Описание структуры управляющего программного обеспечения КИП СПП

# Общее описание

Управляющее программное обеспечение для комплексов измерения параметров силовых полупроводниковых приборов (КИП СПП) предназначено для централизованного управления набором измерительных и вспомогательных блоков, из которых состоит комплекс, реализации задач взаимодействия с пользователем и выдачи необходимых данных на внешние интерфейсы.

Данное программное обеспечение реализовано на базе персонального компьютера с ОС Windows с использованием платформы .NET 4.0. ПО представляет собой взаимосвязанный набор модулей, который может быть запущен на базе одного или нескольких блоков (вариант с выделенным блоком пользовательского интерфейса).

Оглавление

[Общее описание 1](#_Toc426122908)

[Структура программного обеспечения 2](#_Toc426122909)

[Логика работы в рамках комплекса 3](#_Toc426122910)

[Логика работы в рамках сети предприятия 5](#_Toc426122911)

[Удаленная печать протоколов 6](#_Toc426122912)

[Запрос результатов измерений 6](#_Toc426122913)

[Централизованное управление профилями измерений 7](#_Toc426122914)

[Настройки модуля SCME.Service 8](#_Toc426122915)

[Настройки модуля SCME.UI 9](#_Toc426122916)

[Архитектура программного обеспечения комплекса 10](#_Toc426122917)

[Библиотека типов SCME.Types 10](#_Toc426122918)

[Приложение SCME.Service 13](#_Toc426122919)

[Классы для управления оборудованием 15](#_Toc426122920)

[Приложение SCME.UI 16](#_Toc426122921)

[Классы взаимодействия через WCF 17](#_Toc426122922)

[Классы визуального интерфейса 18](#_Toc426122923)

[Классы специализированных элементов управления 19](#_Toc426122924)

[Вспомогательные классы 20](#_Toc426122925)

[Архитектура внешнего программного обеспечения 21](#_Toc426122926)

[Приложение SCME.NetworkPrinting 22](#_Toc426122927)

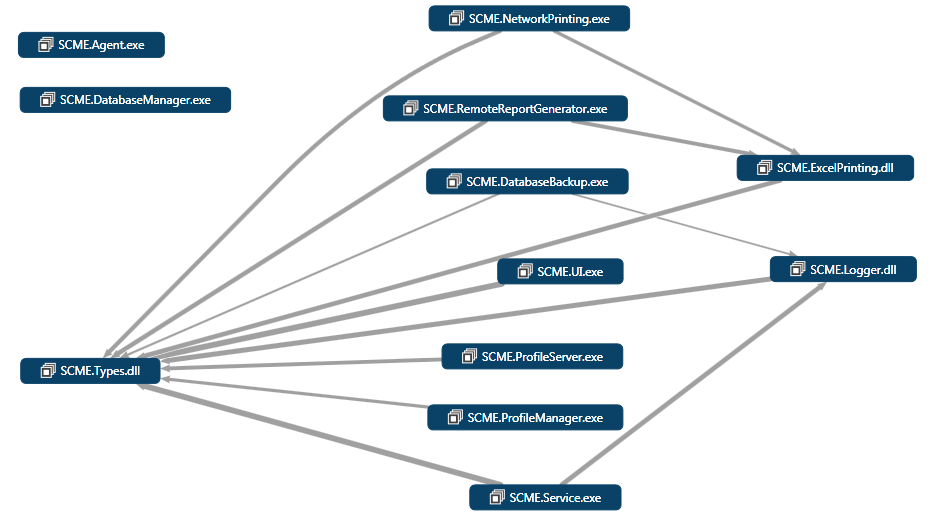
[Приложение SCME.ProfileServer 22](#_Toc426122928)

[Структура БД результатов измерений 23](#_Toc426122929)

# Структура программного обеспечения

Программное обеспечение реализовано на базе платформы .NET и написано в среде MS Visual Studio с использованием C#. Большинство проектов, входящих в структуру ПО объединены в одно решение SCME.

* SCME.Types – библиотека общих типов данных и интерфейсов, используемых совместно различными проектами;
* SCME.Logger – библиотека, предоставляющая функциональность журнала событий для различных приложений;
* SCME.Service – основное управляющее приложение, которое содержит в себе логику инициализации блоков комплекса, проведения измерений и сохранения результатов;
* SCME.UI – приложение, обеспечивающее пользовательский интерфейс комплекса, функциональность профилей измерения и просмотра результатов;
* SCME.NetworkPrinting – приложение, развертываемое на сервере внутренней сети предприятия для реализации функциональности печати результатов измерений;
* SCME.RemoteReportGenerator – приложение, позволяющее получать результаты измерений на комплексе КИП СПП с любого компьютера сети предприятия;
* SCME.ExcelPrinting – библиотека, предоставляющая функции генерации отчетов в формате Excel для SCME.NetworkPrinting и SCME.RemoteReportGenerator;
* SCME.ProfileServer - приложение, развертываемое на сервере внутренней сети предприятия для реализации функциональности централизованного хранения профилей измерений;
* SCME.Agent – вспомогательное приложение для запуска и мониторинга состояния процессов SCME.UI и SCME.Service;
* SCME.ProfileManager – приложение для управления базой профилей измерений (в разработке);
* SCME.DatabaseManager – приложение для инициализации базы данных результатов измерений (устарело);
* SCME.DatabaseBackup – приложение для резервного копирования БД результатов измерений в комплексе (в разработке);
* SCCI\_IO – внешняя библиотека, реализующая интерфейс обмена данными по последовательному интерфейсу с аппаратурой комплекса;



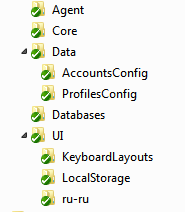
Все перечисленные модули и проекты компилируются для платформы x86 и версии среды исполнения .NET 4.0. Для взаимодействия между удаленными модулями применяются сетевые соединения на основе технологии WCF.

# Логика работы в рамках комплекса

Основным модулем, отвечающим за управление комплексом является SCME.Service. Источником команд для него является клиент, подключающийся к нему по сетевому протоколу и передающий управляющие команды. В случае автономного комплекса с пользовательским интерфейсом роль управляющего клиента реализует SCME.UI, отображающий графический пользовательский интерфейс.

В случае удаленно управляемого комплекса (КИП СПП для АКИМ) приложение SCME.UI не используется и его роль выполняется одним из модулей АКИМ, подключающимся к комплексу удаленно по сетевому интерфейсу.

По умолчанию, запуск SCME.Service и SCME.UI выполняется вспомогательным приложением SCME.Agent, которое следит за их состоянием и перезапускает их в случае непредвиденного сбоя. После запуска, SCME.UI устанавливает сетевое соединение с сервисом и запускает процедуру инициализации. Завершение работы комплекса не предусматривается, достаточно выключить питание комплекса.

 На комплексе программное обеспечение развертывается в следующей структуре:

1. Agent – содержит приложение агента запуска и его конфигурационный файл;
2. Core – содержит приложение SCME.Service;
3. Data -> AccountsConfig – содержит список пользователей комплекса (операторов) и их учетные данные для входа;
4. Data -> ProfilesConfig – содержит список профилей измерений;
5. Databases – содержит файлы баз данных журнала событий и результатов измерений;
6. UI – содержит приложение SCME.UI
7. UI -> KeyboardLayouts – содержит варианты раскладок виртуальной клавиатуры;
8. UI -> LocalStorage – содержит файл хранилища параметров конфигурации блоков;

Помимо основного управляющего сетевого соединения, SCME.Service публикует дополнительный интерфейс, предназначенный для запросов к базе данных результатов измерений. Соединение по этому интерфейсу устанавливается постоянным образом приложением SCME.UI и, дополнительно, внешними приложениями (например, RemoteReportGenerator) при необходимости получения данных.

