

Schema documentation for xml-screen-1.1.xsd

Publication date 7 august 2012

Table of Contents

| | |
|--------------------------------|----|
| Namespace: "" | 1 |
| Schemas | 1 |
| Main schema xml-screen-1.1.xsd | 1 |
| Elements | 1 |
| Element widgets | 1 |
| Element fail-widgets | 2 |
| Element screen | 2 |
| Element macro-template | 4 |
| Element web-settings | 5 |
| Element transition | 5 |
| Element path-parameter | 6 |
| Element conditional-response | 6 |
| Element default-response | 7 |
| Element error-response | 8 |
| Element subscreens | 9 |
| Element conditional-default | 9 |
| Element subscreens-item | 10 |
| Element pre-actions | 11 |
| Element subscreens-menu | 12 |
| Element subscreens-active | 12 |
| Element subscreens-panel | 12 |
| Element section | 12 |
| Element section-iterate | 13 |
| Element container | 14 |
| Element container-panel | 14 |
| Element panel-header | 15 |
| Element panel-left | 15 |
| Element panel-center | 16 |
| Element panel-right | 16 |
| Element panel-footer | 16 |
| Element container-dialog | 17 |
| Element include-screen | 18 |
| Element render-mode | 18 |
| Element text | 18 |
| Element GwtWidgets | 19 |
| Element gwt | 19 |
| Element SwingWidgets | 19 |
| Element swing | 19 |

Namespace: ""

Schemas

Main schema xml-screen-1.1.xsd

| | |
|------------|-------------------------------------|
| Namespace | No namespace |
| Properties | attribute form default: unqualified |
| | element form default: qualified |

Elements

Element widgets

| | |
|-----------|--------------|
| Namespace | No namespace |
|-----------|--------------|

| | |
|------------|---|
| Diagram | |
| Properties | content: complex |
| Used by | Element Group section-elements |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><widgets> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </widgets></pre> |

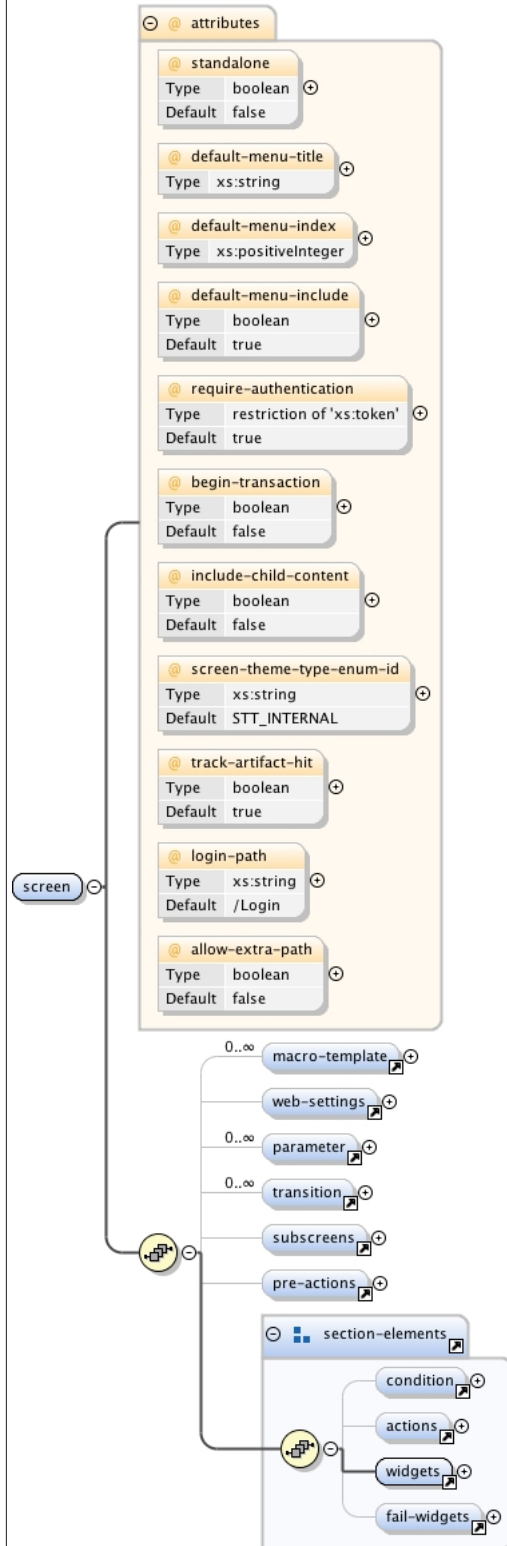
Element fail-widgets

| | |
|------------|---|
| Namespace | No namespace |
| Diagram | |
| Properties | content: complex |
| Used by | Element Group section-elements |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><fail-widgets> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </fail-widgets></pre> |

