

FL MMOG

Версия защиты 5.7

Руководство пользователя

Оглавление

1	Продукт FL MMOG	4
2	Схема установки защиты FL MMOG при разработке и выпуске продукта	5
3	Защита продукта и тестирование	6
3.1	Начало работ по защите	6
	<i>Установка Protection Studio</i>	6
	<i>Запуск Protection Studio</i>	7
3.2	Установка защиты	11
	<i>Интерфейс Protection Studio</i>	11
	Команды меню	11
	Кнопки панели инструментов	14
	Использование быстрых клавиш	14
	Рабочая область	14
	Отображение процесса защиты	15
	<i>Задание параметров защиты</i>	15
	Защищаемые файлы	16
	Папка для выходных данных	20
	<i>Полное описание настроек проекта защиты</i>	21
	Раздел "Проект"	22
	Раздел "Учетная запись"	23
	Раздел "Базовые параметры"	24
	Раздел "Файлы"	25
	Раздел "Функции"	32
	Раздел "Параметры пользовательского интерфейса"	39
	Раздел "Дополнительные параметры"	42
	<i>Защита файлов</i>	43
	<i>Формирование дистрибутива</i>	45
	<i>Запуск сервисов из командной строки</i>	46
3.3	Тестирование защищенного приложения	47
3.4	StarForce SDK	48
	<i>Состав SDK</i>	48
	<i>Генерация SDK</i>	49
	<i>Применение секретных классов</i>	50
	<i>Функции API защиты</i>	51
	<i>Интеграция SDK в разрабатываемое приложение</i>	55
	<i>Пример использования SDK</i>	58
3.5	Дополнительные возможности защиты	61
	<i>Настройка GUI</i>	61
	Назначение модифицируемого GUI	61
	Рекомендации по написанию индивидуальных GUI-проектов	61
	<i>Редактор сообщений</i>	62
	Интерфейс приложения	62
	Команды меню	62
	Кнопки панели команд	63
	Использование быстрых клавиш	63
	Дерево сообщений и языков	63
	Функции редактора	64

4 Подготовка к выпуску и выпуск продукта	66
4.1 Изменение настроек системы защиты.....	66
4.2 Дополнительная возможность инсталляции защищенного продукта.....	68
5 Сопровождение продукта	70
5.1 Драйвер защиты.....	70
<i>Установка драйвера защиты на компьютере конечного пользователя</i>	<i>70</i>
5.2 Обновление защищенного продукта. Защита обновленных версий.....	71
6 Диагностика. Устранение ошибок	72
6.1 Диагностика и устранение ошибок при установке защиты.....	72
6.2 Диагностика и устранение ошибок при запуске защищенного приложения.....	74
6.3 Рекомендации по размещению файлов библиотек защиты.....	78
7 Глоссарий	82
8 Техподдержка	84
Предметный указатель	85

1 Продукт FL MMOG

Продукт FrontLine MMOG представляет собой комплекс, направленный на защиту MMO игр от целого спектра угроз: FL MMOG обеспечивает защиту игровых серверов от анализа, модификации и возможности запуска на неавторизованных площадках, защиту кода игры от анализа и взлома, защиту от запуска и выполнения чит-программ и ботов, защиту трафика между сервером MMOG и клиентом.

Особенности защиты FL MMOG:

- Затруднение неинвазивных и инвазивных атак;
- Отсутствие сильного влияния на работу игры;
- Простота установки защиты;
- Защита MMOG не требует привязки к какому-либо объекту (оптический диск или компьютер);
- Защита MMOG не требует активации.

Решение FL MMOG позволяет осуществить:

1. Проверку целостности приложения на этапе его запуска.

Такая проверка предназначена для дополнительной защиты защищенных модулей игры от модификации. Для этой цели в модуль добавляется цифровая подпись защиты, которая проверяется в своем загрузчике при загрузке модуля (см. [Функции API защиты](#)). Производится автоматически при запуске игры. Кроме того, защита будет срабатывать, если исполняемые модули игры заражены вирусами.

2. Проверку целостности кода и данных в памяти.

Контроль целостности неизменяемого кода и данных в памяти. Этот метод предназначен для проверки защищенного исполняемого файла в процессе работы игры. Проверке целостности подлежат те секции защищенного файла, которые предназначены только для чтения, то есть переменные игры не проверяются. Для этого защищенная игра вызывает соответствующую функцию SF API (см. [Функции API защиты](#)). Данное действие значительно усложняет модификацию защищенного файла в процессе работы игры

3. Защиту отдельных переменных игры от несанкционированного доступа/изменения.

Для защиты переменных вместо встроенных типов (Uint) используются «защищенные типы» из состава SF API (SF Uint) (см. [Применение секретных классов](#)). Подобная защита значительно усложняет возможность модификации переменных игры в процессе ее работы.

4. Обеспечение целостности файлов данных.

Контроль целостности неизменяемых файлов (защищенная игра проверяет цифровые подписи файлов данных). Проверка производится через StarForce API (см. [Функции API защиты](#)). Таким образом, усложняется анализ и модификация файлов данных защищенной игры.

5. Контроль родительского процесса.

Данный метод (включен по умолчанию) разрешает запуск защищенной игры только из-под санкционированных процессов. Если процесс (файл вредоносной программы) находится в черном списке – выдается ошибка. Проверка таких файлов выполняется по сигнатурам. Сам черный список можно со временем пополнять.

6. Защита трафика между клиентом и сервером.

Данный метод заключается в шифровании трафика между клиентской и серверной частями игры и значительно осложняет его модификацию или подмену. Суть защиты заключается в формировании криптографически защищённого канала связи между клиентом и сервером. Для реализации требуется установка дополнительного модуля StarForce на сервере игры (написан на Java) (см. [Функции API защиты](#)).

Для осуществления защиты приложения служит программный комплекс Protection Studio, являющийся одним из компонентов системы защиты StarForce (SF).

Настоящее руководство содержит описание действий по установке защиты и по тестированию и сопровождению защищенного приложения.

Условные обозначения

В данном документе используются следующие условные обозначение:

Обозначение	Описание
Parameter	Код
Сервис	Команда, кнопка, меню, закладка
Базовые параметры	Раздел Protection Studio, название продукта
Привязка	Поле, параметр в Protection Studio; значение параметра
File	Имя файла, папки
Основные параметры	Группа полей, параметров
Информация	Определение; важное примечание
Внимание!	Важная информация

Условные обозначения

2 Схема установки защиты FL MMOG при разработке и выпуске продукта

№ п/п	Этапы	Разделы руководства
1	Создание файла проекта защиты*	Раздел "Проект"
Этап разработки игры		
2	Генерация SDK	StarForce SDK

№ п/п	Этапы	Разделы руководства
3	Внедрение в код защиты необходимых функций	Функции API защиты
4	Тестирование незащищённого клиентского приложения	
Этап защиты		
5	Выбор и добавление в проект защищаемых файлов	Защищаемые файлы; Раздел "Файлы"
6	Выбор защищаемых функций и задание методов их защиты	Раздел "Функции"
7	Установка параметров	Раздел "Базовые параметры"; Раздел "Дополнительные параметры"
8	Настройка графической оболочки интерфейса защиты	Раздел "Параметры пользовательского интерфейса"
9	Задание пути к папке для загрузки файлов с сервера защиты	Папка для выходных данных
10	Защита файлов	Защита файлов
11	Подготовка дистрибутива	Формирование дистрибутива
12	Тестирование работы защищенного программного продукта*	Тестирование защищенного приложения

Порядок действий при установке защиты

* Полу жирным шрифтом выделены обязательные действия.

*При получении неудовлетворительных результатов тестирования повторите этапы 2, 3, 4, и далее по порядку.

3 Защита продукта и тестирование

3.1 Начало работ по защите

3.1.1 Установка Protection Studio

Инсталляционный файл Protection Studio можно загрузить с сайта StarForce: <http://www.star-force.ru/>. Для этого в разделе сайта **Защита ПО** выберите **Вход для клиентов** и войдите в систему с использованием логина и пароля полученной учетной записи.

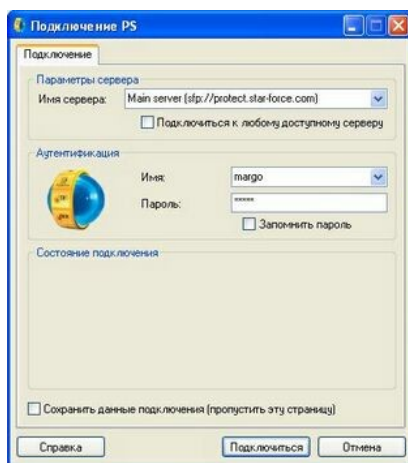
Учетная запись создается в информационном пространстве Protection Studio для каждого пользователя, начинающего работать с системой защиты StarForce. Заданные в ней имя (логин) и пароль передаются пользователю службой технической поддержки.

Учетная запись используется:

- для входа в область веб-сайта StarForce, предназначенную для скачивания клиентами всех необходимых программ и документов для защиты.
- для запуска Protection Studio с подключением рабочих пространств, в которые назначена данная учетная запись.

Установка производится стандартным образом.

3.1.2 Запуск Protection Studio



Запуск Protection Studio

После установки Protection Studio запускается из меню **Start/Пуск** командой **Start/Пуск|Programs/Все программы|Protection Studio 2|Protection Studio**.

В результате на экран выводится окно "Подключение PS" (см. [выше](#)).

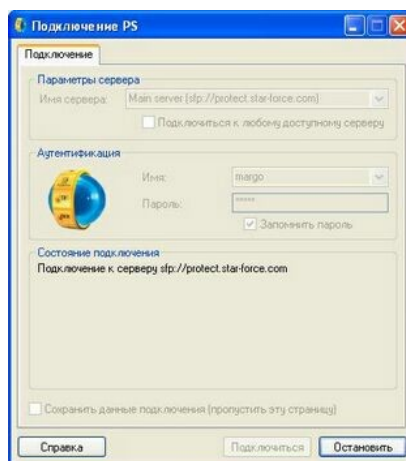
На вкладке **Подключение** необходимо выполнить следующее:

1. Задайте или выберите в поле со списком **Имя сервера** требуемый сервер либо поставьте флажок **Подключиться к любому доступному серверу**, если подходит вариант подсоединения к любому свободному серверу.
2. Введите логин и пароль своей учетной записи. Если выставить флажок в поле **Запомнить пароль**, то в дальнейшем при наборе этого логина пароль будет подставляться автоматически.
3. Нажмите на кнопку **Подключиться**.

Сообщение о процессе подключения выводится в лог **Состояние подключения**.

В том случае, если выставлен флажок **Подключиться к любому доступному серверу**, производится однократная попытка поочередного подключения ко всем имеющимся в списке серверам защиты. Если все серверы заняты, Вы можете сразу возобновить попытку подсоединения, нажав на кнопку **Подключиться**, или закрыть Protection Studio, нажав кнопку **Отмена**, и возобновить попытки позднее.

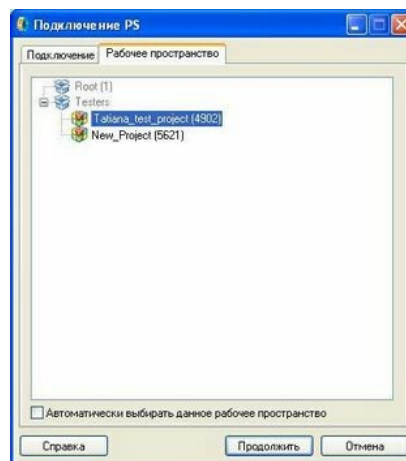
Во время подключения можно остановить процесс, нажав кнопку **Остановить** (см. [ниже](#)).



Остановка подключения

Если установленные в окне подключения Protection Studio данные не будут меняться в течение какого-то временного периода, можно выставить флажок **Сохранить данные подключения (пропустить эту страницу)**. Тогда при дальнейших запусках Protection Studio будет сразу происходить попытка подключения к серверу и перехода к следующему окну – как если бы уже была нажата кнопка **Подключиться**.

После удачного подключения к серверу появляется окно выбора проекта (см. [ниже](#)).



Выбор рабочего пространства

На вкладке **Рабочее пространство** выберите рабочее пространство, в котором требуется производить операции по защите.

Рабочее пространство – это совокупность информационных объектов, доступных пользователю системы защиты StarForce. Проект создается службой технической поддержки StarForce после предоставления менеджеру StarForce данных о параметрах защиты, выбранных клиентом. Проект защиты содержит настройки защиты по умолчанию, а также параметры, относящиеся к административному управлению рабочими пространствами.

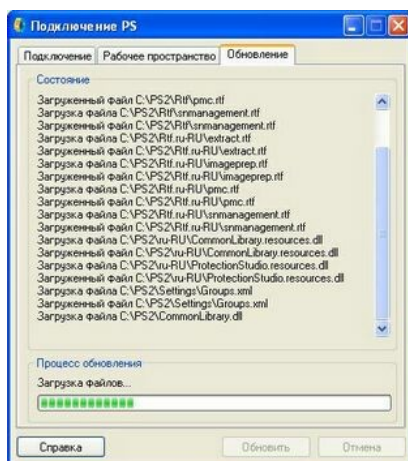
После выбора рабочего пространства нажмите на кнопку **Продолжить**.

Если выбранное рабочее пространство не будет меняться в течение длительного времени, можно выставить флажок **Автоматически выбирать данное рабочее пространство**. Тогда при дальнейших запусках Protection Studio будет сразу

происходить переход к следующему окну – как если бы уже была нажата кнопка **Продолжить**.

Если Вам доступно только одно рабочее пространство, оно будет выбрано автоматически, то есть окно выбора проекта не появится вне зависимости от значения флага «**Автоматически выбирать данное рабочее пространство**».

В результате появится окно автоматического обновления (см. [ниже](#)).

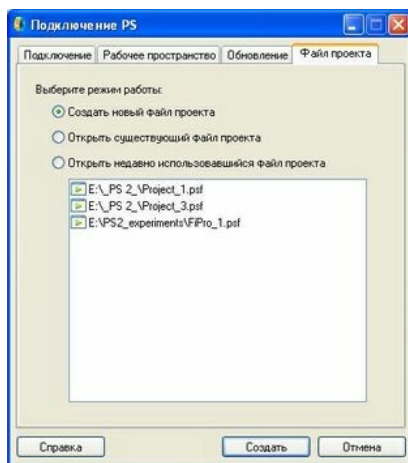


Обновление PS

При каждом обращении к Protection Studio происходит проверка на наличие обновлений (если не была отключена соответствующая опция (см. [рисунок](#))). В том случае, если они были, в Protection Studio автоматически производятся соответствующие изменения. Таким образом, Вы всегда работаете с последней версией программы. После обновления автоматически производится перезапуск Protection Studio. Так будет до тех пор, пока не будет отключен этот сервис (см. [Интерфейс Protection Studio](#)).

Замечание. Если сервис автоматического обновления отключен, вкладка **Обновление** в окне подключения не отображается.

Если обновление не требуется, то происходит автоматический переход на вкладку выбора файла проекта (см. [рисунок](#)).



