

Автоматизация конфигурирования сетевого оборудования Cisco

Создано системой Doxygen 1.8.11

Оглавление

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Классы	5
3.1 Класс InterfaceModel	5
3.2 Класс InterfaceModelList	5
3.3 Класс Ipv4Validator	6
3.3.1 Конструктор(ы)	6
3.3.1.1 Ipv4Validator(QObject *parent=0)	6
3.3.2 Методы	6
3.3.2.1 validate(QString &, int &) const	6
3.4 Класс Ipv6Validator	7
3.4.1 Конструктор(ы)	7
3.4.1.1 Ipv6Validator(QObject *parent=0)	7
3.4.2 Методы	7
3.4.2.1 validate(QString &, int &) const	8
3.5 Класс MainWindow	8
3.6 Класс NetworkEquipmentForm	8
3.6.1 Подробное описание	9
3.7 Класс NetworkEquipmentModel	9
3.7.1 Подробное описание	11

3.7.2	Методы	11
3.7.2.1	bannerMotd()	11
3.7.2.2	hostname()	11
3.7.2.3	loggingSynchronous()	11
3.7.2.4	needBannerMotd()	12
3.7.2.5	needHostname()	12
3.7.2.6	needPasswordConsole()	12
3.7.2.7	needPasswordVty()	12
3.7.2.8	passwordConsole()	12
3.7.2.9	passwordEncryption()	13
3.7.2.10	passwordVty()	13
3.7.2.11	setBannerMotd	13
3.7.2.12	setHostname	13
3.7.2.13	setLoggingSynchronous	13
3.7.2.14	setNeedBannerMotd	14
3.7.2.15	setNeedHostname	14
3.7.2.16	setNeedPasswordConsole	14
3.7.2.17	setNeedPasswordVty	14
3.7.2.18	setPasswordConsole	14
3.7.2.19	setPasswordEncryption	15
3.7.2.20	setPasswordVty	15
3.8	Шаблон класса QList< T >	15
3.9	Класс TextValidator	15
3.9.1	Конструктор(ы)	16
3.9.1.1	TextValidator(QObject *parent=0)	16
3.9.2	Методы	17
3.9.2.1	validate(QString &, int &) const	17
3.10	Класс WordValidator	17
3.10.1	Конструктор(ы)	17
3.10.1.1	WordValidator(QObject *parent=0)	17
3.10.2	Методы	18
3.10.2.1	validate(QString &, int &) const	18

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

InterfaceModel	5
QAbstractTableModel	
InterfaceModelList	5
QList< T >	15
QList< InterfaceModel * >	15
QMainWindow	
MainWindow	8
QObject	
NetworkEquipmentModel	9
QValidator	
Ipv4Validator	6
Ipv6Validator	7
TextValidator	15
WordValidator	17
QWidget	
NetworkEquipmentForm	8

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

InterfaceModel	5
InterfaceModelList	5
Ipv4Validator	6
Ipv6Validator	7
MainWindow	8
NetworkEquipmentForm	8
NetworkEquipmentModel	
Модель сетевого оборудования	9
QList< T >	15
TextValidator	15
WordValidator	17

Глава 3

Классы

3.1 Класс InterfaceModel

Открытые члены

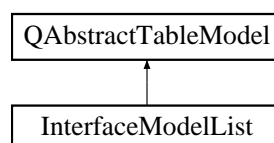
- InterfaceType **type** ()
- void **setType** (InterfaceType type)
- QString **number** ()
- void **setNumber** (QString number)
- QString **description** ()
- void **setDescription** (QString description)
- Duplex **duplex** ()
- void **setDuplex** (Duplex duplex)
- QHostAddress **ipv4** ()
- void **setIpv4** (QHostAddress ip)
- QHostAddress **ipv6** ()
- void **setIpv6** (QHostAddress ip)

Объявления и описания членов классов находятся в файлах:

- interfacemodel.h
- interfacemodel.cpp

3.2 Класс InterfaceModelList

Граф наследования:InterfaceModelList:



Открытые члены

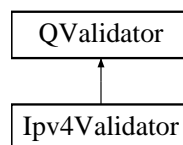
- **InterfaceModelList** (QObject *parent)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const
- void **setInterfaces** (QList< [InterfaceModel](#) * > interfaces)

