gr	Abstract Features	p81I4	(L4) An abstract feature refinement declaration of a feature with a feature prototype reference mus	0	0	0
Feature Groups and Feature (Corpo Tip 82074 Feature Groups and Feature Group Tip 82074 Feature Gr						
Feature Groups and Feature Group Type2572 Feature Groups and Feature Group Type2577 Feature Groups and Feature Group Type257 Feature Groups and Feature	Feature Groups and Feature Group T	Γ ₎ p82n1	(N1) The defining identifier of a feature group type must be unique within the package namespace	0	0	0
Feature Groups and Feature Group 1920-26 Feature Groups and Feature Group 1920-27 Feature Group 1920-27 Feature Groups and Feature Group 1920-27 Feature Groups	Feature Groups and Feature Group T	Γ ₎ p82n2	(N2) Each feature group type provides a local namespace. The defining identifiers of prototype, fe	0	0	0
Feature Groups and Feature Group Typ8275 Feature Groups and Feature Group Typ8275 Feature Groups and Feature Group Typ8275 Feature Groups and Feature Group Typ8277 Feature Groups and Geature Group Typ8277 Feature Groups and Geature Group Typ8	Feature Groups and Feature Group T	Г ₎ p82n3	(N3) The local namespace of a feature group type extension includes the defining identifiers in the	0	0	0
Feature Groups and Feature Group 1 p82nt Call A feature group type men in feature group type. Enter or more elements, i.e., feature or feature groups, if it is a feature group type. (L) A feature group type men in feature Group 1 p82nt Call A feature group type men in feature group type. as indifferent group type. (L) A feature group type men in the feature group type. as indifferent group type. (L) A feature group type men in the feature group type. as indifferent group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type. (L) A feature group type men in the feature group type with features and in the feature group type. (L) A feature group type men in the feature group type with features and in the feature group type with features a	Feature Groups and Feature Group T	Γ ₎ p82n4	(N4) The defining feature identifiers of feature group declarations must be unique in the local nam	0	0	0
Feature Groups and Feature Group 1 p8207 (1) A feature group type may contain zero or more elements, i.e. feature Groups and Feature Group 1 p8202 (2) A feature group type may contain zero or more elements, i.e. feature Groups and Feature Group 1 p8202 (2) A feature group type can be declared to be the inverse of another feature group type. as indicated to be the inverse of another feature group type. as indicated feature Group 1 p8202 (2) A feature group byte without inverse of or feature group type with features and inverse of a feature Group and Feature Group 1 p8203 (4) (4) A feature group byte without inverse of or feature group byte without inverse of or feature group byte without an inverse of or feature Group and Feature Group 1 p8203 (4) (5) The feature group byte with at is an extension of another feature group byte with features and in Feature Group and Feature Group 1 p8207 (6) (6) A feature group byte with a san extension of another feature group byte with features and in Feature Group and Feature Group and Feature Group 1 p8207 (7) A feature group the feature Group and Feature Group 1 p8207 (8) (2) A feature group and the contained in the feature group and its contained in the feature group and group and group and feature Group 1 p8207 (2) (3) Feature Group and feature G	Feature Groups and Feature Group T	Γ ₎ p82n5	(N5) The defining feature group identifier of feature_refinement declarations in component types n	0	0	0
Feature Groups and Feature Group TyB2172 (12) A feature group type can be desclared to be the inverse of another feature group type. as indices for a feature Group TyB212 (12) A feature group type can be desclared to be the inverse of another feature group type. as indices for a feature Group TyB212 (13) Ohly feature group type can be desclared to be the inverse of another feature group type. as indices for a feature Group TyB212 (13) Ohly feature group type can be desclared to be the inverse of another feature group type. as indices for a feature Group TyB212 (14) A feature group type that is an extension of another feature group type with to an inverse of relative Group TyB212 (15) The feature Group and Feature Group TyB213 (16) The feature group type that is an extension of another feature group type with features and inverse of relative Groups and Feature Group TyB217 Feature Groups and Feature Group TyB217 Feature Groups and Feature Group TyB218 Feature Groups and Feature Group TyB219 Feature Groups and Feature Group TyB219 Feature Groups and Feature Group TyB219 Feature Groups and Feature Group TyB211 (14) Feature Groups and Feature Group TyB211 (15) Feature Groups and Feature Group TyB211 (17) A feature group because the seature group desclaration with an inverse of the feature group and tso considered to complement early the state type the access connection rules specified in Sextic Feature Groups and Feature Group TyB211 (18) Feature Groups and Feature Group TyB211 (19) Feature Groups and Feature Group TyB211 (11) Feature Groups and Feature Group TyB211 (12) Feature Groups and Feature Group TyB211 (13) If an in or out direction is specified as part of a feature group declaration, then all features in the feature group declaration, then all features in the feature group declaration, then all features in the feature Group TyB211 (14) Feature Groups and Feature Group TyB2111 (15) Feature Groups and Feature Group TyB211 (16) Feature Groups and Feature Group TyB211 (17) Feature Grou				0	0	0
Feature Groups and Feature Group Tj p8212 (12) A feature group types can be declared to be the inverse of another feature group types, as indic Circloups and Feature Group Tj p8214 (14) A feature group types with features and inverse of a Feature Group Tj p8214 (15) The feature group type that is an extension of another feature group type with features and inverse of Teature Group Tj p8216 (15) The feature group type that is an extension of another feature group type with features and inverse of Teature Group Tj p8216 (15) The feature group type that is an extension of another feature group type with features and inverse of Circloups and Feature Group Tj p8216 (15) The feature group type that is an extension of another feature group type with features and inverse of Circloups and Feature Group Tj p8217 (17) A feature group type there of teature group and its complement of Circloups and Feature Group Tj p8218 (18) The number of feature group contained in the feature group and its case indo Feature Groups and Feature Group Tj p8211 (18) Each of the declared features or leature group so and in the feature group types have zero feature group and its complement and feature Group Tj p8211 (19) Both feature group types have zero feature group and its complement and feature Group Tj p8211 (19) Both feature group types have zero feature group must be a pair-wise complement and feature Group Tj p8211 (19) Both feature group types have zero feature group must be a pair-wise complement and feature Group Tj p8211 (19) Both feature group types have zero feature group declaration, then all features is complement and feature Group Tj p8211 (12) Access features and in the feature group declaration, then all features is complement and feature group types have zero feature group declaration, then all features is complement and feature group types have zero feature group declaration, then all features is complement and the seature group types have zero feature group types have zero feature group types have zero feature group types	Feature Groups and Feature Group T	Γ\ p82n7		0	0	0
Feature Groups and Feature Group TyB222 (L3) A feature group byce can be declared to be the inverse of another feature group byce, as indiverse of Peature Groups and Feature Group TyB221 (L3) Only feature group byce without inverse of or feature group byce with features and inverse of Croups and Feature Group TyB221 (L5) The feature group byce that is an extension of another feature group byce with features and in Peature Group TyB221 Feature Groups and Feature Group TyB222 Feature Groups and Feature Group TyB226 Feature Groups and Feature Group TyB227 Feat	Feature Groups and Feature Group T	Γ\ p82l1	(L1) A feature group type may contain zero or more elements, i.e., feature or feature groups. If it c	0	0	0
Feature Groups and Feature Group Ty R8244 (L5) A feature group type that is an extension of another feature group type without an inverse of concerning feature Group Ty R8256 (L5) A feature group type that is an extension of another feature group type with features and in Feature Group Ty R8257 (L6) A feature group type that is an extension of another feature group by the with features and in Feature Group Ty R8257 (L6) A feature group type that is an extension of another feature group by the with features and in Feature Group Ty R8257 (L6) A feature group type that is an extension of another feature group by the with feature size of the Getariaton with an inverse of statement must only reference feature group type feature of Getariate Group Ty R8257 (L6) A feature group refinement may be refined to only add properly associations. In this case ind Feature Group Ty R8257 (L7) A feature group group contained in the feature group to social tension of the feature of composent of the declared features or feature group type share zero features. The they are considered to complement ear? (L7) Feature Group Ty R8271 (L1) Each of the declared features or feature group type share zero features, then they are considered to complement ear? (L1) Feature Group Ty R8271 (L1) Post are pair-wise complementary if they satisfy the access connection rules in Seature Group Ty R8271 (L1) A defining port identifier must adher to the naming rules specified of spart of a feature group declaration, then all features in Seature Group Type Type Type Type Type Type Type Typ	Feature Groups and Feature Group T	Γ _\ p82l2		0	0	0
Feature Groups and Feature Group Tj p8216 (L5) A Feature group type that is an extension of another feature group type without an inverse of a Copy and Feature Group Tj p8277 (L5) A feature group bye that is an extension of another feature group type with feature and on Copy and Feature Group Tj p8277 (L6) A feature group declaration with an inverse of statement must only reference feature group type (L7) A feature group declaration with an inverse of statement must only reference feature group type (L6) A feature group declaration with an inverse of statement must only reference feature group type (L7) A feature group declaration with an inverse of statement must only reference feature group type (L7) A feature group refinement may be refined to only add properly associations. In this case incl Feature Groups and Feature Group Tj p82810 (L9) Each of the declared feature or feature group must be a pair-wise complement each (L1) If both feature group types have zero features, then they satisfy the port connection rules specified in Sectic (L1) Ports are pair-wise complementary if they satisfy the access connection rules in Section Group and Feature Group Tj p82112 (L1) Access features are pair-wise complementary if they satisfy the access connection rules in Section Group and Feature Group Tj p82112 (L1) A feature group declaration, then all features in Section Group and Feature Group Tj p82112 (L1) A feature group types have zero features, then they satisfy the access connection rules in Section Group and Feature Group Tj p82112 (L1) A feature group types have zero features group must be a pair-wise complementary if they satisfy the access connection rules in Section Group Tj p82112 (L1) A feature group Tj p82112 (L1) A feature group Tj p82114 (L2) Access features are pair-wise complementary if they satisfy the access connection rules in Section Group Tj p82112 (L2) A feature group Tj p82112 (L2) Access features Group Tj p82112 (L2) Access features Group Tj p82112 (L2) Access features Grou	· · · · · · · · · · · · · · · · · · ·	1.		0	0	0
Feature Groups and Feature Group Typ 8213 (L6) The feature group type that is an extension of another feature group type with features and in Ceature Group Typ 8217 Feature Groups and Feature Group Typ 8218 (L7) A feature group type feature group type declaration with an inverse of statement must only reference feature group. Feature Group Typ 8218 (L7) A feature group refinement may be refined to only add property associations. In this case incl (L7) A feature Group Typ 8218 (L8) The number of feature group scotlained in the feature group and its complement or Ceature Group Typ 8218 (L9) Each of the declared features or feature group pust be a pair-wise compliance of the declared features or feature group pust be a pair-wise compliance or Ceature Group Typ 8218 (L10) If both feature group types have zero features, then they are considered to complement each of Ceature Group Typ 82113 (L11) Fortia ser pair-wise complementary if they satisfy the considered to complement each of Ceature Group Typ 82113 (L13) If an in or out direction is specified as part of a feature group declaration, then all features is complementary if they satisfy the access connection rules is complement to Ceature Group Typ 82112 (L13) If an in or out direction is specified as part of a feature group declaration, then all features (see Section Ceature Group Typ 82113 (L13) If an in or out direction is specified as part of a feature group declaration, then all features (see Section Ceature Group Typ 82113 (L13) If an in or out direction is specified as part of a feature group declaration must also appear in a feature declaration Ceature Group Typ 82113 (L13) If an in or out direction of a port refinement declaration must also appear in a feature declaration Ceature Group Typ 82113 (L13) If an in or out direction of a port refinement must be the name of a condition of the feature declaration Ceature Group Typ 82113 (L13) If an in or out direction of a port refinement must be the name of a condition Ceature Group Typ 82113 (L				0	0	0
Feature Groups and Feature Group T) p826 (L.6). A feature group declaration with an inverse of statement must only reference feature group by 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		1.		0	0	0
