Schema documentation for xml-screen-1.3.xsd

20 october 2013

Table of Contents

Namespace: ""

Schemas

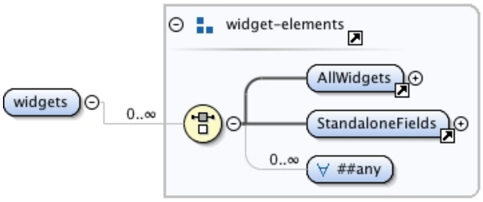
Main schema `xml-screen-1.3.xsd`

Namespace	No namespace

Elements

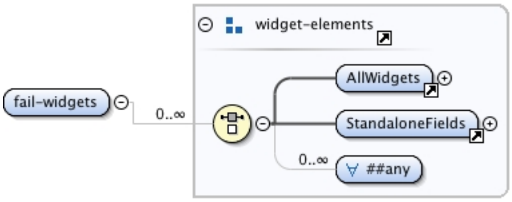
Element `widgets`

Namespace	No namespace

Diagram	 widgets 0..∞ widget-elements AllWidgets StandaloneFields 0..∞ ##any
Used by	Element Group section-elements

Instance	`<widgets>`
	`  <AllWidgets>{1,1}</AllWidgets>`
	`  <StandaloneFields>{1,1}</StandaloneFields>`
	`</widgets>`

Element fail-widgets

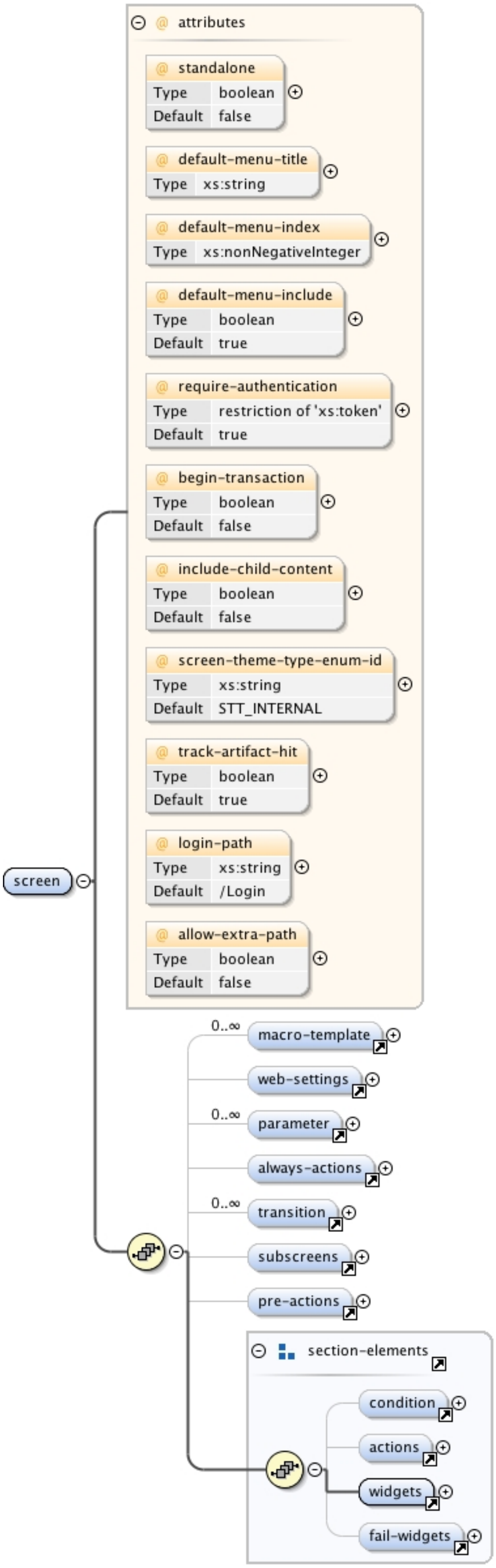
Namespace	No namespace
Diagram	 fail-widgets 0..∞ widget-elements AllWidgets StandaloneFields 0..∞ ##any
Used by	Element Group section-elements

Instance	`<fail-widgets>`
	`  <AllWidgets>{1,1}</AllWidgets>`
	`  <StandaloneFields>{1,1}</StandaloneFields>`
	`</fail-widgets>`

Element screen

Namespace	No namespace
Annotations	The screen is the basic unit of a user interface defines how data, logic, and visual elements fit together. Screen filenames should be camel-cased and start with an upper-case letter (whereas transitions should start with a lower-case letter).

<table>
<tr>
<td>Diagram</td>
<td>



</td>
</tr>
<tr>
<td>Instance</td>
<td>

```
<screen allow-extra-path="false" begin-transaction="false" default-menu-
include="true" default-menu-index="" default-menu-title="" include-child-
content="false" login-path="/Login" require-authentication="true" screen-theme-type-enum-
id="STT_INTERNAL" standalone="false" track-artifact-hit="true">
  <macro-template location="" type="">{0,unbounded}</macro-template>
  <web-settings allow-web-request="true" character-encoding="UTF-8" mime-type="text/
html" require-encryption="true">{0,1}</web-settings>
  <parameter from="" name="" value="">{0,unbounded}</parameter>
  <always-actions>{0,1}</always-actions>
  <transition method="any" name="">{0,unbounded}</transition>
```

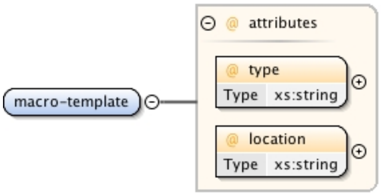
</td>
</tr>
</table>

```
  <subscreens always-use-full-path="false" default-item="">{0,1}</subscreens>
  <pre-actions>{0,1}</pre-actions>
</screen>
```

Attributes

QName	Type	Fixed	Default	Use
allow-extra-path	boolean		false	optional
	If set to true arbitrary path follow this screen's path is allowed. Default is false and an exception will be thrown if there is extra path that does not match a subscreen, a transition, or content below the screen.			
begin-transaction	boolean		false	optional
	Begin a transaction for the screen render if there is not one already in place. Most screens don't need this, but it is useful for greater data consistency in certain cases.			
default-menu-include	boolean		true	optional
	Set this to false to not automatically appear in the parent's subscreens menu based on the directory it is in. If true this screen will automatically be included in the parent's subscreens menu.			
default-menu-index	xs:nonNegativeInteger			optional
default-menu-title	xs:string			optional
include-child-content	boolean		false	optional
	False by default, meaning that child content is sent to the client as they are and nothing else with it. If true then the child content is included in this screen as if it were a subscreen.			
login-path	xs:string		/Login	optional
	If specified will be used as the login screen path for this screen and any subscreens, otherwise defaults to "/Login".			
require-authentication	restriction of xs:token		true	optional
screen-theme-type-enum-id	xs:string		STT_INTERNAL	optional
standalone	boolean		false	optional
	If set to true this screen will be rendered without rendering any parent screens. It can still be referred to as a subscreen of its parent, but when rendered the parent will not run, the rendering will start at this screen. Any non-standalone children will still be treated as normal subscreens.			
track-artifact-hit	boolean		true	optional
	If set to false no ArtifactHit or ArtifactHitBin data will be kept for this screen and for any content or transitions under the screen.			