Element screen

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

Diagram



| | |
|------------|---|
| Properties | content: complex |
| Model | macro-template*, web-settings{0,1}, parameter*, transition*, subscreens{0,1}, pre-actions{0,1}, condition{0,1}, actions{0,1}, widgets, fail-widgets{0,1} |
| Children | actions, condition, fail-widgets, macro-template, parameter, pre-actions, subscreens, transition, web-settings, widgets |
| Instance | <pre> <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> </pre> |

| | | | | | |
|------------|--|---|--------------|----------------|------------|
| | <pre> <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> <transition method="any" name=" ">{0,unbounded}</transition> <subscreens default-item=" ">{0,1}</subscreens> <pre-actions>{0,1}</pre-actions> </screen> </pre> | | | | |
| 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:positiveInteger | | | 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 | | |
| Properties | content: | complex |

| | | | | | |
|------------|--|-------------|--------------|----------------|------------|
| Used by | Element screen | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | location | xs:string | | | required |
| | type | xs:string | | | required |
| | Can be anything. Default supported values include: text, html, xsl-fo, xml, and csv. | | | | |

Element web-settings

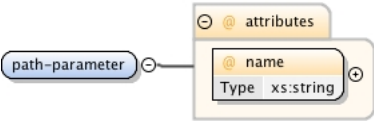
| | | | | | |
|------------|--------------------|-----------|-------|-----------|----------|
| Namespace | No namespace | | | | |
| Diagram | | | | | |
| Properties | content: | complex | | | |
| Used by | Element | screen | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | allow-web-request | boolean | | true | optional |
| | character-encoding | xs:string | | UTF-8 | optional |
| | mime-type | xs:string | | text/html | optional |
| | require-encryption | boolean | | true | optional |

Element transition

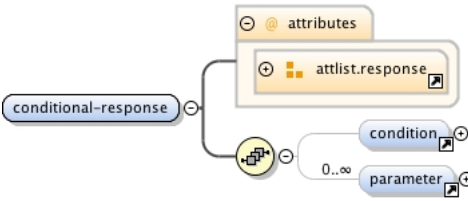
| | | | | | |
|-------------|--------------|---------|--|--|--|
| Namespace | No namespace | | | | |
| Annotations | | | | | |
| Diagram | | | | | |
| Properties | content: | complex | | | |

| | | | | | |
|------------|---|-------------------------|-------|---------|----------|
| Used by | Element screen | | | | |
| Model | path-parameter*, condition{0,1}, (service-call actions), conditional-response*, default-response, error-response{0,1} | | | | |
| Children | actions, condition, conditional-response, default-response, error-response, path-parameter, service-call | | | | |
| Instance | <pre> <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> </pre> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | method | restriction of xs:token | | any | optional |
| | name | xs:string | | | required |
| | <p>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).</p> <p>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.</p> | | | | |

Element path-parameter

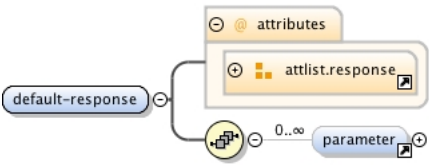
| | | | | | |
|------------|---|------------|-------|---------|----------|
| Namespace | No namespace | | | | |
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Used by | Element | transition | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | name | xs:string | | | required |

Element conditional-response

| | |
|------------|--|
| Namespace | No namespace |
| Diagram |  |
| Properties | content: complex |
| Used by | Element transition |
| Model | condition{0,1} , parameter* |
| Children | condition, parameter |
| Instance | <pre><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></pre> |