Выбор файла проекта

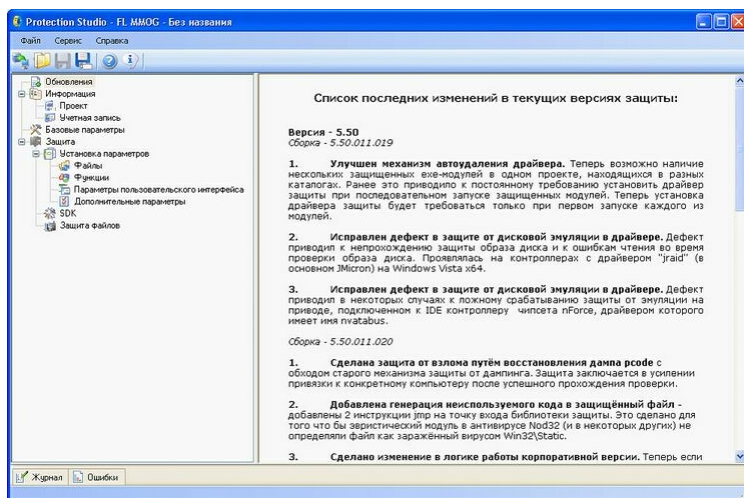
На вкладке **Файл проекта** имеется возможность определить файл проекта, с которым необходимо работать.

Файл проекта создается в Protection Studio в процессе работы над защитой продуктов и хранится на компьютере в виде файла с расширением . PSF. Для сложных проектов может создаваться несколько PSF-файлов.

В окне представлены три способа открытия файла:

- *Создать новый файл проекта.* После нажатия на кнопку выбора **Создать новый файл проекта** становится активной кнопка **Создать**, при нажатии на которую происходит открытие основного окна программы (см. [рисунок](#)) с параметрами по умолчанию (пустым проектом).
- *Открыть уже существующий файл проекта.* После нажатия на кнопку выбора **Открыть существующий файл проекта** становится активной кнопка **Открыть**, при нажатии на которую появляется стандартное диалоговое окно открытия файла. После того как нужный файл открыт, происходит открытие основного окна программы (см. [рисунок](#)) с параметрами открытого файла проекта.
- *Выбрать файл проекта из тех, с которыми уже осуществлялась работа.* Последние пять таких файлов показаны в списке в центре окна. После нажатия на кнопку выбора **Открыть недавно использовавшийся файл проекта** и нажатия в списке на нужный файл становится активной кнопка **Открыть**, при нажатии на которую происходит открытие основного окна программы (см. [рисунок](#)) с параметрами выбранного файла проекта. Также вместо трех вышеописанных действий достаточно двойного нажатия на имя нужного файла в списке.

Основное окно Protection Studio (см. [рисунок](#)) представляет собой интерфейс для задания параметров защиты продукта, доступных в выбранном рабочем пространстве, а также предоставляет возможность запуска сервисов защиты с заданными параметрами.



Интерфейс Protection Studio

3.2 Установка защиты

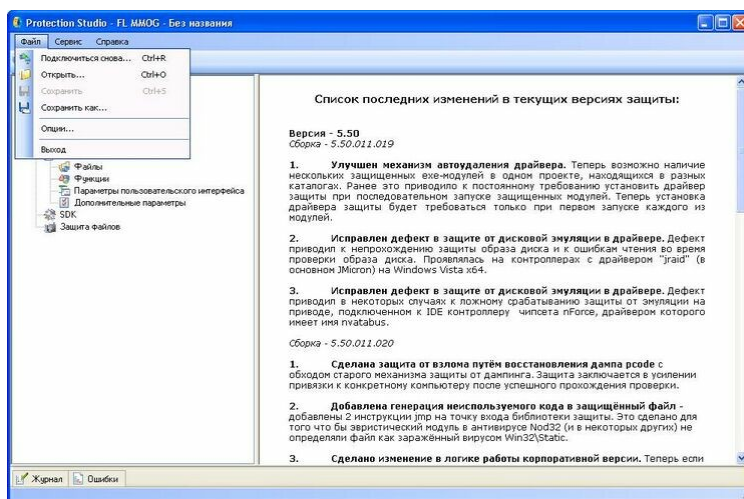
3.2.1 Интерфейс Protection Studio

В заголовке окна Protection Studio указываются название продукта защиты **FL MMOG** и имя открытого файла проекта. Если был создан новый проект и еще не был сохранен, в заголовке он будет иметь имя **Без названия**. Пока не были сохранены результаты последнего редактирования проекта, рядом с его именем будет стоять символ "*", который исчезнет после выполнения команд **Сохранить** или **Сохранить как**.

3.2.1.1 Команды меню

Меню Файл

Внешний вид меню **Файл** показан на [рисунке ниже](#).



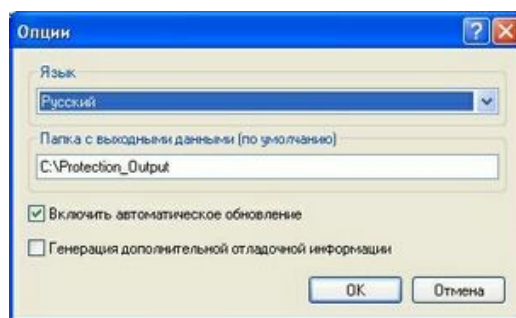
Интерфейс Protection Studio. Меню "Файл"

Команда	Назначение команды
Подключиться снова...	Возврат к процессу подключения для выбора другого сервера защиты, другого рабочего пространства или файла проекта. Окно Protection Studio закрывается и вновь выводится окно подключения.
Открыть	Открытие сохраненного на компьютере пользователя файла проекта (.PSF).
Сохранить	Сохранение текущих настроек в файле проекта (.PSF).
Сохранить как...	Сохранение текущих настроек в другом файле проекта (.PSF).
Опции	Вызов окна "Опции".
Выход	Закрытие Protection Studio.

Команды меню "Файл"

Для надежности рекомендуется сохранять изменения параметров защиты в файле проекта, хотя защиту можно провести и без сохранения изменений.

Опции представлены в дополнительном окне (см. [ниже](#)).



Опции

В нем:

- Предоставляется возможность выбора языка интерфейса Protection Studio. Поддерживается два языка интерфейса Protection Studio: русский или английский. Чтобы изменить язык, в меню **Файл** выберите команду **Опции**, и в появившемся окне (см. [рисунок](#)) укажите желаемый язык и нажмите на кнопку **ОК**.

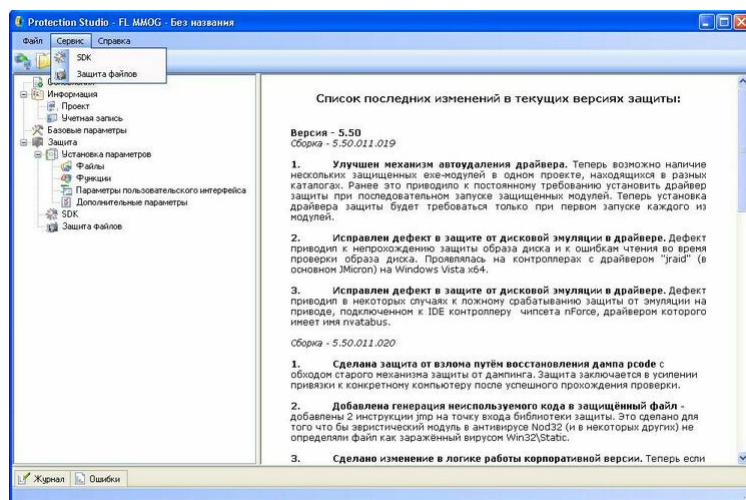
Чтобы изменение языка вступило в силу, необходимо перезапустить приложение, о чем предупреждается в соответствующем сообщении, которое выдается при нажатии на кнопку **ОК** после смены языка.

- Задаётся значение по умолчанию для папки, в которой будут сохраняться результаты работы всех сервисов. Данное значение может быть переопределено для каждого сервиса в соответствующем ему разделе. В случае такого переопределения для конкретного сервиса (см. [Защита файлов](#), [SDK](#)) последующие изменения поля **Папка с выходными данными** в диалоговом окне "Опции" никак на нём не отразятся.
- Предоставляется возможность отключения автоматического обновления Protection Studio (рекомендуется делать только при согласовании со службой технической поддержки), которое по умолчанию включено.

- Предоставляется возможность включения генерации дополнительной отладочной информации, которая будет помещена в файл debug.log в корневой папке Protection Studio. Эти сведения могут быть полезны при возникновении проблем с установкой защиты.

Меню Сервис

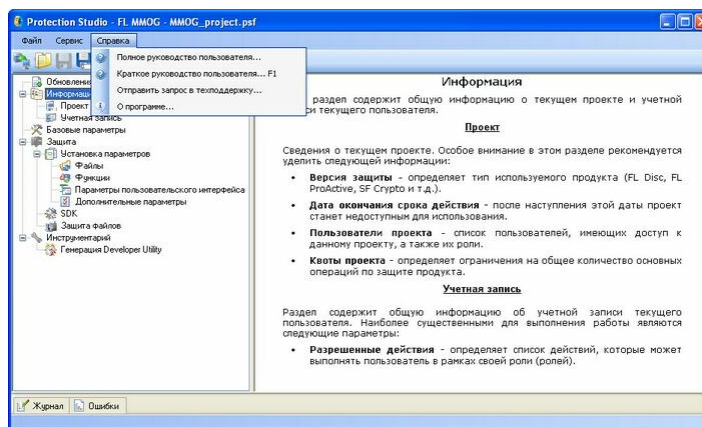
Меню **Сервис** содержит ссылки для перехода к разделам с сервисами. Ссылки совпадают по названию с сервисами, показанными в окне Protection Studio (см. [ниже](#)). Таким образом, соответствующий сервису раздел можно открыть одним из двух способов: из этого меню или путем выбора одноименного элемента дерева непосредственно в окне Protection Studio.



Интерфейс Protection Studio. Сервисы

Меню Справка

Внешний вид меню **Справка** показан на [рисунке ниже](#).









Интерфейс Protection Studio. Меню "Справка"

Команда	Назначение команды
Полное руководство пользователя...	Переход к руководству пользователя по данному продукту в клиентской части сайта StarForce.

Команда	Назначение команды
Краткое руководство пользователя...	Вывод на экран краткой справки по Protection Studio (выберите раздел для просмотра справки по нему).
Отправить запрос в техподдержку...	Возможность написать письмо в техническую поддержку продукта: при выборе этой команды запускается программа для работы с почтой на компьютере пользователя (выбранная как почтовый клиент по умолчанию) и создается шаблон письма.
О программе...	Вывод на экран сведений о версии Protection Studio с кнопкой Обратная связь (описана выше).

Команды меню "Справка"

3.2.1.2 Кнопки панели инструментов

Кнопка	Название	Назначение
	Подключиться снова...	Возврат к процессу подключения для выбора другого сервера защиты, другого рабочего пространства или файла проекта. Окно Protection Studio закрывается, и вновь выводится окно подключения.
	Открыть	Открытие сохраненного на компьютере пользователя файла проекта (.PSF).
	Сохранить	Сохранение текущих настроек проекта в файле проекта (.PSF).
	Сохранить как...	Сохранение текущих настроек в другом файле проекта (.PSF).
	Справка	Вывод на экран Справки по Protection Studio.
	О программе	Вывод на экран сведений о версии Protection Studio.

Кнопки панели инструментов

3.2.1.3 Использование быстрых клавиш

Комбинация клавиш	Команда
Ctrl+R	Подключиться снова
Ctrl+O	Открыть
Ctrl+S	Сохранить

Интерфейс Protection Studio. Быстрые клавиши

3.2.1.4 Рабочая область

Основная часть окна Protection Studio (см. [рисунок](#)) поделена на две части: левая панель в виде дерева содержит список доступных рабочих разделов, а правая отображает содержимое выбранного раздела.

Разделы объединяются в две основные группы: *информационную* и *функциональную*.

Информационная группа включает в себя:

- разделы: **Проект** и **Учетная запись**. В них отображаются основные сведения о

проекте (рабочем пространстве) и учетной записи. Эти данные определяются при создании проекта сотрудниками службы технической поддержки StarForce. В разделах этой группы доступны для редактирования только параметры учетной записи;

Функциональная группа включает в себе:

- раздел **Базовые параметры** - обязательный и общий раздел параметров для всех сервисов, в нем указывается тип платформы, вид привязки, информация о драйвере и путь ключа в реестре, а также представлена информация о неизменяемых, но важных параметрах проекта защиты.
- разделы: **Файлы, Функции, Параметры пользовательского интерфейса и Дополнительные параметры**, составляющие раздел **Установка параметров**. В этих разделах предоставляется возможность задать большинство из возможных для данного проекта настроек защиты;
- разделы: **Защита файлов, SDK**, представляющие соответствующие сервисы Protection Studio. В каждом разделе есть его описание, поля для ввода данных и кнопки, управляющие работой сервиса. Для запуска сервиса нужно определить опции только соответствующего раздела и раздела **Базовые параметры**. Для защиты файлов параметры задаются в разделе **Установка параметров**.

3.2.1.5 Отображение процесса защиты

В нижней части окна Protection Studio находятся две панели: **Журнал** и **Ошибки** (см. [рисунок](#)).

Панель **Журнал** содержит информацию о работе всех сервисов, а также информацию об изменении параметров учетной записи и пользовательских прав при использовании SDK.

Панель **Ошибки** содержит список предупреждений и ошибок, выявленных при запуске последнего сервиса, если они были.

Эти панели закрываются после одного из следующих действий:

1. Нажатие на кнопку мыши после помещения курсора в свободное поле в левой части окна или в какое-либо поле ввода в правой;
2. Двойное нажатие на кнопку мыши на заголовке соответствующей панели.

При запуске сервисов появляются дополнительные панели, содержащие информацию о ходе их выполнения.

3.2.2 Задание параметров защиты

Среди всей информации о проекте выделяются обязательные параметры, без которых невозможно провести процесс защиты. Этим данным посвящен текущий раздел. Остальные сведения приведены в главе **Полное описание настроек проекта защиты**. При создании проекта защиты обязательными параметрами являются: пути к защищаемым файлам и путь к папке для записи защищенных файлов (см. таблицу в разделе [«Схема установки защиты...»](#)).

После ввода в проект необходимых данных можно переходить к самому процессу защиты, для чего следует обратиться к сервису **Защита файлов** (см. [Защита файлов](#)).

3.2.2.1 Защищаемые файлы

Основным назначением защиты файлов является противодействие взлому программного продукта, т.е. раскрытию его алгоритмов и программного кода.

В список файлов, подлежащих защите, могут включаться следующие виды файлов:

- исполняемые файлы (.exe) и файлы динамических библиотек (.dll), иногда объединяемых под общим названием программных файлов;
- файлы данных.

Защита программных файлов

При использовании данного способа защиты программные файлы хранятся в зашифрованном виде. Если проверка запускаемого приложения на легальность прошла успешно, файлы загружаются в оперативную память в расшифрованном виде. Если проверка не прошла, защищенное приложение не запускается.

При установке защиты можно выбрать ее уровень (см. опцию **Уровень защиты** в разделе **Дополнительные параметры** на [рисунке](#)). Повышение уровня защиты повышает ее качество, но увеличивает размер библиотеки защиты. Следует помнить, что в некоторых случаях повышение уровня защиты может привести к замедлению работы приложения.