Схема взаимодействия программного обеспечения комплекса представлена на рисунке ниже:

IExternalControl

SCME.Service

БД

SCME.UI

Профили

Аккаунты

FTDI-адаптер

Интерфейсный адаптер

IDatabaseCommunication

SQLite

RS-232/485

[WCF]

USB

В качестве основы БД применяется технология SQLite, которая обеспечивает следующие преимущества:

* Архитектуру без сервера – движок БД реализован в виде подключаемой библиотеки;
* Поддержку большинства возможностей реляционной модели и SQL;
* Устойчивость к незапланированному выключению питания – полная гарантия сохранения целостности БД;

Последний пункт особенно важен, так как комплекс не предусматривает специальной процедуры выключения, кроме отключения питания. Целостность ОС в данном случае реализуется путем «замораживания» системного раздела диска таким образом, что запуск системы всегда происходит из исходной конфигурации, а любые промежуточные записи кэшируются и не сохраняются в образе тома.

FTDI-адаптер отмечен на схеме светлым цветом, так как является опциональным компонентом, функциональность которого может быть реализована через интерфейсный адаптер в случае если Service и UI исполняются на одном компьютере (совмещенный HMIU).

# Логика работы в рамках сети предприятия

Следующие аспекты функционирования комплекса могут быть интегрированы в настоящий момент в сеть предприятия:

1. Удаленная печать протоколов измерений – по команде из интерфейса измерительного комплекса, протокол измерения выбранной партии приборов будет распечатан на одном из принтеров корпоративной сети;
2. Запрос результатов измерений – данные о результатах измерений можно запросить из внешнего приложения, установленного на любом компьютере, входящем в корпоративную сеть;
3. Централизованное управление профилями измерений – запрос сведений об актуальных профилях измерений для данного комплекса от серверного приложения, развернутого в сети, вместо использования локальных данных (разработка не завершена).
4. Резервное копирования БД результатов измерений – извлечение данных из БД результатов измерений комплекса для их резервного копирования в центральную БД. Реализуется в виде стороннего приложения, развертываемого на компьютере сети предприятия (разработка не завершена).

SCME.Service

БД

SCME.UI

Профили

Remote Report Generator

Database Backup

Network Printing

Profile Server

*Сервер*

INTRANET

## Удаленная печать протоколов

Данная функциональность реализуется следующим образом – по запросу оператора из приложения пользовательского интерфейса передается команда модуля Service, содержащая информацию о настройках печати и целевой партии приборов. Модуль Service устанавливает связь с сервером печати NetworkPrinting и передает ему указанную информацию и идентификатор комплекса.

После получения запроса, сервер печати определяет в своих настройках принтер для печати отчета, сопоставленный с данным комплексом и настройки формирования отчета. После этого осуществляет обратное соединение с измерительным комплексом по интерфейсу IDatabaseCommunication и производится запрос необходимых данных. Далее осуществляется формирование отчета и отправка его на принтер при помощи библиотеки ExcelPrinting.

Команда печати

Передача команды на Service

Установление соединения с NetworkPrinting

Передача команды на сервер

Обработка запроса

Установление соединения с Service

Извлечение данных

Формирование отчета

## Запрос результатов измерений

Для запроса данных о результатах измерений применяется приложение RemoteReportGenerator, которое подключается к выбранному комплексу и производится запрос данных об измеренных партиях в заданный пользователем период.

Для формирования отчета производится запрос данных о выбранной партии, полученные результаты форматируются в виде отчета MS Excel для распечатки, либо сохранения в файле.

Выборка списка партий партий

Обработка списка

Обработка данных

Завершение соединения с Service

Формирование отчета

Установление соединения с комплексом

Выборка данных по партии

## Централизованное управление профилями измерений

Для синхронизации профилей измерения между различными комплексами и централизованного управления ими может применять система с центральным сервером, который хранит словарь профилей измерений и его конфигурацию для конкретного измерительного комплекса.

При старте измерительный комплекс пробует получить список измерительных профилей с сервера профилей. В случае удачной попытки данный список принимается в работу и сохраняется на комплексе в виде локальной копии. В случае неудачной попытки загрузки с сервера для работы используется локальная копия.

Старт

Установление соединения с ProfileServer

Успешное соединение

Загрузка списка профилей

Сохранение в виде локальной копии

Использование последней успешной копии

# Настройки модуля SCME.Service

Как описано выше, данное приложение обеспечивает управление всеми блоками измерительного комплекса. Коммуникация между блоками осуществляется посредством шины CAN, обеспечивающей гальванически изолированную отказоустойчивую среду обмена данными. Доступ управляющего компьютера к этой шине производится через промежуточный интерфейсный контроллер, подсоединенный к компьютеру по последовательному интерфейсу. Протокол обмена данными с интерфейсным контроллером специализированный, базирующийся на протоколе MODBUS, и обеспечивается библиотекой SCCI\_IO.

Для внешней конфигурации (SCME.Service.exe.config) доступны следующие настройки приложения:

* GateNode – адрес блока измерения характеристик управления (Gate) на шине CAN;
* SLNode – адрес блока измерения статических потерь (SL) на шине CAN;
* BVTNode – адрес блока измерения блокирующих характеристик (BVT) на шине CAN;
* CommutationNode – адрес блока коммутации (Commutation) на шине CAN;
* AdapterPort – адрес COM-порта для соединения с интерфейсным контроллером;
* GatewayNode – адрес интерфейсного контроллера (Gateway) на шине CAN;
* IsCommutationType6 – флаг, задающий 2-х или 6-ти позиционный коммутационный блок используется;
* CommutationExNode – адрес дополнительного блока коммутации (CommutationEx) на шине CAN;
* CommutationExEmulation – флаг эмуляции блока CommutationEx;
* IsCommutationExType6 – флаг, задающий 2-х или 6-ти позиционный дополнительный коммутационный блок используется;
* ClampingSystemEmulation – флаг эмуляции пресса;
* ClampingSystemNode – адрес пресса (ClampingSystem) на шине CAN;
* IsClampingSystemConnected – флаг, задающий использование пресса в данном комплексе;
* DBOptionsResults – опции подключения к БД результатов;
* DBOptionsLogs – опции подключения к БД журнала событий;
* IncludeDetailsInLog – задает детализацию ведения журнала событий;
* ForceLogFlush – записывать в журнал события не используя кэш;
* GateReadGraph – читать осциллограммы с модуля Gate;
* SLReadGraph – читать осциллограммы с модуля SL;
* BVTReadGraph – читать осциллограммы с модуля BVT;
* DisableResultDB – выключить запись в БД результатов;
* DisableLogDB – выключить запись в БД событий;
* MMECode – идентификатор комлпекса;
* CommutationEmulation – эмуляция блока Commutation;
* GatewayEmulation – эмуляция блока Gateway;
* AdapterEmulation – эмуляция подключения к интерфейсному адаптеру;
* LogsDatabasePath – путь к БД журнала событий;
* ResultsDatabasePath – путь к БД журнала результатов;
* BVTEmulation – эмуляция блока BVT;
* GateEmulation – эмуляция блока Gate;
* SLEmulation – эмуляция блока SL;
* LogsTracePathTemplate – путь к файлу текстового журнала событий;
* DBServiceExternalEndpoint – точка подключения внешнего интерфейса запроса данных;
* dVdtNode– адрес блока критической скорости нарастания напряжения (dVdt) на шине CAN;
* dVdtEmulation – эмуляция блока dVdt;

# Настройки модуля SCME.UI

Данный модуль осуществляет реализацию пользовательского интерфейса и управление приложением SCME.Service по протоколу WCF.

