Criando logs

Utilizar arquivos de log é uma boa forma de obter informações sobre a execução do seus programas assim como corrigir erros.

Para criar os logs iremos utilizar a classe RFileLogger, essa classe permite gerar um arquivo .txt com as informações que você desejar.

Criando o log

No Symbian OS arquivos de logs são armazenados na pasta **logs** no diretório (\Epoc32\winscw\c), como essa pasta não vem criada por default, ela deve ser criada. Dentro dessa pasta deve ser criada uma outra pasta para armazenar os seus logs, esse diretório será utilizado na construção da classe.

Para utilizar a classe RFileLogger é preciso incluir o cabeçalho flogger.h e dentro do arquivo .mmp do seu projeto deve-se incluir a biblioteca flogger.lib

O primeiro passo é criar uma instância da classe RFileLogger e então fazer a conexão ao servidor de logs.

```
RFileLogger iFileLogger;
iFileLogger.Connect();
```

Iremos então utilizar a função CreateLog para gerar o arquivo de log, nessa função serão passados o diretório onde ficará os logs (a pasta encontrada na pasta logs), o nome do arquivo de log, e o modo no qual o log será criado.

Nota: O arquivo de log será criado apenas se o diretório existir.

Exemplo:

```
_LIT(KLogDir,"projeto");
_LIT(KLogArquivo,"logfile.txt");

iFileLogger.CreateLog(KLogDir, KLogArquivo, EFileLoggingModeOverwrite);
```

Analisando o código:

iFileLogger - instância da classe RFileLogger.
KLogDir - literário com o diretório onde o log será armazenado.
KLogArquivo - literário com o nome do arquivo do log.
EFileLoggingModeOverwrite - modo no qual o arquivo será criado, podendo ser Overwrite para sobrescrever o arquivo (caso o arquivo não exista ele será criado) ou append(adiciona as novas informações no final do arquivo).

Para gravar no log, será utilizada a função Write, ela possui diversas variâncias, é interessante você verificar os documentos que acompanham o SDK para analizar qual função preenche seus requisitos.

Como exemplo será mostrado a utilização da função Write de duas maneiras diferentes:

```
_LIT( KExemplo, "Ola, fazendo um teste");
iFileLogger.Write( KExemplo);
```

Nesse exemplo, será gravado no arquivo logfile.txt a string "Ola, fazendo um teste"

```
_LIT( KLogDir2, "projeto");
_LIT(KLogArquivo2, "logfile2.txt");
_LIT( KExemplo2, "Ola, fazendo um teste2");
iFileLogger.CreateLog( KLogDirectory2, KLogFile2, EFileLoggingModeOverwrite);
iFileLogger.Write( KExemplo2);
```

Nesse exemplo, iremos gravar as informações em outro arquivo de log, diferente do criado no CreateLog, a definição é

(C) Copyright Nokia 2014. All rights reserved.

basicamente a mesma, porém temos uma última referência que passa a string para o arquivo.

Nota: na utilização desse segundo exemplo não é necessário chamar a função Connect().

Após gravar as informações ao arquivo de log, é preciso fechar o arquivo e disconectar do servidor de logs.

Exemplo:

```
iFileLogger.CloseLog();
iFileLogger.Close();
```

Removendo a hora/data no log

Para desabilitar a exibição da hora e data no arquivo de log, basta fazer a chamada a função:

```
iFileLogger.SetDateAndTime( EFalse, EFalse);
```

essa função recebe dois argumentos, ambos booleanos, o primeiro é referente a data e o segundo a hora. Para ativa-los basta substituir por ETrue.

Exibindo valores das variáveis

Exibir valores da variáveis é muito semelhante a utilizada no printf(), pode ser feita utilizando a função WriteFormat() como exibido abaixo:

```
TInt iValor = 123;
_LIT( KValorInt, "Valor Variavel = %d");
iFileLogger.WriteFormat(K ValorInt, iValor);
```

A saída será:

20/05/2007 11:01:53 Valor Variavel = 123

Exemplo

RFileLogger exemplo

Documento SDK

RFileLogger - Referência SDK

(C) Copyright Nokia 2014. All rights reserved.