Element `macro-template`

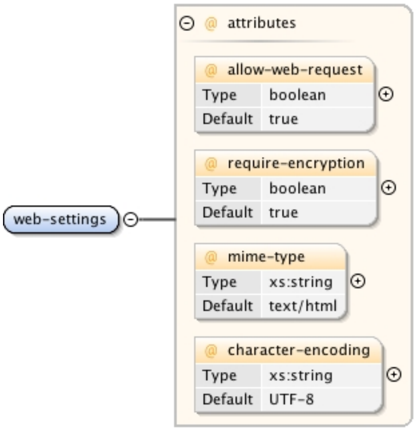
Namespace	No namespace
Annotations	A location here will override the settings in the moqui-conf.screen-facade.screen-text-output, but will be overridden by a value set with the ScreenRender.macroTemplate() method.
Diagram	 macro-template attributes type Type xs:string location Type xs:string
Used by	Element screen

Attributes

QName	Type	Fixed	Default	Use
location	xs:string			required

QName	Type	Fixed	Default	Use
type	xs:string			required
	`Can be anything. Default supported values include: text, html, xsl-fo, xml, and csv.`			

Element `web-settings`

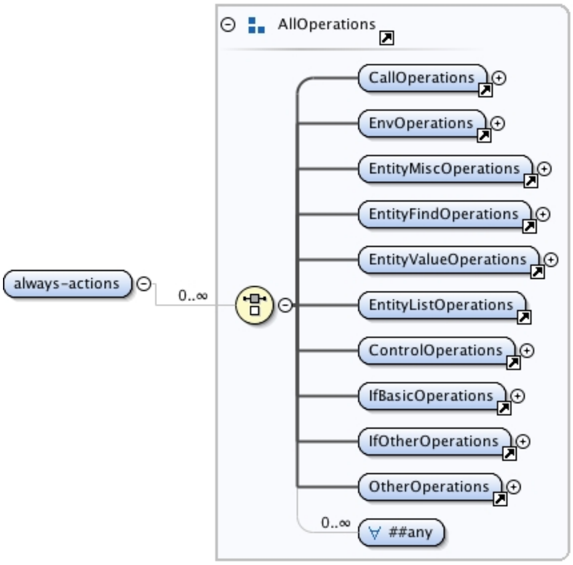
Namespace	No namespace
Diagram	 attributes allow-web-request Type boolean Default true require-encryption Type boolean Default true mime-type Type xs:string Default text/html character-encoding Type xs:string Default UTF-8 web-settings
Used by	Element screen
Attributes	

QName	Type	Fixed	Default	Use
allow-web-request	boolean		true	optional
character-encoding	xs:string		UTF-8	optional
mime-type	xs:string		text/html	optional
require-encryption	boolean		true	optional

Element `always-actions`

Namespace	No namespace

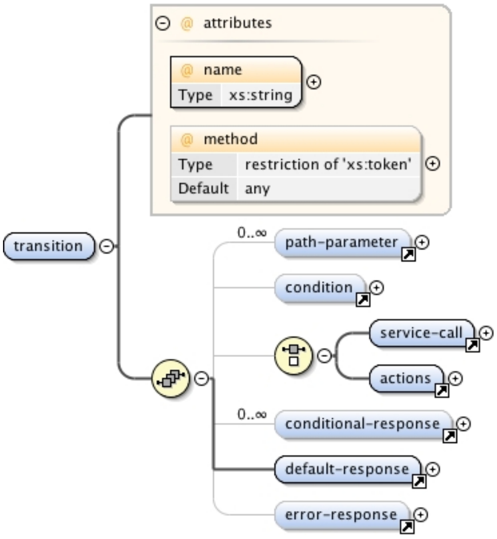
Annotations	`These actions always run when this screen appears in a screen path, including`
	`both screen rendering and transition running. One difference between this and the pre-actions element is`
	`that this runs before transitions are processed while pre-actions do not. The always-actions also run for`
	`all screens in a path while the pre-actions only run for screens that will be rendered.`

Diagram	 AllOperations CallOperations EnvOperations EntityMiscOperations EntityFindOperations EntityValueOperations EntityListOperations ControlOperations IfBasicOperations IfOtherOperations OtherOperations 0..∞ ##any always-actions 0..∞
Used by	Element screen

`Instance`	`<always-actions>`
	`  <CallOperations>{1,1}</CallOperations>`

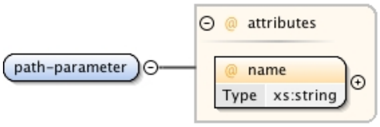
<table>
<tr><td></td><td><pre> &lt;EnvOperations>{1,1}</EnvOperations>
  <EntityMiscOperations>{1,1}</EntityMiscOperations>
  <EntityFindOperations>{1,1}</EntityFindOperations>
  <EntityValueOperations>{1,1}</EntityValueOperations>
  <EntityListOperations>{1,1}</EntityListOperations>
  <ControlOperations>{1,1}</ControlOperations>
  <IfBasicOperations>{1,1}</IfBasicOperations>
  <IfOtherOperations>{1,1}</IfOtherOperations>
  <OtherOperations>{1,1}</OtherOperations>
</always-actions&gt;</pre></td></tr>
</table>

Element `transition`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td></td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Used by</td><td>Element screen</td></tr>
<tr><td>Instance</td><td><pre>&lt;transition method="any" name="">
  <path-parameter name="">{0,unbounded}</path-parameter>
  <condition>{0,1}</condition>
  <conditional-response parameter-map="" save-current-screen="false" save-parameters="false" type="url" url="" url-type="screen-path">{0,unbounded}</conditional-response>
  <default-response parameter-map="" save-current-screen="false" save-parameters="false" type="url" url="" url-type="screen-path">{1,1}</default-response>
  <error-response parameter-map="" save-current-screen="false" save-parameters="false" type="url" url="" url-type="screen-path">{0,1}</error-response>
</transition&gt;</pre></td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>method</b></td><td>restriction of xs:token</td><td></td><td>any</td><td>optional</td></tr>
<tr><td><b>name</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td></td><td colspan="4"><pre>Transition names should be camel-cased and start with an lower-case
 letter (whereas screen filenames
and subscreens-item names start with a upper-case letter).