Для внешней конфигурации (SCME.UI.exe.config) доступны следующие настройки приложения:

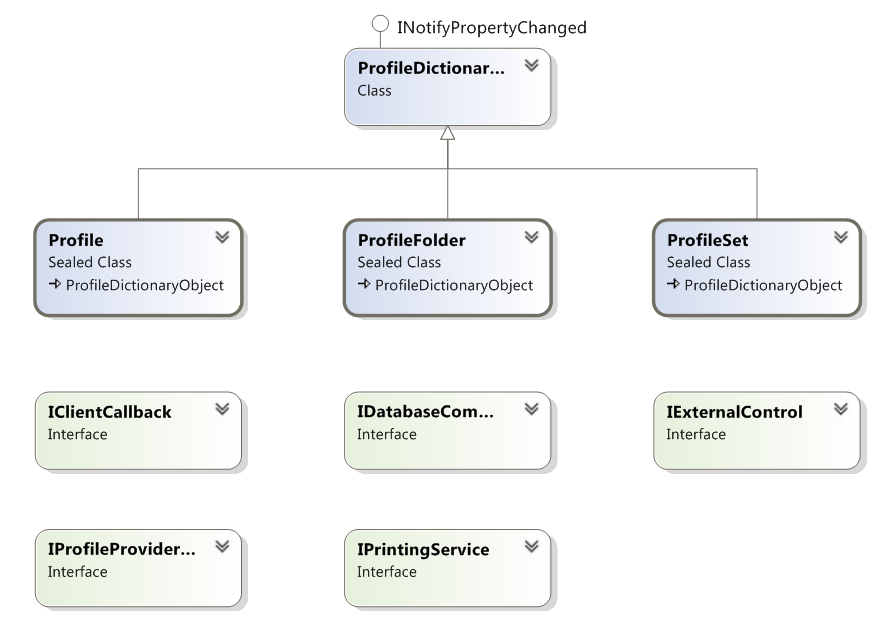
* GateIsVisible – блок Gate включен в работу;
* SLIsVisible – блок SL включен в работу;
* BvtIsVisible – блок BVT включен в работу;
* CommIsVisible – блок Commutation включен в работу;
* ExitButtonIsVisible – отображается кнопка закрытия приложения UI;
* IsAnimationEnabled – включена анимация элементов управления;
* TechPassword – пароль для входа в режим наладчика;
* Localization – текущая локализация приложения;
* PlotUserVTM – отрисовывать график результатов VTM в режиме измерений оператором;
* PlotUserGate – отрисовывать график результатов Gate в режиме измерений оператором;
* PlotUserBVT – отрисовывать график результатов BVT в режиме измерений оператором;
* IsTechPasswordEnabled – запрос пароля при входе в режим наладчика;
* KeyboardsPath – путь к файлу с раскладками клавиатур;
* BVTVoltageFrequency – частота напряжения для блока BVT;
* SinglePositionModuleMode – режим работы с однопозиционными модулями;
* FTDIPresent – присутствует плата расширения на базе чипа FTDI;
* CommExIsVisible – блок CommutationEx включен в работу;
* IgnoreSC – игнорировать цепь безопасности;
* RunExplorer – запускать Explorer при закрытии приложения;
* ProfilesPath – путь к файлу профилей;
* AccountsPath – путь к файлу с учетными записями операторов;
* FTDIIsInUse – плата расширения FTDI используется в работе (обработка светодиодов и кнопок START-STOP на КИП СПП-002);
* ClampIsVisible – блок ClampingSystem включен в работу;
* NormalWindow – отображать в нормальном режиме вместо полноэкранного;
* UseVTMPostProcessing – использовать постобработку результатов VTM;
* StoragePath – путь к файлу локального хранилища параметров;
* dVdtIsVisible – блок dVdt включен в работу;
* UseLocalProfiles – использовать данные профилей измерений из локального файла, а не с сервера профилей;
* MMECode – идентификатор комплекса;

# Архитектура программного обеспечения комплекса

## Библиотека типов SCME.Types

Помимо большого количества классов, описывающих технические детали обмена информацией между приложением интерфейса (UI) и сервисом (Service), в данной библиотеке присутствуют следующие важные типы данных:

* IExternalControl – задает интерфейс управления приложением Service;
* IDatabaseCommunication – задает интерфейс запроса данных из БД измерений;
* IClientCallback – задает интерфейс оповещения управляющего приложения об изменении состояния Service;
* IPrintingService – интерфейс запроса удаленной печати;
* IProfileProvider – интерфейс запроса списка профилей измерений с сервера профилей;
* ProfileDictionaryObject – абстрактный класс, предоставляет общую функциональность для элементов словаря профилей измерений;
* Profile – профиль измерений, содержащий информацию о проводимых измерениях, условиях проведения и нормативов на полученные значения.
* ProfileSet – элемент словаря профилей, объединяющий историю изменения параметров профиля;
* ProfileFolder – элемент словаря профилей, объединяющий профили по произвольным категориям;



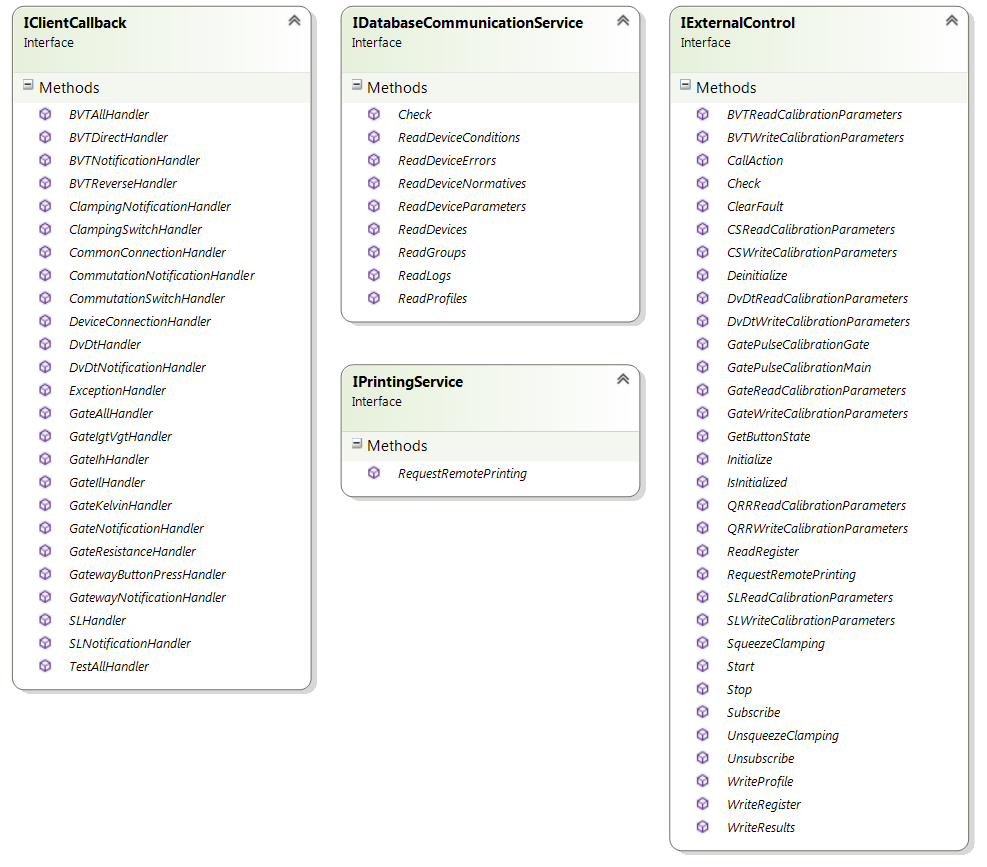
Интерфейс **IExternalControl** содержит список методов, условно группируемых по своему отношению к различным измерительным блокам. Это относится к методам, начинающимся с префиксов SL, BVT, Gate, CS, QRR и пр. Существуют также методы, управляющие общей работой измерительного комплекса: *Initialize*, *Start*, *Stop*. Результаты измерений записываются при помощи методов *WriteProfile*, *WriteResults*.

