

# File input stream

## Obsah

1. Popis.....	1
2. Závislosti .....	1
3. Implementačné informácie .....	1
4. Piny .....	2
5. Konfigurácia .....	2
5.1 Konfiguračná štruktúra.....	2
5.2 Konfiguračný dialóg.....	2
6. Definícia triedy .....	3
7. Príklad použitia .....	3

## 1. Popis



Trieda *File input stream* načíta obsah súboru z disku do pamäťového buffru. Meno a adresu vstupného súboru je možné nastaviť v konfigurácii. Obsah vstupného súboru sa prenáša v binárnom móde (t.j. súbor je prenesený do pamäte bezo zmien).

## 2. Závislosti

Trieda *File input stream* používa tieto externé definičné súbory a knižnice:

**Headers:** *filtergraph.h*, *datatypes.h*

**Libs:** *filtergraph.lib*

## 3. Implementačné informácie

Informácie o triede:

**Názov:** *File input stream*

**Verzia:** *1.0*

**Magic:** *10*

Informácie o definíciách:

**Header:** *fileinputstream.h*

**Konfiguračný header:** *fileinputstreamcfg.h*

**Lib:** *fileinputstream.lib*

Informácie o knižnici obsahujúcej triedu:

**Názov knižnice:** *File input stream Library*

**Verzia knižnice:** *1.0*

**Dll súbor knižnice:** *fileinputstream.dll*

## 4. Piny

Piny sú popísané spôsobom: „*[index pinu na filtri] názov pinu: popis pinu*“.

**[0] Cfg in:** Konfiguračný vstup. Vstupom je serializovaná konfigurácia.

**[1] Cfg out:** Konfiguračný výstup. Výstupom je serializovaná konfigurácia.

**[2] Out:** Dátový výstup. Výstupom je pamäťový buffer, v ktorom je načítaný obsah daného súboru.

## 5. Konfigurácia

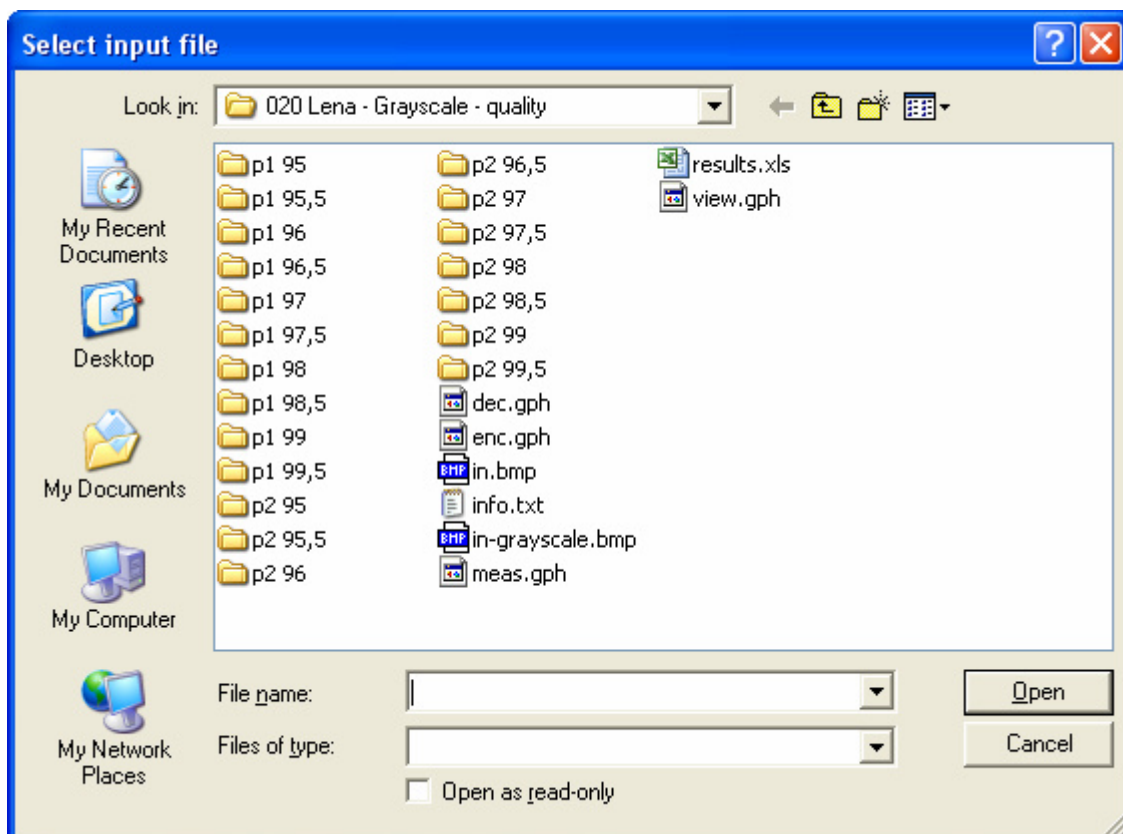
### 5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    char* FileName;           // reťazec s adresou a názvom súboru  
} TFileInputStreamConfig;
```

Reťazec *FileName* obsahuje meno súboru s adresou.

### 5.2 Konfiguračný dialóg



Trieda vyvolá štandardný dialóg systému pre voľbu súboru.

## 6. Definícia triedy

Trieda *File input stream* je definovaná nasledovne:

```
class TFileInputStreamFilter : public TFilter {
private:
    char*      FileName;          // meno súboru
    FILE*      File;              // informácie o otvorenom súbore
    TBuffer    DataBuffer;        // buffer pre dáta
    int        RecordSize;        // veľkosť čítacieho bloku
    bool       StopFlag;          // indikátor zastavenia

    TPin* CfgInPin;                // vstupný konfiguračný pin
    TPin* CfgOutPin;              // výstupný konfiguračný pin
    TPin* OutPin;                 // výstupný pin

    int setConfigFromPin(TPin *pin); // nastaví konfiguráciu z pinu
    int putConfigOnPin(TPin *pin);  // sprístupní konfiguráciu na pine
    int freeConfigOnPin(TPin *pin); // uvoľní konfiguráciu na pine
    int clearOutputPins();          // vyčistí výstupné piny
    int freeOutputData();           // uvoľní výstupné dáta
public:
    TFileInputStreamFilter();        // konštruktor
    ~TFileInputStreamFilter();       // deštruktor

    int setConfigData(TBuffer config, int type); // nastaví konfiguráciu
    TBuffer getConfigData(int type);             // vráti konfiguráciu
    int initialize();                             // inicializácia
    int run();                                    // beh
    int reset();                                  // reset stavu
    int stop();                                   // zastavenie behu
    int finalize();                               // finalizácia
    int showConfigDialog();                      // vyvolá konfiguračný dialóg
};
```

## 7. Príklad použitia

