Описание базовых классов AgavaSCADA/AgavaPLC

В данном документе описываются свойства, поля и методы базовых классов AgavaPLC/AgavaSCADA, взаимодействие с которыми доступно в функциях, классах и программах приложений.

Иерархия классов объектной модели AgavaPLC 1.6

Данный документ применим к AgavaSCADA/AgavaPLC версии 1.5 и старше (новее).

□

Содержание

BasicNode
- Конструкторы
- Методы
- Свойства

SignalValue
- Конструкторы
- Методы
- Примеры

NodeProperty
- Конструкторы
- Методы
- Примеры

BasicSource
- Методы

BasicSignal
- Методы

ModbusRegister
- Поля
- Методы

BasicWidget
- Методы
- Обработчики

BasicWindow
- Методы

1 BasicNode

Базовый класс для всех узлов проекта.

Предок: нет.

1.1 Конструкторы

```
BasicNode::BasicNode()              ///< Конструктор по умолчанию
BasicNode::BasicNode(string path)   ///< Создание узла по пути в проекте
```

1.2 Методы

```
string BasicNode::GetName()         ///< Получение имени узла
string BasicNode::GetShortName()    ///< Получение шифра узла
string BasicNode::GetDescription()  ///< Получение описания узла
```

```
void BasicNode::SetName(string name)
void BasicNode::SetShortName(string shortname)
void BasicNode::SetDescription(string description)
```

```
string BasicNode::GetPath()         ///< Получение пути узла в проекте
```

```
void BasicNode::Start()             ///< Запуск узла
void BasicNode::Stop()              ///< Остановка узла
```

```
NodeProperty@ BasicNode::GetProperty(string& prop)     ///< Получение указателя на свойство узла
void BasicNode::PropertyValueUpdated(string& prop)     ///< Сигнализация об обновлении значения
свойства
```

1.3 Свойства

Имя	Описание	Тип	Доступ	Внутренний идентификатор
Id	Идентификатор узла	int	Чтение/Запись	CONFIGNODE_PARAM_ID
Name	Имя узла	string	Чтение/Запись	CONFIGNODE_PARAM_NAME
Description	Описание узла	string	Чтение/Запись	CONFIGNODE_PARAM_DESCRIPTION
ShortName	Шифр узла	string	Чтение/Запись	CONFIGNODE_PARAM_SHORTNAME
NodeDisabled	Узел выключен	bool	Чтение/Запись	CONFIGNODE_PARAM_DISABLED

2 SignalValue

Класс, хранящий значение, передаваемое между узлами.

Предок: нет.

2.1 Конструкторы

```
SignalValue::SignalValue(string value)
SignalValue::SignalValue(int value)
SignalValue::SignalValue(double value)
SignalValue::SignalValue(bool value)
```

2.2 Методы

```
string SignalValue::ToString()   ///< Преобразовать значение в строку
int SignalValue::ToInt()         ///< Преобразовать значение в int
double SignalValue::ToDouble()   ///< Преобразовать значение в double
bool SignalValue::ToBool()       ///< Преобразовать значение в bool
```

```
void SignalValue::SetValue(string& value)
void SignalValue::SetValue(int value)
void SignalValue::SetValue(double value)
void SignalValue::SetValue(bool value)
```

```
bool SignalValue::IsNumber()     ///< Значение является числовым
bool SignalValue::IsError()      ///< Значение является ошибкой
```

2.3 Примеры

```
void fnIncrementSource(BasicSource_t &source)
{
    SignalValue sv = source.GetValue();
    if (sv.IsNumber() == true)
    {
        int iVal = sv.ToInt();
        iVal++;

        sv = SignalValue(iVal);
    }
    else
    {
        sv = SignalValue(0);
    }

    source.SetValue(sv);
}
```

3 NodeProperty

Класс, предоставляющий доступ к свойствам узлов.

Предок: нет.

