# Databank dokumentation

Databanken er et API der agerer som bindeled mellem Controlleren (og dermed GUI) og Databankens underklasser.

Databankens funktioner er alle wrapper-funktioner, der ikke laver andet end at kalde funktioner i dens underklasser som er beskrevet nedenunder:

## Sensorkonfigurationsbankens funktioner

Sensorkonfigurationsbanken inkluderer headerfilen *SensorConfiguration.h* der indeholder implementeringen af en sensorkonfiguration. Denne klasse har kun get/set-metoder. For alle attributter (navn, sensor ID, akse, mapping scheme, lydpakke) i en sensorkonfiguration findes der én set-metode og én get-metode. Sensorkonfigurationsbanken har en container (*std::map*) med disse sensorkonfigurationer.

|  |  |
| --- | --- |
| **Funktion** | SensorConfigurationBank - *Constructor* |
| **Parametre** |  |
| **Returværdi** |  |
| **Beskrivelse** | ’Loader’ sensorkonfigurationerne, ved at bruge Boost:   * Læser XML-filen ”SensorConfigurationBank.xml” og gemmer hver *configuration*-knude som en *SensorConfiguration*-klasse * XML-filen indeholder en knude *mappingscheme* som er referencen (navnet) til et mapping scheme * Mapping schemet er en underklasse til sensorkonfigurationen. Vi skal have fat i disse tilhørende attributter, og derfor køres *getMappingScheme*   Disse sensorkonfigurationer tilføjes til mappet *SensorConfigurations*, så de kan bruges omgående. |

|  |  |
| --- | --- |
| **Funktion** | getMappingScheme |
| **Parametre** | string id – navnet på mapping schemet vi skal have hentet de tilhørende attributter til. |
| **Returværdi** | MappingScheme – en mapping scheme klasse der skal bruges til en sensorkonfiguration. |
| **Beskrivelse** | Funktionen virker lidt ligesom constructoren *SensorConfigurationBank*.   * Læser XML-filen ”MappingSchemes.xml” * Finder den specifikke mapping scheme med samme navn som den *id* man sendte med som parameter * Gemmer attributterne fra denne *scheme*-knude til en *MappingScheme* klasse * Returnér denne |

|  |  |
| --- | --- |
| **Funktion** | getMappingSchemes – *ej at forveklse med overstående* |
| **Parametre** |  |
| **Returværdi** | Returnerer en *vector<string>* med navnene på alle mapping schemes. |
| **Beskrivelse** | Minder lidt om constructoren og *getMappingScheme*.   * Læser XML-filen ”MappingSchemes.xml” * Finder alle *id­*-knuder og tilføjer indholdet af disse til en *vector* *myMappingSchemeList* * Returnér denne   Bruges i GUI’en for at få et overblik over hvilke mapping schemes der er til rådighed. |

|  |  |
| --- | --- |
| **Funktion** | getSensorIDs |
| **Parametre** |  |
| **Returværdi** | Returnerer en *vector<string>* med ID på alle sensorer der er i brug. |
| **Beskrivelse** | Samme som overstående, bare med en anden XML-fil og anden knude.   * Læser XML-filen ”SensorConfiguration.xml” * Finder alle *sensorid*-knuder i alle *configuration*-knuder og tilføjer indholdet af disse til en *vector mySensorIDs* * Returnér denne |

|  |  |
| --- | --- |
| **Funktion** | edit |
| **Parametre** | En sensorkonfiguration man gerne vil ændre |
| **Returværdi** | void |
| **Beskrivelse** | * Sletter sensorkonfiguration med samme navn * Tilføjer den ’nye’ sensorkonfiguration vha. funktion *add* |

|  |  |
| --- | --- |
| **Funktion** | add |
| **Parametre** | En sensorkonfiguration man gerne vil tilføje |
| **Returværdi** | void |
| **Beskrivelse** | * Indsætter sensorkonfiguration i *sensorConfigurations*-mappet, hvor data er sensorkonfigurationen man gerne vil tilføje, og *key*’en er navnet på sensorkonfigurationen * Kalder metoden *save* |

|  |  |
| --- | --- |
| **Funktion** | removeSensorConf |
| **Parametre** | Navnet (*key*’en) på sensorkonfigurationen man gerne vil fjerne. |
| **Returværdi** | void |
| **Beskrivelse** | * Fjerner sensorkonfigurationen fra *sensorConfigurations*-mappet * Kalder metoden *save* |

|  |  |
| --- | --- |
| **Funktion** | save |
| **Parametre** |  |
| **Returværdi** | void |
| **Beskrivelse** | Gemmer (serialiserer) sensorkonfigurationerne i *sensorConfigurations*-mappet. Disse gemmes til ”SensorConfigurationBank.xml”   * Opret ’træ’ (*boost::property\_tree::ptree)* vi kalder for ’*pt*’ der fungerer som roden til vores XML-dokument der skal gemmes om lidt * For hver sensorkonfiguration der findes i *sensorConfigurations*-mappet:   + Opret endnu et træ *myTree* der vil fungerer som en gren i XML-dokumentet   + Tilføjer alle attributter fra netop den sensorkonfiguration vi kigger på nu som knuder til dette nye træ   + *myTree* sættes som child (gren) til roden/parent-træet *pt* * Gem træet *pt* som XML-filen med navnet “SensorConfigurationBank.xml” |

