

Tradução de [How to package Flash content in a Widget](#)

Um dos maiores problemas na distribuição de conteúdo [Flash Lite](#) é o fato do usuário não ter um método direto para utilização do aplicativo, sendo necessário abrir o [Flash Lite Player](#) ou o gerenciador de arquivos ou até mesmo a galeria para encontrar tal conteúdo, assim não sendo tratado como as demais aplicações que dispõem de ícone e a opção para adicionar atalhos de acesso rápido.

Outra desvantagem é que, caso você tenha recursos externos, será necessário disponibilizá-los separadamente, tornando o processo tedioso para os usuários. Nos dispositivos [s60](#) isso pode ser manipulado com a criação de um arquivo SIS que simplesmente extrai o conteúdo para a pasta correta. Existem algumas questões relacionadas a esta abordagem. O arquivo sis requer passar pelo processo de assinatura e a pasta alvo nem sempre pode ser a mesma nas diferentes versões da plataforma s60.

## Contents

- [1 Por que simplesmente não utilizar o .sis?](#)
- [2 Widgets são a solução!](#)
- [3 Ok. Pronto! Mostre-me como.](#)
  - ◆ [3.1 Modifique o ícone](#)
  - ◆ [3.2 Modifique o info.plist](#)
  - ◆ [3.3 O HTML e JavaScript](#)
    - ◇ [3.3.1 Referência CSS](#)
    - ◇ [3.3.2 JavaScript](#)
    - ◇ [3.3.3 Anexando o .swf](#)
  - ◆ [3.4 Flash Lite App](#)
  - ◆ [3.5 Criando o pacote Widget](#)
- [4 Testando o Widget](#)
- [5 Ajuda](#)

## Por que simplesmente não utilizar o .sis?

Existem opções comerciais, livres e guias para fazer isso com [Symbian C++](#), porém todas requerem no mínimo um computador rodando o Windows e alguns mínimos conhecimentos com a codificação em [Symbian C++](#). Muitos desenvolvedores na plataforma Flash trabalham com computadores Macintosh, não sendo possível instalar o s60 SDK e outras ferramentas necessárias.

## Widgets são a solução!

A Nokia anunciou uma nova plataforma para [widgets](#) ? [s60 Web Runtime](#). Com o WRT é extremamente simples desenvolver aplicações sem a necessidade de aprender [Symbian C++](#). Tudo que você precisa é o conhecimento básico de [HTML](#), [CSS](#) e [JavaScript](#). O conteúdo é zipado para um pacote com a extensão [.wgz](#). Este pacote não exige passar pelo processo de assinatura e é reconhecido como um aplicativo instalável pelos

## Como empacotar conteúdo do Flash em um Widget

dispositivos que suportam a tecnologia. Todo conteúdo dentro do pacote é copiado para uma pasta privada e um ícone para o início do Widget é posto na pasta de aplicações. Sendo o Web Runtime baseado no browser da plataforma s60, tudo suportado pelo browser é também suportado pelo WRT ? incluindo Flash Lite! Utilizando essa facilidade nós estamos aptos a empacotar nosso conteúdo Flash Lite dentro de um Widget.

## Ok. Pronto! Mostre-me como.

Antes de começar, é uma boa idéia ler um pouco sobre o [Web Runtime](#). O melhor lugar para começar é através do endereço do [Widget](#). Você deverá ser capaz de criar um pacote com as instruções abaixo, mas se você desejar se aprofundar no desenvolvimento de Widgets, siga o link abaixo:

- Faça o download do [exemplo](#)
  - ◆ O Link para download direciona para o sistema de compartilhamento [Mosh](#), a rede social da Nokia.
  - ◆ Recomenda-se o uso do Mosh para envio dos seus exemplos de widgets.
- Altere o nome do arquivo para FlashWidget.zip
  - ◆ Sim, o arquivo .wgz é apenas um arquivo zip renomeado
- Extraia o conteúdo
  - ◆ Lembre-se de manter a mesma estrutura de pastas
  - ◆ Selecione a pasta alvo que melhor se adequa as suas necessidades

## Modifique o ícone

Nos arquivos de exemplo você irá encontrar o ícone utilizado na aplicação. Você pode utilizar o que foi criado, mas ele serve apenas para demonstração, não para ser distribuído comercialmente como um FlashWidget. O ícone precisa estar no formato PNG. Ícones SVG ainda não são suportados.