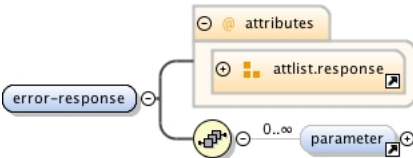
| Attributes | QName | Type | Fixed | Default | Use |
|------------|---|-------------------------|-------|-------------|----------|
| | parameter-map | xs:string | | | optional |
| | save-current-screen | boolean | | false | optional |
| | Save the current screen's path and parameters for future use, 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 |
| | <p>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).</p> <p>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).</p> <p>If the screen-path type url starts with a "/" it will be relative to the root screen instead of relative to the current screen.</p> <p>If the url-type is "plain" then this can be any valid URL (relative on current server or absolute).</p> | | | | |
| | url-type | restriction of xs:token | | screen-path | optional |

Element default-response

| | | | | | |
|------------|---|---|-------|---------|----------|
| Namespace | No namespace | | | | |
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Used by | Element | transition | | | |
| Model | parameter* | | | | |
| Children | parameter | | | | |
| Instance | <pre><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></pre> | | | | |
| 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 | | | |

| QName | Type | Fixed | Default | Use |
|-----------------|---|-------|-------------|----------|
| | <p>the value here is a path from the current screen to the desired screen (or transition).</p> <p>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).</p> <p>If the screen-path type url starts with a "/" it will be relative to the root screen instead of relative to the current screen.</p> <p>If the url-type is "plain" then this can be any valid URL (relative on current server or absolute).</p> | | | |
| url-type | restriction of xs:token | | screen-path | optional |

Element error-response

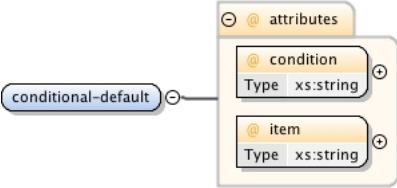
| | | | | | |
|------------|---|--|-------|-------------|----------|
| Namespace | No namespace | | | | |
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Used by | Element | transition | | | |
| Model | parameter* | | | | |
| Children | parameter | | | | |
| Instance | <pre><error-response 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></pre> | | | | |
| 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 subscreens

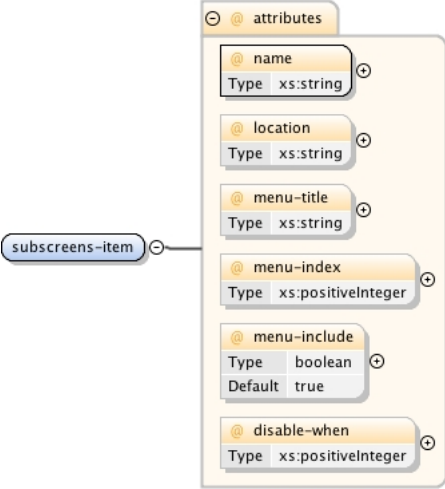
| | | | | | |
|-------------|---|---|-------|---------|----------|
| Namespace | No namespace | | | | |
| Annotations | <p>Declare subscreens for this screen. One subscreen at a time is active, based on the "screen path" used to access this screen. The parent screen (this screen) will be the current element in the screen path and the next screen path element will be the name of the subscreen of this screen to use.</p> <p>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.</p> <p>There are three ways to add subscreens to a screen:</p> <ol style="list-style-type: none">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)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)for adding screens, removing screens, or changing order and title of screens to an existing application: a record in the moqui.screen.SubscreensItem entity <p>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.</p> | | | | |
| Diagram | <pre>graph LR subgraph subscreens [subscreens] direction TB subgraph attributes [attributes] default-item["@ default-item Type xs:string"] end conditional-default["conditional-default"] subscreens-item["subscreens-item"] end conditional-default --- subscreens subscreens-item --- subscreens</pre> | | | | |
| Properties | content: | complex | | | |
| Used by | Element | screen | | | |
| Model | conditional-default*, subscreens-item* | | | | |
| Children | conditional-default, subscreens-item | | | | |
| Instance | <pre><subscreens 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></pre> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | default-item | xs:string | | | optional |
| | | The name of the default subscreen-item. Used when then screen-path ends on this screen so we know which subscreen-item to activate. If empty the first subscreen-item will be the default. | | | |