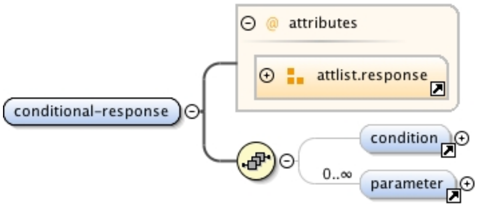
The transition name is used in link and other elements in place of URLs
 when going to another
screen within this application. The transition name will appear briefly
 as the URL before the
redirect is done for the transition response.</pre></td></tr>
</table></td></tr>
</table>

Element `path-parameter`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Used by</td><td>Element transition</td></tr>
</table>

Attributes	QName	Type	Fixed	Default	Use
	name	xs:string			required

Element `conditional-response`

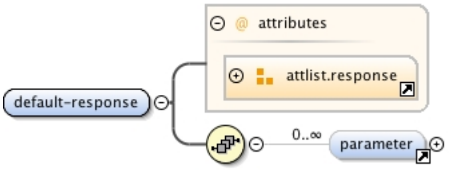
Namespace	No namespace
Diagram	 attributes attlist.response conditional-response condition 0..∞ parameter
Used by	Element transition

Instance

```
<conditional-response parameter-map="" save-current-screen="false" save-
parameters="false" type="url" url="" url-type="screen-path">
  <condition>{0,1}</condition>
  <parameter from="" name="" value="">{0,unbounded}</parameter>
</conditional-response>
```

Attributes	QName	Type	Fixed	Default	Use
	parameter-map	xs:string			optional
	save-current-screen	boolean		false	optional
		Save the current screen's path and parameters for future use, generally with the screen-last type of response.			
	save-parameters	boolean		false	optional
		Save the current parameters (and request attributes) before doing a redirect so that the screen rendered after the redirect renders in a context similar to the original request to the transition.			
	type	restriction of xs:token		url	optional
	url	xs:string			optional
		The URL to follow in response, based on the url-type. The default url-type is "screen-path" which means the value here is a path from the current screen to the desired screen (or transition). You can use "." to represent the current screen, and ".." to represent the parent screen on the runtime screen path. The ".." can be used multiple times, such as "../.." to get to the parent screen of the parent screen (the grand-parent screen). If the screen-path type url starts with a "/" it will be relative to the root screen instead of relative to the current screen. If the url-type is "plain" then this can be any valid URL (relative on current server or absolute).			
	url-type	restriction of xs:token		screen-path	optional

Element `default-response`

Namespace	No namespace
Diagram	 attributes attlist.response default-response 0..∞ parameter
Used by	Element transition

Instance

```
<default-response parameter-map="" save-current-screen="false" save-
parameters="false" type="url" url="" url-type="screen-path">
  <parameter from="" name="" value="">{0,unbounded}</parameter>
```

```
</default-response>
```

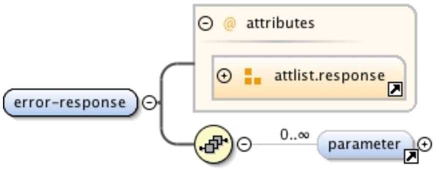
Attributes

QName	Type	Fixed	Default	Use
parameter-map	xs:string			optional
save-current-screen	boolean		false	optional
	Save the current screen's path and parameters for future use, generally with the screen-last type of response.			
save-parameters	boolean		false	optional
	Save the current parameters (and request attributes) before doing a redirect so that the screen rendered after the redirect renders in a context similar to the original request to the transition.			
type	restriction of xs:token		url	optional
url	xs:string			optional
	The URL to follow in response, based on the url-type. The default url-type is "screen-path" which means the value here is a path from the current screen to the desired screen (or transition). You can use "." to represent the current screen, and ".." to represent the parent screen on the runtime screen path. The ".." can be used multiple times, such as "../.." to get to the parent screen of the parent screen (the grand-parent screen). If the screen-path type url starts with a "/" it will be relative to the root screen instead of relative to the current screen. If the url-type is "plain" then this can be any valid URL (relative on current server or absolute).			
url-type	restriction of xs:token		screen-path	optional

Element `error-response`

Namespace: No namespace

Diagram



Used by

Element	transition

Instance

```
<error-response parameter-map="" save-current-screen="false" save-parameters="false" type="url" url="" url-type="screen-path">
  <parameter from="" name="" value="">{0,unbounded}</parameter>
</error-response>
```

Attributes

QName	Type	Fixed	Default	Use
parameter-map	xs:string			optional
save-current-screen	boolean		false	optional
	Save the current screen's path and parameters for future use, generally with the screen-last type of response.			
save-parameters	boolean		false	optional
	Save the current parameters (and request attributes) before doing a redirect so that the screen rendered after the redirect renders in a context similar to the original request to the transition.			
type	restriction of xs:token		url	optional
url	xs:string			optional
	The URL to follow in response, based on the url-type. The default url-type is "screen-path" which means the value here is a path from the current screen to the desired screen (or transition). You can use "." to represent the current screen, and ".." to represent the parent screen on the runtime			

<table>
<tr><td></td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td></td><td colspan="4">screen path. The ".." can be used multiple times, such as "../.." to get to the parent screen of the
parent screen (the grand-parent screen).

If the screen-path type url starts with a "/" it will be relative to the root screen instead of relative
to the current screen.