## Modifique o info.plist

Abra o arquivo `info.plist` e edite os valores chave como marcados abaixo com a cor verde. O `DisplayName` é usado como título da aplicação e a chave `AllowNetworkAccess` define se o widget pode acessar a internet ou não. Se a sua aplicação não utiliza a rede, então você pode setar o valor para falso e o usuário não será perguntado sobre isso na utilização do Widget. Mais informações sobre o `info.plist` são encontradas no endereço do [Widget](#).

```
<key>DisplayName</key>
<string>MyFlashWidget</string>
<key>Identifier</key>
<string>textcom.forum.widgets.MyFlashWidget</string>
<key>Version</key>
<string>1.0</string>
<key>MainHTML</key>
```

## Como empacotar conteúdo Flash em um Widget

```
<string>main.html</string>
<key>AllowNetworkAccess</key>
<true/>
```

## O HTML e JavaScript

Quando você abrir o arquivo `Main.html` em um editor de texto ou HTML, você irá se deparar com basicamente três coisas que você poderá modificar caso ache necessário:

### Referência CSS

É recomendado utilizar arquivos CSS externos. Em nosso caso isto é insignificante, já que estamos criando conteúdo Flash Lite para ser exibido no modo fullscreen, tela inteira. O CSS será utilizado para setar a cor do background para preto e posicionar da forma adequada o conteúdo Flash Lite para fullscreen, tela inteira. Novamente, não existem razões para entrar em detalhes aqui. Você pode encontrar mais informações em como o HTML e o CSS funcionam no escopo de Widget no endereço do [Widget](#).

```
body {
    background: rgb(0,0,0);
    position: absolute;
    left: 0px;
    top: 0px;
    width: 100%;
    height: 100%;
    margin: 0px;
    overflow: hidden;
}
```

### JavaScript

De acordo com a [documentação do Widget], a convenção preferida é utilizar a codificação separada em um arquivo `.js`. Neste projeto o JavaScript serve apenas para setar o modo de navegação. Se você desejar utilizar mais funcionalidades do Widget, siga para o endereço do [Widget](#).

### Setando o modo de navegação

```
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
<!--
    widget.setNavigationEnabled(false);
//-->
</SCRIPT>
```

O código acima adiciona um comando JavaScript para a página `Main.html`: `widget.setNavigationEnabled(false)`; Isso diz ao ambiente Web Runtime para utilizar a navegação baseada em guias. Se você desejar utilizar a navegação no modo ponteiro, altere o valor para `true`, como a seguir:

### Setando a navegação no modo ponteiro

```
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
<!--
```

## Como empacotar conteúdo Flash em um Widget

```
        widget.setNavigationEnabled(true);  
    //-->  
</SCRIPT>
```

Este arquivo de exemplo tem apenas uma animação, mas os testes provaram que ambos os tipos de navegação funcionam perfeitamente com conteúdo [Flash Lite](#). Sinta-se livre para escolher o que melhor se aplica para você.

### Anexando o .swf

O código abaixo foi gerado pelo [Adobe Flash Professional](#). Se você utilizar o arquivo Main.html fornecido no exemplo, você precisa se certificar em mudar os valores correspondentes à sua aplicação. Principalmente o valor `src` e a diretiva `allowScriptAccess`.

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="238" height="318" id="MyFlash"  
<param name="allowScriptAccess" value="sameDomain" />  
<param name="movie" value="MyFlash.swf" />  
<param name="loop" value="false" />  
<param name="menu" value="false" />  
<param name="quality" value="high" />  
<param name="wmode" value="opaque" />  
<param name="bgcolor" value="#ffffff" />  
<embed src="MyFlash.swf" loop="false" menu="false" quality="high" wmode="opaque" bgcolor="#ffffff"  
</object>
```

### Flash Lite App

Se você não tem um aplicativo Flash Lite para efetuar o teste, agora é um bom momento para criar, ou editar um criado anteriormente. O exemplo foi criado utilizando 2 pixels a menos em ambos os lados, apenas para ter certeza de não existir qualquer problema de antialiasing. Neste caso a resolução foi setada para 238 x 318.

**Dica:** *Novos aparelhos suportam o Web Runtime e também o Flash Lite 3!*

### Criando o pacote Widget

Agora que você efetuou todas as modificações necessárias, ou criou todos os arquivos necessários, é hora de testar seu Widget. Como mencionado anteriormente, o arquivo `.wgz` é apenas um arquivo zip renomeado, então tudo que você precisa fazer é utilizar seu método favorito para zipar o arquivo e mudar sua extensão para `wgz`. Você deve incluir no pacote zipado a pasta onde estão seus arquivos `.html`, `.css`, `.swf`, etc.

### Testando o Widget

O emulador existente no kit do desenvolvedor não suporta Flash Lite, portanto você poderá testar a validade do pacote, porém não verá o conteúdo Flash Lite. Caso você não tenha um dispositivo que suporte o Web Runtime da plataforma s60 você pode utilizar o serviço [Remote Device Access](#).

O método de instalação é sempre o mesmo. Use Bluetooth, cartão de memória ou qualquer outro método de transferência do pacote `.wgz` para seu dispositivo e execute ele. O dispositivo ou emulador irá iniciar o

## Como\_empacotar\_conteÃdo\_Flash\_em\_um\_Widget

processo de instalao, que quando finalizado ir adicionar um ícone para o seu FlashWidget na pasta de aplicaes.

## Ajuda

Se voc encontrou qualquer problema com este documento, utilize a guia de *comentrios* para post-lo. Se voc precisa de ajuda para desenvolver seu pacote Widget, a melhor opo  postar sua questo no [Web runtime discussion board](#).