Интерфейс **IDatabaseCommunication** предоставляет возможность запроса данных о результатах измерений с помощью методов:

* *ReadGroup* – список партий за некоторый период;
* *ReadDevices* – список приборов в заданной партии;
* *ReadDeviceParameters* – результаты измерения прибора;
* *RadDevuceNormatives* – нормативы на результаты измерения;
* *ReadDeviceConditions* – условия измерения приборов;
* *ReadDeviceErrors* – ошибки измерения прибора;

Интерфейс **IClientCallback** используется для функций обратного вызова в паре с интерфейсом IExternalControl. С его помощью Service уведомляет ведущее приложения (UI) об изменениях состояния. Методы также можно сгруппировать по именам, отнеся их к различным измерительным блокам.

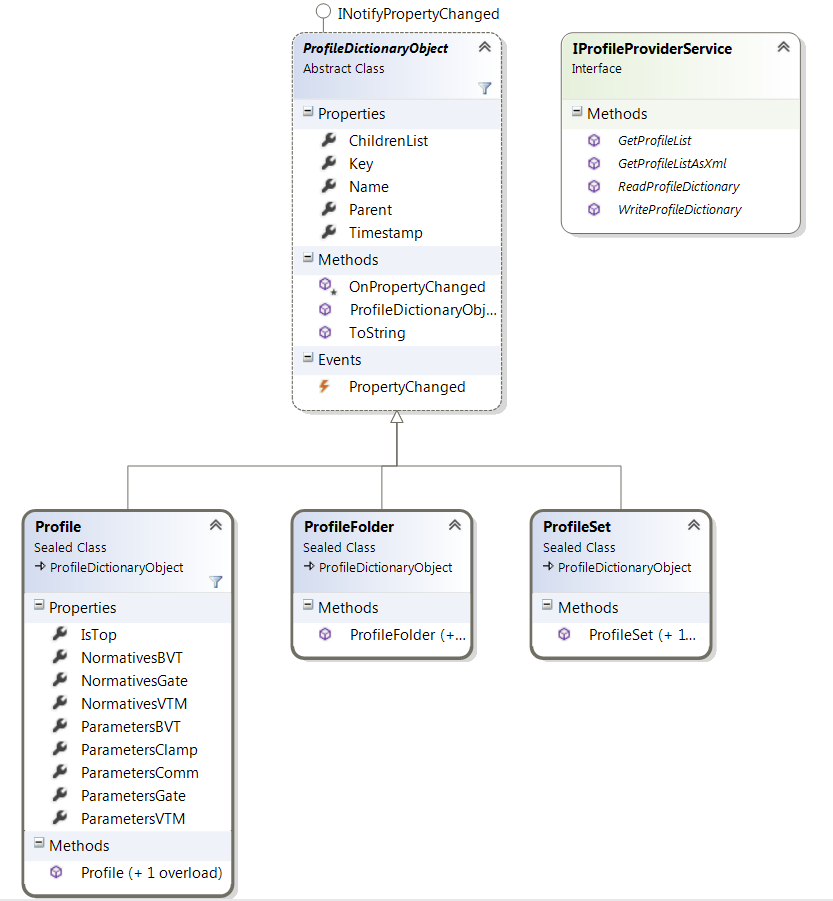
Интерфейс **IPrintingService** содержит единственный метод *RequestRemotePrinting*, который передает всю необходимую информацию: название партии, идентификатор комплекса, настройки печати.



Структура семейства классов Profile ориентирована на иерархическую структуру хранения. Базовый класс предоставляет свойства для хранения имени элемента, его GUID, времени создания (необходимо для версионирования профилей с помощью ProfileSet), ссылки на родителя дерева и коллекции дочерних элементов.

Класс **Profile** содержит поля для хранения *Normatives{}*, *Parameters{}* для различных блоков.

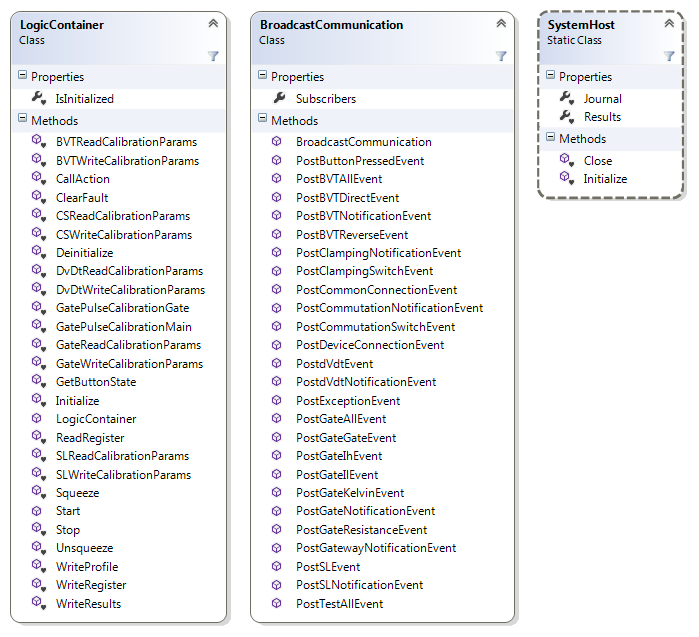
Интерфейс **IProfileProviderService** предназначен для получения списка профилей измерений с сервера. Данная функция реализуется при помощи метода *GetProfileList*. Остальные функции нужны для редактирования серверного словаря профилей при помощи приложения SCME.ProfileManager (в разработке).