Feature Groups and Feature Group Ty B2727 (L7) A feature group refinement may be refined to only add property associations. In this case ind 0 0 Feature Groups and Feature Group Ty B2728 (L8) The number of feature groups contained in the feature group and its complement in 0 Feature Groups and Feature Group Ty B2719 (L9) Each of the declared features or feature group must be a pair-wise complement each of Feature Group and Feature Group Ty B2710 (L10) if both feature group types have zero features; then they are considered to complement each of Feature Groups and Feature Group Ty B2711 (L10) Ports are pair-wise complementary if they satisfy the port connection rules specified in section of Feature Group Ty B2712 (L13) if an in or out direction is specified as part of a feature group declaration, then all features in complements of the pass of the feature group declaration, then all features (see Section pass and Feature Group Ty B2712 (L13) if an in or out direction is specified as part of a feature group declaration, then all features (see Section pass and Feature Group Ty B2712 (L13) if an in or out direction is specified as part of a feature group declaration, then all features (see Section pass and Feature Group Ty B2712 (L13) if an in or out direction is specified as part of a feature group declaration, then all features (see Section pass and Feature Group Ty B2712 (L13) if an in or out direction is specified for all features (see Section pass and pa	Feature Groups and Feature Group T	Γ ₁ p82l6	(L6) A feature group declaration with an inverse of statement must only reference feature group ty	0	0	0
Feature Groups and Feature Group Ty p8218 (L8) The number of feature groups contained in the feature group and its complement and presture Group Ty p8219 (L9) Each of the declarated returues or feature groups in a feature group must be a pair-wise complement each of the declarated feature group Typ p8210 (L10) if both feature group Typ sea have zero features, then they are considered to complement each of the declaration from the satisfy the port connection rules specified in Sectic Groups and Feature Group Typ p82112 (L11) Ports are pair-wise complementary if they satisfy the port connection rules specified in Sectic Groups and Feature Group Typ p82112 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section Groups and Feature Group Typ p82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features in Section Groups and Feature Group Typ p82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features in Section Groups and Feature Group Typ p82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features in Section Groups and Feature Group Typ p82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features (see Section Groups and Feature Group Typ p82113 (L13) if an in or out direction is specified for all features (see Section Groups and Feature Group Typ p82113 (L2) The unique component type identifier of a port defining intensity of a port refinement declaration must also appear in a feature declaration of a port of the data classifier reference must be the name of a condition of a port of the data classifier reference must be the name of a condition of a port of the data classifier reference must be the name of a condition of a port of the data classifier reference must be the name of a condition of a port of the data component of the data component of the data component of the d	Feature Groups and Feature Group T	Γ _\ p82l7		0	0	0
Feature Groups and Feature Group Tip 8219 (L19) Each of the declared features or feature group in a feature group must be a pair-wise complement each Group Tip 82110 (L11) broth feature group types have zero features, then they are considered to complement each Group Tip 82111 (L11) Prots are pair-wise complementary if they satisfy the port cases connection rules in Signature Group Tip 82112 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Signature Group Tip 82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features in Signature Group Tip 82113 (L13) if an in or out direction is specified as part of a feature group declaration, then all features (see Sective Ports Ports Pa31 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Sective Ports Pa33 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of Ports Pa33 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of Ports Pa33 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of Ports Pa31 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtu Ports Ports Pa31 (L2) Data and event data ports may be incompletely defined by not specifying the data component Ports Pa31 (L2) Data, event, and event data ports may be refined by adding a property association. The data of Ports Pa31 (L4) The port direction of a port refinement must be the same as the category of the port being refused to Ports Pa31 (L4) The port direction of a port refinement must be the same as the category of the port being refused to Ports Pa31 (L4) The port direction of a port refinement must be the same as the category of the port being refused to Ports Pa32 (L4) The port direction of a port refinement must be the same as the direction of the feature b		1		0	0	0
Feature Groups and Feature Group Typ82l12 (L11) Ports are pair-wise complementary if they satisfy the access connection rules specified in Sectic Groups and Feature Group Typ82l12 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in S 0 0 Feature Groups and Feature Group Typ82l12 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Groups and Feature Group Typ82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Group Typ82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Group Typ83l14 (N1) A defining port identifier of a port refinement declaration must also appear in a feature declaration S 0 0 Forts p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a 0 0 Forts p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of 0 0 Forts p83n4 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual Decrease p83n4 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 0 Forts p83n2 (L2) Data and event data ports may be refined by adding a property association. The data 0 0 Forts p83n4 (L4) The port category of a port refinement must be the same as the category of the port being ref 0 0 Forts p83n4 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 0 CSubprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group access dec 0 0 CSubprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 CSubprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 CSubpro		1.		0	0	0
Feature Groups and Feature Group Typ82l12 (L11) Ports are pair-wise complementary if they satisfy the access connection rules in S 0 0 Feature Group and Feature Group Typ82l13 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in S 0 0 Feature Group Typ82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Group Typ82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Group Typ82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features in S 0 0 Feature Group Typ82l13 (N1) A defining port identifier of a port refinement declaration must also appear in a feature declaration Declaration Typ82l13 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration Declaration Typ82l13 (N3) The unique component type identifier of the data classifier reference must be the name of a 0 0 Feature Group Typ82l13 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of 0 0 Feature Group Typ82l13 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual Declaration Typ82l13 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 0 Feature Group Typ83l2 (L2) Data and event data ports may be refined by adding a property association. The data 0 0 Feature Group Typ83l3 (L3) Data, event, and event data ports may be refined by adding a property association. The data 0 0 Feature Group Typ83l3 (L5) The port direction of a port refinement must be the same as the direction of the feature being of 0 0 Feature Group Typ83l3 (N2) The defining identifier of a provides or requires subprogram group access dec 0 0 0 CSubprogram Group Typ84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 0 0 0 0	· · · · · · · · · · · · · · · · · · ·	1		0	0	0
Feature Groups and Feature Group Ty Re2112 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in S 0 0 0 Feature Group Ty Re2113 (L13) If an in or out direction is specified as part of a feature group declaration, then all features ins 0 0 0 Ports Re311 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Sectic 0 0 0 Ports Re312 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of the data classifier reference must be the name of a c 0 0 Ports Re314 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of 0 0 Ports Re314 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtu 0 Ports Re312 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 Ports Re313 (L3) Data, event, and event data ports may be refined by adding a property association. The data 0 0 0 Ports Re314 (L4) The port category of a port refinement must be the same as the category of the port being ref 0 0 Ports Re315 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 0 0 Subprogram Group / p8411 (N1) The defining identifier of a provides or requires subprogram or subprogram group access dec 0 0 0 Subprogram Group / p8402 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	·	1.			0	0
Feature Groups and Feature Group Ty p82113 (L13) If an in or out direction is specified as part of a feature group declaration, then all features ins 0 0 Ports p83n1 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Sectix 0 0 Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a construction of a port refinement pye identifier of the data classifier reference must be the name of a construction of a port refinement pye identifier of the data classifier reference must be the name of a construction of a port refinement pye identifier of a prototype reference, if specified, must exist in the namespace of construction of a port per port of ports pass construction of a port refinement must be the name of a construction of a port refinement must be the name of a construction of a port refinement must be the same as the category of the port being reformant pass of construction of a port refinement must be the same as the category of the port being reformant pass of construction of a port refinement must be the same as the category of the port being reformant pass of construction of a port refinement must be the same as the category of the port being reformant pass of construction of the feature being construction of construction of a provides or requires subprogram or subprogram group access deconstruction construction of a provides or requires subprogram or subprogram group refinement construction construction of construction of a provides or requires subprogram or subprogram group refinement construction construct		1		0	0	0
Ports p83n1 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Sectic 0 0 0 Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration 0 0 Ports p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of 0 0 Ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of 0 0 Ports p83n3 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtu 0 0 Ports p83n2 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 Ports p83n3 (L3) Data, event, and event data ports may be refined by adding a property association. The data p83n4 (L4) The port category of a port refinement must be the same as the category of the port being ref 0 Ports p83n3 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 Ports p83n5 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 Ports p83n5 (N2) The defining identifier of a provides or requires subprogram or subprogram group access dec 0 0 Subprogram Group / p84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 Ports Pa4n1 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 0 Ports Pa4n1 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 0 Ports Pa4n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 Ports Pa4n2 (N2) The defining identifier of a provides or requires subprogram group refinement 0 Ports Pa4n2 (N2) The defining identifier of a provides or requires subprogram group refinement 0 Ports Pa4n2 (N2) The defining identifier of a provides or requires subprogram group refinement 0 Ports Pa4n2 (N2) The				0	0	0
Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a c p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The port data ports may be incompletely defined by not specifying the data componen ports protocomposition. The data ports may be refined by adding a property association. The data ports por			(· , · · · · · · · · · · · · · · · · ·			
Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a c p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of ports p83n4 (N4) The port data ports may be incompletely defined by not specifying the data componen ports protocomposition. The data ports may be refined by adding a property association. The data ports por	Ports	p83n1	(N1) A defining port identifier must adhere to the naming rules specified for all features (see Section	0	0	0
Ports p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a c 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports	-			0	0
Ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports					0
Ports p831 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtu 0 0 0 Ports p832 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0						
Ports p8312 (L2) Data and event data ports may be incompletely defined by not specifying the data componen 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports				0	
Ports p8313 (L3) Data, event, and event data ports may be refined by adding a property association. The data 0 0 0 Ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being ref 0 0 0 Ports p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports				•	
Ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being ref 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports					
Ports p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being 0 Subprogram and Subprogram Group / p84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group access dec 0 Subprogram and Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement 0 O O O O O O O O O O O O O						
Subprogram and Subprogram Group / p84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group access dec Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram Group / p84n2 (N2) The defining identifier of a provides or requir	Ports					
Subprogram and Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement 0 0		p0010	(2-) p should be a portional mark be the during the direction of the leature being	-		-