Внимание! Защищать можно только незашифрованные и несжатые исполняемые файлы. Здесь имеется в виду паковка кода программами типа UPX и шифрование другими системами защиты.

Защита файлов данных

При защите файлов данных производится шифрование их содержимого, а сами файлы скрываются в файле контейнера. Содержимое файла автоматически расшифровывается при доступе к нему из защищенного приложения.

Главным принципом здесь является не шифрование содержимого файлов, а скрытие имен этих файлов путем помещения файлов в контейнер. Если имя файла неизвестно, его нельзя никакими способами достать из контейнера. Поэтому файлы, лежащие в контейнере, не находятся при помощи функций FindFirst/FindNext и им подобных. Если же имя файла известно, его достаточно легко извлечь из контейнера.

В тех случаях, когда имена файлов, хранящихся в контейнере, нельзя скрыть, можно рекомендовать помещать пустые или заполненные несущественным содержимым файлы с такими же именами вне контейнера. Защищенная программа будет обращаться только к файлам в контейнере, а открыто хранящиеся файлы отвлекут внимание взломщика программы. Следует, однако, отметить, что это достаточно слабый способ защиты и использовать его следует весьма ограниченно.

Основным ограничением, накладываемым на защищаемые файлы данных, является то, что они могут быть доступны только для чтения.

В связи с тем, что затраты времени на расшифровку файла данных перед обращением к нему невелики, данный способ защиты можно применять практически к любым

файлам, используемым приложением. Исключением здесь являются лишь файлы, время доступа к которым критично.

Защищаемыми файлами также могут являться и файлы, доступ к которым ведется через отображение в память (File Mapping).

Защита файлов данных эффективна только в том случае, если имена этих файлов скрыты, а значит:

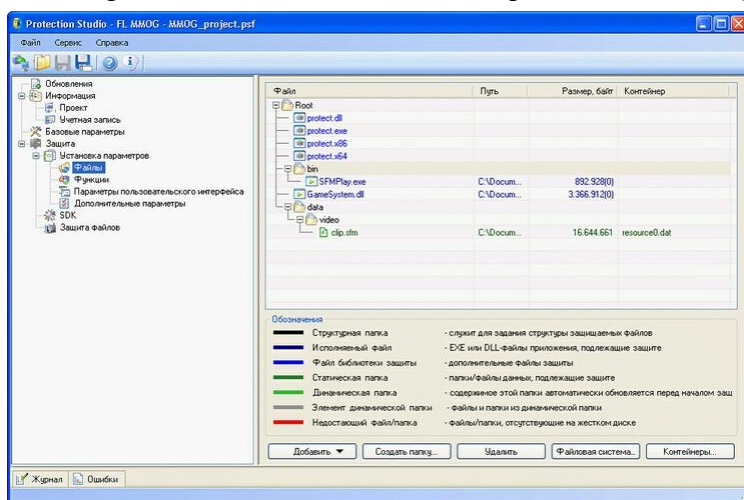
- Имена защищенных файлов не должны лежать в защищенном приложении в открытом виде.
- Доступ к файлам не должен осуществляться каждый раз при запуске программы.

Имена защищаемых файлов данных не должны быть доступны в других версиях защищаемого приложения.

Файлы данных помещаются в контейнеры, создаваемые пользователем, причем после проведения операции защиты все файлы, содержащиеся в отдельном контейнере, преобразуются в один файл с именем контейнера.

Внимание! Создание контейнеров, как и защита файлов данных, возможно только при одновременной защите исполняемых файлов и использовании драйвера защиты (см. [Драйвер защиты](#)). Установка этой опции производится службой технической поддержки при создании проекта защиты.

Добавление файлов в проект защиты выполняется в разделе **Файлы** (см. [ниже](#)).



Раздел "Файлы"

При включении файлов в список задаются их пути относительно корневого каталога защищаемого программного продукта, или относительно каталога, в котором будет находиться библиотека защиты.

Более подробные сведения и рекомендации по защите файлов см. **Раздел "Файлы"**.

Добавление исполняемых файлов в проект защиты

Внимание! Имена исполняемых файлов, включаемых в проект защиты, контейнеров, библиотеки защиты и структурных папок должны содержать только буквы латинского алфавита, цифры и допустимые символы, за исключением

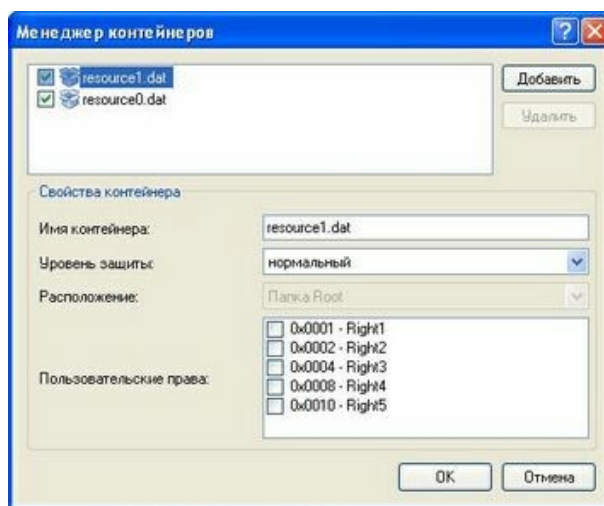
следующих: * ? / \ | : < > " № Также имена структурных папок, контейнеров и библиотеки защиты не могут быть следующими: CON, PRN, AUX, CLOCK\$, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, в т.ч. любыми комбинациями типа CON.* (для всех вышеперечисленных имен; без учета регистра).

1. Выделите папку, в которую необходимо добавить исполняемый файл(ы).
2. Нажмите на кнопку **Добавить** (см. [рисунок выше](#)).
3. В контекстном меню выделите команду **Исполняемый файл(ы)....**
4. В появившемся стандартном диалоговом окне выберите файл (или, удерживая клавишу **Ctrl**, несколько файлов) и нажмите кнопку **ОК**.
5. Выбранные файлы будут добавлены в список защищаемых исполняемых файлов.

Добавление файлов данных в проект защиты

Создание контейнера

1. В окне раздела **Файлы** нажмите на кнопку **Контейнеры...** (см. [рисунок выше](#)).



Раздел "Файлы". Окно "Менеджер контейнеров"

2. В открывшемся окне "Менеджер контейнеров" (см. [рисунок](#)) нажмите на кнопку **Добавить** и задайте имя для создаваемого контейнера.
3. Определите уровень защиты контейнера. "**Нормальный**" – оптимальный уровень. При повышении уровня защиты увеличивается размер контейнера.
4. При ограничении прав на контейнеры отметьте нужные права.
5. Нажмите на кнопку **ОК** для завершения операции.

Добавление файлов

1. Нажмите на кнопку **Добавить** (см. [рисунок выше](#)).
2. В контекстном меню выделите команду **Файл данных в контейнер....**
3. В появившемся стандартном диалоговом окне выберите файл (или, удерживая клавишу **Ctrl**, несколько файлов) и нажмите кнопку **ОК**.

4. Определите контейнер, в который надо добавить файл. При этом следует иметь в виду, что контейнер однозначно связан с той папкой, в которой расположен объект (защищаемый файл или папка), первым добавленный в данный контейнер. В дальнейшем в этот контейнер можно будет поместить только защищаемые объекты, расположенные в этой или во вложенных в нее папках. Для таких объектов этот контейнер считается *подходящим*. Объекты, расположенные в папках одного или более высоких уровней относительно данной, могут быть размещены только в других контейнерах.

Принцип добавления:

- 1) Если до сих пор не было создано ни одного подходящего для данного файла контейнера, определите параметры нового контейнера, используя появившееся окно "Создание контейнера".
 - 2) Если несколько созданных контейнеров являются подходящими для данного файла, то в появившемся окне "Выбор контейнера" надо будет выбрать один из них.
 - 3) Если существует только один подходящий контейнер, то он будет использоваться автоматически.
5. Выбранные файлы будут добавлены в контейнер.

Замечание. Как правило, для защиты используется один контейнер. Более одного контейнера создается в том случае, если размер первого контейнера превышает 2 Гб (например, при обновлении защищенного продукта, см. раздел [Обновление...](#)).

Удаление файла

1. Выделите файл, предназначенный для удаления.
2. Выполните любое из трех действий:
 - 1) нажмите на кнопку **Удалить** в рабочем поле раздела;
 - 2) нажатием на правую кнопку мыши вызовите контекстное меню, в котором выберите команду **Удалить**;
 - 3) нажмите на клавишу **Delete** на клавиатуре.

Примечания

1. Если в результате удаления защищаемых объектов из проекта один или несколько контейнеров станут пустыми, то в появившемся окне "Удаление контейнера" будет предложено их удалить, что будет сделано в случае нажатия на кнопку **Удалить**. Если какие-либо контейнеры необходимо оставить, снимите флажки около их названий.
В случае выбора кнопки **Отмена** контейнеры не удалятся и могут быть снова использованы для размещения защищаемых объектов в папках рабочего поля любого уровня.

Внимание! На момент запуска сервиса Защита файлов пустых контейнеров быть не должно: с ними защита невозможна.
--

2. Можно изменить путь, по которому расположен файл, записанный в проект защиты. Достаточно отметить строку с именем файла, чтобы на пересечении ее со

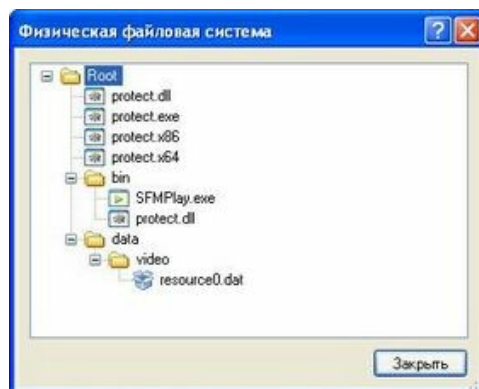
столбцом **Путь** появился переключатель, открывающий стандартное окно поиска папки.

Если будет выбрана папка, в которой не окажется файла с указанным именем, то все символы этой строки будут выделены красным цветом.

3. Можно изменить местоположение файла данных. Для этого надо отметить строку с именем файла; тогда на пересечении ее со столбцом **Контейнер** появится переключатель, открывающий список контейнеров. В нем можно выбрать для файла контейнер из уже созданных или определить новый, выбрав запись: **Контейнеры..** и вызвав тем самым появление окна "Менеджер контейнеров" (см. [рисунок](#)).

4. Если структура защищаемого приложения такова, что исполняемый файл и файлы данных, подлежащие защите, располагаются в разных папках относительно корневого каталога, то сначала необходимо выстроить цепочку вложенных папок при помощи кнопки **Создать папку...**, а затем, выделив папку нужного уровня, приступить к добавлению файлов в проект.

Так, на [рисунке выше](#) видно, что относительно корневого каталога исполняемый файл имеет путь: \bin, а файл данных находится в контейнере: \data\video\resource0.dat. Эта структура хорошо видна в окне "Физическая файловая система", которое появляется при нажатии на кнопку **Файловая система...** (см. [ниже](#)).



Физическая файловая система

3.2.2.2 Папка для выходных данных

В заданную папку будут помещаться файлы, полученные в результате установки защиты (зашифрованные файлы, библиотека защиты, контейнеры данных, и т.д.). По умолчанию значение данного поля соответствует значению в диалоге «Опции» (см. [рисунок](#)).

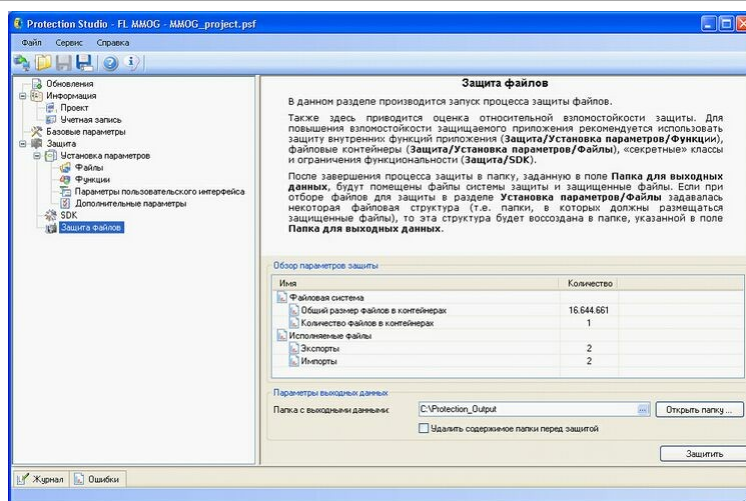
Папка на компьютере для записи защищенных файлов может быть переопределена в разделе **Защита файлов** в поле **Папка с выходными данными** (см. [рисунок](#)). Значение поля сохраняется в файле проекта.

Папка может быть определена одним из двух способов:

1. явное задание полного пути и имени папки;
2. задание пути относительно места размещения файла проекта:

- 1) <имя папки> или .\<имя папки> – папка будет помещена туда же, где находится файл проекта;
- 2) ..\<имя папки> – папка будет помещена на один уровень выше файла проекта;
- 3) ...\\.\<имя папки> – папка будет помещена на два уровня выше файла проекта.

Внимание! Использование относительного пути возможно только в том случае, если файл проекта был хотя бы один раз сохранен. Иначе при запуске сервиса Protection Studio сразу выдаст сообщение: "Папка с указанным путем не может быть создана!"



Раздел "Защита файлов"

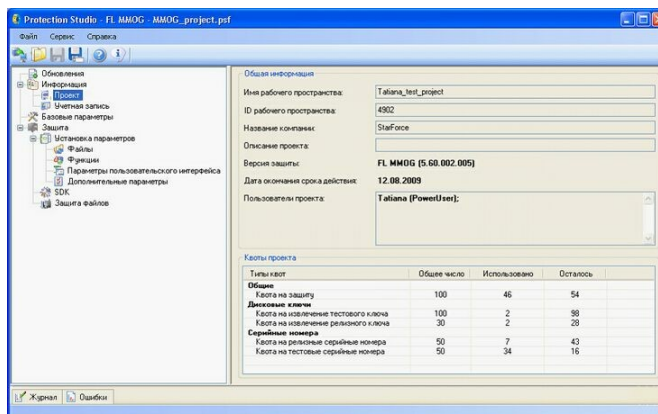
Если нужно полностью обновить содержимое папки, в поле **Удалить содержимое папки перед защитой** выставляется флажок.

3.2.3 Полное описание настроек проекта защиты

В этом разделе более подробно рассказывается о проекте защиты и его настройках. В нем даются сведения, не вошедшие в главу [Задание параметров защиты](#). Таким образом, в этих двух главах представлена исчерпывающая информация о возможных установках в проекте защиты.

3.2.3.1 Раздел "Проект"

Информация о проекте защиты (рабочем пространстве) выводится в разделе **Проект** (см. [ниже](#)).



Раздел "Проект"

Эта информация задается при создании проекта сотрудниками Службы технической поддержки StarForce. Изменение также может производиться только Службой технической поддержки StarForce.