If the url-type is "plain" then this can be any valid URL (relative on current server or absolute).</td></tr>
<tr><td><b>url-type</b></td><td>restriction of xs:token</td><td></td><td>screen-path</td><td>optional</td></tr>
</table>
</td></tr>
</table>

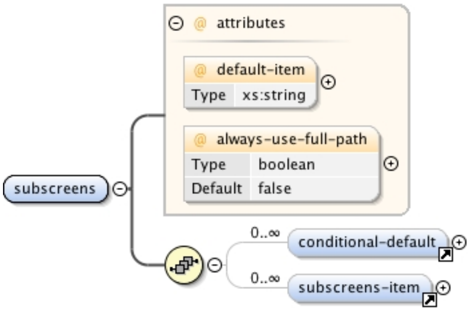
Element subscreens

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>Declare subscreens for this screen. One subscreen at a time is active, based on the "screen path" used to
access this screen. The parent screen (this screen) will be the current element in the screen path and the
next screen path element will be the name of the subscreen of this screen to use.

If there is no additional element in the screen path or the next element is not a valid subscreen-item.name
then the default-item will be the active subscreen.

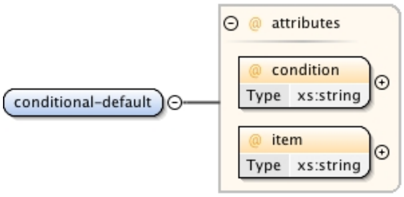
There are three ways to add subscreens to a screen:

1. for screens within a single application:
by directory structure: create a directory in the directory where the parent screen is named the same as
the parent screen's filename and put XML Screen files in that directory (name=filename up to .xml,
title=screen.default-title, location=parent screen minus filename plus directory and filename for
subscreen)
2. for including screens that are part of another application, or shared and not in any application:
subscreens-item elements below the screen -> subscreens element (this element)
3. for adding screens, removing screens, or changing order and title of screens to an existing application:
a record in the moqui.screen.SubscreensItem entity

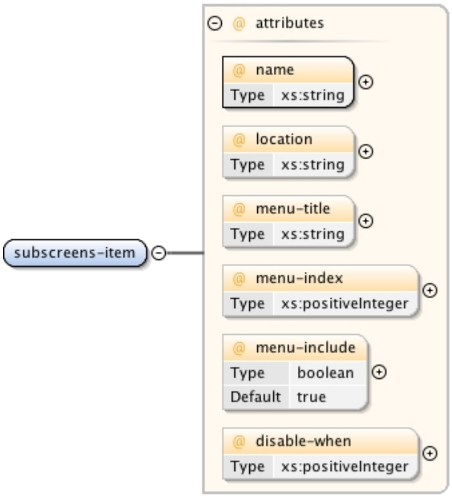
There are two visual elements (widgets) that come from the subscreens, a menu and the active subscreen.
Those are included with the widgets using the "subscreens-menu" and "subscreens-active" elements, or the
"subscreens-panel" element.</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Used by</td><td>Element screen</td></tr>
<tr><td>Instance</td><td><subscreens always-use-full-path="false" default-item="">
  <conditional-default condition="" item="">{0,unbounded}</conditional-default>
  <subscreens-item disable-when="" location="" menu-include="true" menu-index="" menu-title="" name="">{0,unbounded}</subscreens-item>
</subscreens></td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>always-use-full-path</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
<tr><td><b>default-item</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td></td><td colspan="4">The name of the default subscreen-item. Used when then screen-path ends on this screen so we
know which subscreen-item to activate.</td></tr>
</table>
</td></tr>
</table>

	QName	Type	Fixed	Default	Use
		If empty the first subscreen-item will be the default.			

Element conditional-default

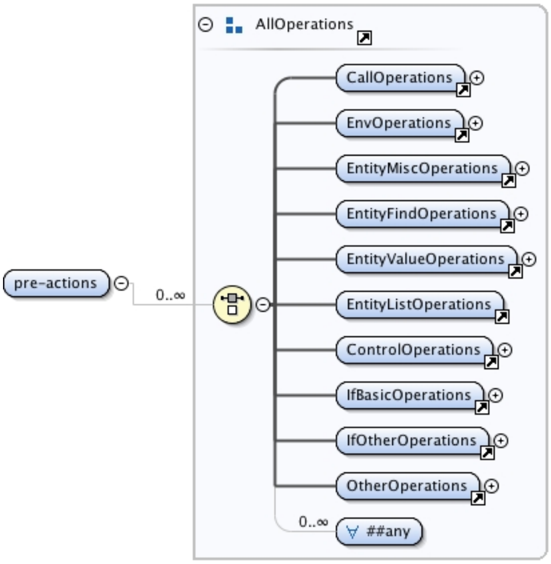
Namespace	No namespace				
Diagram	 attributes condition Type xs:string item Type xs:string conditional-default				
Used by	Element	subscreens			
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	condition	xs:string			required
		Groovy condition expression (evaluates to a boolean) used to determine if the specified subscreens item is the one to use by default instead of the on specified in the subscreens.@default-item attribute.			
	item	xs:string			required
		The subscreens item to make the default.			

Element subscreens-item

Namespace	No namespace				
Annotations	One way to add a subscreen. This is most commonly used to refer to a subscreen that is located in another application, another part of this application, that is not in any application and is meant to be shared, or is in a different type of location than the parent screen. One subscreens-item is active at a time, meaning that screen is shown and the tab/etc for that screen is highlighted.				
Diagram	 attributes name Type xs:string location Type xs:string menu-title Type xs:string menu-index Type xs:positiveInteger menu-include Type boolean Default true disable-when Type xs:positiveInteger subscreens-item				
Used by	Element	subscreens			
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	disable-when	xs:positiveInteger			optional
		This condition is run the subscreens menu is rendered to see if the item is available (otherwise the button/link/etc is disabled).			
	location	xs:string			optional
		Subscreen location can include various prefixes to support including from a file, http, component,			

QName	Type	Fixed	Default	Use
	or a content repository. If empty defaults to the value of the name attribute under the current screen (in the directory with the same name as the current screen), and can be a screen or sub-content.			
menu-include	boolean		true	optional
menu-index	xs:positiveInteger			optional
	If specified this item will be inserted in existing list of subscreens at this index (1-based). If empty this item will be added to the end of the list (after the directory load, before the entity load).			
menu-title	xs:string			optional
name	xs:string			required
	The name of the subscreens item for use in the screen path. The screen path element following the one for the parent screen of the item will match on this name. Subscreen Item names should be camel-cased and start with a upper-case letter (just like screen filenames start with a upper-case letter).			