Subprogram and Subprogram Group / p84n2 (N2) The defining identifier of a provides or requires subprogram group refinement 0 0	Subprogram and Subprogram Group	/ n84n1	(N1) The defining identifier of a provides or requires subprogram or subprogram group access der	0		0
Casep. Gram. and Casep. Gram. Group. The Component type definition of Component imperioritation in Component in Comp		-				
	Capping and Gapping and Group	POTITO	(170) The component type identifier of component implementation name of a subprogram of subpr	U		U

Subprogram and Subprogram 0	Group / p84n4	(N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if	0	0	0
Subprogram and Subprogram 0	Group / p84l1	(L1) If a subprogram access refers to a component classifier or a component prototype, then the c	0	0	0
Subprogram and Subprogram G	Group / p84l2	(L2) If a subprogram group access refers to a component classifier or a component prototype, the	0	0	0
Subprogram and Subprogram 0	Group / p84l3	(L3) An abstract feature can be refined into a subprogram access or a subprogram group access.	0	0	0
Subprogram and Subprogram 0	Group / p84l4	(L4) A subprogram or subprogram group access declaration that does not specify a component cla	0	0	0
Subprogram and Subprogram 0	Group / p84l5	(L5) A subprogram or subprogram group access declaration may be refined by adding a property	0	0	0
Subprogram and Subprogram 0	Group / p84l6	(L6) A provides subprogram access cannot be refined to a requires subprogram access and a req	0	0	0
Subprogram and Subprogram 0	Group / p84c1	(C1) A provides subprogram access feature indicates that a subprogram is made available to be re	0	0	0
Subprogram Parameters	p85n1	(N1) The defining identifier of a parameter must be unique within the namespace of the subprogra	0	0	0
Subprogram Parameters	p85n2	(N2) The defining parameter identifier of a parameter refinement declaration must also appear in $\boldsymbol{\epsilon}$	0	0	0
Subprogram Parameters	p85n3	(N3) The data classifier reference must refer to a data component type or a data component imple	0	0	0
Subprogram Parameters	p85n4	(N4) The prototype identifier, if present, must exist in the namespace of the subprogram classifier	0	0	0
Subprogram Parameters	p85l1	(L1) Parameters can be declared for subprogram component types.	0	0	0
Subprogram Parameters	p85l2	(L2) A parameter declaration that does not specify a data classifier reference is incomplete. Such	0	0	0
Subprogram Parameters	p85l3	(L3) A parameter declaration may be refined by adding a property association. Inclusion of the dat	0	0	0
Subprogram Parameters	p85l4	(L4) The parameter direction of a parameter refinement must be the same as the direction of the f	0	0	0
Data Component Access	p86n1	(N1) The defining identifier of a provides or requires data access declaration must be unique within	0	0	0
Data Component Access	p86n2	(N2) The defining identifier of a provides or requires data access refinement must exist as a defini	0	0	0
Data Component Access	p86n3	(N3) The component type identifier or component implementation name of a data access classifier	0	0	0
Data Component Access	p86n4	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains	0	0	0
Data Component Access	p86l1	(L1) If a data access refers to a component classifier or a component prototype, then the category	0	0	0
Data Component Access	p86l2	(L2) A data access declaration may be refined by refining the data classifier, by adding a property	0	0	0
Data Component Access	p86l3	(L3) A provides data access cannot be refined to a requires data access and a requires data acce	0	0	0
Data Component Access	p86l4	(L4) An abstract feature can be refined into a data access. In this case, the abstract feature must	0	0	0
Data Component Access	p86c1	(C1) A data access declaration that does not specify a data classifier reference is incomplete. Such	0	0	0
Data Component Access	p86c2	(C2) If the source code of a component does access shared data, then the component type declara-	0	0	0
Data Component Access	p86c3	(C3) A data access refinement may refine an abstract feature declaration. If the abstract feature de	0	0	0
Bus Component Access	p87n1	(N1) The defining identifier of a provides or requires bus access declaration must be unique within	0	0	0
Bus Component Access	p87n2	(N2) The defining identifier of a provides or requires bus refinement must exist as a defining identifier	0	0	0
Bus Component Access	p87n3	(N3) The component type identifier or component implementation name of a bus access classifier	0	0	0
Bus Component Access	p87n4	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains	0	0	0
Bus Component Access	p87I1	(L1) If a bus access refers to a component classifier or a component prototype, then the category	0	0	0
Bus Component Access	p87l2	(L2) A bus access declaration may be refined by refining the bus classifier, by adding a property a	0	0	0
Bus Component Access	p87l3	(L3) A provides bus access cannot be refined to a requires bus access and a requires bus access	0	0	0
Bus Component Access	p87l4	(L4) An abstract feature can be refined into a bus access. In this case, the abstract feature must n	0	0	0
Bus Component Access	p87c1	(C1) A bus access declaration that does not specify a bus classifier reference is incomplete. Such	0	0	0
Bus Component Access	p87c2	(C2) If a bus access feature is a refinement of an abstract feature, then the direction of the abstrac	0	0	0
Bus Component Access	p87n1	(N1) The defining identifier of a defined connection declaration must be unique in the local names	0	0	0
Bus Component Access	p87n2	(N2) The connection identifier in a connection refinement declaration must refer to a named conne	0	0	0
Bus Component Access	p87l1	(L1) A connection refinement must contain at least one of the following: a connection source and (0	0	0
Bus Component Access	p87l2	(L2) If a semantic connection may be active in a particular mode, then the ultimate source and ulti-	0	0	0
Bus Component Access	p87l3	(L3) If a semantic connection may be active in a particular mode transition, then the ultimate source	0	0	0
Feature Connections	p91n1	(N1) A source or destination reference in a feature connection or feature connection refinement de	0	0	0
Feature Connections	p91n2	(N2) The subcomponent reference may refer to a subcomponent or a subcomponent array.	0	0	0
Feature Connections	p91l1	(L1) If the feature connection declaration represents a connection between features of sibling com	0	0	0
Feature Connections	p91l2	(L2) If the feature connection declaration represents a connection between features up the contain	0	0	0
Feature Connections	p91l3	(L3) If the feature connection declaration represents a connection between features down the con	0	0	0
Feature Connections	p91l4	(L4) If the feature connection declaration specifies a directional connection, then the direction of the	0	0	0

Feature Connections	p91I5	(L5) The individual connections of a semantic connection must be bidirectional or have the same (0	0	0
Port Connections	p92n1	(N1) The connection identifier in a port connection refinement declaration must refer to a named p	0	0	0
Port Connections	p92n2	(N2) A source or destination reference in a port connection or port connection refinement declarat	0	0	0
Port Connections	p92n3	(N3) The subcomponent reference may also consist of a reference to a subcomponent array.	0	0	0
Port Connections	p92n4	(N4) The event_or_event_data identifier of event source specifications (self.event_or_event_data	0	0	0
Port Connections	p92l1	(L1) In the case of a directional port connection the connection end representing the source of the	0	0	0
Port Connections	p92l2	(L2) In the case of a bidirectional port connection either connection end can be the source. If the b	0	0	0
Port Connections	p92l3	(L3) If the source connection end is a data access feature it must have read access rights; if the d	0	0	0
Port Connections	p92l4	(L4) The feature identifier of a subcomponent reference may refer to a feature array, if the subcomponent reference may refer to a feature array.	0	0	0
Port Connections	p92l5	(L5) The following are acceptable sources and destinations of port connections. The left column sl	0	0	0
Port Connections	p92l6	(L6) If the port connection declaration represents a connection between ports of sibling componen	0	0	0
Port Connections	p92l7	(L7) If the port connection declaration represents a connection between ports up the containment	0	0	0
Port Connections	p92l8	(L8) If the port connection declaration represents a connection between ports down the containme	0	0	0
Port Connections	p92l9	(L9) The individual connections of a semantic port connection must be bidirectional or have the sa	0	0	0
Port Connections	p92l10	(L10) Self. <identifier> must only be referenced as the source of a connection.</identifier>	0	0	0
Port Connections	p92l11	(L11) A data port cannot be the destination of more than one semantic port connection unless each	0	0	0
Port Connections	p92l12	(L12) A semantic connection cannot contain connection declarations with both immediate and dela	0	0	0
Port Connections	p92l13	(L13) For connections between data ports, event data ports and data access, the data classifier of	0	0	0
Port Connections	p92l14	(L14) The following rules are supported: въў въў въў Сlassifier_Match: The source data tyr	0	0	0
Port Connections	p92l15	(L15) If more than one port connection declaration in a semantic port connection has a property as	0	0	0
Port Connections	p92l16	(L16) A processor port specification must only be used in event connections within threads and sul	0	0	0
Port Connections	p92c1	(C1) There cannot be cycles of immediate connections between threads, devices, and processors.	0	0	0
Port Connections	p92c2	(C2) The processor port identifier of a processor port specification (processor processor port_iden	0	0	0
Port Connections	p92c3	(C3) The Supports_Classifier_Subset_Matches property may be associated with a bus or virtual but	0	0	0
Port Connections	p92c4	(C4) The Supports_Type_Conversions property may be associated with a bus or virtual bus. This s	0	0	0
Parameter Connections	p93n1	(N1) The connection identifier in a parameter connection refinement declaration must refer to a na	0	0	0
Parameter Connections	p93n2	(N2) A source (destination) reference in a parameter connection declaration must reference a parameter	0	0	0
Parameter Connections	p93I1	(L1) The source of a parameter connection must be an incoming data or event data port of the cor	0	0	0
Parameter Connections	p93I2	(L2) The following source/destination pairs are acceptable for parameter connection declarations:	0	0	0
Parameter Connections	p93I3	(L3) A parameter cannot be the destination feature reference of more than one parameter connec	0	0	0
Parameter Connections	p93l4	(L4) The data classifier of the source and destination must match. The matching rules as specified	0	0	0
		444 7			_
Access Connections	p94n1	(N1) The connection identifier in an access connection refinement declaration must refer to a nam	0	0	0
Access Connections	p94n2	(N2) An access reference in an access connection declaration must reference an access feature of	0	0	0
Access Connections	p94l1	(L1) The category of the source and the destination of a access connection declaration must be the	0	0	0
Access Connections	p94l2	(L2) In the case of a bidirectional semantic access connection either connection end can be the sc	0	0	0
Access Connections	p94I3	(L3) In the case of a directional data or bus access connection the connection end representing the	0	0	0
Access Connections	p94I4	(L4) In a partial AADL model the ultimate source or destination may be a provides access feature	0	0	0
Access Connections	p94I5	(L5) If the access connection declaration represents an access connection between access feature	0	0	0
Access Connections	p94l6	(L6) If the access connection declaration represents a feature mapping up the containment hierard	0	0	0
Access Connections	p94I7	(L7) If the access connection declaration represents a feature mapping down the containment hier	0	0	0
Access Connections	p94l8	(L8) A requires access cannot be the source or destination feature reference of more than one acc	0	0	0
Access Connections	p94I9	(L9) For access connections the classifier of the provider access must match to the classifier of the	0	0	0
Access Connections	p94I10	(L10) If more than one access feature in a semantic access connection has an Access_Right prop	0	0	0
Access Connections	p94I11	(L11) The category of the access connection source and destination must be identical. If the comp	0	0	0
Feature Group Connections	p95n1	(N1) The connection identifier in a feature group connection refinement declaration must refer to a	0	0	0
Feature Group Connections	p95n2	(N2) A source or destination reference in a feature group connection declaration must reference a	0	0	0
Feature Group Connections	p95l1	(L1) If the feature group connection declaration represents a component connection between sibli	0	0	0
Feature Group Connections	p95l2	(L2) The Classifier Matching Rule property specifies the rule to be applied to match the feature g	0	0	0
reature Group Connections	paoiz	(LE2) The Grassmer_watering_rule property specifies the rule to be applied to match the leadure g	U	o,	U

Feature Group Connections	p95l3	(L3) The following rules are supported for feature group connection declarations that represent a c	0	0	0
Feature Group Connections	p95l4	(L4) The following rules are supported for feature group connection declarations that represent a c	0	0	0
Feature Group Connections	p95l5	(L5) If the feature group connection declaration represents a connection between feature group of	0	0	0
Feature Group Connections	p95l6	(L6) If the feature group connection declaration represents a connection between feature groups u	0	0	0
Feature Group Connections	p95l7	(L7) If the feature group connection declaration represents a connection between feature groups of	0	0	0
Feature Group Connections	p95l8	(L8) A feature group connection must be bidirectional or be consistent with the direction of the sou	0	0	0

SECTION NAME	ID	RULE TEXT	POS	IEG COMMENTARY
AADL Specifications	p41n1	(N1) An AADL specification has one global namespace. The package and property set identifiers reside in this space and	0	0 Checked by DFS during processing package nodes
AADL Specifications	p41n2	(N2) These package and property set identifiers qualify the names of individual elements contained in them when they are	0	0
AADL Specifications	p41n3	(N3) Package declarations represent labeled namespaces for component type, component implementation, feature group	0	0