В раздел включены следующие поля:

Название поля	Содержимое
Имя рабочего пространства	Имя выбранного рабочего пространства, в котором в данный момент возможно производить действия по защите с помощью Protection Studio. Обычно совпадает с названием защищаемого продукта.
ID рабочего пространства	ID данного рабочего пространства в системе StarForce.
Название компании	Название компании-издателя/разработчика защищаемого программного продукта, либо компании-заказчика проекта защиты.
Описание проекта	Обычно в данном поле содержится краткое описание назначения данного проекта, но также может быть указана любая другая информация для удобства пользователя при работе с разными проектами.
Версия защиты	Тип используемого продукта (в данном случае FL MMOG) и его версия.
Дата окончания срока действия	Дата, после наступления которой проект станет недоступен для использования.
Пользователи проекта	Список пользователей (и их роли), имеющих доступ к данному проекту.
Квоты проекта	Ограничения на общее количество основных операций по защите продукта, ограничения на извлечение ключей и генерацию серийных номеров.

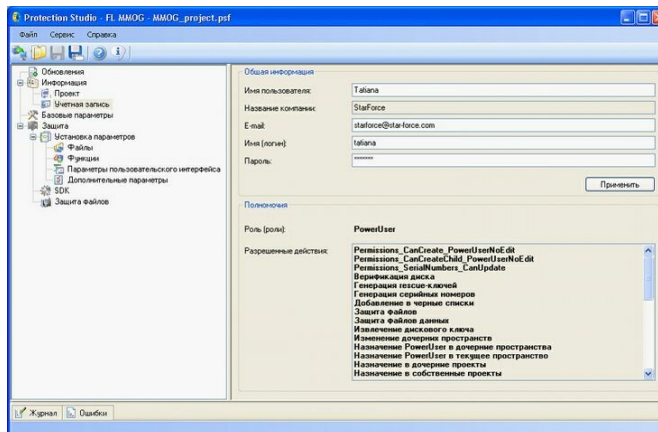
Поля раздела "Проект"

Квоты – это ограничения на количество выполнений определенных действий по защите продукта (защита файлов и т.д.).

Квоты назначаются рабочим пространствам.

3.2.3.2 Раздел "Учетная запись"

В разделе **Учетная запись** выводится информация о пользователе и его полномочиях (см. [ниже](#)).



Раздел "Учетная запись"

В раздел включены следующие поля:

Название поля	Содержимое
Имя пользователя	Обычно имя и фамилия пользователя.
Название компании	Название компании, которую представляет пользователь.
E-mail	Электронный адрес пользователя – для получения информации об истечении проектов и сообщений по подписке.
Имя (логин)	Логин, используемый для доступа в систему.
Пароль	Пароль, который используется для доступа в систему.
Роль (роли)	Название ролей, предоставленных пользователю.
Разрешенные действия	Список действий, которые может выполнять пользователь в данном рабочем пространстве в рамках своей роли (ролей).

Поля раздела "Учетная запись"

Назначение учетной записи в рабочее пространство означает, что соответствующее лицо получает доступ к рабочему пространству и его квотам. Важным атрибутом назначения учетной записи является **роль**.

Роль – это набор статических прав, определяющий спектр возможных действий пользователя в системе относительно конкретного рабочего пространства. Одному пользователю может быть назначено любое количество уникальных ролей в одном рабочем пространстве, в этом случае права, разрешённые каждой из ролей, суммируются.

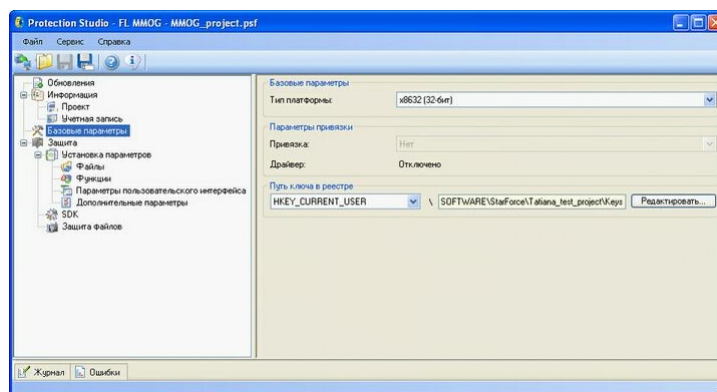
Пользователю недоступны механизмы назначения ролей для конкретных учетных записей, однако он может обратиться в Службу технической поддержки StarForce, если у него возникнет потребность в дополнительном ограничении или расширении прав доступа лиц, назначенных в рабочее пространство.

Пользователь может задавать по своему усмотрению имя пользователя, электронный адрес, логин и пароль. Чтобы изменения вступили в силу, надо нажать ставшую

активной кнопку **Применить**. Следует иметь в виду, что при подключении к Protection Studio надо будет указывать актуальные логин и пароль.

3.2.3.3 Раздел "Базовые параметры"

В разделе **Базовые параметры** также помимо данных, установленных службой технической поддержки, представлены параметры, доступные для изменения (см. [ниже](#)).



Раздел "Базовые параметры"

Поле	Возможные значения	Назначение
Тип платформы	<ul style="list-style-type: none"> x8632 x8664 	Задание аппаратной платформы продукта: 32-разрядной или 64-разрядной.
Привязка	<ul style="list-style-type: none"> Нет 	Выставляется службой поддержки и не подлежит изменению.
Драйвер	<ul style="list-style-type: none"> Отключено 	Драйвер обеспечивает более высокую надежность защиты, а также позволяет защищать файлы данных. Данный параметр по умолчанию имеет значение Отключено и включается по запросу в службу поддержки.
Путь ключа в реестре (поле со списком)	<ul style="list-style-type: none"> HKEY_LOCAL_MACHINE HKEY_CURRENT_USER 	Выбор раздела системного реестра, в котором будут храниться ключи и данные защиты продукта. Не рекомендуется использовать HKEY_LOCAL_MACHINE.
Путь ключа в реестре (текстовое поле)	Разрешенные в именах ключей системного реестра символы (латинский алфавит (прописные и строчные буквы), цифры, \ () { } [] ' , ' , , подчёркивание, пробел, дефис)	Задание точного адреса местоположения ключей в выбранном разделе реестра.

Поля раздела "Базовые параметры"

Данные защиты в системном реестре

Ключи, параметры лицензии и настройки защиты, необходимые для запуска защищенного программного продукта, сохраняются в файле реестра Windows при первом запуске продукта (реже – при последующих запусках).

Выбор раздела реестра для хранения параметров лицензии определяется видом защищенного приложения. *Рекомендуется для всех видов приложений, за исключением сервисов, использовать раздел* HKEY_CURRENT_USER.

Для защищаемых сервисов рекомендуется использовать раздел HKEY_LOCAL_MACHINE.

Такое разграничение обусловлено тем, что, как правило, для записи в раздел HKEY_LOCAL_MACHINE приложение должно обладать правами администратора, не всегда имеющимися у конечного пользователя защищенного продукта. В таком случае использование указанного раздела приведет, например, к невозможности активации приложения этим пользователем. Также существует ряд ограничений при работе с этим разделом реестра в операционных системах Windows Vista.

Если защищенное приложение будет запускаться, например, в целях тестирования, на той же машине, на которой установлен Protection Studio, то для последующей очистки реестра от настроек защиты надо воспользоваться кнопкой **Редактировать** (см. также [Изменение настроек системы защиты](#)).

3.2.3.4 Раздел "Файлы"

Этот раздел относится к разделу **Установка параметров**.

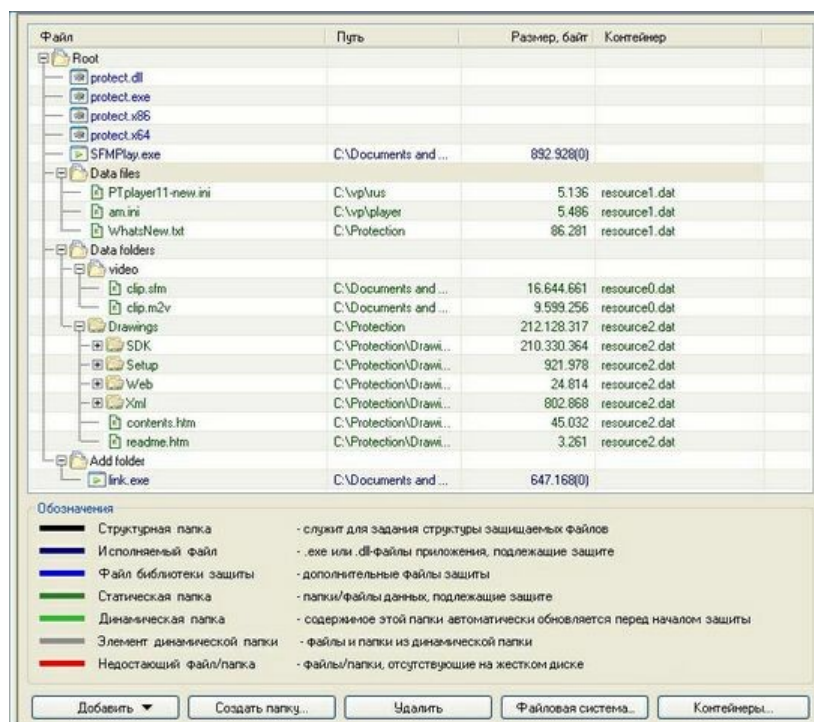
В этом разделе производится выбор файлов, подлежащих защите (см. [рисунок](#)) (см. также [Защищаемые файлы](#)).

Раздел представляет собой дерево файлов защиты (protect.dll, protect.exe, protect.x86, protect.x64), исполняемых файлов, файлов данных и папок с файлами данных, подлежащих защите. Для исполняемого файла указывается его размер, размер оверлея (в скобках) и путь к нему; для файла данных – его размер, путь к нему и контейнер, в который этот файл заключается при защите. Для папки с файлами данных и для файла данных – размер папки/файла, путь до папки/файла и контейнер, в который эта папка/файл заключается при защите. При отсутствии оверлея у исполняемого файла в скобках отображается 0.

Объекты разных типов выделены различающимися цветами, которые описаны в поле **Обозначения**.

Если для файла, указанного в разделе, требуется изменить имя папки, то следует выделить его, затем с помощью переключателя, появившегося в столбце "Путь", вызвать стандартное окно выбора папки и определить для этого файла новый путь.

Файлы, для которых требуется одинаково переопределить имена папок или задать один контейнер, можно выбрать сразу: удерживать клавишу **Shift** для выделения связной группы файлов или клавишу **Ctrl** для выборочного выделения. В этом случае достаточно будет задания пути или контейнера для одного из отмеченных файлов, чтобы это распространилось на всю выделенную группу.



Раздел "Файлы"

Кнопки

Элементами управления в разделе **Файлы** (см. [выше](#)) являются кнопки, расположенные внизу рабочего поля, и соответствующее им контекстное меню.

Кнопка Добавить

Можно добавить (см. [выше](#)):

- исполняемый файл – в выбранную (или созданную) папку;
- файлы данных – в выбранную (или созданную) папку и в выбранный контейнер, или в защищаемую папку, находящуюся внутри контейнера;
- папку, все или большинство файлов из которой подлежат защите, – в выбранную (или созданную) папку и в выбранный контейнер, или в защищаемую папку, находящуюся внутри контейнера; папка добавляется с сохранением структуры файлов и вместе со всеми поддиректориями;
- динамическую папку, содержимое которой автоматически обновляется перед началом защиты, – в выбранную (или созданную) папку и в выбранный контейнер, или в защищаемую папку, находящуюся внутри контейнера.

Смотрите также главу [Защищаемые файлы](#).

Кнопка Создать папку...

Папки создаются (см. [выше](#)) с целью задания относительных путей для защищаемых файлов. Таким образом определяется структура защищаемых файлов относительно корневой папки Root. Для создания папок служит окно "Свойства папки", появляющееся при нажатии на кнопку **Создать папку...** Цепочку из вложенных

папок можно задать одним из двух способов:

1. Последовательно создавать по одной папке и, отметив ее, переходить к созданию папки следующего уровня.
2. Сразу задать в окне свойств папки всю цепочку вложений с помощью указания пути.

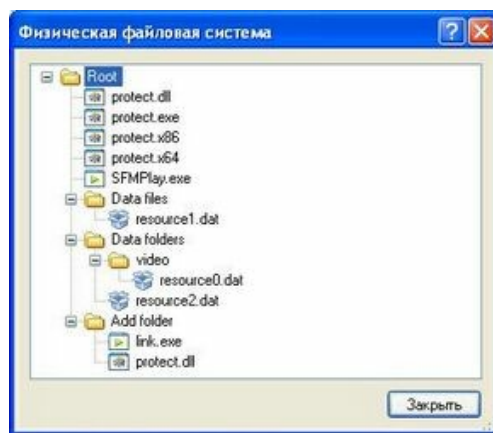
Кнопка Удалить

Кнопка становится активной, когда выделен объект для удаления. Не подлежат удалению файлы библиотеки защиты (по умолчанию PROTECT.DLL, PROTECT.EXE или PROTECT.X86/X64) и файлы/папки, входящие в состав динамических папок.

Кнопка Файловая система...

Нажатие на эту кнопку предоставляет возможность просмотреть файловую структуру защищенных файлов в защищаемом приложении.

В появившемся окне, показанном на [рисунке ниже](#), отражается структура защищенных исполняемых файлов и контейнеров относительно папки, определенной в поле **Папка с выходными данными** раздела **Защита файлов** (см. [рисунки](#)), и обозначаемой Root в разделе **Файлы** (см. [рисунки](#)). В дистрибутиве защищенного приложения в каждой папке, где есть исполняемый модуль, будет размещен экземпляр файла библиотеки защиты PROTECT.DLL, что обязательно отображается в окне "Физическая файловая система".



Файловая система

Кнопка Контейнеры... (см. [рисунки](#))

В окне "Менеджер контейнеров", открывающемся после нажатия на эту кнопку, можно создать контейнер для добавления в него файлов данных и папок, содержащих файлы данных. Кроме того, в этом окне можно установить уровень защиты контейнера и пользовательские права на него.

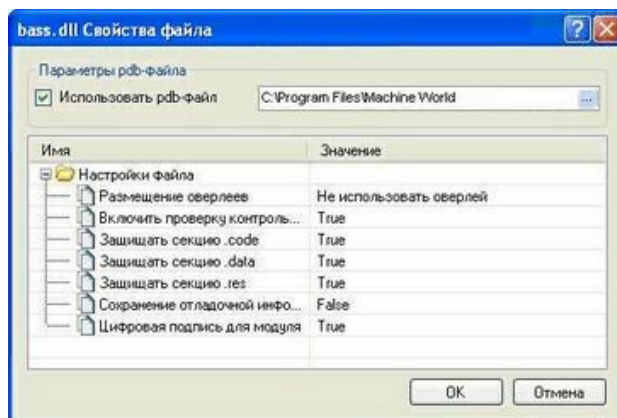
Команды контекстного меню

При нажатии на правую кнопку мыши появляется контекстное меню, вид которого зависит от того, отмечено ли на рабочем поле что-нибудь, и что именно.