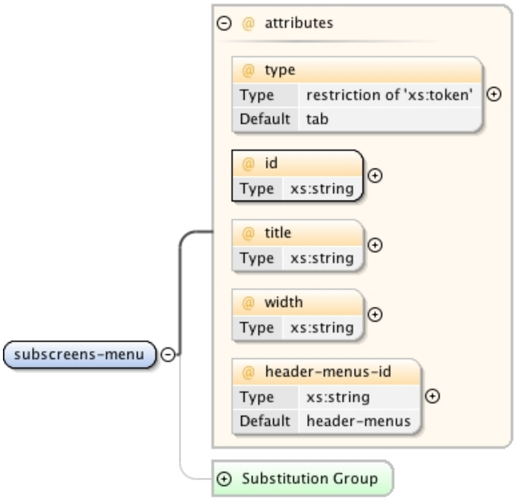
Element `pre-actions`

Namespace	No namespace
Annotations	These actions run before any of the screens (this screen or any parent screens) are rendered, allowing you to set parameters used by parent screens or other general reasons.
Diagram	 AllOperations CallOperations EnvOperations EntityMiscOperations EntityFindOperations EntityValueOperations EntityListOperations ControlOperations IfBasicOperations IfOtherOperations OtherOperations 0..∞ ##any pre-actions 0..∞
Used by	Element screen

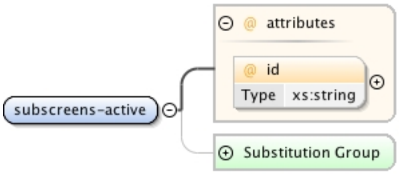
Instance	`<pre-actions>`
	`  <CallOperations>{1,1}</CallOperations>`
	`  <EnvOperations>{1,1}</EnvOperations>`
	`  <EntityMiscOperations>{1,1}</EntityMiscOperations>`
	`  <EntityFindOperations>{1,1}</EntityFindOperations>`
	`  <EntityValueOperations>{1,1}</EntityValueOperations>`
	`  <EntityListOperations>{1,1}</EntityListOperations>`
	`  <ControlOperations>{1,1}</ControlOperations>`
	`  <IfBasicOperations>{1,1}</IfBasicOperations>`
	`  <IfOtherOperations>{1,1}</IfOtherOperations>`
	`  <OtherOperations>{1,1}</OtherOperations>`
	`</pre-actions>`

Element `subscreens-menu`

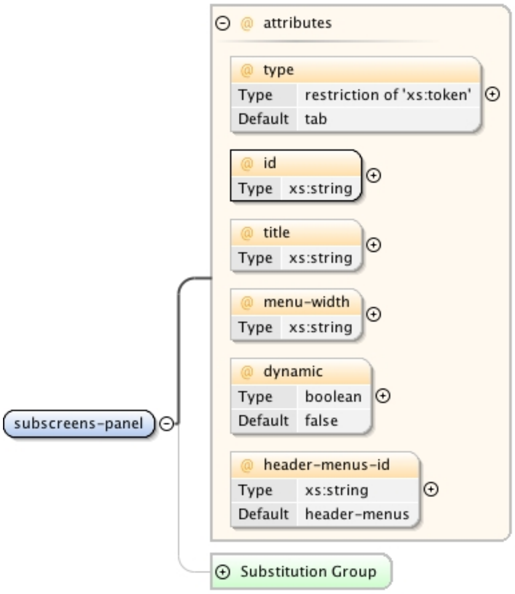
Namespace	No namespace

<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>header-menus-id</b></td><td>xs:string</td><td></td><td>header-menus</td><td>optional</td></tr>
<tr><td><b>id</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>title</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>type</b></td><td>restriction of xs:token</td><td></td><td>tab</td><td>optional</td></tr>
<tr><td><b>width</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
</table>
</td></tr>
</table>

Element `subscreens-active`

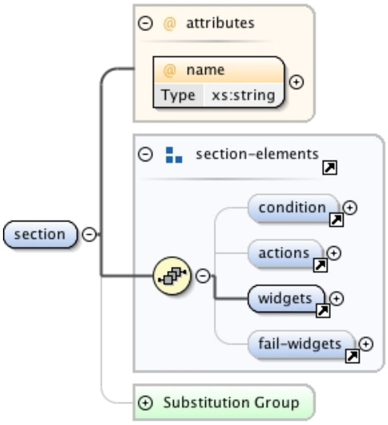
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>id</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
</table>
</td></tr>
</table>

Element `subscreens-panel`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
</table>

Attributes	QName	Type	Fixed	Default	Use
	dynamic	boolean		false	optional
	header-menus-id	xs:string		header-menus	optional
	id	xs:string			required
	menu-width	xs:string			optional
	title	xs:string			optional
	type	restriction of xs:token		tab	optional

Element section

Namespace	No namespace
Diagram	 attributes name Type xs:string section-elements condition actions widgets fail-widgets section Substitution Group

Instance

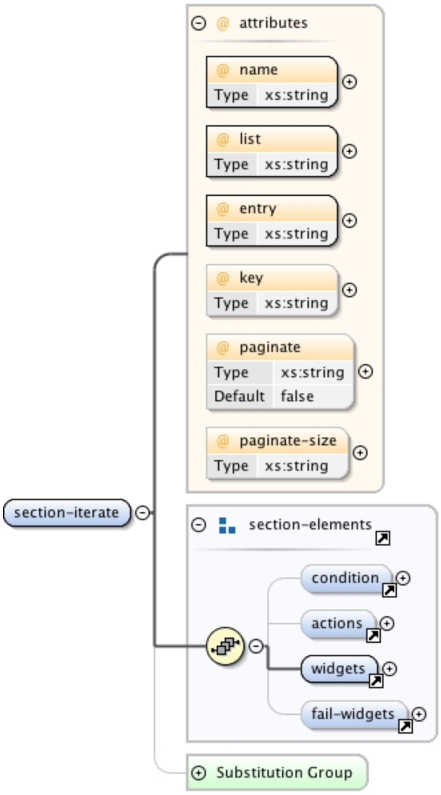
```
<section name="">
  <condition>{0,1}</condition>
  <actions>{0,1}</actions>
  <widgets>{1,1}</widgets>
  <fail-widgets>{0,1}</fail-widgets>
</section>
```

Attributes	QName	Type	Fixed	Default	Use
	name	xs:string			required
		A name for the section, used for reference within the screen. Must be specified and must be unique within the screen.			