AADL Specifications	p41n4	(N4) Property set declarations represent labeled namespaces for property type and property definition declarations.	0	0
ADL Specifications	p41n5	(N5) Packages and property sets may be separately stored. Those packages and property sets are considered to be part	0	0 Provided by parser
AADL Specifications	p41n6	(N6) Defining identifiers in AADL must not be one of the reserved words of the language (see Section 15.7).	0	0 Provided by parser
AADL Specifications	p41n7	(N7) The AADL identifiers and reserved words can be in upper or lower case (or a mixture of the two) (see Section 15).	0	0 Provided by realization of AADLIdentifer class
AADL Specifications	p41n8	(N8) The AADL does not require that an identifier be declared before it is referenced.	0	
Packages	p42n1	(N1) A defining package name consists of a sequence of one or more package identifiers separated by a double colon (B	0	0 Checked by counting private and public package declarations in DFS during processing package nodes
Packages	p42n2	(N2) The public and private section of a package may be declared in separate package declarations; these two declarations	0	0 Provided by parser
ackages	p42n3	(N3) Associated with every package is a package namespace that contains the names for all the elements defined within	0	
Packages	p42n4	(N4) The package namespace is divided into a public section and a private section. Items declared in the public section of	0	
ackages	p42n5	(N5) The reference to an item declared in another package must be an item name qualified with a package name separa	0	0 Can be checked after all possible references are known, impossible in current realization
Packages	p42n6	(N6) The reference to a property other than predeclared properties must be an property name qualified with a property se		0 Can be checked after all possible references are known, impossible in current realization
ackages	p42n7	(N7) The package name in a import declaration must exist in the global name space.	0	Checked by DFS during processing package nodes - imports
Packages	p42n8	(N8) The property set identifier in a import declaration must exist in the global name space.	0	Checked by DFS during processing package nodes - imports
ackages	p42n9	(N9) Items declared in the private section of the package can only be referenced from within the private section of the pa	0	Can be checked after all possible references are known, impossible in current realization
Packages	p42n10	(N10) If the qualifying package identifier of a qualified reference is missing, the referenced component classifier, feature q		0
Packages	p42n11	(N11) The package name referenced in an alias_declaration must exist in the global namespace and must be listed in the		0 Checked by DFS during processing package nodes - package aliases
Packages	p42n11	(N12) The classifier referenced by the alias declaration must exist in the name space of the public section of the package		Checked by DFS during processing package nodes - classifier aliases
Packages	p42n13	(N13) The classifier referenced by the alias declaration must refer to a component type or a feature group type.	0	0 Provided by parser
Packages	p42n14	(N14) The defining identifier of an alias_declaration must be unique in the namespace of the package containing the alias		Checked by DFS during processing package nodes - package and classifier aliases
Packages	p42n15	(N15) The alias_declaration makes the publicly visible identifier of classifiers declared in another package accessible in t		
Packages	p42n16	(N16) If the alias_declaration renames all publicly visible identifiers of component types and feature group types by naming		
Packages	p42n17	(N17) The identifiers introduced by the alias_declaration are only accessible within the package. When declared in the public transfer in	_	0 Not compatible with current realization of N14 check (alias ids == component ids), not all possible references are known
Packages	p42n18	(N18) The alias declared for a component type can be used instead of a qualified component type in a reference to a component type in a reference type in a reference to a component type in a reference type in a component type		
Packages	p42l1	(L1) The defining package name following the reserved word end must be identical to the defining package name following		0 Provided by parser
Packages	p42l2	(L2) For each package there may be at most one public section declaration and one private section declaration. These to		0 Checked with p42n1
Packages	p42l3	(L3) A component implementation may be declared in both the public and private part of a package. In that case the decl		0 Should be checked in component implementations
ackages	p42l4	(L4) The component category in an alias declaration must match the category of the referenced component type.	0	
Component Types	p43n1	(N1) The defining identifier for a component type must be unique in the namespace of the package within which it is decl	0	0 Checked when creating local namespaces of the packages
component Types	p43n2	(N2) Each component type has a local namespace for defining identifiers of prototypes, features, modes, mode transition	0	0
Component Types	p43n3	(N3) The component type identifier of the ancestor in a component type extension, i.e., that appears after the reserved w	0	0 Checked by DFS for each component type node
Component Types	p43n4	(N4) When a component type extends another component type, a component type namespace includes all the identifiers	0	
component Types	p43n5	(N5) A component type that extends another component type does not include the identifiers of the implementations of its	0	
Component Types	p43n6	(N6) The defining identifier of a feature, flow specification, mode, mode transition, or prototype must be unique in the nan	0	
omponent Types	p43n7	(N7) The refinement identifier of a feature, flow specification, or prototype refinement refers to the closest refinement or to	0	
Component Types	p43n8	(N8) The prototypes referenced by prototype binding declarations must exist in the local namespace of the component ty	0	
Component Types	p43n9	(N9) Mode transitions declared in the component type may not refer to event or event data ports of subcomponents.	0	0
Component Types	p43l1	(L1) The defining identifier following the reserved word end must be identical to the defining identifier that appears after the	0	0 Provided by parser
Component Types	p43l2	(L2) The prototypes, features, flows, modes, and properties subclauses are optional. If a subclause is present but empty,	0	0 Provided by parser(kinda, error is - "No viable alternative")
omponent Types	p43l3	(L3) The category of the component type being extended must match the category of the extending component type, i.e.,	0	0 Checked by DFS for each component type node
Component Types	p43l4	(L4) The classifier being extended in a component type extension may include prototype bindings. There must be at mos	0	0
Component Types	p43l5	(L5) A component type must not contain both a requires_modes_subclause and a modes_subclause.	0	0 Provided by parser(kinda, error is - extraneous input 'requires' expecting {'ANNEX', 'END', 'PROPERTIES', IDENTIFIER
component Types	p43l6	(L6) If the extended component type and an ancestor component type in the extends hierarchy contain modes subclauses	0	0 Checked by DFS for each component type node
Component Implementat	tior p44n1	(N1) A component implementation name consists of a component type identifier and a component implementation identif	0	0
component Implemental		(N2) The defining identifier of the component implementation must be unique within the local namespace of the component	0	0
		(N3) Every component implementation defines a local namespace for all defining identifiers of prototypes, subcomponent		

Component Implementation	tion n44m4	(NA) This level paragraphs inherits the paragraphs of the appropriate temporary times in the first in the US		
Component Implementat		(N4) This local namespace inherits the namespace of the associated component type, i.e., defining identifiers must be ur	0	
Component Implementat		(N5) Refinement identifiers of features must exist in the namespace of the associated component type or one of the component type of the component type or one of	U	0
Component Implementat		(N6) In a component implementation extension, the component type identifier of the component implementation being ex	0	0
Component Implementat		(N7) When a component implementation extends another component implementation, the local namespace of the extens	0	0
Component Implementat	tior p44n8	(N8) Within the scope of the component implementation, subcomponent declarations, connections, subprogram call sequ	0	0
Component Implementat	tior p44n9	(N9) The prototype referenced by the prototype binding declaration must exist in the local namespace of the component i	0	0
Component Implementat	tior p44I1	(L1) The pair of identifiers separated by a dot (вЪњ.вЪќ) following the reserved word end must be identical to the pair of	0	0
Component Implementat	tior p44l2	(L2) The prototypes, subcomponents, connections, calls, flows, modes, and properties subclauses are optional. If they are	0	0
Component Implementat		(L3) The category of the component implementation must be identical to the category of the component type for which the	0	0
Component Implementat		(L4) If the component implementation extends another component implementation, the category of both must match, i.e.,	0	0
Component Implementat		(L5) The classifier being extended in a component implementation extension may include prototype bindings. There must	0	0
Component Implementat		(L6) If the component type of the component implementation contains a requires modes subclause then the component	0	0
			0	0
Component Implementat		(L7) If modes are declared in the component type, then modes cannot be declared in component implementations.	0	
Component Implementat		(L8) If modes or mode transitions are declared in the component type, then mode transitions can be added in the compo	U	
Component Implementat		(L9) The category of a subcomponent being refined must match the category of the refining subcomponent declaration, i.	0	
Component Implementat	tior p44I10	(L10) For all other refinement declarations the categories must match (see the respective sections).	0	0
Component Implementat	tior p44I11	(L11) Component implementations and component implementation extensions must not refine prototypes declared in a c	0	0
Subcomponents	p45n1	(N1) The defining identifier of a subcomponent declaration placed in a component implementation must be unique within	0	0
Subcomponents	p45n2	(N2) The defining identifier of a subcomponent refinement must exist as a defining subcomponent identifier in the local na	0	0
Subcomponents	p45n3	(N3) The component type identifier or the component implementation name of a component classifier reference must exit	0	0
	p45n4		0	0
Subcomponents		(N4) The prototype identifier of a prototype reference must exist in the local name space of the component implementation	. 0	
Subcomponents	p45n5	(N5) The prototype referenced by the prototype binding declarations must exist in the local namespace of the component	U	0
Subcomponents	p45n6	(N6) The modes named in the in modes statement of a subcomponent must refer to modes in the component implement	0	0
Subcomponents	p45l1	(L1) The category of the subcomponent declaration must match the category of its corresponding component classifier re	0	0
Subcomponents	p45l2	(L2) The component classifier reference of a subcomponent declaration may include prototype bindings for a subset or a	0	0
Subcomponents	p45l3	(L3) In a subcomponent refinement declaration the component category may be refined from abstract to one of the concr	0	0
Subcomponents	p45l4	(L4) The Classifier_Substitution_Rule property specifies the rule to be applied when a refinement supplies a classifier and	0	0
Subcomponents	p45l5	(L5) In the case of a signature match, the component type of the subcomponent being refined must have a subset of the	0	0
Subcomponents	p45l6	(L6) The component category and optional component classifier or prototype reference can be followed by a set of array	0	0
Subcomponents	p45I7	(L7) The array size specification for the dimensions is optional. In this case the array declaration is considered incomplete		0
			0	
Subcomponents	p45l8	(L8) When refining a subcomponent array the number of dimensions of the array cannot be changed, but the array size c	0	0
Subcomponents	p45l9	(L9) When the subcomponent is declared as an array with array dimension sizes then a list of component implementation	0	0
Subcomponents	p45l10	(L10) Selecting index ranges in one or more dimensions of an array is only possible if the size of the array for these dime	0	0
Subcomponents	p45l11	(L11) An array element implementation list is valid only if (a) the subcomponent classifier is a component type and (b) all	0	0
Subcomponents	p45c1	(C1) The classifier of a subcomponent cannot recursively contain subcomponents with the same component classifier. In	0	0
Abstract Components	p46l1	(L1) An abstract component type declaration can contain feature declarations (including abstract feature declarations), flo	0	0
Abstract Components	p46l2	(L2) An abstract component implementation can contain subcomponent declarations of any category. Certain combinatio	0	0
Abstract Components	p46l3	(L3) An abstract component implementation can contain a modes subclause, a connections subclause, a flows subclause	0	0
			0	
Abstract Components	p46l4	(L4) An abstract subcomponent can be contained in the implementation of any component category.	0	
Abstract Components	p46l5	(L5) If an abstract subcomponent is refined to a concrete category, the concrete category must be acceptable to the com	0	0
Abstract Components	p46l6	(L6) An abstract subcomponent can be declared as an array of subcomponents.	0	0
Abstract Components	p46I7	(L7) If an abstract component type is refined to a concrete category, the features, modes, and flow specifications of the a	0	0
Abstract Components	p46l8	(L8) If an abstract component implementation is refined to a concrete category, the subcomponents, call sequences, mod	0	0
Prototypes	p47n1	(N1) The prototype identifier on the left-hand side of a prototype binding must exist in the local namespace of the classific	0	0
Prototypes	p47n2	(N2) The prototype identifier on the right-hand side of a prototype binding, if present, must exist in the local namespace of	0	0
	p47112		0	0
Prototypes		(N3) Unique component classifier references must exist in the public section of the package being identified in the references.		