Команда	Назначение
Добавить	Добавление в выделенный элемент: <ul style="list-style-type: none"> исполняемого файла в структурную папку; файла в контейнер или в статическую папку; статической или динамической папки в контейнер; аналогична по действию кнопке Добавить .
Создать папку...	Создание структурной папки внутри выделенного элемента: <ul style="list-style-type: none"> корневой папки Root; структурной папки; аналогична по действию кнопке Создать папку...
Удалить	Удаление выделенного элемента: <ul style="list-style-type: none"> файла, не входящего в динамическую папку и не являющегося файлом библиотеки защиты (по умолчанию PROTECT.DLL, PROTECT.EXE или PROTECT.X86/X64); папки, не входящей в динамическую папку и не содержащей файлы библиотеки защиты; аналогична по действию кнопке Удалить .
Менеджер контейнеров...	Вызов окна Менеджер контейнеров (см. рисунок) для создания контейнера; аналогична по действию кнопке Контейнеры...
Просмотреть файловую систему...	Просмотр файловой структуры защищенных файлов в защищаемом приложении; аналогична по действию кнопке Файловая система...
Развернуть все	Полное разворачивание дерева папок и файлов в рабочем поле раздела.
Свернуть все	Полное сворачивание дерева папок и файлов в рабочем поле раздела.
Свойства	Вызов окна свойств выделенного объекта: <ul style="list-style-type: none"> файла библиотеки защиты (по умолчанию PROTECT.DLL, PROTECT.EXE или PROTECT.X86/X64) – окна "Свойства библиотеки защиты"; исполняемого файла, не помещенного в контейнер – окна "Свойства файла"; Это же окно появляется, если, выделив файл одного из указанных типов, дважды нажать на левую кнопку мыши.
Переименовать	Вызов окна для задания имени структурной папки – окна "Свойства папки".

Раздел "Файлы". Команды контекстного меню

Свойства элементов рабочего поля



Окно "Свойства файла"

1. Окно "Свойства файла" показано на рисунке выше.

В группе **Параметры pdb-файла** предоставлена возможность добавлять в проект отладочную информацию, содержащуюся в pdb-файле, чтобы использовать ее при установке защиты. Pdb-файл должен иметь то же имя, что и исполняемый модуль. Следует помнить, что разбор pdb-файла может занять много времени.

Если при запуске сервиса **Защита файлов** pdb-файла не окажется, хотя он будет указан в проекте, то появится сообщение об ошибке, и сервис не начнет работу.

При открытии файла проекта, содержащего pdb-файл, проверка на наличие изменений будет производиться не только для исполняемого модуля, но и для соответствующего ему pdb-файла. Таблица **Настройки файла** позволяет переопределить опции, связанные с качеством защиты исполняемого файла.

Параметр	Возможное значение	Назначение
Размещение оверлеев	Не использовать оверлей; Дописывать оверлей в конец исполняемого кода; Сохранять оверлей в защищенном файле в той же позиции	Задаёт вариант обработки оверлеев при защите файла: оверлей не сохраняются; оверлей сохраняются, но положение их относительно начала файла может не совпадать с исходным; сохраняются оверлей и их положение в файле (рекомендуемое значение).
Оверлей - дополнительные данные в конце исполняемого модуля, не предусматриваемые форматом исполняемого файла. Обычно оверлей создаются архиваторами и инсталляторами. Protection Studio определяет отсутствие или наличие оверлея и в первом случае устанавливает по умолчанию значение: Не использовать оверлей , а во втором – значение: Сохранять оверлей в защищенном файле в той же позиции . Пользователь имеет возможность выбрать другое значение из списка возможных.		
Включить проверку контрольной суммы	True False	Проверка при запуске приложения: не подвергся ли изменениям защищенный файл
Защищать секцию .code	True False	Шифровать секцию .code исполняемого файла.
Защищать секцию .data	True False	Шифровать секцию .data исполняемого файла.
Защищать секцию .resources	True False	Шифровать секцию .resources исполняемого файла.
Сохранение отладочной информации	True False	Ссылка на файл PDB удаляется из модуля или нет.
Цифровая подпись для модуля	True False	Ставить цифровую подпись на модуль. Примечание. По умолчанию данный параметр имеет значение True для модулей, которые были подписаны в незащищенном виде, и значение False для модулей, не имевших подписи.

Свойства файла

2. Окно "Свойства библиотек защиты" позволяет переопределить имя библиотек

защиты (protect по умолчанию).

Внимание! В имени библиотек защиты допустимы только буквы латинского алфавита, цифры и специальные символы, за исключением следующих: * ? / \ | : < > " № `

3. Окно "Свойства папки" позволяет переименовать структурную папку.

Задание структуры защищенных файлов

Если выбранные для защиты файлы в защищенном программном продукте должны находиться в разных папках, т.е. образуют некоторую файловую структуру, то информация об этой структуре должна храниться в защищенных файлах. Для этого при формировании списка файлов для защиты можно задать пути выходных (защищенных) файлов относительно папки Root.

Поскольку после выполнения операции защиты все защищенные файлы и файлы библиотеки защиты загружаются в папку, заданную пользователем (см. [Папка для выходных данных](#)), то при заданных путях выходных файлов эта загрузка происходит следующим образом:

- При указании путей исполняемых файлов относительно корневого каталога для этих файлов и контейнеров в выходной папке будут автоматически созданы папки, указанные в путях соответствующих файлов и контейнеров, и защищенные файлы будут загружены по папкам в соответствии с заданными путями.
- Для отдельных защищенных файлов данных информация об их путях будет сохранена внутри контейнеров.

Для того чтобы корректно защитить файлы данных, необходимо сохранить их оригинальную структуру, то есть фактически воссоздать данную структуру в списке файлов. Для этого необходимо представлять, как будет строиться абсолютный путь к файлу данных внутри контейнера:

Абсолютный путь к защищенному файлу внутри контейнера равен пути до директории Root + путь от директории Root до контейнера, то есть до первой по иерархии директории, в которой встречается файл или папка, помещенная в данный контейнер, + путь внутри контейнера до данного файла (иерархия поддиректорий).

Основным способом задания путей внутри контейнера является добавление папки с файлами в контейнер.

Добавление папки с файлами в контейнер

1. Выделите нужную папку.
2. Нажмите на кнопку **Добавить** (см. [рисунок](#)).
3. В контекстном меню выделите команду **Папку в контейнер...**
4. В появившемся стандартном диалоговом окне выберите папку, большинство файлов в которой подлежат защите, и нажмите кнопку **ОК**.
5. Если создано несколько контейнеров, то в появившемся окне "Выбор контейнера" выберите контейнер. Если не было ни одного, то выберите имя для создаваемого контейнера.

6. Удалите из списка файлы, не подлежащие защите.

Основным способом задания пути до контейнера является создание в директории Root требуемого дерева папок, в нижний уровень которого добавляется папка или файлы, помещаемые в контейнер. Дерево папок строится путем создания вложенных папок.

Создание папки

1. Выберите объект (Root или папку для относительных путей), внутри которого будет создаваться папка, и нажмите на кнопку **Создать папку**.
2. На экране появится окно "Свойства папки" с полем для ввода имени папки. После ввода имени и нажатия кнопки **ОК** папка с этим именем появится внутри соответствующего объекта.
3. Для перемещения уже имеющегося в списке файла в нужную папку его необходимо отметить, нажав левую кнопку мыши, а затем, не отпуская кнопку, передвинуть его в требуемую папку (метод "*drag and drop*").

Рекомендации по защите файлов

1. Эффективно защищаются приложения, написанные на:
 - C/C++ под Visual Studio, Builder и GCC;
 - Delphi;
 - Pascal;
 - Visual Basic 6.0, за исключением защиты экспортов;
 - Visual Basic старше 6.0;
 - Microsoft .NET.
2. Для .Net-приложений managed dll загружаются только в тот момент, когда используется код из них. Соответственно, protect.dll загружается только в момент обращения к функциям из защищенного модуля, и весь интерфейс защиты выводится только после загрузки модуля. Таким образом, незащищенное приложение, использующее защищенную dll, может работать, пока не вызовет функцию из защищенного модуля. Этого можно избежать, если незащищенное приложение будет вызывать в начале работы функцию для загрузки protect.dll (например, LoadLibrary). Другой способ – защищать исполняемый модуль вместе с dll.
3. При отборе файлов, подлежащих защите, и определении уровня защиты необходимо учитывать, что чрезмерная защита тех или иных компонентов программного продукта может привести к недопустимой потере производительности приложения, и модифицировать параметры защиты по результатам тестирования защищенного продукта.

Рекомендации по защите игр

1. Многие игры позволяют пользователям создавать свои модификации, переделывать внутриигровую механику (геймплей) под себя, использовать новые текстуры,

делать новые карты и т.п. – то есть осуществлять так называемый моддинг.

Но установка защиты именно на те ресурсы, которые могут подвергаться модифицированию, лишает пользователя такой возможности.

Поэтому если архитектура игры поддерживает создание для нее продолжений или дополнений (модов), или предполагается после выпуска продукта выпустить какие-то утилиты для написания собственных карт, скинов, модов, то защищать ресурсы в таких играх не рекомендуется.

2. После выпуска демо-версии при подготовке финальной версии защищенного приложения рекомендуется выполнить следующие действия:
 - сделать модули приложения (демо- и финальной версии) несовместимыми по таблицам экспортов (см. раздел ниже);
 - объявить и защитить дополнительные функции (см. раздел ниже);
 - изменить формат некоторых файлов данных, например, уровней.

3.2.3.5 Раздел "Функции"

В этом разделе задаются процедуры (функции) исполняемых файлов, подлежащие защите.

Защита функций является опциональной и призвана значительно повысить уровень защиты кода исполняемых файлов от взлома. Защита функций включает два метода:

- Защита внутренних (экспортируемых) функций исполняемого файла;
- Защита таблиц импорта функций, обычно называемая *защитой импортов*.

Защита внутренних функций

Принципы работы

Защищенные внутренние функции являются *наиболее важным элементом* защиты исполняемых модулей, применяемым в системе **StarForce**.

Защищаемая внутренняя функция удаляется из программного кода модуля разработчика и переносится в ядро защиты. На этапе исполнения в момент вызова защищенной функции приложение передает управление ядру, и функция выполняется "скрыто". Для того чтобы система защиты определила точку входа в функцию, эта функция в исходной программе должна быть предварительно помещена в таблицу экспортов, или к исполняемому файлу должен прилагаться pdb-файл.

Типы защищаемых функций

Защищаемые функции можно разделить на три группы:

Loopback-функции. При защите помимо удаления функции из модуля, ссылка на нее удаляется из таблицы экспортов данного модуля. Защита функции в качестве loopback является наиболее рекомендуемым вариантом.

Public-функции. Отличаются от loopback-функций тем, что ссылка на функцию не удаляется из таблицы экспортов. Такой способ защиты применяется к функции, если она используется и другими модулями приложения, то есть является «настоящим» экспортом.

Callback-функции. Их вызов производится самим ядром защиты после расшифровки защищенного модуля, но перед его запуском. Таким образом, вызов callback-функции является неявным и отсутствует в защищенном модуле. Такой способ защиты функции является наиболее взломоустойчивым.

Объявление экспортируемых функций

Использование специального кода

Для объявления функции как экспортируемой в исходный проект следует включить специальный код.

- При программировании на языке C/C++ используется ключевое слово `__declspec (dllexport)`. Например:

```
__declspec( dllexport) <тип функции> <заголовок функции>
```

```
__declspec( dllexport) void InitAVICodec( void) ; - для callback-функции;
```

```
__declspec( dllexport) int RunMasterMenu( CMenu &, int ); - для loopback-функции.
```

- При программировании на языке PASCAL используется ключевое слово `exports`. Например:

```
exports
```

```
    YourFirstExportedFunction name ' YourFirstExportedFunction' ;
```

```
    YourSecondExportedFunction name ' YourSecondExportedFunction' ;
```

```
    YourThirdExportedFunction name ' YourThirdExportedFunction' ;
```

Внимание!

Компилировать проект нужно с включенной опцией генерирования fixups (relocation table) линковщика.

- В среде MS Visual C++ для включения данной опции следует добавить строку **/FIXED:NO** в список опций проекта **Project Options**, расположенный на вкладке **Link** в диалоговом окне свойств проекта.
- В среде Visual Studio.NET данная опция включается в поле **Additional Options** раздела **Linker -> Command Line** в диалоговом окне свойств проекта.
- В среде Borland Builder / Delphi компилятор генерирует fixups по умолчанию. Если требуется принудительная генерация fixups, необходимо добавить опцию **/b:xxxx** линковщика (Command-line switch = /b:xxxx). Для .EXE-файлов в Win32 обычно указывается адрес 00400000h, для .DLL - 10000000h.

После компиляции необходимо убедиться в наличии строки ".reloc" в защищаемом модуле.

Использование префиксов

Существует возможность задания параметров функций для автоматической установки их системой защиты. Эти параметры оформляются в префикс, который прикрепляется конкатенацией впереди к имени функции. Если префикс задан, то при защите данного исполняемого файла функция с префиксом автоматически будет защищена с использованием заданного метода, а также (если определено) с заданной скоростью

исполнения. Таким образом, не обязательно задавать параметры защиты функций вручную в Protection Studio – для автоматизации процесса разработки защиты удобнее использовать префиксы. Соответственно трем типам защищаемых функций разделяют три типа префиксов:

- для callback-функций: SFINIT<k>_<v>;
- для loopback-функций: SFLB_<v>;
- для public-функций: SFPROT_<v>;

где <k> - порядковый номер, под которым вызывается callback-функция;
<v> - скорость выполнения функции (может принимать значения от 0 до 4, что соответствует значениям от **очень медленно** до **очень быстро** (соответственно), см. [таблицу](#)).

Пример: SFPROT_3foo, где SFPROT – префикс, 3 – скорость выполнения функции, foo – имя функции.

Использование pdb-файла

Можно сделать защищаемыми внутренние функции и без объявления их экспортами. Для этого необходимо в разделе **Файлы** Protection Studio (см. [Раздел "Файлы"](#)) добавить в проект защиты соответствующий данному исполняемому модулю pdb-файл с отладочной информацией. В результате появляется возможность устанавливать параметры защиты для внутренних функций данного файла: таблица экспортов (узел "Экспорты") заменяется таблицей всех его внутренних функций (узел "Внутренние функции").

При этом исчезают настройки для функций, экспортировавшихся по ординалу. Параметры защиты для внутренних функций устанавливаются так же, как для экспортируемых функций.

Если в окне "Свойства файла" раздела **Файлы** снимается флажок **Использовать pdb-файл**, то узел «Внутренние функции» заменяется на узел «Экспорты», соответственно, показываются только экспортируемые функции, а настройки непроэкспортированных или проэкспортированных по ординалу функций пропадают.

Область применения

Область применения защищенных функций ограничена задержками, вносимыми на этапе исполнения. Время задержки зависит от профиля защиты, параметров защищенной функции и типа привязки. Если функция имеет циклы, время ее исполнения может увеличиться.

Внимание! Время исполнения сложных функций после защиты может значительно увеличиваться. Во избежание замедления работы защищенного приложения StarForce рекомендует особенно внимательно выбирать защищаемые функции и обращаться за консультацией в службу технической поддержки.

Кроме того, при защите функций следует обратить внимание на следующие моменты:

- Защита функций основывается на сокрытии их кода. Поэтому недопустимо распространение демонстрационных и других версий программ, в которых эти функции присутствуют в открытом виде.

