

Contents

- [1 O que é o Shared Object](#)
- [2 Como criar, abrir e ler dados do Shared Object](#)
- [3 Salvando e excluindo dados](#)
- [4 Exemplo em funcionamento](#)

O que é o Shared Object

O *Shared Object* é o sistema proposto pela tecnologia *Flash Lite* para armazenamento de dados no dispositivo em um conjunto de pares nome-valor.

Sendo uma tarefa lenta a leitura destes dados, para assegurar que os mesmos estejam imediatamente disponíveis diante da requisição *Flash Lite 2.x* e o mais recente *Flash Lite 3.0* utilizam manipuladores de eventos para efetuar esta tarefa.

O limite de armazenamento pode ser consultado utilizando o método *SharedObjects.getMaxSize()*.

Em *Flash Lite* não é possível compartilhar o mesmo *Shared Object* entre vários arquivos SWF, sendo o mesmo apenas utilizado por um único arquivo SWF de mesma versão. O arquivo é considerado de versão diferente quando existir uma modificação no arquivo original, mesmo que haja alteração no nome do arquivo, o mesmo continua sendo de mesma versão.

Como criar, abrir e ler dados do Shared Object

O primeiro passo é criar/abrir os dados e efetuar a leitura para utilização. Para esta tarefa contamos com duas funções do tipo *static*.

A função *getLocal()* cria e retorna a referência de um objeto de armazenamento local caso não exista nenhum atualmente criado.

Definição:

```
public static getLocal(name:String) : SharedObject
```

A função *addListener()* adiciona uma função a ser executada ao carregar um determinado objeto.

Definição:

```
public static addListener(nome_do_objeto:String,  
                          funcao_notificadora:Function) : Void
```

Exemplo:

```
function getSharedObject(info_obj:Object):Void {  
    trace("arquivo carregado");  
}
```

Como utilizar o Shared Objects para armazenar informações com Flash Lite

```
}  
SharedObject.addListener("meusDados", getSharedObject);  
var persist_so:SharedObject = SharedObject.getLocal("meusDados");
```

Salvando e excluindo dados

Para armazenar dados no formato nome/valor basta utilizar a propriedade *data* da variável instância da classe *SharedObject*. Os valores armazenados podem ser dos tipos básicos do Flash (*Array*, *Number*, *Boolean*, etc).

Exemplo:

```
if(persist_so.getSize() <= 0) {  
    persist_so.data.numeroAcessos = 1;  
} else {  
    persist_so.data.numeroAcessos += 1;  
}
```

Para garantir que os dados sejam salvos no dispositivo utiliza-se a função *flush()* à partir da variável instância da classe *SharedObject*.

Exemplo:

```
// imediatamente armazena os dados localmente no dispositivo  
persist_so.flush();
```

A função *clear()* é utilizada à partir da variável instância da classe *SharedObject* para limpar os dados e excluir permanentemente o arquivo de armazenamento do dispositivo, porém caso a exclusão seja apenas para um conjunto nome/valor utiliza-se a estrutura **delete so.data.attributeName;**

Exemplo:

```
// limpa os dados e efetua a exclusão do arquivo  
persist_so.clear();
```

Exemplo:

```
// efetua a exclusão do conjunto par/valor  
delete persist_so.data.numeroAcessos;
```

Exemplo em funcionamento

Crie um novo documento Flash (*Mobile*), no *Device Central* selecione o *Flash Lite Player 2.1* e mantenha a versão do *ActionScript* em 2.0.

Com o novo documento criado, salve como **sharedObjects.fla**, clique no primeiro frame da timeline e vá para o menu *Window > Actions [F9]*, cole o código abaixo e execute a aplicação (CTRL + ENTER). Para verificar o funcionamento basta voltar para *IDE* do Adobe Flash e executar novamente a aplicação (CTRL + ENTER)

```
// define a execução em tela cheia  
fscommand2("FullScreen", true);
```

Como utilizar o Shared Objects para armazenar informações com Flash Lite

```
// objeto ouvinte para manipular os eventos de tecla
var handler_obj:Object = new Object();

// variavel que irá receber os dados do arquivo de armazenamento local
var persist_so:SharedObject;

// criação de um campo de texto dinâmico para receber os valores
// do arquivo de armazenamento

// Stage.width e Stage.height são variáveis do tipo static que indicam a largura e altura.

// this.getNextHighestDepth() = retorna o próximo nível disponível na lista de display
this.createTextField("display_txt",
    this.getNextHighestDepth(),
    0,
    0,
    Stage.width,
    Stage.height);

function getSharedObject(info_obj:SharedObject) {
// verifica o tamanho atual do arquivo em bytes
// caso seja o primeiro acesso atribui o valor 1,
// caso contrário incrementa o valor atual mais 1
if(persist_so.getSize() <= 0) {
    data.numeroAcessos = 1;
} else {
    data.numeroAcessos += 1;
}

// Escreve no campo de texto o tamanho atual, número de acessos
// e valor máximo de armazenamento no dispositivo
display_txt.text = "Tamanho: " + persist_so.getSize() + "\n";
display_txt.text = "Número de acessos: " + persist_so.data.numeroAcessos + "\n";
display_txt.text = "Armazenamento máximo: " + SharedObject.getMaxSize();

// imediatamente armazena os dados localmente no dispositivo
persist_so.flush();
}

// Ouvinte para o evento onKeyDown da classe Key
handler_obj.onKeyDown = function() {
// Caso o código da tecla seja igual ao
// código correspondente a Soft Key da esquerda,
// os dados são limpos e o arquivo é excluído
if(Key.getCode() == ExtendedKey.SOFT1) {
    textDisplay.text = "Dados limpos";
    clearPersist_so();
} else {
// Caso a Soft Key da direita seja acionada, fecha a aplicação
if(Key.getCode() == ExtendedKey.SOFT2) {
    ("quit");    fscommand2
}
}
}

// adiciona o objeto para manipular os eventos de tecla
Key.addListener(handler_obj);

// adiciona uma manipulador de evento a ser executado no carregamento do arquivo
SharedObject.addListener("meusDados", getSharedObject);
```

Como utilizar o Shared Objects para armazenar informações com Flash Lite

```
// Retorna a referência do objeto de armazenamento 'meusDados'  
persist_so = SharedObject.getLocal("meusDados");
```