Element conditional-default

| | |
|-----------|--------------|
| Namespace | No namespace |
|-----------|--------------|

| Diagram |  | | | | |
|------------|---|------------|-------|---------|----------|
| Properties | content: | complex | | | |
| 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 | <p>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.</p> <p>One subscreens-item is active at a time, meaning that screen is shown and the tab/etc for that screen is highlighted.</p> | | | | |
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| 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 |
| | <p>Subscreen location can include various prefixes to support including from a file, http, component, or a content repository.</p> <p>If empty defaults to the value of the name attribute under the current screen (in the directory</p> | | | | |

| QName | Type | Fixed | Default | Use |
|---------------------|--|-------|---------|----------|
| | 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

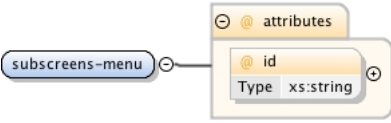
| | |
|-------------|--|
| Namespace | No namespace |
| Annotations | These actions run before any of the screens (this screen or any parent screens) are rendered, allowing you to set parameters used by parent screens or other general reasons. |
| Diagram | |
| Properties | content: complex |
| Used by | Element screen |
| Model | CallOperations EnvOperations EntityMiscOperations EntityFindOperations EntityValueOperations EntityListOperations ControlOperations IfBasicOperations IfOtherOperations OtherOperations ANY element from ANY namespace |
| Children | CallOperations, ControlOperations, EntityFindOperations, EntityListOperations, EntityMiscOperations, EntityValueOperations, EnvOperations, IfBasicOperations, IfOtherOperations, OtherOperations |
| Instance | <pre><pre-actions> <CallOperations>{1,1}</CallOperations></pre> |

```

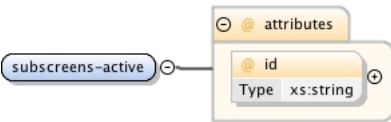
<EnvOperations>{1,1}</EnvOperations>
<EntityMiscOperations>{1,1}</EntityMiscOperations>
<EntityFindOperations>{1,1}</EntityFindOperations>
<EntityValueOperations>{1,1}</EntityValueOperations>
<EntityListOperations>{1,1}</EntityListOperations>
<ControlOperations>{1,1}</ControlOperations>
<IfBasicOperations>{1,1}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

```

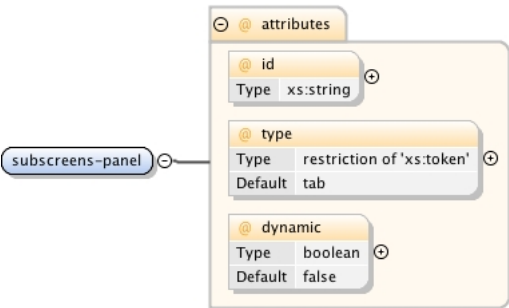
Element subscreens-menu

| Namespace | No namespace | | | | |
|------------|---|-----------|-------|---------|----------|
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | id | xs:string | | | optional |

Element subscreens-active

| Namespace | No namespace | | | | |
|------------|--|-----------|-------|---------|----------|
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | id | xs:string | | | optional |

Element subscreens-panel

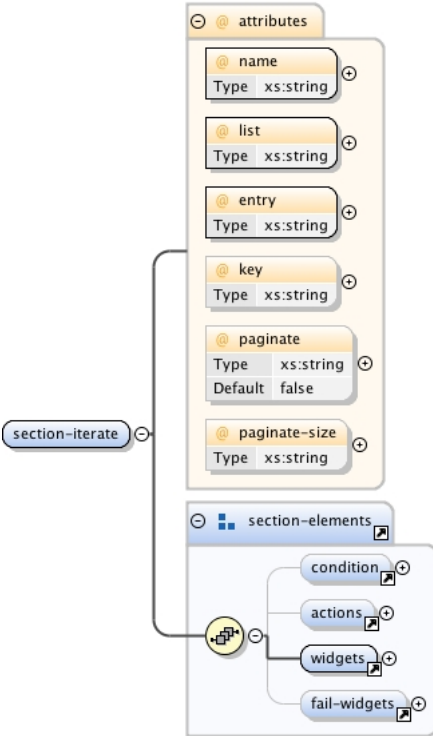
| Namespace | No namespace | | | | |
|------------|---|-------------------------|-------|---------|----------|
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | dynamic | boolean | | false | optional |
| | id | xs:string | | | optional |
| | type | restriction of xs:token | | tab | optional |

