

File output 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	2
7. Príklad použitia	3

1. Popis



Trieda *File output stream* zapíše obsah pamäťového buffra do súboru na disku. Meno a adresu súboru je možné nastaviť v konfigurácii. Obsah pamäťového buffra sa prenáša v binárnom móde (t.j. obsah buffra sa zapíše na disk bezo zmien).

2. Závislosti

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

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

Libs: *filtergraph.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *File output stream*

Verzia: *1.0*

Magic: *20*

Informácie o definíciách:

Header: *fileoutputstream.h*

Konfiguračný header: *fileoutputstreamcfg.h*

Lib: *fileoutputstream.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *File output stream Library*

Verzia knižnice: *1.0*

Dll súbor knižnice: *fileoutputstream.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] **In:** Dátový vstup. Vstupom je pamäťový buffer, ktorý má byť zapísaný do súboru.

5. Konfigurácia

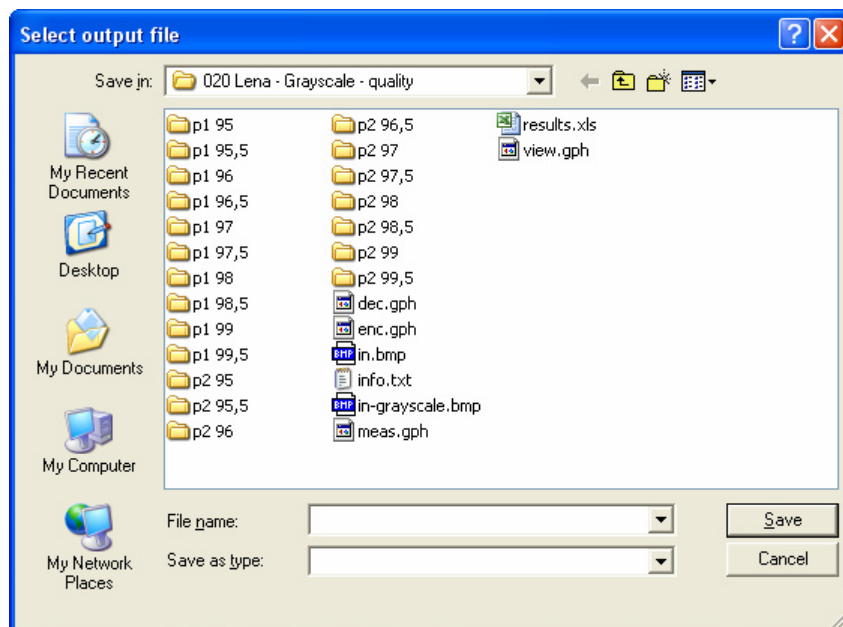
5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    char* FileName;           // reťazec s menom a adresou súboru  
} TFileOutputStreamConfig;
```

Reťazec *FileName* obsahuje meno súboru s adresou. Ak tento súbor už existuje, jeho obsah bude nahradený novým. Ak súbor ešte neexistuje, bude vytvorený nový súbor. Ak adresa k danému súboru neexistuje alebo je neplatná, beh sa ukončí s chybou.

5.2 Konfiguračný dialóg



Trieda vyvolá štandardný dialóg systému pre voľbu súboru.

6. Definícia triedy

Trieda *File output stream* je definovaná nasledovne:

```

class TFileOutputStreamFilter : public TFilter {
private:
    char*      FileName;           // meno súboru
    FILE*      File;               // informácie o otvorenom súbore
    int        RecordSize;         // veľkosť zapisovacieho bloku
    bool       StopFlag;           // indikátor zastavenia

    TPin* CfgInPin;                // vstupný konfiguračný pin
    TPin* CfgOutPin;               // výstupný konfiguračný pin
    TPin* InPin;                   // vstupný 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:
    TFileOutputStreamFilter();       // konštruktor
    ~TFileOutputStreamFilter();      // 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();                                   // ukončenie behu
    int finalize();                               // finalizácia
    int showConfigDialog();                      // vyvolá konfiguračný dialóg
};

```

7. Príklad použitia

