

This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. The article is believed to be still valid for the original topic scope.



## Contents

- [1 Template widget model: Syndication](#)
  - ◆ [1.1 Widget views](#)
  - ◆ [1.2 Scripted elements](#)
  - ◆ [1.3 Static styles](#)
  - ◆ [1.4 Widget services](#)
  - ◆ [1.5 Widget resources](#)
  - ◆ [1.6 Widget parameters](#)
- [2 See also](#)

## Template widget model: Syndication

The syndication script, the syndication service, and widget skins are used to create syndication widgets from RSS and Atom feeds. This section explains how the syndication script works, and how you can create custom skins.

A generic syndication script uses server-side services to synchronize feed items every time a client goes online. An online server notifies the client when a new item becomes available, and a script downloads the new item automatically. The script saves synchronized and loaded content on the phone so it can also be accessed offline. Note that only the first 500 characters from each item are downloaded automatically. Images are downloaded only by request.

## Widget views

A syndication widget consists of four views:

- **Minimized view "viewIndex"**  
Normally displaying the icon of the feed viewed by the widget.



- **Minimized view "viewNew"**  
Viewed when there are unread items in the feed. The WidSets framework automatically places a green plus-sign at the bottom right corner of the minimized view when the widget has new feed items.



- **List view "viewList"**

Displays a list of the newest items. In this view, you can mark messages as read.



- **Item view "viewItem"**

Displays an individual feed item, along with pictures if they have been enabled in the widget options. If the feed contains enclosures such as attached picture links, they are rendered between the subject and the first paragraph.



Inline HTML `<img>`-tagged images are rendered at the same place where they exist in the description content.

Only 500 characters of each feed item's content is loaded in the initial synchronization. As the user scrolls down, more content is loaded. This is done to optimize fast content delivery, and to minimize unnecessary use of resources.

In the maximized view, you can save the feed item link as *bookmark*, or send the link to a friend as *email*.

## Scripted elements

The following scripted elements are available to be used in [widget.xml](#):

- **feedItemList**

Creates a list of items displaying subject for the "viewList" view.

- **feedItem**

Creates an item flow for the "viewItem" view. This flow consists of *date* (Text component), *subject* (Text component), *author* (Text) if available, attached images that are not inline HTML, and the actual content.

## Syndication\_widgets

- **feedTitle**  
Creates a *Label* component containing the parameter *title* or *widgetname* as content. Used in minimized views. Deprecated, use `<label>${widgetname}</label>` to display feed names.
- **newestItem**  
Creates a *Text* component displaying the subject of the newest feed item. Used in minimized views.
- **feedNewCount**  
Creates a *Label* component displaying the number of unread (or old) items the feed currently has. Used in minimized views.
- **feedNewCountNumber**  
Same as above, but displays only the number.

## Static styles

The following styles are used by the syndication widget:

- **feedItem**  
Item subject (read) in the list view.
- **feedItemNew**  
Item subject (unread) in the list view.
- **subject**  
Subject field in the item view.
- **date**  
Date field in the item view. If not defined, the subject style is used.
- **author**  
Author in the item view. If not defined, the subject style is used.
- **content**  
Text content in the item view. Note that the item text may contain more than one *Text* component, so if you define text decoration in the beginning, it may appear more than once.
- **image**  
Used on attached images.
- **input.choice**  
Used for the "Send as email" text field.

## Widget services

To create a syndication widget, set up the following services in [widget.xml](#):

```
<services>
  <service type="syndication" id="feed1">
    <reference from="feedurlrss" to="feedurl"/>
  </service>
  <service type="picviewer" id="widgetImages"/>
</services>
```

- Picviewer service is not mandatory if your feed does not contain images, or if you do not want to display images within the items.

## Widget resources

Templated syndication widget will use Syndication template-class to handle the widget logic, so instead of having code-resource the widget needs to have a class-resource:

```
<resources>
  <class src="Syndication"/>
</resources>
```

- If your widget has img- or stylesheet-resources you'll need to defined them in this same resources-section.

## Widget parameters

You also need to set up the following parameters in widget.xml:

```
<parameters>
  <parameter type="string"
    name="widgetname"
    description="Name of the widget "
    editable="false"/>
  <parameter type="url"
    name="feedurlrss"
    description="Feed URL"
    editable="true">
    http://www.examplesite.com/yourfeed.xml
  </parameter>
  <parameter type="number"
    name="alert.sound"
    description="Alert sound"
    editable="true"/>
  <parameter type="number"
    name="alert.volume"
    description="Alert volume"
    editable="true"/>
  <parameter type="string"
    name="show.images"
    description="Show images"
    editable="true">
    <value>
      <option value="yes" selected="true">Yes</option>
      <option value="no">No</option>
      <option value="placeholder">Placeholders</option>
    </value>
  </parameter>
  <parameter type="string"
    name="wide.images"
    editable="true">
    <value>
      <option value="true" selected="true">Horizontal</option>
      <option value="false">Vertical</option>
    </value>
  </parameter>
</parameters>
```

- **alert.sound** (number) and **alert.volume** (number)

These parameters are not mandatory. If they are specified, the user can change the alert sound played

## Syndication\_widgets

when new content arrives. The parameters do not need to be bound to any value, and default to no sound and zero volume.

- **wide.images** (string)  
If this parameter is set to "true", the item view will be laid out horizontally. This is ideal for feeds that have only one wide image as content, for example cartoons.
- **show.images** (string)  
A parameter that controls the image viewing capabilities in the RSS reader widgets. The possible values are "placeholder", "yes", and "no".
- **open.browser.link** (string)  
If defined but empty, a "Open in browser" link will be added at the end of each RSS item. Clicking this link will open browser to RSS items url (if it has one). If value is not empty the text will be used as link text.
- **max.items** (string)  
Number of RSS items to be shown by the widget.
- **image.minwidth** (number)  
Minimum width for embedded images. If images are smaller they will not be shown. (Many RSS feeds contain basic set of image links for some actions or visit-counts that can be this way skipped). Default is 10.
- **image.minheight** (number)  
Minimum height for embedded images. If images are smaller they will not be shown. (Many RSS feeds contain basic set of image links for some actions or visit-counts that can be this way skipped). Default is 10.
- **image.minsize** (number)  
Minimum size for embedded images. If images are smaller they will not be shown. (Many RSS feeds contain basic set of image links for some actions or visit-counts that can be this way skipped). Default is 10.
- **title** (string)  
Title text shown in "feedTitle" element in minimized view instead of widget name.
- **use.banner** (boolean)  
If "true", banners will be shown in the beginning of feed item list.

## See also

- [Syndication widgets](#)
- [PicViewer widgets](#)