## Приложение SCME.Service

Основная функциональность данного приложения содержится в трех группах классов:

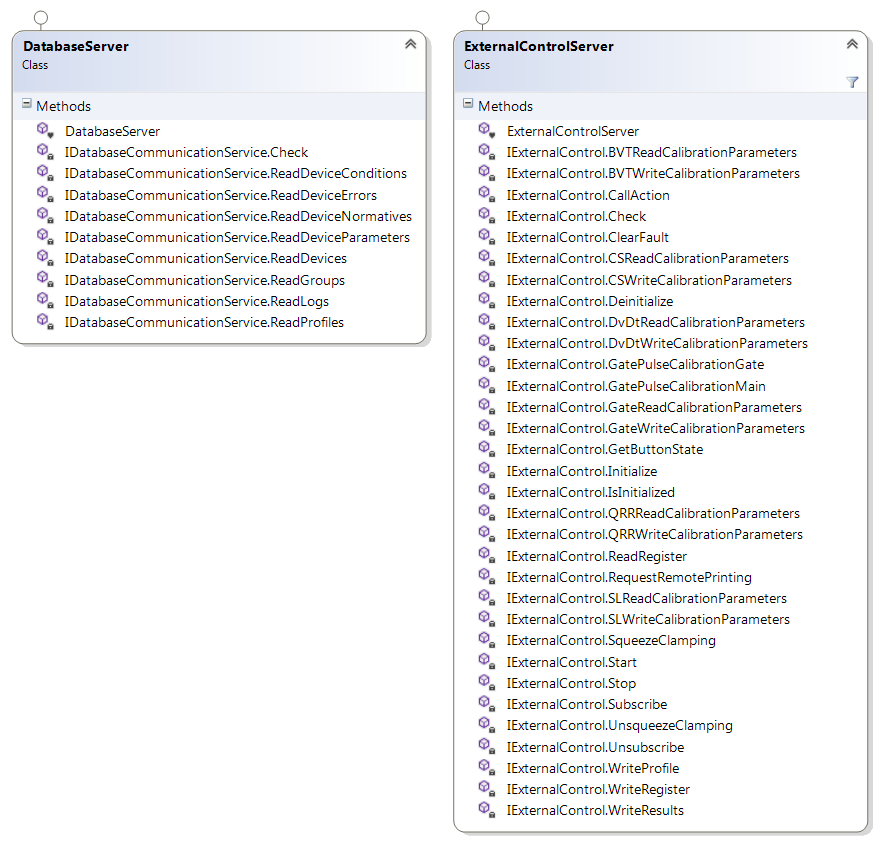
1. Классы основного функционала: LogicContainer, BroadcaseCommunication, SystemHost
2. Классы взаимодействия с управляющим приложением: DatabaseServer, ExternalControlServer
3. Классы взаимодействия с оборудованием: IOAdapter, IOGateway, IOCommutation, IOGate, IOStLs, IOBvt, IOClamping



Статический класс **SystemHost** отвечает за инициализацию журнала событий, базы данных результатов и запуск приложения. После запуска приложения Service, внутреннее управление передается классу LogicContainer.

Класс **LogicContainer** содержит реализацию функциональности IExternalControl, передаваемую через ExternalControlServer (см. ниже). Последовательность процесса измерения и логика реализованы в паре методов Start/Stop.

Класс **BoardcastCommunication** предназначен для трансляции сообщений об изменении состояния Service через интерфейс IClientCallback. Данное сообщение рассылается всем клиентам, подписавшимся на обновления (интерфейс предусматривает возможность получения информации мониторинговыми сервисами).

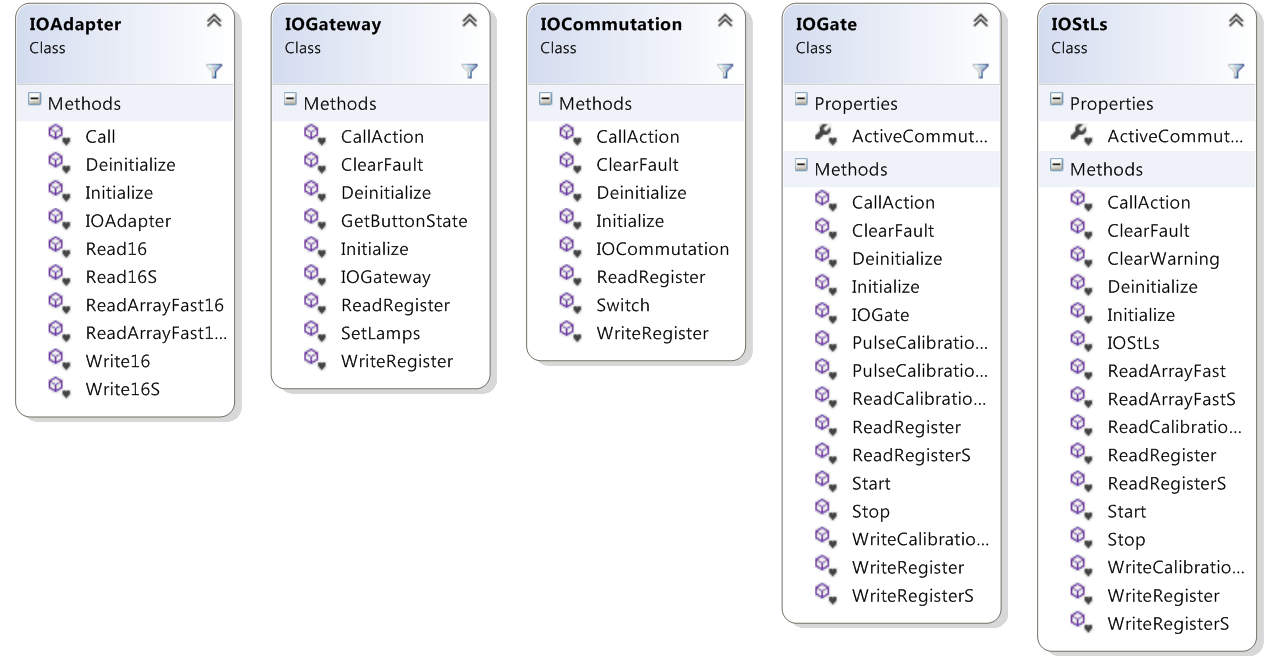


Класс **ExternalControlServer** непосредственно реализует интерфейс IExternalControlServer, выполняя технические действия по обработке запросов, необходимые для инфраструктуры WCF (обработка и упаковка исключений). Логика работы **DatabaseServer** аналогична – функциональность реализуется классом LogicContainer, а этот класс обеспечивает выполнение требований сетевой инфраструктуры.

### Классы для управления оборудованием

Каждому блоку измерительного комплекса соответствует специальный класс, реализующий необходимые функции через запросы от LogicContainer.

1. IOAdapter – реализует функциональность подключения к интерфейсному адаптеру и обмен данными с блоками комплекса через шину;
2. IOGateway – доступ к интерфейсному адаптеру как самостоятельному устройству и чтение его портов ввода-вывода;
3. IOCommutation – представляет функциональность блока коммутации;
4. IOGate – представляет функциональность блока измерения управления;
5. IOStLs – представляет функциональность блока измерения статических потерь;
6. IOBvt – представляет функциональность блока классификатора;
7. IOClamping – представляет функциональность пресса;



## Приложение SCME.UI

Данное приложение предоставляет графический пользовательский интерфейс для управления комплексом со стороны оператора.