Prototypes	p47n4	(N4) Unique feature group type references must exist in the public section of the package being identified in the reference	U	0
Prototypes	p47l1	(L1) The component category declared in the component prototype binding must match the component category of the p	0	9
Prototypes	p47l2	(L2) The component category of the optional component classifier reference in the prototype declaration must match the	0	0

Destatement	4710	(10) Kills and the state of the			
Prototypes Prototypes	p47l3	(L3) If the component prototype only specifies a component category, then any component type and component impleme	0		0
Prototypes Brototypes	p47l4	(L4) If the component prototype declaration includes a component classifier reference, then the classifier supplied in the	0		0
Prototypes	p47l5	(L5) The category of the component implementation that contains the prototype declaration places restrictions on the set	0		0
Prototypes	p47l6	(L6) If the direction is declared for feature prototypes, then the prototype actual satisfies the direction according to the sai	0	0	
Prototypes	p47I7	(LT) In the case of feature group prototypes, the supplied feature group types must match the declared feature group type	0		0
Prototypes	p47l8	(L8) A classifier supplied in a feature prototype binding must match the classifier of the prototype declaration, if present, a	0	0	
Prototypes	p47l9	(L9) Component prototypes declared with square brackets specify that they expect a list of component classifiers. These	0	0	
Prototypes	p47l10	(L10) The component category of the classifier reference or prototype reference in a prototype binding declaration must r	0		0
Prototypes	p47l11	(L11) If a direction is specified for an abstract feature in a prototype declaration, then the direction of the prototype actual	0	0	0
Prototypes	p47l12	(L12) Component prototype bindings must only bind component prototypes, feature group prototype bindings must only bindings mus	0	0	0
Prototypes	p47l13	(L13) Component prototype refinements must only refine component prototypes, feature group prototype refinements mu	0	0	0
Annex Subclauses and A		(N1) The annex identifier must be the name of an approved annex or a project-specific identifier different from the approv	0		0
Annex Subclauses and A		(N2) The mode identifiers in the in_modes statement must refer to modes in the component type or component implemen	0	0	-
Annex Subclauses and A	nr p48l1	(L1) Annex subclauses can only be declared in component types, component implementations, and feature group types.	0	0	0
Annex Subclauses and A	nr p48l2	(L2) A component type, component implementation, or feature group type declaration may contain at most one annex su	0	0	0
Annex Subclauses and A	nr p48l3	(L3) Annex libraries must be declared in packages.	0	0	0
Annex Subclauses and A	nr p48l4	(L4) A package declaration may contain at most one annex library declaration for each annex.	0	0	0
Data	p51l1	(L1) A data type declaration can contain provides subprogram access declarations as well as property associations.	0	0	0
Data	p51l2	(L2) A data type declaration must not contain a flow specification or modes subclause.	0	0	0
Data	p51l3	(L3) A data implementation can contain abstract, data and subprogram subcomponents, access connections, and data p	0	0	0
Data	p51l4	(L4) A data implementation must not contain a flow implementation, an end-to-end flow specification, or a modes subclau	0	0	0
Subprograms and Subpro	ogi p5 <u>2n1</u>	(N1) The defining identifier of a subprogram call sequence declaration must be unique within the local namespace of the	0	0	0
Subprograms and Subpro		(N2) The defining identifier of a subprogram call declaration must be unique within the local namespace of the componen	0	0	0
Subprograms and Subpro		(N3) If the called subprogram name is a subprogram classifier reference, its component type identifier or component impl	0	0	
Subprograms and Subpro		(N4) The subprogram classifier reference of a subprogram call may be a subprogram type reference.	0	0	0
Subprograms and Subpro		(N5) If the called subprogram name is a subprogram subcomponent reference, the subprogram subcomponent must exis	0	0	0
Subprograms and Subpro		(N6) If the called subprogram name is a requires subprogram access reference, the requires subprogram access must en	0	0	
Subprograms and Subprograms		(L1) A subprogram type declaration can contain parameter, out event port, out event data port, and feature group declara	0		0
		(L2) A subprogram implementation can contain abstract, subprogram, and data subcomponents, a subprogram calls sub-	0	0	Ö
Subprograms and Subpro					Č
Subprograms and Subprograms and Subprograms		(L3) Only one subprogram call sequence can apply to a given mode. (C1) The reference to a provides subprogram access of a processor in a subprogram call (processor, provides subprogram).	0		
Subprograms and Subpro		(C1) The reference to a provides subprogram access of a processor in a subprogram call (processor . provides_subprogram (C2) A subprogram call provides_subprogram access of a processor in a subprogram call (processor . provides_subprogram call processor . provides_subpr	0		
Subprograms and Subprograms	ogi p52c2	(C2) A subprogram call may reference a subprogram classifier. A project may enforce a consistency rule that this referen	U	- 0	U
0	0 - 50 -	AM The deficient design of a share constraint of the state of the stat			
Subprogram Groups and		(N1) The defining identifier of a subprogram group type must be unique within the package namespace of the package w	0		0
Subprogram Groups and		(N2) Each subprogram group provides a local namespace. The defining subprogram identifiers of subprogram declaration	0	0	-
Subprogram Groups and		(N3) The local namespace of a subprogram group type extension includes the defining identifiers in the local namespace	0	0	
Subprogram Groups and		(N4) The defining subprogram identifiers of subprogram access feature declarations in feature group refinements must need to the control of t	0		0
Subprogram Groups and		(N5) The package name of the unique subprogram group type reference must refer to a package name in the global nam	0	0	0
Subprogram Groups and	S(p53l1	(L1) A subprogram group type can contain provides and requires subprogram access, and provides and requires subprog	0	0	0
Subprogram Groups and	St p53l2	(L2) A subprogram group implementation can contain abstract, data, subprogram group, and subprogram subcomponent	0	0	0
Subprogram Groups and	St p53l3	(L3) A subprogram group type or implementation may contain zero or more subcomponent declarations. If it contains zer	0	0	0
Threads	p54l1	(L1) A thread type declaration can contain port, feature group, requires data access declarations, as well as requires and	0	0	0
Threads	p54l2	(L2) A thread component implementation can contain abstract, data, subprogram, and subprogram group subcomponent	0	0	0
Threads	p54l3	(L3) The Complete out event port, and Error out event data port are predeclared, i.e., are implicitly identifiers in the name	0	0	0
Threads	p54c3	(C3) Either the Compute_Entrypoint, Compute_Entrypoint_Source_Text Compute_Entrypoint_Call_Sequence property n	0	0	0
Threads	p54c4	(C4) The Period property must have a value if the Dispatch_Protocol property value is periodic, sporadic, timed, or hybrid	0	0	0
	p. 101	(1.) The policies of the control of			
Thread Groups	p55l1	(L1) A thread group component type can contain provides and requires data access, as well as port, feature group, provides	0		
rineau Groups	poort	ter) A unicad group component type can contain provides and requires data access, as well as port, feature group, provides	- 0	U	V

Thread Groups	p55l2	(L2) A thread group component implementation can contain abstract, data, subprogram, subprogram group, thread, and	0	C
Thread Groups	p55l3	(L3) A thread group implementation can contain a connections subclause, a flows subclause, a modes subclause, and pr	0	
Thread Groups	p55l4	(L4) A thread group must not contain a subprogram calls subclause.	0	
Processes	p56l1	(L1) A process component type can contain port, feature group, provides and requires data access, provides and require	0	
Processes	p56l2	(L2) A process component implementation can contain abstract, data, subprogram, subprogram group, thread, and threa	0	4
Processes	p56l3	(L3) A process implementation can contain a connections subclause, a flows subclause, a modes subclause, and a propi	0	
Processes	p56l4	(L4) A thread group must not contain a subprogram calls subclause.	0	C
Processes	p56c1	(C1) The complete source text associated with a process component must form a complete and legal program as defined	0	C
Processors	p61I1	(L1) A processor component type can contain port, feature group, provides subprogram access, provides subprogram gru	0	1
Processors	p61l2	(L2) A processor component implementation can contain declarations of memory, bus, virtual bus, virtual processor, and	0	1
Processors	p61l3	(L3) A processor implementation can contain a modes subclause, flows subclause, and a properties subclause.	0	
Processors	p61I4	(L4) A processor implementation can contain bus access, subprogram access, subprogram group access, port, feature, a	0	C
Processors	p61l5	(L5) A processor implementation must not contain a subprogram calls subclause.	0	
Virtual Processors	p62l1	(L1) A virtual processor component type can contain port, feature group, provides subprogram access, and subprogram (0	C
Virtual Processors	p62l2	(L2) A virtual processor component implementation can contain declarations of virtual bus, virtual processor, and abstract	0	
Virtual Processors	p62l3	(L3) A virtual processor implementation can contain a modes subclause, flows subclause, and a properties subclause.	0	C
Virtual Processors	p62l4	(L4) A virtual processor implementation must not contain a subprogram calls subclause.	0	
Virtual Processors	p62I5	(L5) A virtual processor implementation can contain subprogram access, subprogram group access, port, feature, and fe	0	C
Virtual Processors	p62c1	(C1) In a fully bound system every virtual processor must be directly or indirectly bound to, or directly or indirectly contain	0	0
Virtual Processors	p62c1	(C2) In a fully deployed system a requires virtual bus binding of a virtual processor specified by the Required_Virtual_Bus	0	
VIIIual P100855015	pozcz	(62) In a fully deployed system a requires virtual bus billoting of a virtual processor specified by the Required_virtual_bus	U	
Memory	p63I1	(L1) A memory type can contain bus access declarations, feature groups, a modes subclause, and property associations	0	C
			0	
Memory	p63l2	(L2) A memory implementation can contain abstract, memory, and bus subcomponent declarations.	0	
Memory	p63l3	(L3) A memory implementation can contain a modes subclause and property associations.	0	
Memory	p63l4	(L4) A memory implementation can contain bus access connection declarations. Bus access connections can connect a	0	C
Memory	p63l5	(L5) A memory implementation must not contain flows subclause, or subprogram calls subclause.	0	C
Buses	p64I1	(L1) A bus type can have requires bus access declarations, a modes subclause, and property associations.	Δ.	C
	p64I2		0	
Buses		(L2) A bus type must not contain any flow specifications.	0	
Buses	p64I3	(L3) A bus implementation can contain virtual bus and abstract subcomponent declarations.	0	
Buses	p64I4	(L4) A bus implementation can contain a modes subclause and property associations.	0	
Buses	p64l5	(L5) A bus implementation must not contain flows subclause, or subprogram calls subclause.	0	9
Virtual Buses	p65l1	(L1) A virtual bus type can have property associations.	0	C
Virtual Buses	p65l2	(L2) A virtual bus type must not contain flow specifications.	0	
Virtual Buses	p65l3	(L3) A virtual bus implementation can contain virtual bus subcomponent declarations.	0	
Virtual Buses	p65l4	(L4) A virtual bus implementation can contain a modes subclause and property associations.	0	
Virtual Buses	p65l5	(L5) A virtual bus implementation must not contain a connections subclause, flows subclause, or subprogram calls subclause.	0	C
Virtual Buses	p65c1	(C1) In a fully deployed system virtual buses must be directly or indirectly bound to processors or buses that support thes	0	Ç
Devices	p66l1	(L1) A device type can contain port, feature group, provides subprogram access, provides subprogram group access, but	0	C
Devices	p66l2	(L2) A device component implementation must not contain a subprogram calls subclause.	0	
Devices	p66l3	(L3) A device implementation can contain abstract, data, virtual bus, and bus subcomponents, bus access connections, a	0	C
Systems	p71l1	(L1) A system component type can contain subprogram, subprogram group, data and bus access declarations, port, feat	0	C
Systems	p71l2	(L2) A system component implementation can contain abstract, data, subprogram, subprogram group, process, and system	0	C
Systems	p71I3	(L3) A system implementation can contain a modes subclause, a connections subclause, a flows subclause, and propert	0	C
Systems	p71l4	(L4) A thread group must not contain a subprogram calls subclause.	0	C
Systems	p71n1	(N1) The defining identifier of a feature must be unique within the namespace of the associated component type.	0	C
Systems	p71n2	(N2) Thread features may not be declared using the predeclared ports names Complete or Error.	0	C
0,0.0.110	pr mz	(1.2) The decision of the decision of the production per to further complete of Effor.	J	

Systems p. P.T4 (NA A Feature is professed in 2014 (NE) 40 (1) Feature in contraction from the component important place in a support of professed in 2014 (NE) 50 (1) Feature in contraction of the contraction professed in 2014 (NE) 50 (1) Feature in contraction of the contraction professed in 2014 (NE) 50 (1) Feature in contraction of the contraction professed in 2014 (NE) 50 (1) Feature in contraction of the contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction of the contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in 2014 (NE) 50 (1) Feature in contraction professed in contraction professed in contraction contr						
Systems 9.71 (Is. 15). Each final of a contributed properly secrecision for lefeture must before to an element of a feature group. 9 (9) (9) (9) (9) (9) (9) (9) (9) (9) (Systems	p71n3	(N3) Each refining feature identifier that appears in a feature refinement declaration must also appear in a feature declaration	0	0	0
Systems p.11 (c.) Earl feature can be refined a first order to the same type extension. Systems p.12 (c.) A first care array remains for a feature and the order of a feature and the	Systems	p71n4	(N4) A feature is referenced in one of two ways. Within the component implementations for a component type, a feature (0	0	0
Systems p. 1750 (3) A closure informated displaced on a failure and the original shoutown and both the displaced sourty parameters and complete the systems p. 1741 (4) of the balation enfiltement agreeding and entire production of the production	Systems	p71n5	(N5) The path of a contained property association for a feature must refer to an element of a feature group.	0	0	0
Systems p. 178. (a) Excitor wange manual or by the decision for threshold, devices, and processors. Systems p. 179. (a) Excitor wange manual or by the decision for the feature being priced must always an array dimensions. Systems p. 179. (b) Excitor threshold process processors area, the feature being priced must and be one a many dimensions on the feature being priced must and be one a many dimensions. Systems p. 179. (b) Excitor threshold process priced priced must only be one and priced must and be not a many dimensions. Systems p. 179. (c) In the case of a hashing with a caseabler offerance, the designed or an interference to a size of the case of the size of th	Systems	p71l1	(L1) Each feature can be refined at most once in the same type extension.	0	0	0
Systems p. 17 M (L.) Filte features operationes an any diversion St., then he feature degraded are supported in the feature of systems p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of degraded are supported by the system p. 17 M (L.) in the destinant of destinant of destinant of destinant of the destinant of d	Systems	p71l2	(L2) A feature refinement declaration of a feature and the original feature must both be declared as port, parameter, acce	0	0	0
Systems 9716 (6) A contractor percent processor mater only towns on the format in the register of the format percent p	Systems	p71I3	(L3) Feature arrays must only be declared for threads, devices, and processors.	0	0	0
Systems 9718 (CB) A contrained property association must be lander in the feature group. Assistance Features 9111 (C) The feature direction in a reprined feature decideration in a component by 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Systems	p71I4	(L4) If the feature refinement specifies an array dimension, then the feature being refined must have an array dimension.	0	0	0