- С точки зрения защиты наиболее эффективны callback-функции, однако, поскольку callback-функция может вызываться только один раз до запуска защищенного модуля, подходящую callback-функцию выбрать сложно. Поэтому особое внимание следует уделить loopback-функциям.
- Функция не должна просто инициализировать переменные константами. В противном случае хакеру достаточно результатов работы для того, чтобы восстановить защищенную функцию.
- Функции, защищаемые как callback, не должны пользоваться функциями из RTL компилятора.
- Хорошего уровня защиты можно достичь, защитив около 10-20 экспортируемых функций, критичных для работы программы, размером несколько сотен байт каждая.

Замечание. Не рекомендуется защищать более 20-25 экспортируемых функций. Превышение указанного значения может привести к недостатку ресурсов сервера защиты и невозможности обработать данный запрос; в этом случае выводится сообщение «Exception of type 'SystemOutOfMemoryException' was thrown».

Внимание! При защите функций необходимо помнить, что некоторые функции при небольшом объеме кода автоматически преобразуются компилятором в inline. Поскольку защита inline-функций невозможна, необходимо, чтобы все защищаемые функции были заданы в исходном коде защищаемого приложения как noline.

Пример для Visual C++

```
__declspec( noline ) void MyFunction()
{
    ...
}
```

Эффективность

Защита внутренних функций является чрезвычайно эффективным способом защиты от взлома программы.

Защита импортов

Принципы работы

Исполняемые модули могут использовать функции других модулей с помощью механизма *динамического связывания*. В Windows почти каждый исполняемый модуль использует функции из различных динамических библиотек (DLL).

Динамическое связывание реализуется с использованием механизмов *экспортирования* и *импортирования* функций.

При экспортировании функций в модуле прописывается таблица их имен с указанием адресов точек входа. При импортировании функций указывается имя модуля, из которого берется функция, имя функции и адрес ячейки, в которую загрузчик поместит адрес этой функции. При запуске модуля загрузчик помещает его в память и проходит по всей таблице импортов. Если он встречает имя модуля, еще не загруженного в контекст данного процесса, он загружает этот модуль и ищет в его

таблице экспортов соответствующую функцию. Если функция найдена, ее адрес помещается в соответствующую ячейку таблицы импортов, если нет – выдается сообщение об ошибке.

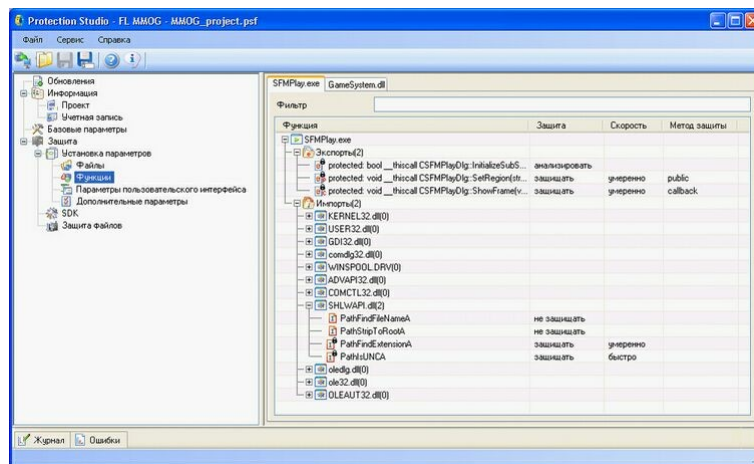
Идея защиты вызовов импортируемых функций заключается в том, что если не знать, какая импортируемая функция вызывается при обращении к данной ячейке таблицы импортов, восстановить работоспособный модуль по дампу памяти достаточно сложно. Для обеспечения сокрытия вызовов импортируемых функций защита StarForce при обработке модуля удаляет его таблицу импортов, а после успешного прохождения проверки на легальность запускаемой копии восстанавливает импорты самостоятельно. Сокрытие вызовов обеспечивается тем, что часть вызовов импортируемых функций заменяется вызовами функций ядра.

Область применения

Защита вызовов импортируемых функций имеет одно существенное ограничение: для обеспечения надежного сокрытия вызова время от передачи управления ядру до вызова защищенной функции должно быть достаточно большим. Это требует аккуратного выбора импортов, вызовы которых будут защищены. Целесообразно придерживаться следующего ограничения: импортируемая функция, вызов которой защищен, не должна вызываться чаще 20 раз в секунду.

Эффективность

Данная защита эффективна только при достаточном количестве защищенных вызовов импортируемых функций. Минимальное рекомендуемое количество защищенных импортов составляет 10-15 штук для одного модуля.



Раздел "Функции"

Для раздела **Функции** правая панель состоит из набора вкладок. Каждая вкладка является рабочим полем для отдельного программного файла из списка защищаемых файлов. На ярлыках вкладок выводятся имена соответствующих файлов.

Для удобства поиска функций в разделе находится **Фильтр**, в поле которого вводится имя или его фрагмент.

Рабочее поле представляет собой таблицу экспортируемых (узел "Экспорты") и импортируемых (узел "Импорты") функций программного файла. Импортируемые функции сгруппированы по библиотекам, из которых они вызываются.

После названия узла/библиотеки в скобках указывается общее количество функций в узле/библиотеке, которые защищаются.

В таблице выводятся следующие данные по каждой функции:

Заголовок столбца	Возможное значение	Назначение
Функция	не задается	Имя функции
Защита	не защищать анализировать защищать	Вариант установки защиты (для экспортов и импортов соответственно)
	не защищать защищать	
Скорость	очень медленно медленно умеренно быстро очень быстро	Требуемая скорость выполнения функции (чем выше скорость выполнения, тем ниже уровень защиты, и наоборот)
Метод защиты	public callback loopback	Тип защиты функции (только для экспортов)

Параметры функции

В столбцах **Скорость** и **Метод защиты** информация выводится только при выборе значения **защищать** в столбце **Защита**.

В каждом программном файле можно использовать следующие варианты установки защиты функций.

Вариант установки защиты	Кем производится
Анализировать функцию	Сервером защиты при выполнении операции защиты. Этот режим применяется ко всем экспортируемым функциям по умолчанию. При выборе этого значения функция анализируется, и в ней могут быть защищены некоторые внутренние переходы.
Защищать функцию	Пользователем. Рекомендуется для квалифицированного пользователя. Для импортов в этом случае можно настроить только скорость.
Указание не защищать данную функцию	Пользователем. Этот режим применяется ко всем импортам по умолчанию.

Варианты установки защиты функции

Выбор параметров защиты при установке значения **защищать** требует достаточно хорошего знания кода продукта и рекомендуется только для квалифицированного пользователя.

Порядок защиты функции

1. Выберите функцию или несколько функций, для которых будут задаваться параметры защиты (см. [рисунки выше](#)). Функции, для которых требуется определить одинаковые параметры, можно выбрать сразу: нажмите на клавишу

Shift для выделения связной группы функций или на клавишу **Ctrl** для выборочного их выделения. В этом случае достаточно будет задания параметров защиты для одной из отмеченных функций, чтобы они распространились на всю выделенную группу.

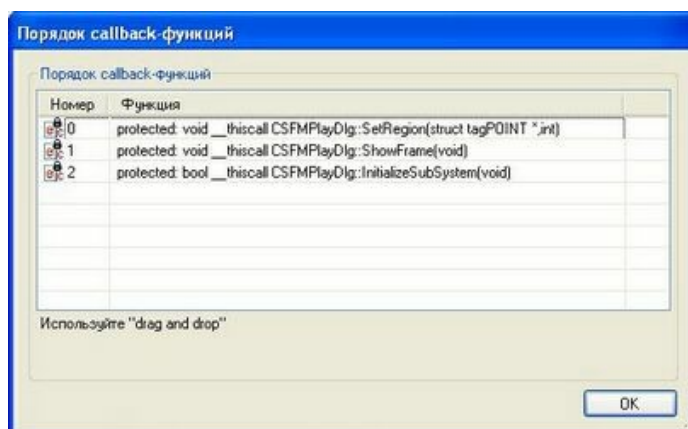
2. Выберите в столбце "Защита" одно из трех значений:
 - **анализировать** (по умолчанию, только для экспортов)
 - **защищать**
 - **не защищать** (значение по умолчанию для импортов).
3. При выборе значения **защищать** выберите одно из пяти значений в столбце "Скорость":
 - **очень медленно**
 - **медленно**
 - **умеренно**
 - **быстро**
 - **очень быстро**.
4. *Только для экспортируемых функций* при выборе значения **защищать** выберите одно из трех значений для способа защиты в столбце "Метод защиты":
 - **public**
 - **loopback**
 - **callback**.

Контекстные меню рабочего поля

При нажатии на правую кнопку мыши появляется контекстное меню, вид которого зависит от того, где именно в таблице вызывается контекстное меню.

1. Команды **Развернуть все** и **Свернуть все** появляются всегда и управляют представлением структуры функций в рабочем поле окна раздела, соответственно разворачивая и сворачивая полностью деревья импортируемых и экспортируемых функций.
2. Команда **Порядок callback...** появляется, если не менее чем для двух экспортов выбран тип `callback`. Использование этой команды вызывает появление окна "Порядок callback-функций" (см. [рисунок](#)).

Здесь можно менять порядок вызова `callback` функций, используя метод "*drag and drop*", то есть отметить функцию, нажав левую кнопку мыши, а затем, не отпуская кнопку, передвинуть функцию в нужное место. Данный порядок задает очередность вызова защищенных функций из библиотеки защиты.



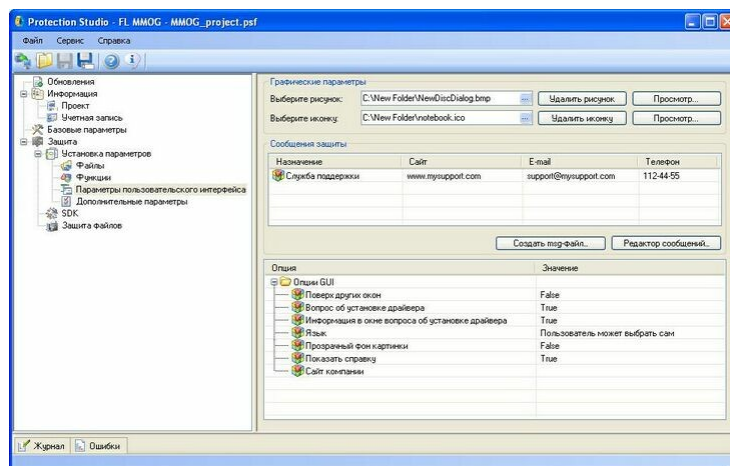
Раздел "Функции". Окно "Порядок callback-функций"

3. Команда **Свойства** присутствует в контекстном меню, если отмечена экспортируемая функция со значением **защитить** в столбце **Защита**. Выбор этой команды вызывает появление окна "Свойства функций", в котором определяются пользовательские права экспортируемых функций.

Следует иметь в виду, что пользовательские права, определенные для какой-либо функции при установке переключателя в положение **защитить**, не сохраняются, если потом для нее выбирается значение **анализировать** или решается вообще не защищать ее. Если впоследствии для этой функции происходит возврат к значению **защитить**, необходимо снова указать нужные права в окне "Свойства функций".

3.2.3.6 Раздел "Параметры пользовательского интерфейса"

В этом разделе определяются три группы параметров: графические параметры, сообщения защиты и дополнительные опции графического интерфейса (см. [ниже](#)).

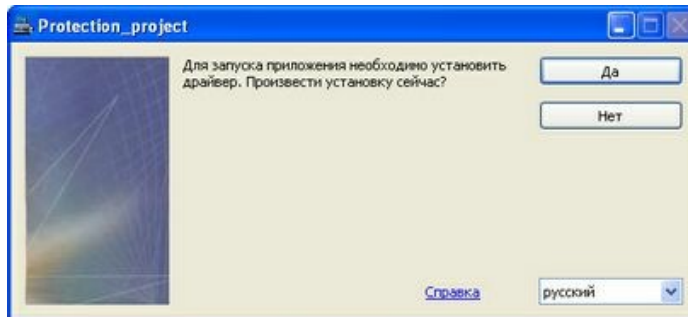


Раздел "Параметры пользовательского интерфейса"

- В группе **Графические параметры** определяются рисунок и иконка графического интерфейса, которые затем можно просмотреть, нажав на соответствующие кнопки справа.

Пример интерфейсного окна защищенного приложения показан на [рисунке ниже](#); здесь:

- иконка – в левом верхнем углу, в заголовке (формат .ico, содержащий конфигурацию 16x16 и 32x32 пикселей);
- рисунок – слева в основной части окна (формат .bmp, 113x195 пикселей).



Окно системы защиты

- В группе **Сообщения защиты** предоставлена возможность определять контактную информацию, которая будет отображаться в сообщениях защиты, а также редактировать сами сообщения защиты:
- в таблице контактной информации могут быть заданы адрес сайта, электронный почтовый адрес и номер телефона для обращения пользователя по различным вопросам:

Значение	Назначение параметров
Служба поддержки	Контактная информация для обращения в Службу поддержки продукта.

Группа контактной информации для пользователя

для редактирования сообщений графического интерфейса защиты нажмите на кнопку **Редактор сообщений**. При этом открывается редактор сообщений, описание которого представлено в главе [Редактор сообщений](#). При необходимости генерации дополнительных msg-файлов сообщений нажмите на кнопку **Создать msg-файл...** и укажите название файла в стандартном диалоговом окне.

Внимание! Сообщения GUI появляются на компьютере конечного пользователя в одном из нижеперечисленных случаев:

1. При установке драйвера;
2. При возникновении ошибок.

- В группе **Опции GUI** можно переопределить дополнительные параметры графического интерфейса, значения которых выставлены по умолчанию:

Параметр	Возможное значение	Назначение
Поверх других окон	False True	Диалоговые окна интерфейса защиты будут всегда находиться поверх других окон.
Вопрос об установке драйвера	True False	При запуске приложения в случае, если не установлен драйвер защиты, на экран выводится сообщение о том, что происходит процедура установки драйвера защиты. Эту опцию отключать не рекомендуется в связи с негативной реакцией пользователей защищенного

Параметр	Возможное значение	Назначение
		продукта на скрытую установку драйвера.
Информация в окне вопроса об установке драйвера	True False	В русской и английской версии интерфейса защиты в случае, если эта опция и опция Вопрос об установке драйвера включены, в окне с вопросом будет отображаться ссылка Справка , позволяющая пользователю просмотреть информацию о назначении и поведении устанавливаемого драйвера защиты.
Показать справку	True False	В русской и английской версии интерфейса защиты в окне сообщения об ошибке появляется ссылка Справка , позволяющая пользователю просмотреть краткую справку по проблеме.
Язык	Пользователь может выбирать сам Автоматически Конкретный язык	В окне защиты сообщения отображаются на языке региональных настроек операционной системы, но пользователь может выбрать любой другой язык из списка в этом окне; Отличается от предыдущего отсутствием возможности выбора языка из списка; Сообщения всегда отображаются на данном языке вне зависимости от настроек пользователя.
Прозрачный фон картинки	False True	Сделать цвет фона (левого верхнего пикселя) картинки прозрачным.
Сайт компании	-	Ссылка на описание продуктов на сайте компании.