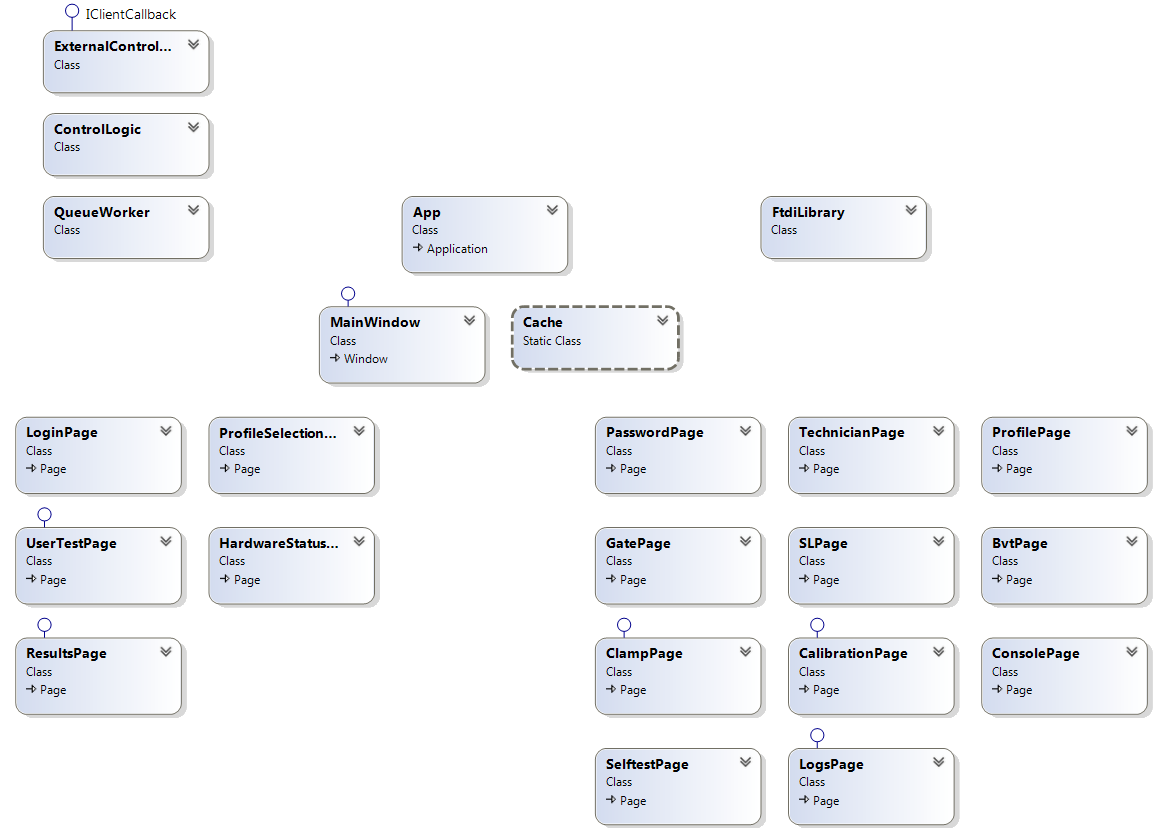
Интерфейс приложения реализован при помощи технологии WPF для обеспечения удобного использования на устройствах с сенсорным экраном. Коммуникация с приложением SCME.Service осуществляется по двум каналам: IExternalControl – для обеспечения функций управления, IDatabaseCommunication – для обеспечения функций получения данных о результатах измерений.

Структура проекта SCME.UI состоит из следующих групп классов:

1. Классы взаимодействия с SCME.Service через механизмы WCF: ControlLogic, ExternalControlCallbackHost, QueueWorker;
2. Классы визуального интерфейса пользователя;
3. Классы специализированных графических элементов управления;
4. Вспомогательные классы;

В соответствии с настройками в конфигурационном файле приложение формирует необходимые страницы пользовательского интерфейса, включая их конфигурирование под одно- или двухпозиционный режим измерений.

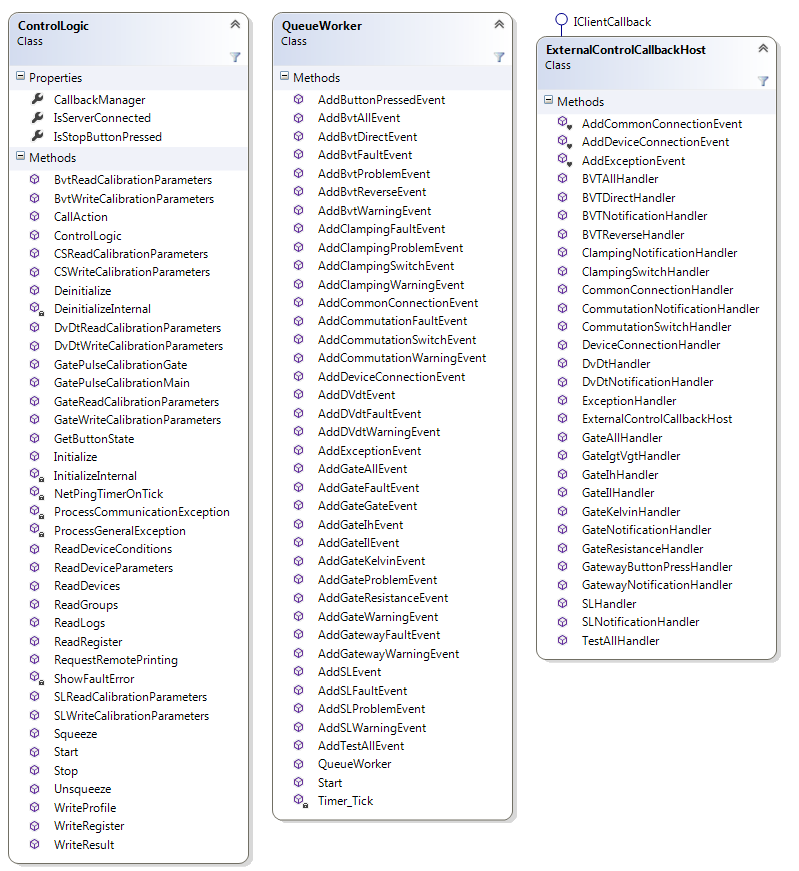
### 



### Классы взаимодействия через WCF

Функции установления соединения с приложением SCME.Service реализуются классом **ControlLogic**. Данный класс выполняет установление двух соединений с сервисным приложением и проведение процедуры инициализации комплекса. Для получения обновлений состояния сервисного приложения используется интерфейс обратного вызова IClientCallback, который реализуется классом **ExternalControlCallbackHost**. Для трансляции этих вызовов в пользовательский интерфейс применяется класс **QueueWorker**. Данный механизм необходимо для того, чтобы обновления пользовательского интерфейса выполнялись в основном потоке приложения, в то время как обработка сообщений WCF происходит в служебных потоках.

Список методов, реализуемых классом **ControlLogic** в целом совпадает с методами интерфейс IExternalControl. Метод *NetPingTimerOnTick* предназначен для регулярного обновления состояния коммуникационного канала и детектирования обрыва связи. Функции установления связи и запуска процесса инициализации оборудования реализуются в методах *Initialize* и *InitializeInternal*, которые визуализируют состояние этого процесса через обратные вызовы и экран HardwareStatusPage (см. ниже). Функции выключения блоков реализуются методами *Deinitialize* и *DeinitializeInternal*. Метод *ProcessCommunicationException* предназначен для обработки ситуации потери связи с сервисным приложением и проведения процедуры повторной установки соединения с учетом того, что в случае сбоя сервисное приложение будет перезапущено агентом.



Класс **QueueWorker** осуществляет добавление методов обратного вызова, инициированных классом ExternalControlCallbackHost во внутреннюю очередь, которая опустошается в процедуре обработки внутреннего таймера *Timer\_Tick*.

Класс **ExternalControlCallbackHost** реализует интерфейс IClientCallback и транслирует вызовы с минимальной внутренней обработкой в QueueWorker.

### Классы визуального интерфейса

Визуальный интерфейс приложения организован при помощи различных функциональных вкладок (наследников класса Page), отображаемых во фрейме главного окна **MainWindow**. Для ускорения процесса старта приложения вкладки не инстанцируются при запуске, а создаются динамически по мере необходимости. Этим процессом (инстанцирование и хранение созданных объектов) управляет статический класс **Cache**.