Объявления и описания членов классов находятся в файлах:

- `interfacemodel.h`
- `interfacemodel.cpp`

3.3 Класс Ipv4Validator

Граф наследования:Ipv4Validator:



Открытые члены

- [Ipv4Validator](#) (QObject *parent=0)
[Ipv4Validator::Ipv4Validator.](#)
- State [validate](#) (QString &, int &) const
[Ipv4Validator::validate.](#)

3.3.1 Конструктор(ы)

3.3.1.1 `Ipv4Validator::Ipv4Validator (QObject * parent = 0)`

[Ipv4Validator::Ipv4Validator.](#)

Аргументы

<i>parent</i>	
---------------	--

3.3.2 Методы

3.3.2.1 `QValidator::State Ipv4Validator::validate (QString & text, int & pos) const`

[Ipv4Validator::validate.](#)

Аргументы

<i>text</i>	
<i>pos</i>	

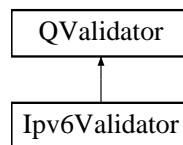
Возвращает

Объявления и описания членов классов находятся в файлах:

- customvalidator.h
- customvalidator.cpp

3.4 Класс Ipv6Validator

Граф наследования:Ipv6Validator:



Открытые члены

- [Ipv6Validator](#) (QObject *parent=0)
[Ipv6Validator::Ipv6Validator.](#)
- State [validate](#) (QString &, int &) const
[Ipv6Validator::validate.](#)

3.4.1 Конструктор(ы)

3.4.1.1 Ipv6Validator::Ipv6Validator (QObject * parent = 0)

[Ipv6Validator::Ipv6Validator.](#)

Аргументы

<i>parent</i>	
---------------	--

3.4.2 Методы

3.4.2.1 `QValidator::State` `Ipv6Validator::validate (QString & text, int & pos) const`

[Ipv6Validator::validate.](#)

Аргументы

<i>text</i>	
<i>pos</i>	

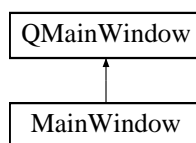
Возвращает

Объявления и описания членов классов находятся в файлах:

- `customvalidator.h`
- `customvalidator.cpp`

3.5 Класс MainWindow

Граф наследования:MainWindow:



Открытые члены

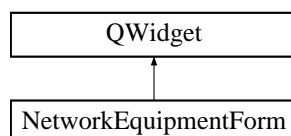
- **MainWindow** (`QWidget *parent=0`)

Объявления и описания членов классов находятся в файлах:

- `mainwindow.h`
- `mainwindow.cpp`

3.6 Класс NetworkEquipmentForm

Граф наследования:NetworkEquipmentForm:



Открытые слоты

- virtual void **setNeedHostname** (bool needHostname)
- virtual void **setHostname** (QString hostname)
- virtual void **setPasswordEncryption** (bool passwordEncryption)
- virtual void **setNeedBannerMotd** (bool needBannerMotd)
- virtual void **setBannerMotd** (QString bannerMotd)
- virtual void **setLoggingSynchronous** (bool loggingSynchronous)
- virtual void **setNeedPasswordConsole** (bool needPasswordConsole)
- virtual void **setPasswordConsole** (QString password)
- virtual void **setNeedPasswordVty** (bool needPasswordVty)
- virtual void **setPasswordVty** (QString password)

Открытые члены

- **NetworkEquipmentForm** (QWidget *parent=0)
- void **setModel** ([NetworkEquipmentModel](#) *model)

3.6.1 Подробное описание

Представление (View) в триаде MVP сетевого оборудования

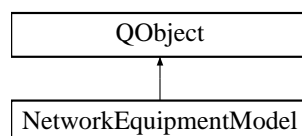
Объявления и описания членов классов находятся в файлах:

- networkequipmentform.h
- networkequipmentform.cpp

3.7 Класс NetworkEquipmentModel

Модель сетевого оборудования

Граф наследования:NetworkEquipmentModel:



Открытые слоты

- virtual void **setNeedHostname** (bool **needHostname**)
Оболочка для установления необходимости называть оборудование
- virtual void **setHostname** (QString **hostname**)
Оболочка для установки нового названия оборудования
- virtual void **setPasswordEncryption** (bool **passwordEncryption**)
Оболочка для изменения необходимости скрывать пароли
- virtual void **setNeedBannerMotd** (bool **needBannerMotd**)
Оболочка для установление необходимости указывать приветственное сообщение
- virtual void **setBannerMotd** (QString **bannerMotd**)
Оболочка для изменения текущего текста приветствия
- virtual void **setLoggingSynchronous** (bool **loggingSynchronous**)
Оболочка для изменения запрета вывода каких-либо консольных сообщений при вводе команд в консольном режиме
- virtual void **setNeedPasswordConsole** (bool **needPasswordConsole**)
Оболочка для изменения необходимости ввода пароля при консольном подключении
- virtual void **setPasswordConsole** (QString **password**)
Оболочка для изменения пароля при консольном подключении
- virtual void **setNeedPasswordVty** (bool **needPasswordVty**)
Оболочка для изменения необходимости ввода пароля при терминальном подключении
- virtual void **setPasswordVty** (QString **password**)
Оболочка для изменения пароля при терминальном подключении

Сигналы

- virtual void **needHostnameChanged** (bool **needHostname**)
- virtual void **hostnameChanged** (QString **hostname**)
- virtual void **passwordEncryptionChanged** (bool **passwordEncryption**)
- virtual void **needBannerMotdChanged** (bool **needBannerMotd**)
- virtual void **bannerMotdChanged** (QString **bannerMotd**)
- virtual void **loggingSynchronousChanged** (bool **loggingSynchronous**)
- virtual void **needPasswordConsoleChanged** (bool **needPasswordConsole**)
- virtual void **passwordConsoleChanged** (QString **password**)
- virtual void **needPasswordVtyChanged** (bool **needPasswordVty**)
- virtual void **passwordVtyChanged** (QString **password**)

Открытые члены

- **NetworkEquipmentModel** (QObject *parent=0)
- virtual bool **needHostname** ()
Оболочка для получения необходимости называть оборудование
- virtual QString **hostname** ()
Оболочка для получения текущего названия оборудования
- virtual bool **passwordEncryption** ()
Оболочка для получения сведений о необходимости скрывать пароли
- virtual bool **needBannerMotd** ()
Оболочка для получения необходимости указывать приветственное сообщение
- virtual QString **bannerMotd** ()
Оболочка для получения текущего текста приветствия
- virtual bool **loggingSynchronous** ()

Оболочка для получения сведений о запрете вывода каких-либо консольных сообщений при вводе команд в консольном режиме

- virtual bool `needPasswordConsole` ()

Оболочка для получения сведений о необходимости ввода пароля при консольном подключении

- virtual QString `passwordConsole` ()

Оболочка для получения пароля при консольном подключении

- virtual bool `needPasswordVty` ()

Оболочка для изменения необходимости ввода пароля при терминальном подключении

- virtual QString `passwordVty` ()

Оболочка для получения пароля при терминальном подключении

3.7.1 Подробное описание

Модель сетевого оборудования

3.7.2 Методы

3.7.2.1 QString NetworkEquipmentModel::bannerMotd () [virtual]

Оболочка для получения текущего текста приветствия

Возвращает

текущий текст приветствия

3.7.2.2 QString NetworkEquipmentModel::hostname () [virtual]

Оболочка для получения текущего названия оборудования

Возвращает

название оборудования

3.7.2.3 bool NetworkEquipmentModel::loggingSynchronous () [virtual]

Оболочка для получения сведений о запрете вывода каких-либо консольных сообщений при вводе команд в консольном режиме

Возвращает

сведения о запрете вывода каких-либо консольных сообщений при вводе команд в консольном режиме

3.7.2.4 `bool NetworkEquipmentModel::needBannerMotd () [virtual]`

Оболочка для получения необходимости указывать приветственное сообщение

Возвращает

необходимость указывать приветственное сообщение

3.7.2.5 `bool NetworkEquipmentModel::needHostname () [virtual]`

Оболочка для получения необходимости называть оборудование

Возвращает

необходимость называть оборудование

3.7.2.6 `bool NetworkEquipmentModel::needPasswordConsole () [virtual]`

Оболочка для получения сведений о необходимости ввода пароля при консольном подключении