Asstract Features part of the Carear of a feature with a classifier reference, the classifier of the referred feature declaration in a component type 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Systems	p71I5	(L5) If the refinement specifies an array dimension size, then the feature being refined must not have an array dimension	0	0	0
Abstract Features gill 11 11 The feature direction in a refined feature decisation must be denteral to the feature decisation 0 0 Abstract Features gill 12 21 The direction of an abstract feature is supported, then the direction must be standard by the refinement (because in the Co.) 0 Abstract Features gill 12 21 The direction of an abstract feature with a feature protopyle refinement must be longer the protopyle the feature 0 0 Abstract Features gill 12 21 The direction of a feature with a feature with a feature protopyle refinement must be longer the protopyle and Feature (Co.) 21 The direction of the control of	Systems	p71I6	(L6) A contained property association must only be used when the feature is a feature group.	0	0	0
Abstract Features p3112 (3) A makester feature was specified, then the direction must be satisfactly by the reference canable 0 0 0 0 0 0 0 0 0	Systems	p71I7	(L7) In the case of a feature with a classifier reference, the classifier of the refined feature declaration in a component typ	0	0	0
Abstract Features p3112 (3) A makester feature was specified, then the direction must be satisfactly by the reference canable 0 0 0 0 0 0 0 0 0						
Abstract Features p. p	Abstract Features	p81I1	(L1) The feature direction in a refined feature declaration must be identical to the feature direction in the feature declaration	0	0	0
Abstract Features Groups and Featury pRDM (N1) The defining identifier of a feature group type must be unique within the package manespace of the package withers of the package within the package provides a local readers of provides and the package within the package provides a local readers of provides package within the package provides and the package within the package provides a local readers from provides, because and package within the local readers of provides package within the local readers provided in the package provides a local readers from provides package within the local readers provided in the package provides and package package within the local readers provided in the package provides and package packag	Abstract Features	p81I2	(L2) If the direction of an abstract feature is specified, then the direction must be satisfied by the refinement (see also the	0	0	0
Feature Groups and Featury 62020. (N1) The defining identifier of al feature group type must be unique within the package namerapose of the package where (N2) Each feature group with per provises a local namerapose. The defining identifiers of prophopye, feature group year Feature Groups and Featury (62020) (N2) Each feature group with per provises a local namerapose. The defining identifiers of prophopye, feature group year Feature Groups and Featury (62020) (N3) The defining feature scientifiers of feature group year group decidentifiers must be unique in the local names space of the compon. Feature Groups and Featury (62020) (N3) The pedicing feature scientifiers of feature group year group identifiers feature group year registers in the local names space of the compon. Feature Groups and Featury (62020) (N3) The policing name of the unique Mentine group year registers in Feature group decidentifiers must refer to a policing entire feature group year. Feature Groups and Featury (6202) (N4) The policing amount of the unique Mentine group year decidentifiers or feature group year. Feature Groups and Featury (6202) (N4) Realture group year with group year without inverse of or feature group year. Since the groups group is group year without inverse of or feature group year. Since the groups year with features and inverse of can be extended. Feature Groups and Featury (6202) (L3) Only feature group year without inverse of or feature group year with features and inverse of can be extended. Feature Groups and Featury (6202) (L3) A Realture group on the group year without inverse of or feature group year with features and inverse of can be extended. Feature Groups and Featury (6202) (L3) A Realture group informant may be refined to only add group year with features and inverse of can be extended. Feature Groups and Featury (6202) (L3) A Realture group informant may be refined to only add group year with features and inverse of the feature group year with features and inverse of the feature gro	Abstract Features	p81I3	(L3) An abstract feature with a feature prototype identifier and the prototype being referenced must both specify the same	0	0	0
Feature Groups and Featur p8203 Feature Scropus and Featur p8203 Seature Scropus and Featur p8203 S	Abstract Features	p81I4	(L4) An abstract feature refinement declaration of a feature with a feature prototype reference must only add property ass	0	0	0
Feature Groups and Featur p8203 Feature Scropus and Featur p8203 Seature Scropus and Featur p8203 S						
Feature Groups and Featur 8014 Feature Groups and Featur 8017 Feature Groups and Featur 8018 Gettle Groups and F	Feature Groups and Fea	atur p82n1	(N1) The defining identifier of a feature group type must be unique within the package namespace of the package where	0		0
Feature Groups and Featur p8267.	Feature Groups and Fea	atur p82n2	(N2) Each feature group type provides a local namespace. The defining identifiers of prototype, feature, and feature grou	0	0	0
Feature Groups and Featur p826 (NE) The defining feature group identifier of teature, grindeness of column and the relative feature group of the paragraph and the fluid paragraph and the paragraph and the surple setting of the paragraph and the surple setting of the paragraph and the surple setting paragraph and the paragraph	Feature Groups and Fea	atur p82n3	(N3) The local namespace of a feature group type extension includes the defining identifiers in the local namespace of the	0	0	0
Feature Croups and Featur p820 (1). A feature group byte meter contains a training and featur p820 (1). A feature group byte may contain zero or more elements 1, efeature or feature groups (1). A feature group byte may contain zero or more elements 1, efeature or feature groups and Featur p820 (1). A feature group byte may contain zero or more elements 1, efeature or feature groups and Featur p820 (1). Only feature group byte may contain zero or more elements 1, efeature or feature groups and Featur p820 (1). Only feature group byte may contain zero or more elements 1, efeature or feature group byte in the seature group byte with features and inverse of carnot contain an inv	Feature Groups and Fea	atur p82n4	(N4) The defining feature identifiers of feature group declarations must be unique in the local name space of the compon	0	0	0
Feature Groups and Featur p8210 (12) A feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to be the inverse of another feature group by each be declared to the third by the state are destined from the feature group by each by the state are destined from the feature group by each be declared to the time of the feature group by each by the state are destined group by each by the state group by each by the state are destined group by each by the state group by the state group by each	Feature Groups and Fea	atur p82n5	(N5) The defining feature group identifier of feature_refinement declarations in component types must exist in the local national management of the component types must exist in the local national management.	0	0	0
Feature Groups and Featur p8211 (1.1) A feature group type may contain zeo or more elements, i.e., feature of carbon groups, as indicated by the reviewed w Clay A feature group type who be declared to be the inverse of anothe feature group, pse, as indicated by the reviewed w Clay A feature group type who have been declared to be the inverse of anothe feature group, pse, as indicated by the reviewed w Clay A feature group type who have the inverse of relative group bype with out an inverse of cam be extended. Clay A feature group type that is an extension of another feature group type who with our another feature group type who with our another feature group type with feature and inverse of cambo contain an inverse of cambo group type and part of the date of t	Feature Groups and Fea	atur p82n6	(N6) The package name of the unique feature group type reference must refer to a package name in the global namespa	0	0	0
Feature Groups and Featur p8203 (1.2) A feature group types without inverse of another feature group type. as indicated by the reserved w (1.2) A feature group types without inverse of or feature group types without an inverse of cannot contain an inverse of teature group type with feature group type and feature group (6). A feature group declaration with an inverse of statement must only reference feature group type without an inverse of the feature group type and feature group feature groups contained in the feature group type to the feature group seaton feature group group seaton and its congelement that the feature group group groups and feature group group group seaton groups in a feature group must be a pair-wise complement with that in the feature group and feature group group group group group feature group group group group group group feature group gro	Feature Groups and Fea	atur p82n7	(N7) The prototype reference in a feature group declaration must refer to a prototype of the component type or feature gr	0	0	0
Feature Groups and Featur p8210 (L10) for feature group types without inverse of can be extended. Feature Groups and Featur p8224 (L4) A feature group type that is an extension of another feature group type with features and inverse of feature of the additional to the feature group type that is an extension of another feature group type with features and inverse of feature of the additional feature group type that is an extension of another feature group type with features and inverse of feature group types without an inverse of feature group types without an inverse of feature group section of the feature group types without an inverse of feature group and feature group types without an inverse of feature group and feature group types without an inverse of feature group and feature group and feature group group gro	Feature Groups and Fea	atur p82l1	(L1) A feature group type may contain zero or more elements, i.e., feature or feature groups. If it contains zero elements,	0	0	0
Feature Groups and Featur p8254 (L4) A feature group type that is an extension of another feature group type without an inverse of cannot contain an inverse of Corpus and Featur p8257 Feature Groups and Featur p8267 Feature Groups and Featur p8267 Feature Groups and Featur p8278 Feature Groups and Featur p8279 Feature Groups and Featur p8280 (L9) Each of the declared features group by see without an inverse of L0) 0 Feature Groups and Featur p8280 (L9) Each of the declared features or feature groups contained in the feature group and its complement with that in the feature Groups and Featur p82810 (L10) If both feature group types have zero features, then they are considered to complement with that in the feature Groups and Featur p82810 (L11) Ports are pair-wise complementary if they satisfy the access connection rules in Section 9.2.1. This includes (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. Ports p8381 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8). Ports p8381 (N2) The defining identifier of a port refinement declaration must also appear in a feature group to declaration of a component type. To Ports p8383 (N3) The unique component type identifier of the data classifier reference must be the name of a data component type. To Ports p8383 (N3) The unique component type identifier of a port refinement declaration must also appear in a feature declaration of a component type. To Ports p8383 (N4) The prototype identifier of a port refinement must adhere to the name of a data component type. To Ports p8383 (N4) The prototype identifier of a port refinement must be the name of a data component type. To Ports p8383 (N5) The unique component type identifier of the data classifier reference must be the name of a data component type. To Ports p8383 (N6) The unique component type identifier of a provides or requires subprogram group access declaration must be unique of the	Feature Groups and Fea	atur p82l2	(L2) A feature group type can be declared to be the inverse of another feature group type, as indicated by the reserved w	0	0	0
Feature Groups and Featur p8215 Feature Groups and Featur p8216 (L6) A feature group yet that is an extension of another feature group type with teatures and inverse of that adds feati Feature Groups and Featur p8216 (L7) A feature group refinement may be refined to only add properly associations. In this case inclusion of the feature group and Featur p8210 (L8) The number of feature or feature groups contained in the feature group and its complement must be identical; (L9) Each of the declared features or feature groups and its complement must be identical; (L9) Each of the declared features or feature groups and its complement must be identical; (L9) Each of the declared features or feature groups and feature group must be a pair-wise complement with the in the fix (L1) Fortia series or feature groups and Feature group and Feature group and Feature Groups and Feature group pytes shave zero features, then they are considered to complement each other; (L11) Fortia series are pair-wise complementary if they satisfy the port connection rules specified con 9.2.1. This includes (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group (L13) If a defining identifier of a port refinement declaration must also appear in a feature group to a general part of a feature group declaration, must also appear in a feature group on the feature group of the declaration from the declaration of a component type in the group of the declaration of the group of the group of the g	Feature Groups and Fea	atur p82l3	(L3) Only feature group types without inverse of or feature group types with features and inverse of can be extended.	0	0	0
Feature Groups and Featur p8216 (I.6) A feature group declaration with an inverse of statement must only reference feature group types without an inverse of Croups and Featur p8217 (I.5) The feature Groups and Featur p8218 (I.6) The number of feature or feature groups refined to only add properly associations. In this case inclusion of the feature group of the feature Groups and Featur p8219 (I.6) Each of the declared features or feature group and its complement must be idented; (I.5) The feature Groups and Featur p8219 (I.1) Ports are pair-wise complementary if they satisfy the port connection rules is Section 9.2.1. This includes (I.1) Feature Groups and Featur p8211 (I.1) Ports are pair-wise complementary if they satisfy the port connection rules is Section 9.2.1. This includes (I.1) Section Feature Groups and Featur p8211 (I.1) Ports are pair-wise complementary if they satisfy the access connection rules in Section 9.2.1. This includes (I.1) Feature Groups and Featur p8211 (I.1) Ports are pair-wise complementary if they satisfy the access connection rules in Section 9.2.1. This includes (I.1) Section for out direction is specified as part of a feature group declaration. Here all regions is section 9.4. Ports p8301 (N.1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8). O 0 Ports p8302 (N.2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type Ports p8303 (N.3) The unique component type identifier of the data classifier reference must be the name of a data component type of the port being the protection of a port refinement must be the same as the category of the port being to protect part of the protection of a port refinement must be the same as the category of the port being of the port being fement. If the feature being refined, or the port	Feature Groups and Fea	atur p82l4	(L4) A feature group type that is an extension of another feature group type without an inverse of cannot contain an inver	0	0	0