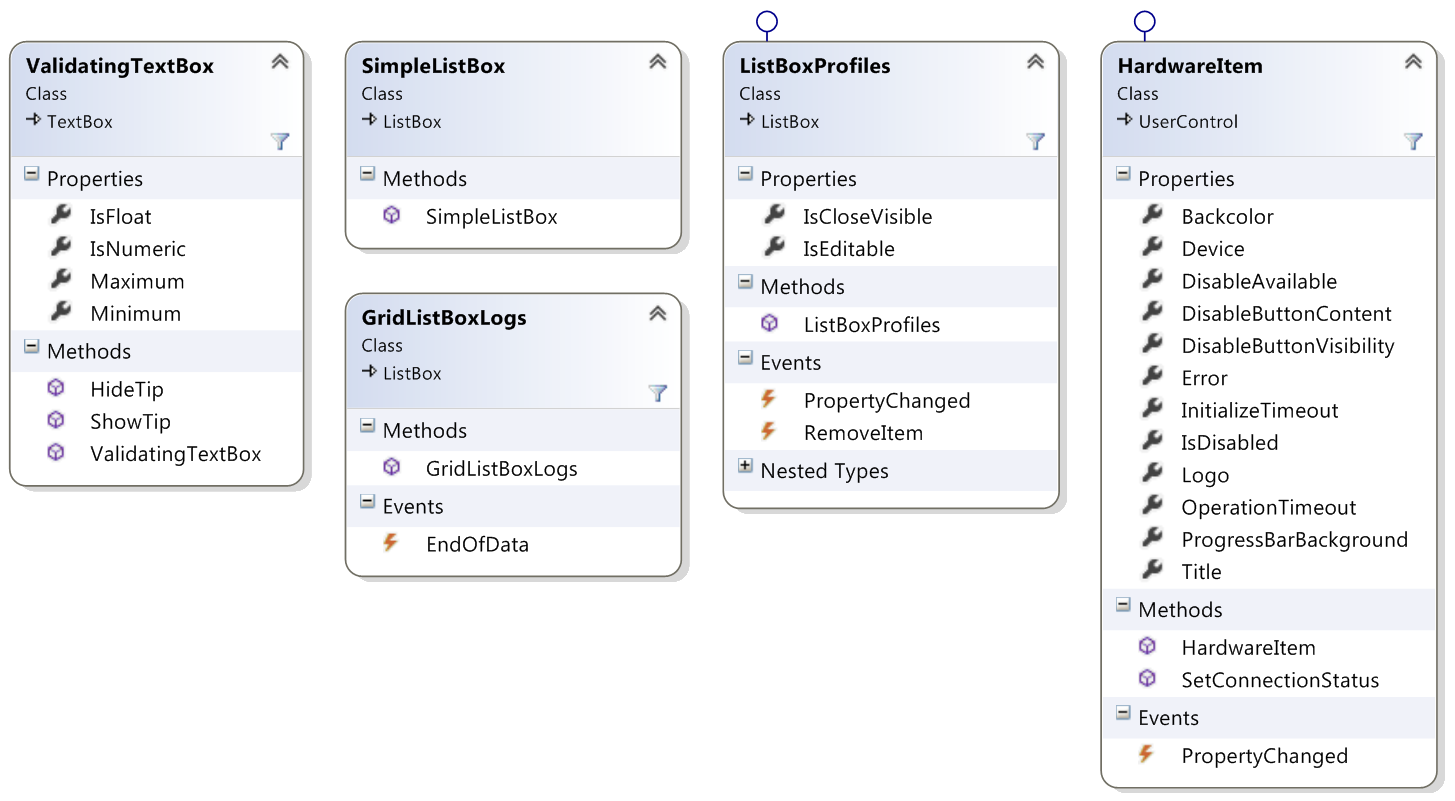
Набор вкладок можно разделить на две группы: вкладки интерфейса оператора и интерфейса наладчика:

1. Интерфейс оператора и общие
   1. **LoginPage** – страница выбора учетной записи оператора и его авторизации;
   2. **ProfileSelectionPage** – страница выбора профиля измерений;
   3. **UserTestPage** – страница проведения поточных измерений приборов;
   4. **ResultsPage** – страница запроса удаленной печати результатов;
   5. **HardwareStatusPage** – страница, отображающая процесс инициализации комплекса;
2. Интерфейс наладчика
   1. **PasswordPage** – страница ввода пароля для входа в режим наладчика;
   2. **TechnicianPage** – страница меню режима наладчика;
   3. **ProfilePage** – страница редактирования локального списка профилей измерений;
   4. **GatePage** – страница измерения характеристик управления в расширенном режиме;
   5. **SLPаge** – страница измерения статических потерь в расширенном режиме;
   6. **BVTPage** – страница измерения блокирующих характеристик в расширенном режиме;
   7. **ClampPage** – страница управления зажимным устройством (прессом);
   8. **CalibratinoPage** – страница доступа к параметрам калибровки блоков;
   9. **ConsolePage** – страница, обеспечивающая функционал прямого доступа к таблицам регистров блоков;
   10. **SeftTestPage** – страница процедуры самотестирования блоков;
   11. **LogsPage** – страница доступа к записям журнала событий;

### Классы специализированных элементов управления

Специализированные элементы управления созданы для специфических задач отображения данных:

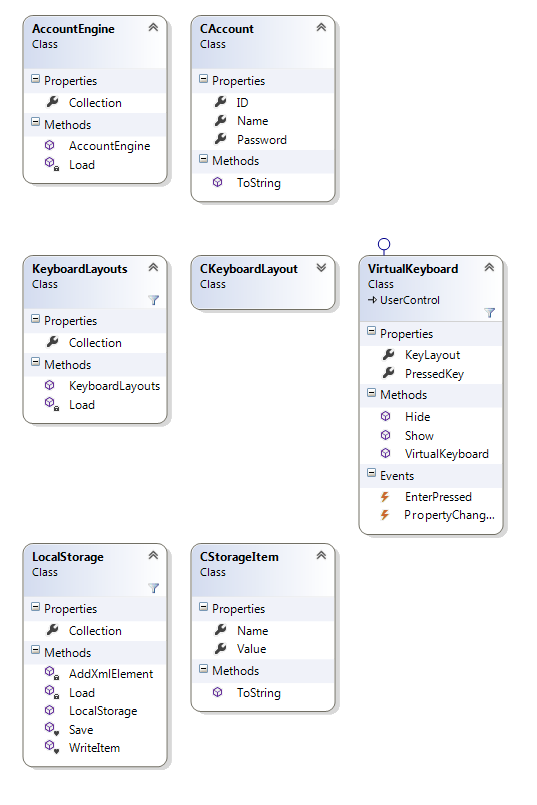
1. **ValidatingTextBox** – предоставляет функциональность полу ввода данных с контролем вводимых значений через свойства *Maximum* и *Minimum*. Формат данных задается свойствами *IsNumeric* и *IsFloat*.
2. **SimpleListBox** – задает простой стилизованный списочный элемент управления, поддерживающий связывание данных через отображение списка строк;
3. **ListBoxProfiles** – список профилей, поддерживающий режим удаления элементов и оповещающий об этом событии через *RemoveItem*;
4. **GridListBoxLogs** – список записей из журнала событий, поддерживающий автоматическую подгрузку содержимого по событию *EndOfData*;
5. **HardwareItem** – графический элемент, отображающий состояние инициализации блока комплекса. Содержит свойства, позволяющие отображать кнопку, отключающую данный блок в текущей сессии работы – *DisableAvailable*, *DisableButtonContent*, *DisableButtonVisibility*. Позволяет управлять внешним видом путем задания свойств *Logo*, *Title*, *Backcolor*, *ProgressBarBackground*. Режим инициализации задается через свойства *OperationTimeout* и *InitializeTimeout*.