Возвращает

необходимость ввода пароля при консольном подключении

3.7.2.7 `bool NetworkEquipmentModel::needPasswordVty () [virtual]`

Оболочка для изменения необходимости ввода пароля при терминальном подключении

Возвращает

необходимость ввода пароля при терминальном подключении

3.7.2.8 `QString NetworkEquipmentModel::passwordConsole () [virtual]`

Оболочка для получения пароля при консольном подключении

Возвращает

пароль при консольном подключении

3.7.2.9 `bool NetworkEquipmentModel::passwordEncryption () [virtual]`

Оболочка для получения сведений о необходимости скрывать пароли

Возвращает

необходимо ли скрывать пароли

3.7.2.10 `QString NetworkEquipmentModel::passwordVty () [virtual]`

Оболочка для получения пароля при терминальном подключении

Возвращает

пароль при терминальном подключении

3.7.2.11 `void NetworkEquipmentModel::setBannerMotd (QString bannerMotd) [virtual], [slot]`

Оболочка для изменения текущего текста приветствия

Аргументы

<i>bannerMotd</i>	новый текст приветствия
-------------------	-------------------------

3.7.2.12 `void NetworkEquipmentModel::setHostname (QString hostname) [virtual], [slot]`

Оболочка для установки нового названия оборудования

Аргументы

<i>hostname</i>	название оборудования
-----------------	-----------------------

3.7.2.13 `void NetworkEquipmentModel::setLoggingSynchronous (bool loggingSynchronous) [virtual], [slot]`

Оболочка для изменения запрета вывода каких-либо консольных сообщений при вводе команд в консольном режиме

Аргументы

<i>loggingSynchronous</i>	сведений о запрете вывода каких-либо консольных сообщений при вводе команд в консольном режиме
---------------------------	--

3.7.2.14 `void NetworkEquipmentModel::setNeedBannerMotd (bool needBannerMotd) [virtual], [slot]`

Оболочка для установление необходимости указывать приветственное сообщение

Аргументы

<i>needBannerMotd</i>	необходимость указывать приветственное сообщение
-----------------------	--

3.7.2.15 `void NetworkEquipmentModel::setNeedHostname (bool needHostname) [virtual], [slot]`

Оболочка для установления необходимости называть оборудование

Аргументы

<i>needHostname</i>	необходимость называть оборудование
---------------------	-------------------------------------

3.7.2.16 `void NetworkEquipmentModel::setNeedPasswordConsole (bool needPasswordConsole) [virtual], [slot]`

Оболочка для изменения необходимости ввода пароля при консольном подключении

Аргументы

<i>needPasswordConsole</i>	необходимость ввода пароля при консольном подключении
----------------------------	---

3.7.2.17 `void NetworkEquipmentModel::setNeedPasswordVty (bool needPasswordVty) [virtual], [slot]`

Оболочка для изменения необходимости ввода пароля при терминальном подключении

Аргументы

<i>needPasswordVty</i>	необходимость ввода пароля при терминальном подключении
------------------------	---

3.7.2.18 `void NetworkEquipmentModel::setPasswordConsole (QString password) [virtual], [slot]`

Оболочка для изменения пароля при консольном подключении

Аргументы

<i>password</i>	новый пароль при консольном подключении
-----------------	---

3.7.2.19 void NetworkEquipmentModel::setPasswordEncryption (bool passwordEncryption)
[virtual], [slot]

Оболочка для изменения необходимости скрывать пароли

Аргументы

<i>passwordEncryption</i>	необходимо ли скрывать пароли
---------------------------	-------------------------------

3.7.2.20 void NetworkEquipmentModel::setPasswordVty (QString password) [virtual], [slot]

Оболочка для изменения пароля при терминальном подключении

Аргументы

<i>password</i>	новый пароль при терминальном подключении
-----------------	---

Объявления и описания членов классов находятся в файлах:

- networkequipmentmodel.h
- networkequipmentmodel.cpp

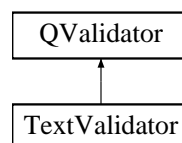
3.8 Шаблон класса QList< T >

Объявления и описания членов класса находятся в файле:

- interfacemodel.h

3.9 Класс TextValidator

Граф наследования:TextValidator:



Открытые члены

- [TextValidator](#) (QObject *parent=0)
[TextValidator::TextValidator.](#)
- State [validate](#) (QString &, int &) const
[TextValidator::validate.](#)

3.9.1 Конструктор(ы)

3.9.1.1 TextValidator::TextValidator (QObject * parent = 0)

[TextValidator::TextValidator.](#)

Аргументы

<i>parent</i>	
---------------	--

3.9.2 Методы

3.9.2.1 QValidator::State TextValidator::validate (QString & text, int & pos) const

[TextValidator::validate.](#)

Аргументы

<i>text</i>	
<i>pos</i>	

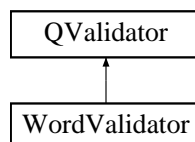
Возвращает

Объявления и описания членов классов находятся в файлах:

- customvalidator.h
- customvalidator.cpp

3.10 Класс WordValidator

Граф наследования:WordValidator:



Открытые члены

- [WordValidator](#) (QObject *parent=0)
[WordValidator::WordValidator.](#)
- State [validate](#) (QString &, int &) const
[WordValidator::validate.](#)

3.10.1 Конструктор(ы)

3.10.1.1 WordValidator::WordValidator (QObject * parent = 0)

[WordValidator::WordValidator.](#)

Аргументы

<i>parent</i>	
---------------	--

3.10.2 Методы

3.10.2.1 `QValidator::State WordValidator::validate (QString & text, int & pos) const`

[WordValidator::validate.](#)

Аргументы

<i>text</i>	
<i>pos</i>	

Возвращает

Объявления и описания членов классов находятся в файлах:

- `customvalidator.h`
- `customvalidator.cpp`

Предметный указатель

- bannerMotd
 - NetworkEquipmentModel, 11
- hostname
 - NetworkEquipmentModel, 11
- InterfaceModel, 5
- InterfaceModelList, 5
- Ipv4Validator, 6
 - Ipv4Validator, 6
 - validate, 6
- Ipv6Validator, 7
 - Ipv6Validator, 7
 - validate, 7
- loggingSynchronous
 - NetworkEquipmentModel, 11
- MainWindow, 8
- needBannerMotd
 - NetworkEquipmentModel, 11
- needHostname
 - NetworkEquipmentModel, 12
- needPasswordConsole
 - NetworkEquipmentModel, 12
- needPasswordVty
 - NetworkEquipmentModel, 12
- NetworkEquipmentForm, 8
- NetworkEquipmentModel, 9
 - bannerMotd, 11
 - hostname, 11
 - loggingSynchronous, 11
 - needBannerMotd, 11
 - needHostname, 12
 - needPasswordConsole, 12
 - needPasswordVty, 12
 - passwordConsole, 12
 - passwordEncryption, 12
 - passwordVty, 13
 - setBannerMotd, 13
 - setHostname, 13
 - setLoggingSynchronous, 13
 - setNeedBannerMotd, 13
 - setNeedHostname, 14
 - setNeedPasswordConsole, 14
 - setNeedPasswordVty, 14
 - setPasswordConsole, 14
 - setPasswordEncryption, 15
 - setPasswordVty, 15
- passwordConsole
 - NetworkEquipmentModel, 12
- passwordEncryption
 - NetworkEquipmentModel, 12
- passwordVty
 - NetworkEquipmentModel, 13
- QList< T >, 15
- setBannerMotd
 - NetworkEquipmentModel, 13
- setHostname
 - NetworkEquipmentModel, 13
- setLoggingSynchronous
 - NetworkEquipmentModel, 13
- setNeedBannerMotd
 - NetworkEquipmentModel, 13
- setNeedHostname
 - NetworkEquipmentModel, 14
- setNeedPasswordConsole
 - NetworkEquipmentModel, 14
- setNeedPasswordVty
 - NetworkEquipmentModel, 14
- setPasswordConsole
 - NetworkEquipmentModel, 14
- setPasswordEncryption
 - NetworkEquipmentModel, 15
- setPasswordVty
 - NetworkEquipmentModel, 15
- TextValidator, 15
 - TextValidator, 16
 - validate, 17
- validate
 - Ipv4Validator, 6
 - Ipv6Validator, 7
 - TextValidator, 17
 - WordValidator, 18
- WordValidator, 17
 - validate, 18
 - WordValidator, 17