

Tables of invocation variants and determinism of initial mode

Invocation variants

XSLT 3.0 defines three invocation variants: apply-templates invocation, call-template invocation and call-function invocation. The following table informally summarizes these variants and the requirements for each invocation method, the details of which are in each respective section:

Invocation variant	Allows unnamed	Default initial name	Initial scope of initial template	Parameters	Result*
Apply-templates	yes**	#unnamed**	First item of initial match selection	global, local, tunnel	raw, tree, serialized
Call-template	no	xsl:initial-template	Context item, defaulting to global context item	global, local, tunnel	raw, tree, serialized
Call-function	no	n/a	n/a	global and function parameters	raw, tree, serialized

* It is up to the API of a processor, and whether the serialization feature is supported, what type of result processing takes place. Historically, apply-templates and call-template invocation resulted in taking the unnamed output definition for serialization, or returning a tree model if serialization was not requested.

** Whether or not the #unnamed mode is the default or is allowed depends on several factors, which is explained in the following paragraphs on apply-templates invocation.

Apply-templates invocation

Determining which modes must be declared

The following table can be used to determine whether a mode must be declared or not. This depends on whether a mode is the unnamed mode or whether the mode is used as the entry point for apply-templates invocation, or whether it is used in the default-mode attribute.

Top level element	Modes must be declared	Initial unnamed mode exists implicitly	Initial named mode must exist implicitly or explicitly	Default mode must be declared
xsl:stylesheet / xsl:transform	no	yes	yes, or XTDE0045	no
xsl:package without declared-modes, or with declared-mode="yes"	yes, raises XTSE3085	no, raises XTDE0045	yes, raises XTDE0045	yes, raises XTSE3085

xsl:package with declared-modes="no"	no	yes	yes, or XTDE0045	no
---	----	-----	------------------	----

Determining whether a mode is considered *available* for use as initial mode

By default, the mode specified as initial mode for a stylesheet transformation is declared if it has an xsl:mode declaration with visibility *public* or *final*, or when declared-modes is "no" and there is an implicit use of the mode by an xsl:template with the mode attribute set to that mode name, or when the initial mode is the unnamed mode.

This table summarizes the rules on whether a mode is available as initial mode. Visibility final can be taken as public.

Inquiry	Declared	visibility	used in xsl:template	used in default-mode	declared-modes	available as initial mode
#unnamed	yes	n/a	-	-	-	yes
	no	n/a	neither	-	no	yes
			either	-	yes	no, XTDE0045
named	yes	private	-	no	-	no, XTDE0045
		private	-	yes	-	maybe*
		public	-	-	-	yes
	no	n/a	either	-	no	yes
			neither	-	no	no, XTDE0045
			either	-	yes	XTSE3085***
named	in used package**	accepted private	-	-	-	no, XTDE0045
		accepted public	-	-	-	yes

* The term "maybe" means that this named mode can only be used if the stylesheet is invoked without an initial mode set, but with apply-templates invocation, in which case it defaults to the default-mode attribute of the outermost element. That this mode is private is irrelevant at that point. Explicitly invoking a private named mode, however, raises XTDE0045.

** A used package, referenced by xsl:use-package, will by default result in all components that are available from the used package to become private. Unless explicitly stated otherwise with an xsl:accept, modes from a used package are not eligible for apply-templates invocation. Note that the unnamed mode from a used package is never available outside that package.

*** A stylesheet with those declarations cannot be compiled and will always raise XTSE3085.

Determining which mode is the initial mode for invocation

The following table shows how a processor determines the initial mode for apply-templates invocation. For a mode to exist implicitly it means that an xsl:template instruction with that mode name must exist. To exist explicitly means that an xsl:mode declaration with that name must exist.

The columns *mode unavailable* can be taken as the result of the previous table. If a mode is not available, for instance because it is not declared, it can sometimes raise a static or

dynamic error. This table only lists the dynamic error as a result of invocation, it assumes the stylesheet can be compiled.

In this table, *ex:init* is an example of a name for a named mode, *#unnamed* means the unnamed mode and *<absent>* means the attribute is absent.

Invoked with initial mode of	default-mode attribute	error if mode unavailable (declared-modes=no)	error if mode unavailable (declared-modes=yes)	Resulting initial mode
<absent>*	ex:init	XTDE0045	not possible**	ex:init
	#unnamed	-	not possible**	#unnamed
	<absent>	-	XTDE0045	#unnamed
#unnamed*	ex:init	-	not possible**	#unnamed
	#unnamed	-	not possible**	
	<absent>	-	XTDE0045	
ex:init	ex:init	-	not possible**	ex:init
	#unnamed	XTDE0045		
	<absent>			
	ex:other			

* It is implementation-dependent how a processor defines explicitly that the #unnamed mode should be used as initial mode. It may, for instance, default to the unnamed mode if it is not supplied. A processor may also provide an option to explicitly use the default mode of the top-level element, for instance with #default.

** If a default-mode attribute uses a non-declared mode and declared-modes=yes, then the stylesheet cannot be compiled regardless of the setting of the initial mode. The error is XTSE3085.