### Вспомогательные классы

Вспомогательные классы обеспечивают техническую реализацию различных функций приложения:

1. **AccountEngine** – обеспечивает загрузку списка аккаунтов пользователей, зарегистрированных для работы на данном измерительном комплексе;
2. **CAccount** – представляет собой пользовательский аккаунт для класса AccountEngine с указание имени (*Name*) и пароля (*Password*);
3. **KeyboardLayouts** – обеспечивает загрузку списка раскладок клавиатур;
4. **CKeyboardLayout** – представляет собой локализованную раскладку клавиатуры;
5. **VirtualKeyboard** – виртуальная клавиатура для сенсорного монитора;
6. **LocalStorage** – обеспечивает загрузку списка сохраняемых параметров настройки блоков комплекса;
7. **CStorageItem** – представляет собой сохраняемый параметр;



# Архитектура внешнего программного обеспечения

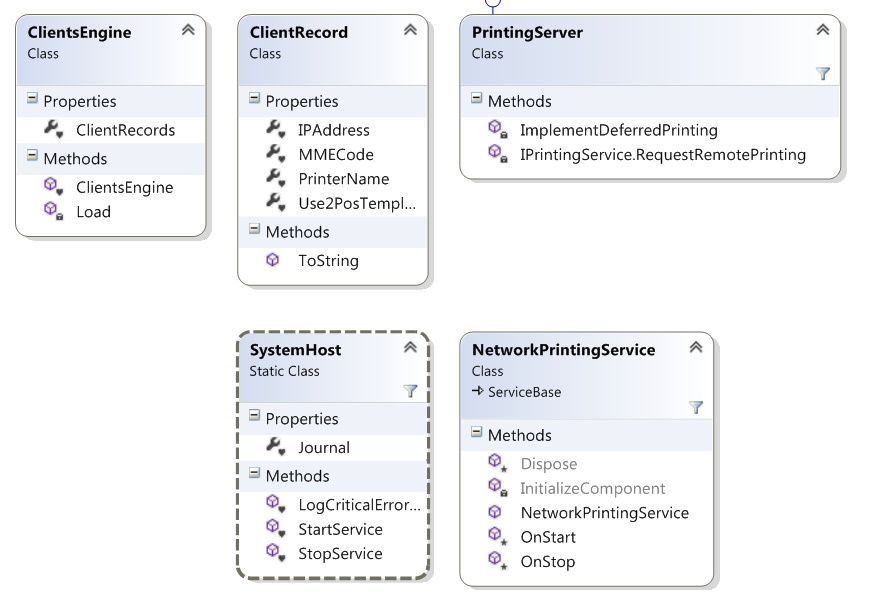
Внешнее программное обеспечение представлено приложениями RemoteReportGenerator, ProfileManager, NetworkPrinting и библиотекой ExcelPrinting.

## Приложение SCME.NetworkPrinting

Данное приложение может быть запущено как в интерактивной сессии пользователя, так и в качестве службы Windows. Отчеты в формате MS Excel формируются при помощи библиотеки **ExcelPrinting**.

При запуске приложение открывает на прослушку интерфейс WCF IPrintingService, который реализуется классом **PrintingServer**. Пара классов **ClientEngine** и **ClientRecord** представляют собой коллекцию зарегистрированных измерительных комплексов в сети предприятия, которая загружается их конфигурационного файла.

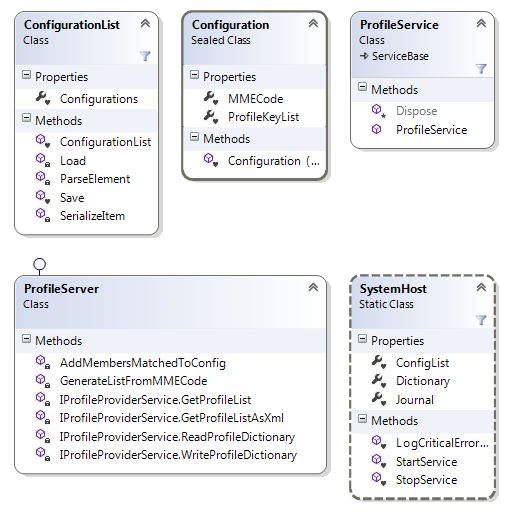
Класс **NetworkPrintingService** предназначен для реализации функции запуска приложения в режиме службы.



## Приложение SCME.ProfileServer

Данное приложение может быть запущено как в интерактивной сессии пользователя, так и в качестве службы Windows.

Пара классов **ConfigurationList** и **Configuration** представляют собой набор настроек списков профилей для комплексов в сети предприятия. Интерфейс IProfileProviderService реализуется классом **ProfileServer**, обеспечивающим реализацию соответствующего интерфейса через технологию WCF.



# Структура БД результатов измерений

База данных результатов измерений содержит в себе данные о результатах тестирования приборов и модулей на измерительном комплексе, включая данные о условиях проведения измерений и нормативах на измеряемые параметры.

Так как для хранения данных выбрана реляционная схема, то для достижения 3-ей нормальной формы, структура данных представлена в виде 10 таблиц, из них 4 предназначены для декомпозиции связей типа «многие-ко-многим».

Список таблиц:

1. **GROUPS** – содержит данные о партиях модулей;
   1. GROUP\_NAME – идентификатор партии модулей или приборов;
2. **DEVICES** – содержит данные об измеряемом модуле
   1. CODE – идентификатор прибора или модуля;
   2. SIL\_N\_1 – данные о партии ППЭ;
   3. SIL\_N\_2 – данные о номере ППЭ;
   4. TS – время измерения и дата;
   5. USR – имя аккаунта оператора;
   6. POS – номер позиции прибора (для приборов с двумя позициями в данной таблице содержатся две записи);
3. **PARAMS** – параметры измеряемые на приборе
   1. PARAM\_NAME– название параметра;
   2. PARAM\_NAME\_LOCAL – локализованное название;
   3. PARAM\_IS\_HIDE – параметр не отображается в отчет;
4. **PROFILES** – данные о профилях измерений
   1. PROF\_NAME – отображаемое имя (не уникально, так как профили могут модифицироваться);
   2. PROF\_GUID – уникальный идентификатор профиля (отражает одну из версий профиля измерений с заданным именем);
   3. PROF\_TS - время измерения профиля и дата;
5. **CONDITIONS** – задает условия проведения измерений
   1. COND\_NAME– имя условия;
   2. COND\_NAME\_LOCAL – локализованное имя условия;
   3. COND\_IS\_TECH – флаг того, что условие является техническим параметром измерительного комплекса;
6. **ERRORS** – ошибки измерений
   1. ERR\_CODE – код ошибки;
   2. ERR\_NAME – интернациональное название;
   3. ERR\_NAME\_LOCAL – локализованное название;
7. **PROF**\_**COND** – значения заданного условия измерений в указанном профиле;
   1. VALUE – значения условия;
8. **PROF\_PARAM** – нормативы на заданный параметр в указанном профиле;
   1. MIN\_VAL – минимальное значение параметра;
   2. MAX\_VAL – максимальное значение параметра;
9. **DEV**\_**PARAM** – значение заданного параметра в указанном приборе;
   1. VALUE – значение параметра;
10. **DEV**\_**ERR** – присутствие заданного кода ошибки для указанного прибора;