|  |  |
| --- | --- |
| **Funktion** | list |
| **Parametre** |  |
| **Returværdi** | Returnerer en *vector<string>* med navnene på alle sensorkonfigurationer. |
| **Beskrivelse** | * Opret ny *vector* der hedder *sensorConfigs* * For alle sensorkonfigurationer i *sensorConfigurations*: * Læg navnet på sensorkonfigurationen ind i *sensorConfigs* * Returnér *sensorConfigs* |

|  |  |
| --- | --- |
| **Funktion** | get |
| **Parametre** | Navnet på sensorkonfiguration der ønskes returneret |
| **Returværdi** | Sensorkonfigurationen man ønsker |
| **Beskrivelse** | * Da navnet på sensorkonfigurationen man gerne vil returnere er det samme som *key*’en i *sensorConfigs*, skal man blot returnere det element der binder sig til *key’*en som er input parameteren. |

## Presetbankens funktioner

|  |  |
| --- | --- |
| **Funktion** | PresetBank – *Constructor* |
| **Parametre** |  |
| **Returværdi** |  |
| **Beskrivelse** | ’Loader’ presets, ved at bruge Boost:   * Læser XML-filen ”PresetBank.xml” og gemmer hver *preset*-knude som en *Preset*-klasse   Disse presets tilføjes til vectoren *presets\_*, så de kan bruges omgående. |

|  |  |
| --- | --- |
| **Funktion** | get |
| **Parametre** | Et tal der repræsenterer pladsen i *presets­\_* som presettet man gerne vil ’gette’ ligger på. |
| **Returværdi** | Presettet man gerne vil have |
| **Beskrivelse** | * Returnér preset på input-parameter pladsen fra *presets\_* |

|  |  |
| --- | --- |
| **Funktion** | edit |
| **Parametre** | Et preset man gerne vil ændre |
| **Returværdi** | void |
| **Beskrivelse** | * Assign det nye preset i vectoren *presets\_* |

|  |  |
| --- | --- |
| **Funktion** | save |
| **Parametre** |  |
| **Returværdi** | void |
| **Beskrivelse** | Gemmer (serialiserer) presets der ligger i *presets\_*. Disse gemmes til ”PresetBank.xml”   * Opret ’træ’ (*boost::property\_tree::ptree)* vi kalder for ’*pt*’ der fungerer som roden til vores XML-dokument der skal gemmes om lidt * For hver preset der findes i *presets­\_*-vectoren:   + Tilføj en ’gren’ (knude) *preset* der indeholder de sensorkonfigurationer der skal bruges. * Gem træet *pt* som XML-filen med navnet “PresetBank.xml” |

## Samplebankens funktioner

|  |  |
| --- | --- |
| **Funktion** | SampleBank – *Constructor* |
| **Parametre** |  |
| **Returværdi** |  |
| **Beskrivelse** | ’Loader’ lydpakkerne, ved at bruge Boost:   * Læser XML-filen ”SampleBank.xml” og gemmer hver *sample*-knude som en *SoundPack*-klasse   Disse lydpakker tilføjes til mappet *soundPacks*, så de kan bruges omgående. |

|  |  |
| --- | --- |
| **Funktion** | list |
| **Parametre** |  |
| **Returværdi** | Returnerer en *vector<string>* med navnene på alle lydpakker |
| **Beskrivelse** | * Opret ny *vector* med navnet *soundsVector* * For hver lydpakke i mappet *soundPacks*: * Tilføj navnet på lydpakken til *soundsVector* * Returnér denne |

|  |  |
| --- | --- |
| **Funktion** | add |
| **Parametre** | Navn på lydpakken |
| **Returværdi** | void |
| **Beskrivelse** | * En ny lydpakke oprettes, hvor stien er defaultstien konkateneret med input parameteren (navnet) på lydpakken * Indsæt denne lydpakke i mappet *soundPacks*, hvor *key*’en er input parameteren, og data’en er selve lydpakken. * Kald funktionen *save* |

|  |  |
| --- | --- |
| **Funktion** | remove |
| **Parametre** | Navn på lydpakke |
| **Returværdi** | void |
| **Beskrivelse** | * Slet lydpakken med input-parameteren (navnet) *key* i mappet *soundPacks* * Kald funktionen *save* |

|  |  |
| --- | --- |
| **Funktion** | save |
| **Parametre** |  |
| **Returværdi** | void |
| **Beskrivelse** | Gemmer (serialiserer) lydpakkerne i *soundPacks*-mappet. Disse gemmes til ”SampleBank.xml”   * Opret ’træ’ (*boost::property\_tree::ptree)* vi kalder for ’*pt*’ der fungerer som roden til vores XML-dokument der skal gemmes om lidt * For hver sensorkonfiguration der findes i *soundPacks*-mappet:   + Tilføj en ’gren’ (knude) *sample* der indeholder stien på lydpakken * Gem træet *pt* som XML-filen med navnet “SampleBank.xml” |