Feature Groups and Featur p8217 (L8) The number of feature group refinement may be refined to only add property associations. In this case inclusion of the feature group (L8) The number of feature or feature groups contained in the feature group and its complement with that in the five (L9) Each of the declared features or feature groups in a feature group must be a pair-wise complement with that in the five (L9) Each of the declared features or feature group in a feature group must be a pair-wise complement each other; Feature Groups and Featur p82110 (L11) If both feature group types have zero features, then they are considered to complement each other; Feature Groups and Featur p82112 (L12) Access features are pair-wise complementary if they satisfy the port connection rules in Section 9.2.1. This includes Feature Groups and Feature p82113 (L13) If an in or out direction is specified as part of a feature group declaration, then all features (see Section 8). Ports p83n1 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8). Ports p83n1 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type. To Ports p83n4 (N3) The unique component type identifier of the dacla classifier reference must be then of a data component type. To Ports p83n4 (N4) The protolype identifier of a protolype reference. If specified, must exist in the namespace of the component type or to Ports p83n1 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual processor, and device or passage. Ports p83n3 (L3) Data, event, and event data ports may be incompletely defined by not specifying the data component classifier of the component of the port being in the protone passage. Ports p83n3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of the port being in the port being in the port be	Feature Groups and Fea	atur p82l5	(L5) The feature group type that is an extension of another feature group type with features and inverse of that adds feat	0	0	0
Feature Groups and Featur p8218 (L8) The number of feature groups contained in the feature group and its complement must be identical; 0 0 0 Feature Groups and Featur p8219 (L9) Each of the declared features or feature groups in a feature group must be a pair-wise complement with that in the file of the feature group purple and reading p8210 (L10) by the feature group person and Featur p8210 (L10) by the feature group person and Feature p8211 (L10) ports are pair-wise complementary if they satisfy the port connection rules in Section 9.2.1. This includes 0 0 0 Feature Groups and Featur p82112 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.2.1. This includes 0 0 0 Feature Groups and Feature p82113 (L13) iff an in or out direction is specified as part of a feature group declaration, then all features inside the feature group in the section 9.2.1. This includes 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Feature Groups and Fea	atur p82l6	(L6) A feature group declaration with an inverse of statement must only reference feature group types without an inverse	0	0	0
Feature Groups and Featur p8219 (L9) Each of the declared features or feature group in a feature group must be a pair-wise complement with that in the feature Groups and Featur p82110 (L1) if both feature group types have zero features, then they are considered to complement each other; 0 0 0 Feature Groups and Featur p82111 (L1) if both feature group types have zero features, then they are considered to complement each other; 0 0 0 Feature Groups and Featur p82112 (L1) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. 0 0 0 Feature Groups and Featur p82112 (L1) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. 0 0 0 Feature Groups and Featur p82113 (L1) if an in or out direction is specified as pan of a feature group declaration, then all features inside the feature group in 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Feature Groups and Fea	atur p82l7	(L7) A feature group refinement may be refined to only add property associations. In this case inclusion of the feature gro	0	0	0
Feature Groups and Featur p82110 (L10) If both feature group types have zero features, then they are considered to complement each other; (Eature Groups and Featur p82111 (L11) Ports are pair-wise complementary if they satisfy the connection rules is pecified in Section 9.2.1. This includes (L12) Research Groups and Featur p82112 (L12) Access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary if they satisfy the access features are pair-wise complementary in they satisfy the access features are pair-wise complementary in they satisfy the access features are pair-wise complementary in they satisfy the access features are pair-wise complementary for all features (see Section 8). O	Feature Groups and Fea	atur p82l8	(L8) The number of feature or feature groups contained in the feature group and its complement must be identical;	0	0	0
Feature Groups and Featur p82111 (L11) Ports are pair-wise complementary if they satisfy the port connection rules specified in Section 9.2.1. This includes (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. 0 (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group Ports p83n1 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8). Ports p83n3 (N2) The ubrique component type identifier of a port refinement declaration must also appear in a feature declaration of a component type. To Ports p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a data component type or Ports p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of the component type or Ports p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of the component type or Ports p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of the component type or Ports p83n3 (N3) The unique component type identifier of a prototype reference, if specified, must exist in the namespace of the component type or Ports p83n3 (N3) Data, event, and event data ports may be incompletely defined by not specifying the data component classifier reference or or Ports p83n3 (N3) Data, event, and event data ports may be incompletely defined by not specifying the data component classifier of or	Feature Groups and Fea	atur p82l9	(L9) Each of the declared features or feature groups in a feature group must be a pair-wise complement with that in the fe	0	0	0
Feature Groups and Featur p82l12 (L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4. 0 0 0 C (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group 0 0 0 0 0 C C (L13) If an in or out direction is specified as part of a feature group declaration, then all features (see Section 8). 0 0 0 0 C C C (L13) If an in or out direction is specified as part of a feature group declaration, then all features (see Section 8). 0 0 0 0 C C C C (L13) If an in or out direction is specified as part of a feature group declaration for a component type of the past of the pas	Feature Groups and Fea	atur p82I10	(L10) If both feature group types have zero features, then they are considered to complement each other;	0	0	0
Feature Groups and Featur p82l13 (L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Feature Groups and Fea	atur p82l11	(L11) Ports are pair-wise complementary if they satisfy the port connection rules specified in Section 9.2.1. This includes	0	0	0
Ports p83n1 (N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8). Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type. To ports p83n3 (N3) The unique component type identifier of a prototype identifier of a prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or ports p83n1 (L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual processor, and device ports p83n2 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or ports p83n3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier reference or ports p83n3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of ports p83n3 (L5) The port direction of a port refinement must be the same as the category of the port being refined, or the port being of p83n4 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature being refined in the feature being refined in the feature being refined by adding a property p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be unique of the port being of the port being refined by adding a property p84n2 (N2) The defining identifier or a provides or requires subprogram or subprogram group access of the port being refined by a defining of the port being refined by adding a property p84n2 (N2) The defining identifier or a provides or requires subprogram or subprogram group access of the port being reference, if present, must exist in the option of the	Feature Groups and Fea	atur p82l12	(L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section 9.4.	0	0	0
Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type 0 0 0 Ports p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a data component type. T 0 0 0 Ports p83n3 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or 0 0 Ports p83l1 (L1) Ports can be declared in subprogram, thread group, process, system, processor, virtual processor, and devic 0 Ports p83l2 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or p83l3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of 0 0 0 Ports p83l3 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being refined. If the feature b 0 0 0 Ports p83l5 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b 0 0 0 Subprogram and Subprogrape p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be uniqu 0 0 Subprogram and Subprogrape p84n4 (N4) The prototype identifier or component implementation name of a subprogram group access classifier reference, if present, must exist in tit 0 0 0	Feature Groups and Fea	atur p82l13	(L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the feature group	0	0	0
Ports p83n2 (N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type 0 0 0 Ports p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a data component type. T 0 0 0 Ports p83n3 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or 0 0 Ports p83l1 (L1) Ports can be declared in subprogram, thread group, process, system, processor, virtual processor, and devic 0 Ports p83l2 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or p83l3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of 0 0 0 Ports p83l3 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being refined. If the feature b 0 0 0 Ports p83l5 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b 0 0 0 Subprogram and Subprogrape p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be uniqu 0 0 Subprogram and Subprogrape p84n4 (N4) The prototype identifier or component implementation name of a subprogram group access classifier reference, if present, must exist in tit 0 0 0						
Ports p83n3 (N3) The unique component type identifier of the data classifier reference must be the name of a data component type. T 0 0 0 Ports p83n4 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or 0 0 Ports p83n4 (L1) Ports can be declared in subprogram, thread, thread group, processo, system, processor, virtual processor, and devic 0 0 Ports p83n2 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or 0 Ports p83n3 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of 0 Ports p83n4 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being refined, or the port being refined. If the feature b 0 Ports p83n5 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b 0 Ports p83n5 (N1) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be uniqu Subprogram p84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be uniqu Subprogram p84n1 (N2) The defining identifier or component implementation name of a subprogram group access of 0 0 Subprogram and Subprogram p84n1 (N3) The component type identifier or component implementation name of a subprogram group access classifier reference, if present, must exist in tit 0 0 D	Ports	p83n1	(N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8).	0	0	0
Ports p8314 (N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports	p83n2	(N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a component type	0	0	0
Ports p8311 (L1) Ports can be declared in subprogram, thread, thread group, processor, virtual processor, and device processor, and device processor. Ports p8312 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or ports p8313 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier of ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being refined. If the feature being refined. If the feature being refined. If the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined by the feature being refined by the feature being refined. If the feature being refined by the feature being refined. If the feature being refined by the feature by the	Ports	p83n3	(N3) The unique component type identifier of the data classifier reference must be the name of a data component type. T	0	0	0
Ports p8312 (L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or p8313 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier do ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b port being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b port being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b port passed	Ports	p83n4	(N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the component type or	0	0	0
Ports p8313 (L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier d Ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being Ports p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b Subprogram and Subprograp84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be uniqu Subprogram and Subprograp84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must exist as a definin Subprogram and Subprograp84n3 (N3) The component type identifier or component implementation name of a subprogram group access classifier reference, if present, must exist in tit to 0 Subprogram and Subprograp84n4 (N4) The prototype identifier of a subprogram group access classifier reference, if present, must exist in tit to 0	Ports	p83l1	(L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual processor, and device	0	0	0
Ports p8314 (L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being perined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature b p8315 (L5) The port direction of a port refinement must be the same as the category of the port being refined. If the feature b p8315 (L5) The port direction of a port refinement must be the same as the category of the port being refined. If the feature b p8315 (N1) The defining identifier of a provides or requires subprogram or subprogram group refinement must be uniqued by p8315 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must exist as a defining by p8415 (N3) The component type identifier or component implementation name of a subprogram group access classifier reference, if present, must exist in the provided of the port being refined. If the port being refined. If the feature b provides or requires subprogram or subprogram group access classifier reference, if present, must exist in the provides p8415 (N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if present, must exist in the provides past past past past past past past pas	Ports	p83l2	(L2) Data and event data ports may be incompletely defined by not specifying the data component classifier reference or	0	0	0
Ports p8315 (L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Ports	p83l3	(L3) Data, event, and event data ports may be refined by adding a property association. The data component classifier d	0		0
Subprogram and Subprogra p84n1 (N1) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be unique of Subprogram and Subprogram p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must exist as a defining of Subprogram and Subprogram p84n3 (N3) The component type identifier or component implementation name of a subprogram or subprogram group access clossifier reference, if present, must exist in tite of subprogram page of the provided of the provided or requires subprogram or subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group access clossifier reference, if present, must exist in tite of subprogram group acces	Ports	p83l4	(L4) The port category of a port refinement must be the same as the category of the port being refined, or the port being	0	0	0
Subprogram and Subprogra p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must exist as a definin 0 0 Subprogram and Subprogra p84n3 (N3) The component type identifier or component implementation name of a subprogram or subprogram group access cl 0 0 Subprogram and Subprograp p84n4 (N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if present, must exist in tit 0 0 0	Ports	p83I5	(L5) The port direction of a port refinement must be the same as the direction of the feature being refined. If the feature b	0	0	0