Element section

| | | | | | |
|-----------|--------------|--|--|--|--|
| Namespace | No namespace | | | | |
|-----------|--------------|--|--|--|--|

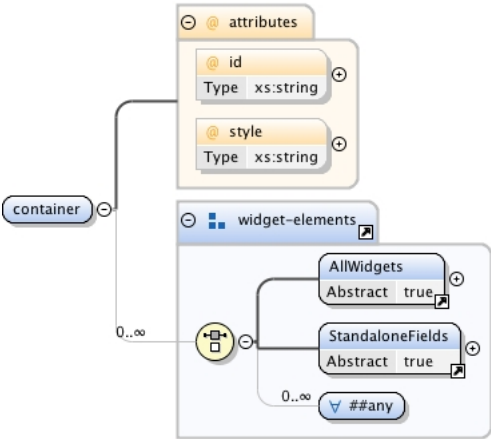
| | | | | | |
|------------|---|-----------|-------|---------|----------|
| Diagram | | | | | |
| Properties | content: | complex | | | |
| Model | condition{0,1} , actions{0,1} , widgets , fail-widgets{0,1} | | | | |
| Children | actions, condition, fail-widgets, widgets | | | | |
| Instance | <pre><section name=""> <condition>{0,1}</condition> <actions>{0,1}</actions> <widgets>{1,1}</widgets> <fail-widgets>{0,1}</fail-widgets> </section></pre> | | | | |
| 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 |  |
| Properties | content: complex |
| Model | condition{0,1} , actions{0,1} , widgets , fail-widgets{0,1} |
| Children | actions, condition, fail-widgets, widgets |
| Instance | <pre><section-iterate entry="" key="" list="" name="" paginate="false" paginate-size=""> <condition>{0,1}</condition> </section-iterate></pre> |

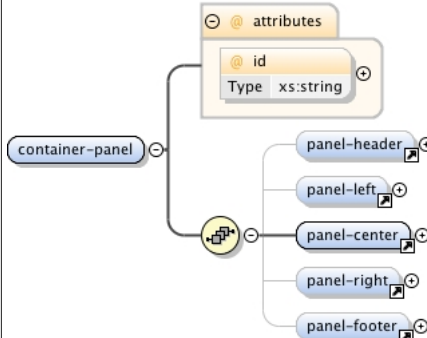
| | | | | | |
|------------|--|--|--------------|----------------|------------|
| | <code><actions>{0,1}</actions></code> <code><widgets>{1,1}</widgets></code> <code><fail-widgets>{0,1}</fail-widgets></code> <code></section-iterate></code> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | 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

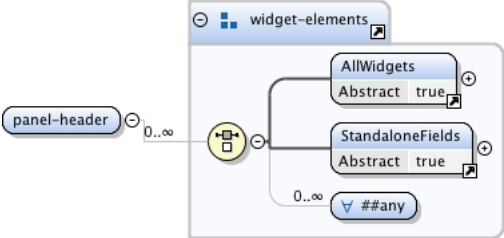
| | | | | | |
|------------|--|-----------|-------|---------|----------|
| Namespace | No namespace | | | | |
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace | | | | |
| Children | AllWidgets, StandaloneFields | | | | |
| Instance | <pre><container id="" style=""> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </container></pre> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | id | xs:string | | | optional |
| | style | xs:string | | | optional |