Опции GUI

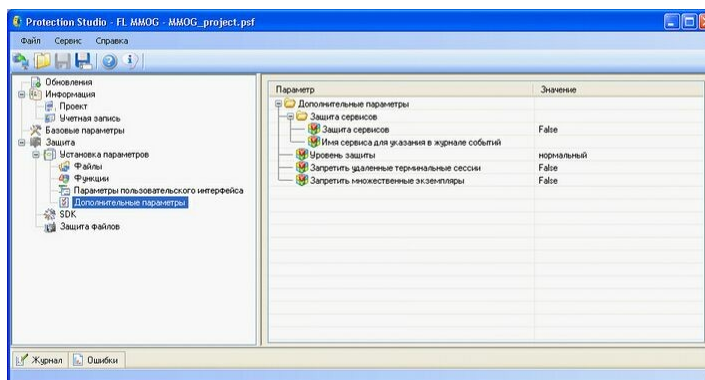
Замечание. Встроенная защита от перевода часов не разрешает запускать приложение, если текущее время меньше времени последнего успешного запуска приложения.

Пробный период

Для продукта FL MMOG поддерживается пробный период. Для включения данной опции отправьте запрос в службу технической поддержки StarForce и укажите продолжительность пробного периода (по умолчанию 30 дней с момента установки защиты). В этом случае защищенное приложение будет работать в течение указанного времени; при каждом запуске выводится сообщение о том, что приложение является некоммерческим. При запуске приложения после изменения системного времени выводится сообщение о некорректной установке времени. По истечении пробного периода выводится сообщение об ошибке.

3.2.3.7 Раздел "Дополнительные параметры"

Страница раздела представлена на [рисунке ниже](#).



Раздел "Дополнительные параметры"

Значения дополнительных параметров определяют уровень защиты и возможности использования приложения.

Параметр	Возможное значение	Назначение
Профиль защиты	низкий уровень и высокая скорость защиты, небольшой размер библиотеки защиты оптимальный высокий уровень и низкая скорость защиты, большой размер библиотеки защиты	Выбор уровня шифрования. Повышение уровня шифрования повышает качество защиты, но увеличивает размер библиотеки защиты и время исполнения защищенных функций.
Запретить удаленные терминальные сессии	False True	При установке этого параметра защищенное приложение невозможно запустить с сервера с использованием терминальных сессий (подключения к удаленному рабочему столу). Таким образом, отсекается возможность использования защищенного программного продукта несколькими людьми одновременно при наличии только одной лицензии.
Запретить множественные экземпляры	False True	Исключается возможность запуска более одного экземпляра приложения на одном компьютере.

Дополнительные параметры защиты

Защита сервисов

Эти параметры задаются при защите Windows-сервисов (служб).

Windows-сервис представляет собой приложение, которое может быть запущено до того, как пользователь осуществит вход в систему. Для реализации защиты сервисов необходимо обеспечить их работу без пользовательского интерфейса. Пользователь получает два продукта: собственно защищенный сервис и активатор, т.е. программу, предназначенную для создания и активации лицензии. При установке сервиса пользователь должен запустить активатор и произвести активацию с использованием обычного интерфейса защищенного программного продукта. После этого сервис при

запуске будет пользоваться готовой активированной лицензией, а в случае возникновения ошибок информация записывается в журнал событий (EventLog) системы.

Т.к. сервисы не выводят диалоговых окон и все их сообщения сохраняются в системном журнале событий, указанные ниже параметры определяют порядок записи сообщений защищенного сервиса в системном журнале событий.

Параметр	Возможное значение	Назначение
Защита сервисов	False True	Установка защиты сервисов
Имя сервиса для указания в журнале событий	Имя	Имя сервиса, указываемое в журнале регистрации событий (Event Log).

Опции защиты сервисов

3.2.4 Защита файлов

Защита файлов выполняется для каждого проекта после установки всех требуемых параметров. Сервис защиты файлов представлен соответствующим разделом, показанным на [рисунке выше](#).

В разделе представлено две группы полей.

В группе **Обзор параметров защиты** в табличной форме приведена информация об общем размере и количестве файлов в контейнерах, о количестве экспортируемых и импортируемых функций, которые были определены для защиты.

В группе **Параметры выходных данных** есть возможность переопределить папку для выходных данных (см. [Папка для выходных данных](#)).

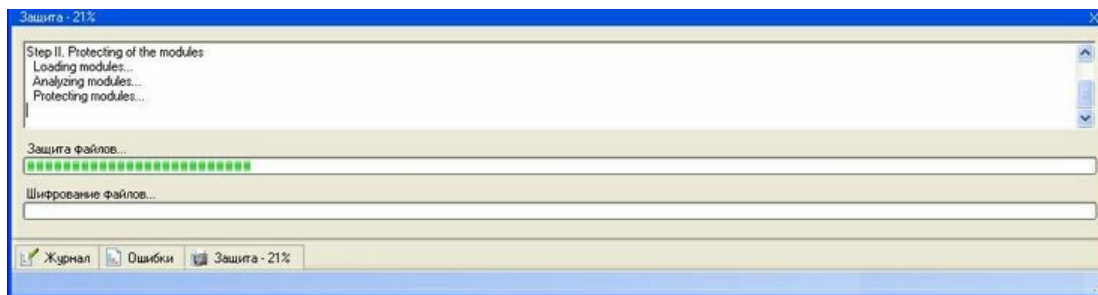
В этой же группе полей есть возможность задать удаление содержимого папки перед защитой путем выставления флажка в соответствующем поле.

Внимание! Содержимое папки удаляется без возможности восстановления.

Запуск сервиса осуществляется нажатием на кнопку **Защитить** или нажатием на клавишу **Enter**.

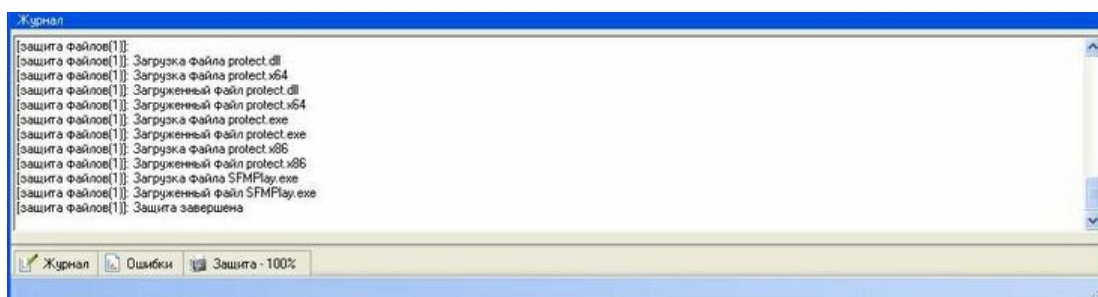
Если в проекте обнаружены ошибки, то на экран выводится окно с сообщением об ошибках (см. [Диагностика и устранение ошибок при установке защиты](#)) и сервис прекращает работу.

Если ошибки не обнаружены, то начинается процесс защиты, который сопровождается сообщениями во вкладке **Защита**, а прогресс защиты указывается в процентах и отображается индикаторами процесса (см. [ниже](#)).



Вкладка "Защита"

Кроме того, вся информация о производимых операциях выводится во вкладке **Журнал** (см. [ниже](#)).



Вкладка "Журнал"

После завершения процесса защиты при нажатии на кнопку **Открыть папку...** (см. [рисунок](#)) открывается папка, заданная в разделе **Защита файлов** (см. [Папка для выходных данных](#)) и содержащая следующие файлы:

Виды файлов	Комментарии
Защищенные программные файлы	Перечень защищенных файлов задавался в разделе Файлы
Файлы контейнеров, содержащие защищенные файлы данных (если таковые есть)	
Файл библиотеки защиты (по умолчанию PROTECT.DLL)	Все эти файлы имеют одинаковое имя, заданное в разделе Файлы (но разное расширение)
Исполняемый файл интерфейса защиты (по умолчанию PROTECT.EXE)	
Файлы установки защиты для различных ОС (PROTECT.X64 и PROTECT.X86)	
Файл текстовых сообщений интерфейса защиты	Формируется на основе информации, введенной в Редакторе сообщений (см. Редактор сообщений), и только при наличии таковой в проекте. Имя файла совпадает с именем библиотеки защиты. Расширение -. msg. Если изменения сообщений интерфейса защиты не вносились, такой файл не генерируется.

Файлы, сформированные сервисом "Защита файлов"

Эти файлы размещаются в папке в соответствии с файловой структурой раздела **Файлы** (см. [рисунок](#)), а файл библиотеки защиты PROTECT.DLL помещается в каждую папку, где есть исполняемый файл.

Следует помнить, что перед новым запуском сервиса **Защита файлов** необходимо

закрыть вкладку **Защита** (см. [выше](#)), и только после этого кнопка **Защитить** снова станет активной.

3.2.5 Формирование дистрибутива

Перед тестированием приложения следует сформировать дистрибутив защищенного приложения.

В дистрибутив приложения необходимо включить следующие файлы.

1. Защищенные исполняемые .EXE и .DLL файлы приложения.
2. Файлы библиотек защиты (по умолчанию PROTECT.DLL, PROTECT.EXE, PROTECT.X86/X64), которые должны быть размещены в соответствии со структурой защищенного приложения. Файл PROTECT.DLL должен быть помещен во все папки, содержащие защищенные исполняемые файлы.

Внимание! Не рекомендуется размещать защищаемый продукт таким образом, чтобы файл библиотеки защиты оказывался в каком-либо системном каталоге.

3. Файлы контейнеров данных (если таковые есть). Файлы контейнеров должны размещаться в папках в соответствии со структурой в разделе **Файлы**.
4. Файл с расширением .GUI (по умолчанию PROTECT.GUI), если настройки графической оболочки интерфейса защиты хранятся во внешнем файле (файл создается с помощью StarForce SDK, см. [StarForce SDK](#)).
5. Файл с расширением .MSG (по умолчанию PROTECT.MSG); создается только если в стандартный текст сообщений были внесены какие-либо изменения (см. [Редактор сообщений](#)). Такой файл должен быть помещен в папку, содержащую файлы PROTECT.EXE и PROTECT.X86/X64.

Все указанные выше файлы уже имеются в папке, заданной в разделе **Защита файлов**, после окончания операции защиты.

Также дистрибутив должен содержать все остальные файлы приложения, кроме защищенных исполняемых модулей и защищенных файлов данных.

На финальных этапах защиты, после того как дистрибутив собран, если для приложения создается специальная программа-инсталлятор для автоматизированной установки приложения на компьютер пользователя, инсталлятор необходимо перекомпилировать с учетом изменений внутри дистрибутива приложения (замена оригинальных версий файлов защищенными и добавление служебных файлов защиты).

Сценарий создаваемого инсталлятора также может содержать действия по переопределению опций защиты ключами системного реестра (для обеспечения гибкости настройки защищенного приложения). Подробное описание и список опций, которые могут быть переопределены, — в главе [Изменение настроек системы защиты](#).

3.2.6 Запуск сервисов из командной строки

Сервисы могут запускаться из командной строки с параметрами, позволяющими производить подключение к серверам защиты, не используя интерфейс Protection Studio. Таким образом, они могут запускаться из командных файлов и скриптов, что позволяет частично автоматизировать процесс защиты приложения.

Название сервиса	Параметр командной строки	Описание
Генерация SDK	-Action:GenerateSDK	Запускает сервис генерации SDK
Защита файлов	-Action:ProtectFiles	Запускает сервис защиты файлов
Обновление	-Action:Update	Запускает сервис автоматического обновления PS2

Сервисы с параметрами командной строки

Параметры запуска сервисов из командной строки:

№	Название	Описание
1	-Login:	Имя (login) учетной записи StarForce.
2	-Password:	Пароль учетной записи StarForce.
3	-Host:	Сервер защиты.
4	-WorkspaceId:	ID рабочего пространства (проекта защиты).
5	-Project:	Полный путь и имя файла проекта (PSF).
6	-Action:	Действие (сервис), которое необходимо выполнить. Список доступных действий указан выше.
7	-Log:	Полный путь и имя файла, в который будет записываться журнал (лог) работы сервиса.
8	-OutputFolder:	Папка, в которую будут помещены все файлы, которые являются результатом работы сервиса.
9	-FolderWithExeFiles:	Путь к папке, исполняемые файлы (exe и dll) которой подлежат защите.

Параметры запуска сервисов из командной строки

Примечания:

1. После знака ":" кавычек ставить не надо (правильный пример -Login:Roman).
2. Только два параметра (-Action: и -Log:) являются обязательными в общем случае.
3. Если в окне "Подключение PS" выставлен флажок **Сохранять данные подключения** (см. [рисунок](#)), то параметры -Login, -Password, -Host нужно задавать только в том случае, если необходимо переопределить эти значения.
4. Если в окне "Подключение PS" выставлен флажок в поле **Автоматически выбирать данное рабочее пространство** (см. [рисунок](#)), то параметр -WorkspaceId: нужно задавать только в том случае, если необходимо переопределить это значение.

5. Порядок параметров не имеет значения.

Коды возврата:

Код	Значение
0	Процесс завершён успешно.
1	Ошибка командной строки (если нет двух обязательных параметров, или указан неверный путь к журналу событий или файлу проекта).
10	Неправильно задана учетная запись.
11	Неправильно задан сервер защиты.
12	Указан неверный ID рабочего пространства (его невозможно перевести в цифры).
13	Невозможно получить информацию о данном рабочем пространстве (возможно, его не существует, или оно не назначено в данную учётную запись).
14	Невозможно найти файл проекта по указанному пути.
15	Log-файл не указан.
16	Ошибка в указании имени папки для выходных данных.
21	Действие, которое выбрал пользователь, либо не существует, либо недоступно для пользователя.
22	Произошла ошибка во время работы сервиса. Более подробная информация должна храниться в журналах событий.
23	Произошла ошибка во время автоматического обновления. Более подробная информация должна храниться в журналах событий.
24	Приложение закрыто пользователем (если пользователь сам закрывает Protection Studio, запущенное из командной строки без параметров или вообще не из командной строки).
25	Папка по указанному пути не существует.

Коды возврата

3.3 Тестирование защищенного приложения

Тестирование защищенного продукта является крайне важным этапом, так как только по его результатам можно оценить приемлемость выбранного уровня защиты программного продукта и возможности поставки защищенного продукта конечному пользователю.

Тестирование работы защищенного приложения производится для того, чтобы убедиться, что установленная защита не влияет на основные рабочие характеристики приложения, т.е. необходимо убедиться, что скорость работы и размеры защищенного приложения соответствуют установленным требованиям.

Поэтому действия должны осуществляться по типовой программе тестирования нового продукта, и необходимо обязательно убедиться в том, что вся функциональность незащищенного приложения полностью поддерживается в защищенной версии. В связи с этим рекомендуется проводить полный тестовый прогон продукта как до, так и после проведения финальной защиты его файлов.

3.4 StarForce SDK

3.4.1 Состав SDK

StarForce SDK предназначен для разработки программ, которые защищаются системой StarForce и используют ее специальные возможности, доступные через API.