Subprogram and Subprogra p84n2 (N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must exist as a definin 0 0 Subprogram and Subprogra p84n3 (N3) The component type identifier or component implementation name of a subprogram or subprogram group access cl 0 0 Subprogram and Subprograp84n4 (N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if present, must exist in tit 0 0 0	Subprogram and Subpro	ogra p84n1	(N1) The defining identifier of a provides or requires subprogram or subprogram group access declaration must be unique	0	0	0
Subprogram and Subprograp84n3 (N3) The component type identifier or component implementation name of a subprogram or subprogram group access cl 0 0 Subprogram and Subprograp84n4 (N4) The prototype identifier of a subprogram group access classifier reference, if present, must exist in tr 0 0		•		0	0	0
Subprogram and Subprograp84n4 (N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if present, must exist in the subprogram of the prototype identifier of a subprogram or subprogram group access classifier reference, if present, must exist in the subprogram or subprog				0	0	0
		• .		0	0	0
		-		0		0
	Casprogram and Subpro	-git po-iii	(2-1) in a susprogram access releas to a component diagonier of a component prototype, filed the category of the classific	U		

Subprogram and Subprogr	-	(L2) If a subprogram group access refers to a component classifier or a component prototype, then the category of the cl	0	0	0
Subprogram and Subprogr	-	(L3) An abstract feature can be refined into a subprogram access or a subprogram group access. In this case, the abstra	0	0	0
Subprogram and Subprogr	ra p84l4	(L4) A subprogram or subprogram group access declaration that does not specify a component classifier reference is inc	0	0	0
Subprogram and Subprogr	ra p84l5	(L5) A subprogram or subprogram group access declaration may be refined by adding a property association. Inclusion of	0	0	0
Subprogram and Subprogr	ra p8416	(L6) A provides subprogram access cannot be refined to a requires subprogram access and a requires subprogram acce	0	0	0
Subprogram and Subprogr	ra p84c1	(C1) A provides subprogram access feature indicates that a subprogram is made available to be referenced. A project material of the control o	0	0	0
Subprogram Parameters	p85n1	(N1) The defining identifier of a parameter must be unique within the namespace of the subprogram type containing the g	0	0	0
Subprogram Parameters	p85n2	(N2) The defining parameter identifier of a parameter refinement declaration must also appear in a feature declaration of	0	0	0
Subprogram Parameters	p85n3	(N3) The data classifier reference must refer to a data component type or a data component implementation.	0	0	0
Subprogram Parameters	p85n4	(N4) The prototype identifier, if present, must exist in the namespace of the subprogram classifier that contains the paran	0	0	0
Subprogram Parameters	p85l1	(L1) Parameters can be declared for subprogram component types.	0	0	0
Subprogram Parameters	p85l2	(L2) A parameter declaration that does not specify a data classifier reference is incomplete. Such a reference can be add	0	0	0
Subprogram Parameters	p85l3	(L3) A parameter declaration may be refined by adding a property association. Inclusion of the data classifier reference is	0	0	0
Subprogram Parameters	p85l4	(L4) The parameter direction of a parameter refinement must be the same as the direction of the feature being refined. If	0	0	0
Data Component Access	p86n1	(N1) The defining identifier of a provides or requires data access declaration must be unique within the namespace of the	0	0	0
Data Component Access	p86n2	(N2) The defining identifier of a provides or requires data access refinement must exist as a defining identifier of a provide	0	0	0
Data Component Access	p86n3	(N3) The component type identifier or component implementation name of a data access classifier reference must exist it	0	0	0
Data Component Access	p86n4	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains the data access declar	0	0	0
Data Component Access	p86I1	(L1) If a data access refers to a component classifier or a component prototype, then the category of the classifier or prot	0	0	0
Data Component Access	p86l2	(L2) A data access declaration may be refined by refining the data classifier, by adding a property association, or by doin	0	0	0
Data Component Access	p86I3	(L3) A provides data access cannot be refined to a requires data access and a requires data access cannot be refined to	0	0	0
Data Component Access	p86l4	(L4) An abstract feature can be refined into a data access. In this case, the abstract feature must not have a direction spe	0	0	0
Data Component Access	p86c1	(C1) A data access declaration that does not specify a data classifier reference is incomplete. Such a reference can be a	0	0	0
Data Component Access	p86c2	(C2) If the source code of a component does access shared data, then the component type declaration must specify a re	0	0	0
Data Component Access	p86c3	(C3) A data access refinement may refine an abstract feature declaration. If the abstract feature declaration specifies a d	0	0	0
Bus Component Access	p87n1	(N1) The defining identifier of a provides or requires bus access declaration must be unique within the namespace of the	0	0	0
Bus Component Access	p87n2	(N2) The defining identifier of a provides or requires bus refinement must exist as a defining identifier of a requires or pro	0	0	0
Bus Component Access	p87n3	(N3) The component type identifier or component implementation name of a bus access classifier reference must exist in	0	0	0
Bus Component Access	p87n4	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains the bus access declar	0	0	0
Bus Component Access	p87I1	(L1) If a bus access refers to a component classifier or a component prototype, then the category of the classifier or proto	0	0	0
Bus Component Access	p87l2	(L2) A bus access declaration may be refined by refining the bus classifier, by adding a property association, or by doing	0	0	0
Bus Component Access	p87I3	(L3) A provides bus access cannot be refined to a requires bus access and a requires bus access cannot be refined to a	0	0	0
Bus Component Access	p87l4	(L4) An abstract feature can be refined into a bus access. In this case, the abstract feature must not have a direction spe	0	0	0
Bus Component Access	p87c1	(C1) A bus access declaration that does not specify a bus classifier reference is incomplete. Such a reference can be add	0	0	0
Bus Component Access	p87c2	(C2) If a bus access feature is a refinement of an abstract feature, then the direction of the abstract feature, if specified, in	0	0	0
Bus Component Access	p87n1	(N1) The defining identifier of a defined connection declaration must be unique in the local namespace of the component	0	0	0
Bus Component Access	p87n2	(N2) The connection identifier in a connection refinement declaration must refer to a named connection declared in an ar	0	0	0
Bus Component Access	p87l1	(L1) A connection refinement must contain at least one of the following: a connection source and destination subclause,	0	0	0
Bus Component Access	p87l2	(L2) If a semantic connection may be active in a particular mode, then the ultimate source and ultimate destination composition	0	0	0
Bus Component Access	p87I3	(L3) If a semantic connection may be active in a particular mode transition, then the ultimate source component must be	0	0	0
	1				
Feature Connections	p91n1	(N1) A source or destination reference in a feature connection or feature connection refinement declaration must reference	0	0	0
Feature Connections	p91n2	(N2) The subcomponent reference may refer to a subcomponent or a subcomponent array.	0	0	0
Feature Connections	p91I1	(L1) If the feature connection declaration represents a connection between features of sibling components, then the sour	0	0	0
Feature Connections	p91l2	(L2) If the feature connection declaration represents a connection between features up the containment hierarchy, then the	0	0	0
Feature Connections	p91I3	(L3) If the feature connection declaration represents a connection between features down the containment hierarchy, the	0	0	0
Feature Connections	p91I4	(L4) If the feature connection declaration specifies a directional connection, then the direction of the connection must be	0	0	0
	*				
Feature Connections	p91I5	(L5) The individual connections of a semantic connection must be bidirectional or have the same direction. The direction	0	0	0

Port Connections	p92n1	(N1) The connection identifier in a port connection refinement declaration must refer to a named port or feature connection	0	0	0
Port Connections	p92n2	(N2) A source or destination reference in a port connection or port connection refinement declaration must reference a po	0	0	0
Port Connections	p92n3	(N3) The subcomponent reference may also consist of a reference to a subcomponent array.	0	0	0
Port Connections	p92n4	(N4) The event_or_event_data identifier of event source specifications (self.event_or_event_data_identifier) must not cor	0	0	0
Port Connections	p92l1	(L1) In the case of a directional port connection the connection end representing the source of the flow must be the source	0	0	0
Port Connections	p92l2	(L2) In the case of a bidirectional port connection either connection end can be the source. If the bidirectional connection	0	0	0
Port Connections	p92l3	(L3) If the source connection end is a data access feature it must have read access rights; if the destination connection e	0	0	0
Port Connections	p92l4	(L4) The feature identifier of a subcomponent reference may refer to a feature array, if the subcomponent is a thread, dev	0	0	0
Port Connections	p92l5	(L5) The following are acceptable sources and destinations of port connections. The left column shows connections betw	0	0	0
Port Connections	p92l6	(L6) If the port connection declaration represents a connection between ports of sibling components, then the source must	0	0	0
Port Connections	p92l7	(L7) If the port connection declaration represents a connection between ports up the containment hierarchy, then the sou	0	0	0
Port Connections	p92l8	(L8) If the port connection declaration represents a connection between ports down the containment hierarchy, then the s	0	0	0
Port Connections	p92l9	(L9) The individual connections of a semantic port connection must be bidirectional or have the same direction. The direction	0	0	C
Port Connections	p92l10	(L10) Self. <identifier> must only be referenced as the source of a connection.</identifier>	0	0	0
Port Connections	p92l11	(L11) A data port cannot be the destination of more than one semantic port connection unless each semantic port connection	0	0	0
Port Connections	p92l12	(L12) A semantic connection cannot contain connection declarations with both immediate and delayed Timing property vi	0	0	0
Port Connections	p92l13	(L13) For connections between data ports, event data ports and data access, the data classifier of the source port must r	0	0	0
Port Connections	p92l14	(L14) The following rules are supported: въў въў въў въў Classifier_Match: The source data type and data implementa	0	0	0
Port Connections	p92l15	(L15) If more than one port connection declaration in a semantic port connection has a property association for a given $ \circ $	0	0	0
Port Connections	p92l16	(L16) A processor port specification must only be used in event connections within threads and subprograms.	0	0	0
Port Connections	p92c1	(C1) There cannot be cycles of immediate connections between threads, devices, and processors.	0	0	0
Port Connections	p92c2	(C2) The processor port identifier of a processor port specification (processor.processor_port_identifier) must name a portion of the processor port identifier of a processor port specification (processor.processor_port_identifier) must name a portion of the processor port identifier of a processor port specification (processor.processor_port_identifier) must name a portion of the processor port identifier of a processor port iden	0	0	0
Port Connections	p92c3	(C3) The Supports_Classifier_Subset_Matches property may be associated with a bus or virtual bus. This specifies the s	0	0	0
Port Connections	p92c4	(C4) The Supports_Type_Conversions property may be associated with a bus or virtual bus. This specifies the subset may	0	0	0
Parameter Connections	p93n1	(N1) The connection identifier in a parameter connection refinement declaration must refer to a named parameter or feati	0	0	0
Parameter Connections	p93n2	(N2) A source (destination) reference in a parameter connection declaration must reference a parameter of a preceding (0	0	0
Parameter Connections	p93l1	(L1) The source of a parameter connection must be an incoming data or event data port of the containing thread, an inco	0	0	0
Parameter Connections	p93l2	(L2) The following source/destination pairs are acceptable for parameter connection declarations: threadport -> call.parar	0	0	0
Parameter Connections	p93l3	(L3) A parameter cannot be the destination feature reference of more than one parameter connection declaration unless	0	0	0
Parameter Connections	p93l4	(L4) The data classifier of the source and destination must match. The matching rules as specified by the Classifier_Matc	0	0	0
Access Connections	p94n1	(N1) The connection identifier in an access connection refinement declaration must refer to a named access or feature of	0	0	0
Access Connections	p94n2	(N2) An access reference in an access connection declaration must reference an access feature of a subcomponent, sut	0	0	0
Access Connections	p94l1	(L1) The category of the source and the destination of a access connection declaration must be the same, i.e., they must	0	0	0
Access Connections	p94I2	(L2) In the case of a bidirectional semantic access connection either connection end can be the source.	0	0	0
Access Connections	p94I3	(L3) In the case of a directional data or bus access connection the connection end representing the component being acc	0	0	0
Access Connections	p94I4	(L4) In a partial AADL model the ultimate source or destination may be a provides access feature of a component instead	0	0	0
Access Connections	p94I5	(L5) If the access connection declaration represents an access connection between access features of sibling componen	0	0	0
Access Connections	p94I6	(L6) If the access connection declaration represents a feature mapping up the containment hierarchy, then one connection	0	0	0
Access Connections	p94I7	(L7) If the access connection declaration represents a feature mapping down the containment hierarchy, then one conne	0	0	0
Access Connections	p94I8	(L8) A requires access cannot be the source or destination feature reference of more than one access connection declar	0	0	0
Access Connections	p94I9	(L9) For access connections the classifier of the provider access must match to the classifier of the requires access acco	0	0	0
Access Connections	p94I10	(L10) If more than one access feature in a semantic access connection has an Access_Right property association, then t	0	0	0
Access Connections	p94l11	(L11) The category of the access connection source and destination must be identical. If the component category is spec	0	0	0
Feature Group Copposition	nen05n1	(N1) The connection identifier in a feature group connection refinement declaration must refer to a feature group named (0	0	0
Feature Group Connectionsp95n1 Feature Group Connectionsp95n2		(N1) The connection identifier in a feature group connection refinement declaration must refer to a feature group named (0	0	C
Feature Group Connectio		(N2) A source or destination reference in a feature group connection declaration must reference a feature group declared (L1) If the feature group connection declaration represents a component connection between sibling components, the feature	0	0	C
•	-		0	0	0
Feature Group Connectio	-	(L2) The Classifier_Matching_Rule property specifies the rule to be applied to match the feature group classifier of a con-	0	0	0
Feature Group Connectio		(L3) The following rules are supported for feature group connection declarations that represent a connection up or down	-	•	
Feature Group Connectio	11509014	(L4) The following rules are supported for feature group connection declarations that represent a connection between two	0	0	0

Feature Group Connectionsp95l5	(L5) If the feature group connection declaration represents a connection between feature group of sibling components, th	0	0 0	0
Feature Group Connectionsp95l6	(L6) If the feature group connection declaration represents a connection between feature groups up the containment hier	0	0 0	0
Feature Group Connectionsp95I7	(L7) If the feature group connection declaration represents a connection between feature groups down the containment h	0	0 0	0
Feature Group Connectionsp95l8	(L8) A feature group connection must be bidirectional or be consistent with the direction of the source and destination fea	0	0 0	0