Element container-panel

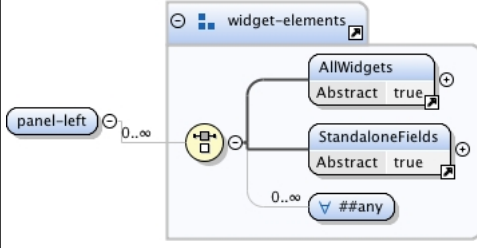
| | | | | | |
|-------------|--|--|--|--|--|
| Namespace | No namespace | | | | |
| Annotations | <p>This panel can have up to five areas: header, left, center, right, footer. Only the center area is required.</p> <p>This can be re-used within the different areas as well, usually just the center area but could be used to split up even the header and footer.</p> <p>If there is an id for the outer container, and each area will have an automatic id as well (with a suffix of: <code>_header</code>, <code>_left</code>, <code>_center</code>, <code>_right</code>, <code>_footer</code>).</p> | | | | |

| | | | | | |
|------------|--|-----------|-------|---------|----------|
| Diagram |  | | | | |
| Properties | content: complex | | | | |
| Model | panel-header{0,1} , panel-left{0,1} , panel-center , panel-right{0,1} , panel-footer{0,1} | | | | |
| Children | panel-center, panel-footer, panel-header, panel-left, panel-right | | | | |
| Instance | <pre><container-panel id=""> <panel-header>{0,1}</panel-header> <panel-left>{0,1}</panel-left> <panel-center>{1,1}</panel-center> <panel-right>{0,1}</panel-right> <panel-footer>{0,1}</panel-footer> </container-panel></pre> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | id | xs:string | | | optional |

Element panel-header

| | |
|------------|---|
| Namespace | No namespace |
| Diagram |  |
| Properties | content: complex |
| Used by | Element container-panel |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><panel-header> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </panel-header></pre> |

Element panel-left

| | | | | | |
|------------|--|---------|--|--|--|
| Namespace | No namespace | | | | |
| Diagram |  <p>The diagram shows the structure of the <code>panel-left</code> element. It has a cardinality of <code>0..∞</code>. The content is a complex type consisting of a sequence of elements: <code>AllWidgets</code>, <code>StandaloneFields</code>, and <code>##any</code>. Each of these elements is optional (indicated by a circle with a plus sign).</p> | | | | |
| Properties | content: | complex | | | |

| | |
|----------|---|
| Used by | Element container-panel |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><panel-left> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </panel-left></pre> |

Element panel-center

| | |
|------------|---|
| Namespace | No namespace |
| Diagram | |
| Properties | content: complex |
| Used by | Element container-panel |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><panel-center> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </panel-center></pre> |

Element panel-right

| | |
|------------|---|
| Namespace | No namespace |
| Diagram | |
| Properties | content: complex |
| Used by | Element container-panel |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre><panel-right> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </panel-right></pre> |

Element panel-footer

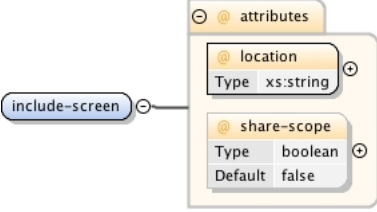
| | |
|-----------|--------------|
| Namespace | No namespace |
|-----------|--------------|

| | |
|------------|---|
| Diagram | |
| Properties | content: complex |
| Used by | Element container-panel |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace |
| Children | AllWidgets, StandaloneFields |
| Instance | <pre> <panel-footer> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </panel-footer> </pre> |

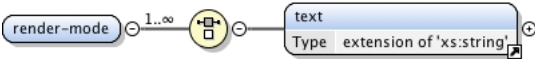
Element container-dialog

| | | | | | |
|-------------|--|-----------|-------|---------|----------|
| 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 | <p>The diagram illustrates the structure of the <code>container-dialog</code> element. It is a container element (indicated by a yellow circle icon) with a cardinality of <code>0..∞</code>. It contains two main sections: <code>attributes</code> and <code>widget-elements</code>. The <code>attributes</code> section includes four attributes: <code>id</code> (Type <code>xs:string</code>), <code>button-text</code> (Type <code>xs:string</code>), <code>width</code> (Type <code>xs:string</code>, Default <code>600</code>), and <code>height</code> (Type <code>xs:string</code>, Default <code>600</code>). The <code>widget-elements</code> section contains three elements: <code>AllWidgets</code> (Abstract <code>true</code>), <code>StandaloneFields</code> (Abstract <code>true</code>), and <code>##any</code> (cardinality <code>0..∞</code>).</p> | | | | |
| Properties | content: | complex | | | |
| Model | AllWidgets StandaloneFields ANY element from ANY namespace | | | | |
| Children | AllWidgets, StandaloneFields | | | | |
| Instance | <pre><container-dialog button-text="" height="600" id="" width="600"> <AllWidgets>{1,1}</AllWidgets> <StandaloneFields>{1,1}</StandaloneFields> </container-dialog></pre> | | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | button-text | xs:string | | | required |
| | height | xs:string | | 600 | optional |
| | id | xs:string | | | required |
| | width | xs:string | | 600 | optional |