Element section-iterate

Namespace	No namespace

Diagram



Instance

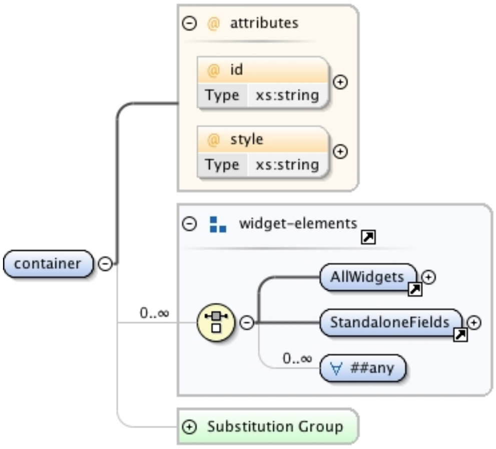
```
<section-iterate entry="" key="" list="" name="" paginate="false" paginate-size="">
  <condition>{0,1}</condition>
  <actions>{0,1}</actions>
  <widgets>{1,1}</widgets>
  <fail-widgets>{0,1}</fail-widgets>
</section-iterate>
```

Attributes

QName	Type	Fixed	Default	Use
entry	xs:string			required
	The name of the field that will contain each entry as we iterate through the list.			
key	xs:string			optional
	If list points to a Map or List of MapEntry the key will be put where this refers to, the value where the entry attribute refers to.			
list	xs:string			required
	The name of the field that contains the list to iterate over.			
name	xs:string			required
	A name for the section, used for external reference within the screen.			
paginate	xs:string		false	optional
	Indicate if this section is paginated or not, false by default.			
paginate-size	xs:string			optional

Element container

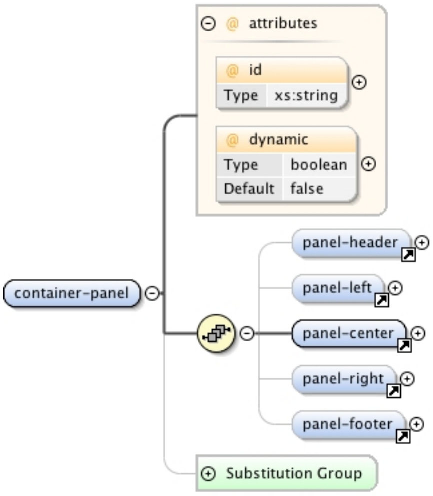
Namespace	No namespace

<table>
<tr><td>Diagram</td><td>
attributes
id
Type xs:string
style
Type xs:string
widget-elements
container
0..∞
AllWidgets
StandaloneFields
0..∞
##any
Substitution Group

</td></tr>
<tr><td>Instance</td><td>&lt;container id="" style="">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</container&gt;</td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>id</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>style</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
</table></td></tr>
</table>

Element `container-panel`

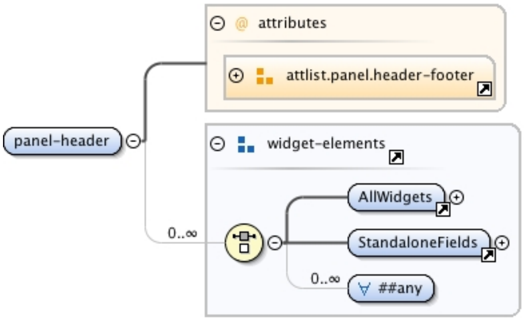
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td>This panel can have up to five areas: header, left, center, right, footer. Only the center area is required.
This can be re-used within the different areas as well, usually just the center area but could be used to
split up even the header and footer.

If there is an id for the outer container, and each area will have an automatic id as well (with a suffix
of: _header, _left, _center, _right, _footer).</td></tr>
<tr><td>Diagram</td><td>
attributes
id
Type xs:string
dynamic
Type boolean
Default false
container-panel
panel-header
panel-left
panel-center
panel-right
panel-footer
Substitution Group

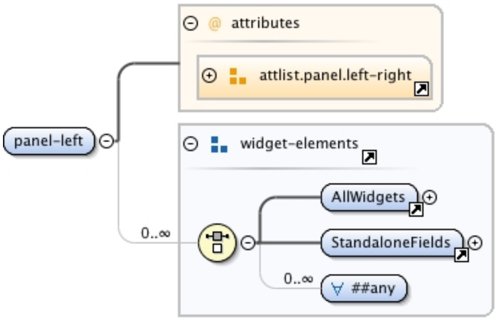
</td></tr>
<tr><td>Instance</td><td>&lt;container-panel dynamic="false" id="">
  <panel-header closable="true" resizable="false" size="auto" size-max="" size-min="" spacing="5">{0,1}</panel-header>
  <panel-left closable="true" resizable="true" size="180" size-max="" size-min="" spacing="5">{0,1}</panel-left>
  <panel-center>{1,1}</panel-center>
  <panel-right closable="true" resizable="true" size="180" size-max="" size-min="" spacing="5">{0,1}</panel-right>
  <panel-footer closable="true" resizable="false" size="auto" size-max="" size-min="" spacing="5">{0,1}</panel-footer>
</container-panel&gt;</td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>dynamic</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
</table></td></tr>
</table>

QName	Type	Fixed	Default	Use
	When true uses a dynamic layout, by default with jQuery Layout (see http://layout.jquery-dev.net/). When false (default) uses a static HTML/CSS layout.			
id	xs:string			optional

Element `panel-header`

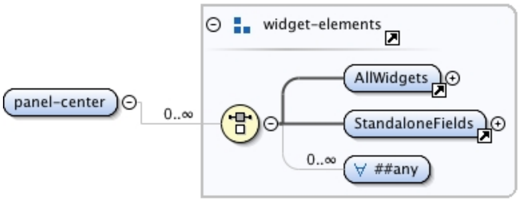
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Used by</td><td><table><tr><td>Element</td><td>container-panel</td></tr></table></td></tr>
<tr><td>Instance</td><td><pre>
<panel-header closable="true" resizable="false" size="auto" size-max="" size-
min="" spacing="5">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</panel-header>
</pre></td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>closable</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td><b>resizable</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
<tr><td><b>size</b></td><td>xs:string</td><td></td><td>auto</td><td>optional</td></tr>
<tr><td><b>size-max</b></td><td>xs:float</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>size-min</b></td><td>xs:float</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>spacing</b></td><td>xs:string</td><td></td><td>5</td><td>optional</td></tr>
</table></td></tr>
</table>