3.1 Конструкторы

```
NodeProperty::NodeProperty()
NodeProperty::NodeProperty(NodeProperty& other)
```

3.2 Методы

```
string NodeProperty::GetName()         ///< Получение имени свойства
string NodeProperty::GetShortName()    ///< Получение шифра свойства
string NodeProperty::GetDescription()  ///< Получение описания свойства
string NodeProperty::GetTitle()        ///< Получение заголовка свойства
```

```
string NodeProperty::ToString()        ///< Преобразование значения свойства в строку
int NodeProperty::ToInt()              ///< Преобразование значения свойства в число типа int
double NodeProperty::ToDouble()        ///< Преобразование значения свойства в число типа double
bool NodeProperty::ToBool()            ///< Преобразование значения свойства в булевское значение
```

```
int NodeProperty::SetValue(string val) ///< Установка значения свойства
int NodeProperty::SetValue(int val)    ///< Установка значения свойства
int NodeProperty::SetValue(double val) ///< Установка значения свойства
int NodeProperty::SetValue(bool val)   ///< Установка значения свойства
```

3.3 Примеры

```
BasicNode node("/Root/node");
NodeProperty@ prop = node.GetProperty("ShortName");

string sn2 = prop.ToString();
prop.SetValue(string(sn2+"ShortName1"));
```

4 BasicSource

Предок: BasicNode.

4.1 Методы

```
SignalValue BasicSource::GetValue()          ///< Получение значения.
int BasicSource::SetValue(SignalValue sv)    ///< Установка собственного значения с установкой
значений связанных узлов (источников).
int BasicSource::UpdateValue(SignalValue sv) ///< Установка собственного значения.
```

5 BasicSignal

Предок: BasicSource.

5.1 Методы

```
SignalValue BasicSignal::GetValue()          ///< Получение значения.
int BasicSignal::SetValue(SignalValue sv)    ///< Установка собственного значения с установкой
```

```
значений связанных узлов (источников).
int BasicSignal::UpdateValue(SignalValue sv) ///< Установка собственного значения.
```

6 ModbusRegister

Предок: BasicSource.

6.1 Поля

```
uint8  m_ucDeviceNo        ///< Номер устройства.
uint8  m_ucReadFunctionNo  ///< Номер функции чтения.
uint16 m_wReadRegisterNo   ///< Номер регистра чтения.
uint8  m_ucWriteFunctionNo ///< Номер функции записи.
uint16 m_wWriteRegisterNo  ///< Номер регистра записи.
uint8  m_ucSize            ///< Размер регистра в битах.
uint8  m_ucPriority        ///< Приоритет. Не используется.
```

6.2 Методы

```
int BasicSource::WriteValue(SignalValue sv) ///< Установка собственного значения с записью значения в
устройство.
```

7 BasicWidget

Предок: BasicNode.

7.1 Методы

```
void BasicWidget::Repaint()                ///< Перерисовка виджета
```

7.2 Обработчики

```
void BasicWidget::OnPaint(Painter@ painter)              ///< Обработчик отрисовки виджета
void BasicWidget::OnMousePress(MouseEvent@ event)        ///< Обработчик нажатия кнопки мыши
void BasicWidget::OnMouseRelease(MouseEvent@ event)      ///< Обработчик отпускания кнопки мыши
void BasicWidget::OnMouseDoubleClick(MouseEvent@ event)  ///< Обработчик двойного нажатия кнопки мыши
void BasicWidget::OnMouseMove(MouseEvent@ event)         ///< Обработчик перемещения курсора мыши над
виджетом
void BasicWidget::OnMouseEnter()                         ///< Обработчик входа курсора мыши в область
над виджетом
void BasicWidget::OnMouseLeave()                         ///< Обработчик выхода курсора мыши из
области над виджетом
void BasicWidget::OnWheel(WheelEvent@ event)             ///< Обработчик прокрутки колеса мыши при
нахождении курсора мыши в области над виджетом
```

8 BasicWindow

Предок: BasicWidget.

8.1 Методы

```
void BasicWindow::Show()  ///< Отображение окна.
void BasicWindow::Hide()  ///< Сокрытие окна.
```

Источник — https://docs.kb-agava.ru/index.php?title=Описание_базовых_классов_AgavaSCADA/AgavaPLC&oldid=2991

Эта страница в последний раз была отредактирована 22 апреля 2025 в 15:44.