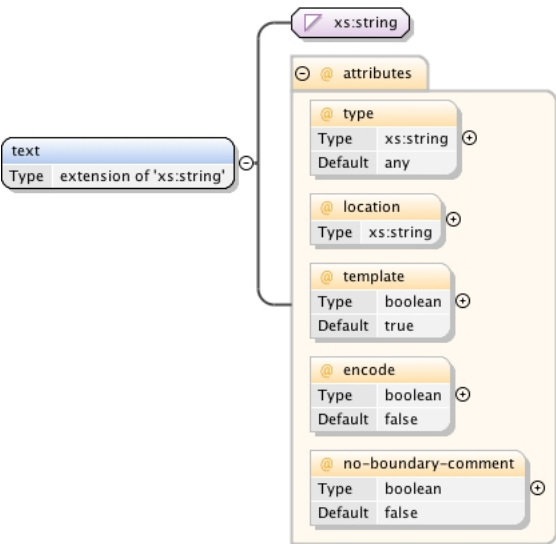
Element include-screen

| Namespace | No namespace | | | | |
|------------|---|-----------|-------|---------|----------|
| Diagram |  | | | | |
| Properties | content: | complex | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | location | xs:string | | | required |
| | share-scope | boolean | | false | optional |

Element render-mode

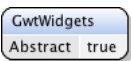
| | |
|------------|---|
| Namespace | No namespace |
| Diagram |  |
| Properties | content: complex |
| Model | text |
| Children | text |
| Instance | <pre><render-mode> <text encode="false" location="" no-boundary- comment="false" template="true" type="any">{1,1}</text> </render-mode></pre> |

Element text


| Namespace | No namespace | | | | |
|------------|---|-------------|-------|---------|----------|
| Diagram |  | | | | |
| Type | extension of xs:string | | | | |
| Properties | content: | complex | | | |
| Used by | Element | render-mode | | | |
| Attributes | QName | Type | Fixed | Default | Use |
| | encode | boolean | | false | optional |

| QName | Type | Fixed | Default | Use |
|----------------------------|---|-------|---------|----------|
| | <p>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.</p> <p>For example, if output is HTML then data presented will be HTML encoded so that all HTML-specific characters are escaped.</p> | | | |
| location | xs:string | | | optional |
| | Template location can include various prefixes to support including from a file, http, component, or a content repository. | | | |
| no-boundary-comment | boolean | | false | optional |
| | Don't ever put boundary comments before this (for opening ?xml tag, etc). | | | |
| template | boolean | | true | optional |
| | Interpret the text at the location as an FTL template? Defaults to true, set to false if you want the text included literally. | | | |
| type | xs:string | | any | optional |
| | 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. | | | |

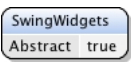
Element GwtWidgets

| | | |
|------------|--|------|
| Namespace | No namespace | |
| Diagram |  | |
| Properties | abstract: | true |
| Used by | Element | gwt |


Element gwt

| | | |
|------------|--|---------|
| Namespace | No namespace | |
| Diagram |  | |
| Properties | content: | complex |
| Model | GwtWidgets* | |
| Children | GwtWidgets | |
| Instance | <pre><gwt> <GwtWidgets>{0,unbounded}</GwtWidgets> </gwt></pre> | |

Element SwingWidgets

| | | |
|------------|---|-------|
| Namespace | No namespace | |
| Diagram |  | |
| Properties | abstract: | true |
| Used by | Element | swing |

Element swing

| | | |
|-----------|---|--|
| Namespace | No namespace | |
| Diagram |  | |

| | |
|------------|--|
| Properties | content: complex |
| Model | SwingWidgets* |
| Children | SwingWidgets |
| Instance | <pre><swing> <SwingWidgets>{0,unbounded}</SwingWidgets> </swing></pre> |