Element `panel-left`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Used by</td><td><table><tr><td>Element</td><td>container-panel</td></tr></table></td></tr>
<tr><td>Instance</td><td><pre>
<panel-left closable="true" resizable="true" size="180" size-max="" size-
min="" spacing="5">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</panel-left>
</pre></td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>closable</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td><b>resizable</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td><b>size</b></td><td>xs:string</td><td></td><td>180</td><td>optional</td></tr>
</table></td></tr>
</table>

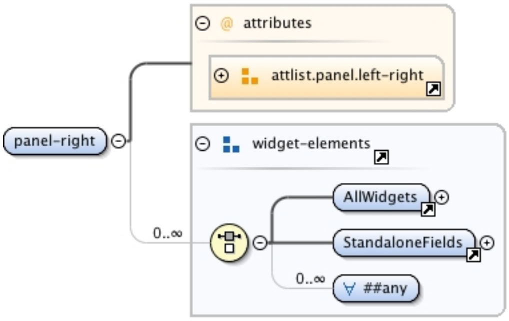
QName	Type	Fixed	Default	Use
size-max	xs:float			optional
size-min	xs:float			optional
spacing	xs:string		5	optional

Element panel-center

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
panel-center
0..∞
widget-elements
AllWidgets
StandaloneFields
0..∞
##any

</td></tr>
<tr><td>Used by</td><td>Element container-panel</td></tr>
<tr><td>Instance</td><td>&lt;panel-center>
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</panel-center&gt;</td></tr>
</table>

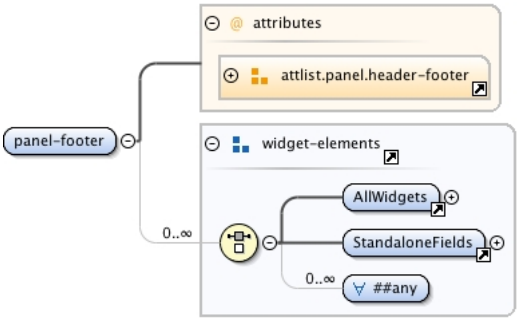
Element panel-right

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
attributes
attlist.panel.left-right
panel-right
0..∞
widget-elements
AllWidgets
StandaloneFields
0..∞
##any

</td></tr>
<tr><td>Used by</td><td>Element container-panel</td></tr>
<tr><td>Instance</td><td>&lt;panel-right closable="true" resizable="true" size="180" size-max="" size-min="" spacing="5">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</panel-right&gt;</td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>closable</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td><b>resizable</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td><b>size</b></td><td>xs:string</td><td></td><td>180</td><td>optional</td></tr>
<tr><td><b>size-max</b></td><td>xs:float</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>size-min</b></td><td>xs:float</td><td></td><td></td><td>optional</td></tr>
<tr><td><b>spacing</b></td><td>xs:string</td><td></td><td>5</td><td>optional</td></tr>
</table>
</td></tr>
</table>

Element panel-footer

Namespace	No namespace

Diagram	 attributes attlist.panel.header-footer panel-footer widget-elements AllWidgets StandaloneFields 0..∞ 0..∞ ##any
Used by	Element container-panel

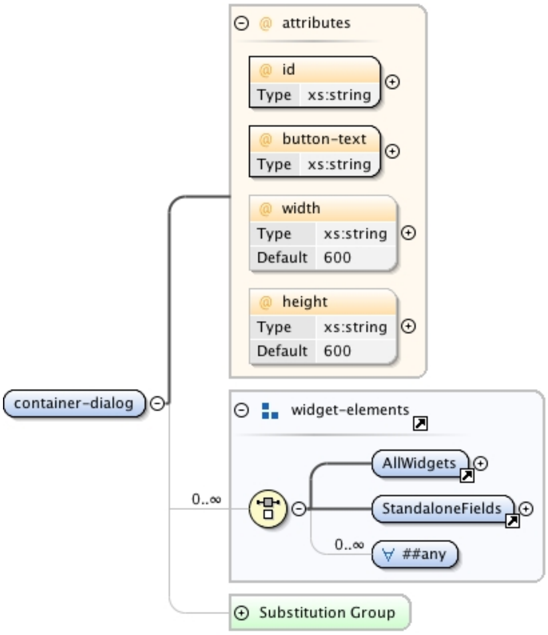
Instance

```
<panel-footer closable="true" resizable="false" size="auto" size-max="" size-
min="" spacing="5">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</panel-footer>
```

Attributes

QName	Type	Fixed	Default	Use
closable	boolean		true	optional
resizable	boolean		false	optional
size	xs:string		auto	optional
size-max	xs:float			optional
size-min	xs:float			optional
spacing	xs:string		5	optional

Element `container-dialog`

Namespace	No namespace
Annotations	The contents start out hidden with only a button with the button-text on it. When the button is clicked on a dialog opens to show the contents.
Diagram	 attributes id Type xs:string button-text Type xs:string width Type xs:string Default 600 height Type xs:string Default 600 container-dialog widget-elements AllWidgets StandaloneFields 0..∞ 0..∞ ##any Substitution Group

Instance

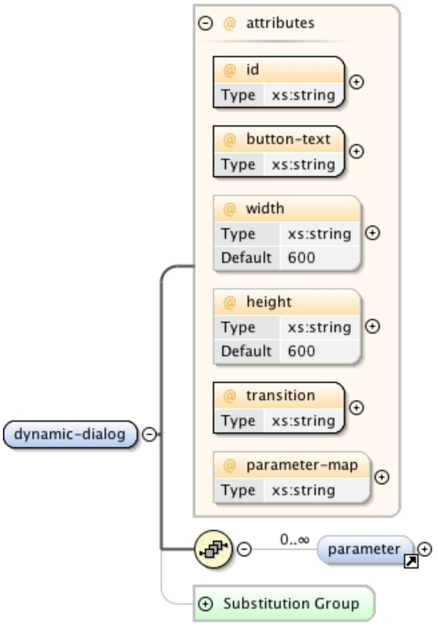
```
<container-dialog button-text="" height="600" id="" width="600">
  <AllWidgets>{1,1}</AllWidgets>
  <StandaloneFields>{1,1}</StandaloneFields>
</container-dialog>
```

Attributes

QName	Type	Fixed	Default	Use
button-text	xs:string			required
height	xs:string		600	optional
id	xs:string			required

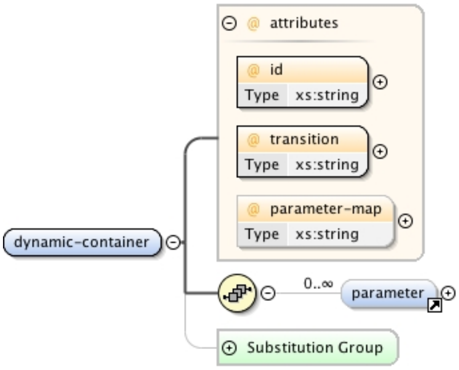
QName	Type	Fixed	Default	Use
width	xs:string		600	optional

Element `dynamic-dialog`

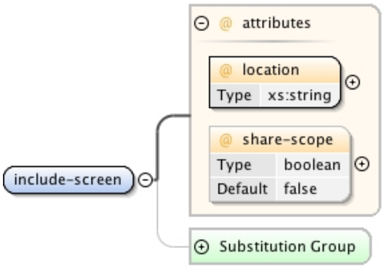
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td><pre>When the button is pressed the dialog contents are loaded from the server at
the given transition.</pre></td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Instance</td><td><pre>&lt;dynamic-dialog button-text="" height="600" id="" parameter-
map="" transition="" width="600">
  <parameter from="" name="" value="">{0,unbounded}</parameter>
</dynamic-dialog&gt;</pre></td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>button-text</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>height</b></td><td>xs:string</td><td></td><td>600</td><td>optional</td></tr>
<tr><td><b>id</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>parameter-map</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>A Map to get parameter
names and values from in addition to the parameter sub-elements.</pre></td></tr>
<tr><td><b>transition</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>width</b></td><td>xs:string</td><td></td><td>600</td><td>optional</td></tr>
</table>
</td></tr>
</table>

Element `dynamic-container`

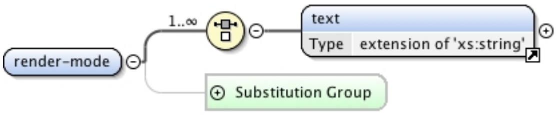
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Annotations</td><td><pre>Container contents are immediately loaded from the server at the given
transition. Contents may be reloaded based on other actions such as background form
 submissions.</pre></td></tr>
</table>

<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Instance</td><td><dynamic-container id="" parameter-map="" transition="">
  <parameter from="" name="" value="">{0,unbounded}</parameter>
</dynamic-container></td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>id</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>parameter-map</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td></td><td colspan="4">A Map to get parameter
names and values from in addition to the parameter sub-elements.</td></tr>
<tr><td><b>transition</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
</table>
</td></tr>
</table>

Element `include-screen`

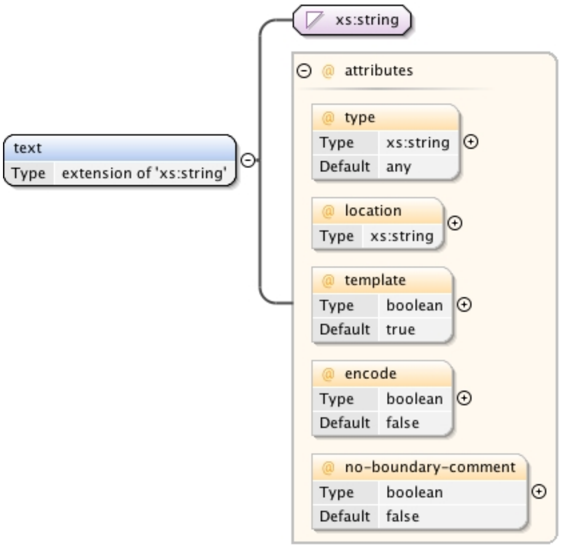
<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>location</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td><b>share-scope</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
</table>
</td></tr>
</table>

Element `render-mode`

<table>
<tr><td>Namespace</td><td>No namespace</td></tr>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Instance</td><td><render-mode>
  <text encode="false" location="" no-boundary-comment="false" template="true" type="any">{1,1}</text>
</render-mode></td></tr>
</table>

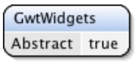
Element `text`

Namespace	No namespace

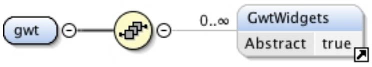
<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Type</td><td>extension of xs:string</td></tr>
<tr><td>Used by</td><td><table><tr><td>Element</td><td>render-mode</td></tr></table></td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>encode</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>If true text will be encoded so that it does not interfere with markup
 of the target output.
Templates ignore this setting and are never encoded.

For example, if output is HTML then data presented will be HTML encoded
 so that all
HTML-specific characters are escaped.</pre></td></tr>
<tr><td><b>location</b></td><td>xs:string</td><td></td><td></td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>Template location can include various prefixes to support including
 from a file, http, component,
or a content repository.</pre></td></tr>
<tr><td><b>no-boundary-comment</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>Don't ever put boundary comments before this (for opening ?xml tag,
 etc).</pre></td></tr>
<tr><td><b>template</b></td><td>boolean</td><td></td><td>true</td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>Interpret the text at the location as an FTL template?
Defaults to true, set to false if you want the text included literally.</pre></td></tr>
<tr><td><b>type</b></td><td>xs:string</td><td></td><td>any</td><td>optional</td></tr>
<tr><td></td><td colspan="4"><pre>Can be anything. Default supported values include: text, cwiki, html,
 xsl-fo, xml, and csv.
A value of "any" will cause it to be used if no other element matches
 the current output type.</pre></td></tr>
</table></td></tr>
</table>

Element `GwtWidgets`

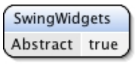
Namespace	No namespace
Diagram	
Used by	Element gwt

Element `gwt`

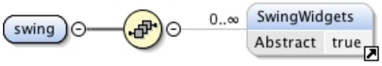
Namespace	No namespace
Diagram	
`Instance`	`<gwt>`

	<GwtWidgets>{0,unbounded}</GwtWidgets> </gwt>

Element SwingWidgets

Namespace	No namespace
Diagram	
Used by	Element swing

Element swing

Namespace	No namespace
Diagram	
Instance	<swing> <SwingWidgets>{0,unbounded}</SwingWidgets> </swing>