StarForce SDK можно использовать для моделирования поведения защищенного приложения в разных ситуациях на этапе разработки и тестирования.

В состав SDK также входят секретные классы, позволяющие повысить уровень защиты программного продукта от взлома.

SDK включает пакет файлов, подлежащих интеграции в защищаемое приложение:

- `protect.dll` – отладочная версия библиотеки защиты. Функциональность библиотеки защиты, включаемой в приложение при финальной защите, полностью сохранена в этой версии, при этом отладочная библиотека включает некоторые дополнительные проверки, в то же время позволяя пропускать некоторые (медленные) этапы инициализации, такие, как проверка диска и активация. Также отладочная версия не содержит защиты от отладчиков, что делает возможной полноценную отладку защищаемого приложения;
- `protect.lib` – библиотека экспортов из `protect.dll` (используется для компоновки `protect.dll` и разрабатываемого программного продукта);
- `protect.exe` – GUI защищенного приложения. Данный файл идентичен генерируемому при финальной защите, в частности, содержит те же проверки версии ОС и т.п.;
- `protect.x86` – для поддержки платформы Win32 на процессорах x8632, `protect.x64` – для поддержки платформы Win64 на процессорах x8664; в случае защиты с драйвером могут иметься оба этих файла.
- `PsaApi.h` – файл для C/C++ с объявлениями функций API, используемых для получения информации о лицензиях, а также объявления защищённых классов для C++ (если они используются).
- `PscApi.h` – файл для C/C++ с объявлениями функций API, используемых для инициализации приложения.
- `PsConstants.h` – файл для C/C++ с описаниями констант, используемых в SDK (главным образом, возвращаемые значения функций API).
- `PsConfig.h` – конфигурационный файл для C/C++, содержащий настройки проекта защиты. Этот файл содержит набор `#define`-директив, таких, как путь ключей защиты в реестре и т.п.
- Файлы в папке Delphi (`PsaApi.pas`, `PscApi.h`, `PsConstants.pas`, `PsConfig.pas`) – аналоги файлов `PsaApi.h`, `PscApi.h`, `PsConstants.h`, `PsConfig.h` для интеграции SDK с приложениями, написанными на Borland Delphi.

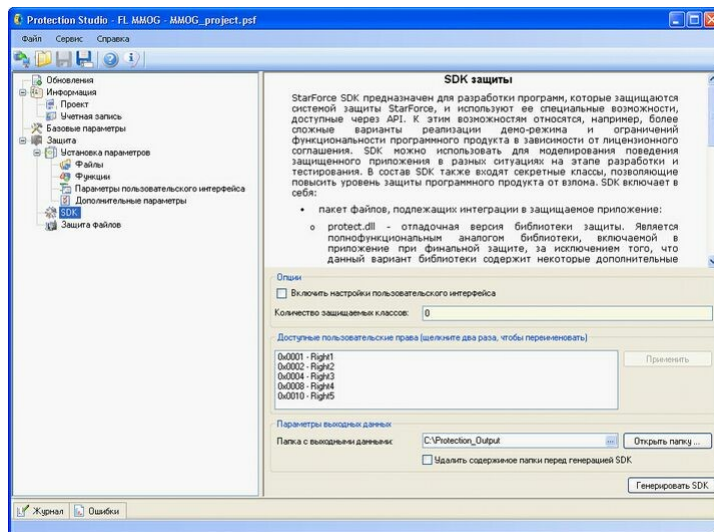
Примечание. Файлы с именем `protect` будут на самом деле иметь имя, заданное в настройках проекта (см. ниже).

LIB-файл можно создать вручную, написав свою DLL, которая имела бы пустые функции с именами, совпадающими с именами функций StarForce API. Это может быть необходимо, если используется компилятор, который не может использовать файл `protect.lib`, входящий в состав SF SDK, например, Microsoft Visual Studio 6.0.

Примечание. Некоторые компиляторы (например, Visual C++ 7.1) не поддерживают исходные тексты в Unicode. Файл `PsConfig.h` поставляется в кодировке Unicode, поэтому для работы с Visual C++ 7.1 данный файл необходимо сконвертировать в ANSI (при этом надо следить за корректностью определённых в этом файле названий продукта и компании, если они заданы не на латинице).

Файлы SDK генерируются индивидуально для каждого проекта защиты. В зависимости от настроек проекта изменяется и состав SDK.

3.4.2 Генерация SDK



Сервис "SDK"

Для генерации и получения SDK выполните следующее:

1. Загрузите требуемый файл проекта или создайте новый.
2. В разделе **Файлы** (см. [рисунок](#)) измените (при необходимости) имя библиотеки защиты.
3. Задайте необходимые параметры в разделах **Параметры пользовательского интерфейса**, **Дополнительные параметры**.
4. В разделе **SDK** (см. [выше](#)):
 - а) включите (при необходимости) настройки пользовательского интерфейса, выставив флажок в одноименном поле;
 - б) определите количество защищаемых классов, которые используются разработчиками защищаемого продукта, работающими со StarForce SDK (в поле **Количество защищаемых классов**); можно определить не больше 4 классов, при превышении этого значения поле ввода выделяется красной рамкой;
 - с) задайте папку, в которую будут загружены файлы SDK (в поле **Папка с**

выходными данными); также есть возможность задать удаление содержимого папки перед генерацией SDK путем выставления флажка в соответствующем поле, причем содержимое папки удаляется без возможности восстановления.

d) нажмите на кнопку **Генерировать SDK** или клавишу **Enter**.

Замечание. Доступные пользовательские права можно переименовать, щелкнув два раза по выбранному элементу в соответствующем поле и введя новое имя.

Если в проекте обнаружены ошибки, то на экран выводится окно с сообщением об ошибках (см. [Диагностика и устранение ошибок при установке защиты](#)), и сервис прекращает работу.

Если ошибки не обнаружены, то начинается работа сервиса.

В процессе работы сервиса информация о производимых операциях выводится во вкладке **Генерация SDK**, а ход процесса отображается индикаторами выполнения.

После завершения процесса в окне появляется соответствующее сообщение.

При нажатии на кнопку **Открыть папку** открывается папка, заданная в поле **Папка с выходными данными** и содержащая файлы, подлежащие интеграции в защищаемое приложение.

Следует помнить, что перед новым запуском сервиса **SDK** необходимо закрыть вкладку **Генерация SDK**, и только после этого кнопка **Генерировать SDK** снова станет активной.

3.4.3 Применение секретных классов

Библиотека секретных классов предназначена для более глубокой интеграции защиты и защищаемого приложения. Она обеспечивает защиту внутренних переменных от изменений и подстановки и защиту исходного кода от дизассемблирования и анализа, являясь эффективным методом борьбы с взломщиками. Библиотека секретных классов представляет собой набор C++ классов, являющихся надстройками над встроенными целочисленными типами данных, и использующихся в математических операциях.

Секретные классы – это надстройки, заменяющие встроенные типы данных, при этом значения соответствующих переменных шифруются и хранятся внутри секретного класса:

```
class PsUInt1
{
private:
    unsigned int var1;
};
```

В настоящее время секретные классы поддерживаются только для типа данных unsigned int.

Число секретных классов определяется разработчиком проекта защиты, а их объявления помещаются в файле PsaApi.h (PscApi.h) в процессе генерации StarForce SDK.

Использование секретных классов вместо обычных целочисленных типов вызывает дополнительные обращения к ядру защиты, что существенно усложняет анализ кода.

Внимание! Использование секретных классов в критичных по времени выполнения участках кода может серьезно замедлить работу приложения, так как вызовы секретных классов происходят относительно медленно (до десятков миллисекунд на операцию).

Для того чтобы встроить секретные классы в приложение до генерации StarForce SDK, задайте необходимое количество секретных классов в одноименном поле раздела **SDK** (см. [рисунок](#)). По умолчанию это число равно 0.

После этого можно заменять объявления переменных типа unsigned int на переменные с типами секретных классов.

Пример:

До замены

```
#include <stdio.h>
unsigned int MyFunction(unsigned int i, unsigned int j )
{
    return i * i + j * j + i * j;
}
```

После замены

```
#include <stdio.h>
#include <PsaApi.h>
unsigned PsUInt1 MyFunction(PsUInt1 i, PsUInt1 j )
{
    return i * i + j * j + i * j;
}
```

Замена встроенных типов данных на секретные классы не изменяет способов обращения к функции, например, вызов формата MyFunction(5, 3) будет легален. Все необходимые трансформации будут автоматически вставлены при компиляции приложения.

Внимание! При компиляции приложения, использующего секретные классы, необходимо включить подстановку inline-функций, иначе защита с использованием секретных классов не будет иметь эффекта. Однако поскольку при защите loopback-функций подстановка inline должна быть отключена, следует быть внимательным при расстановке в исходном коде программы директивы компилятора #inline.

3.4.4 Функции API защиты

PSA_CheckProtectedModulesReadOnlyMem

Функция для контроля целостности кода и данных.

PSA_FsOpenFile, PSA_FsCloseFile, PSA_FsGetFileSize,
PSA_FsSetFilePosition, PSA_FsReadFile

Функции для чтения файла данных, если в контейнер помещён сам файл (в режиме SDK функции работают с файлами на диске, а не в контейнере).

PSA_FsVerifyFileSignature

Функция для проверки целостности цифровой подписи файла данных, если в контейнер помещена цифровая подпись файла. В режиме SDK функция всегда возвращает PSC_STATUS_SUCCESS.

```
PSA_CryptedTrafficOpen, PSA_CryptedTrafficEncrypt,
PSA_CryptedTrafficDecrypt, PSA_CryptedTrafficClose
```

Функции для шифрования трафика на стороне клиента. В составе SDK также будут присланы модули для реализации шифрования трафика на сервере (модули написаны на Java).

```
void __stdcall PSA_DummyFunction();
```

Имеет нулевой функционал и предназначена для предотвращения оптимизации при компоновке и удаления ссылки на protect.dll, если других функций API защиты в данный момент не вызывается.

```
unsigned __int32 __stdcall PSA_IsDemoMode ( bool *isDemoMode);
```

Проверяет факт запуска приложения в демо-режиме.

При использовании режима упрощенной инициализации (см. ниже **Моделирование различных сценариев поведения защищенного программного продукта с помощью SDK**) значение берется из системного реестра, ключ IsDemoMode, тип ключа – REG_DWORD.

```
unsigned __int32 __stdcall PSA_GetFeaturesGrantedByLicense
( unsigned __int32 *features );
```

Определяет набор опций, разрешенных для текущей лицензии, в виде битовой маски (5 бит, [PSA_GrantedFeature0 ... PSA_GrantedFeature]).

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ FeatureSetGrantedByLicense, тип ключа – REG_DWORD.

```
unsigned __int32 __stdcall PSA_CheckFeaturesGrantedByLicense
( unsigned __int32 features );
```

Отладочная функция - игнорируется при защите.

Предназначена для отладки функций, доступ к которым ограничивается видом лицензии. Сравнивает битовую маску возможностей текущей лицензии с определенными требованиями (т.е. эмулирует проверку, осуществляемую системой защиты при выполнении функции с ограничением доступа в зависимости от вида лицензии), и выдает сообщение об ошибке, если нет хотя бы одного бита, совпадающего в масках возможностей лицензии и требований функции.

```
unsigned __int32 __stdcall PSA_GetLicenseStoragePath ( wchar_t
*pathBuffer, size_t *pathBufferSizeInWideChars, HANDLE
*registryBaseHandle );
```

Определяет путь к лицензиям в системном реестре.

```
unsigned __int32 __stdcall PSA_GetLicenseLifetimeLimit ( unsigned
__int64 *licenseLifetimeLimit );
```

Определяет время действия лицензии в интервалах длительностью 100 нс.

Если время действия не задано или не ограничено, задает значение 0xFFFFFFFFFFFFFFFF.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `LicenseLifeTimeLimit`, тип ключа – `REG_BINARY`.

```
unsigned __int32 __stdcall PSA_GetRemainingNumberOfRuns ( unsigned  
__int32 *remainingNumberOfRuns );
```

Определяет оставшееся количество допустимых запусков программы.

Если допустимое максимальное количество запусков не задано или не ограничено, задает значение 0xFFFFFFFFFFFFFFFF.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `RemainingNumberOfRuns`, тип ключа – `REG_DWORD`.

```
unsigned __int32 __stdcall PSA_GetLicenseNumberOfRunsLimit  
( unsigned __int32 *licenseNumberOfRunsLimit );
```

Определяет допустимое максимальное количество запусков программы.

Если оно не задано или не ограничено, задает значение 0xFFFFFFFFFFFFFFFF.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `LicenseNumberOfRunsLimit`, тип ключа – `REG_DWORD`.

```
unsigned __int32 __stdcall PSA_GetRemainingExecutionTime ( unsigned  
__int64 *remainingExecutionTime );
```

Определяет оставшееся суммарное время исполнения программы в интервалах длительностью 100 нс.

Если лимит суммарного времени исполнения не задан или не ограничен, задает значение 0xFFFFFFFFFFFFFFFF.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `RemainingExecutionTime`, тип ключа – `REG_BINARY`.

```
unsigned __int32 __stdcall PSA_GetRemainingExecutionTimeAtStart  
( unsigned __int64 *remainingExecutionTimeAtStart );
```

Определяет оставшееся суммарное время исполнения программы в интервалах длительностью 100 нс, вычисленное от момента ее старта.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `RemainingExecutionTimeAtStart`, тип ключа – `REG_BINARY`.

```
unsigned __int32 __stdcall PSA_GetLicenseExecutionTimeLimit  
( unsigned __int64 *licenseExecutionTimeLimit );
```

Определяет суммарное время исполнения программы в интервалах длительностью 100 нс.

Если лимит суммарного времени исполнения не задан или не ограничен, задает значение 0xFFFFFFFFFFFFFFFF.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ `LicenseExecutionTimeLimit`, тип ключа –

REG_BINARY.

```
unsigned __int32 __stdcall PSA_IsTrialMode ( bool *isTrialMode );
```

Функция задает значение true параметру isTrialMode, если приложение запущено в пробном режиме.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ IsTrialMode, тип ключа – REG_DWORD.

```
unsigned __int32 __stdcall PSA_GetTimeToLicenseExpiration  
( unsigned __int64 *timeToLicenseExpiration );
```

Функция определяет время до истечения лицензии, в сотнях наносекунд.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ TimeToLicenseExpiration, тип ключа – REG_BINARY (реализуется 8-байтным Binary).

```
unsigned __int32 __stdcall PSA_GetLicenseExpirationDateTime  
( unsigned __int64 *LicenseExpirationDateTime );
```

Функция определяет дату и время истечения лицензии в формате FILETIME (количество интервалов длиной 100 наносекунд, прошедших с начала 1601 года).

Для конвертирования возвращаемого значения в другие форматы можно использовать API Windows (например, FileTimeToSystemTime, FileTimeToLocalTime).

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ LicenseExpirationDateTime, тип ключа - REG_BINARY.

```
unsigned __int32 __stdcall PSA_GetUserDefinedField16Bits ( unsigned  
__int16 *userDefinedField );
```

Функция определяет значение поля.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ UserDefinedField16Bits, тип ключа – REG_DWORD.

```
unsigned __int32 __stdcall PSA_DisableFeaturesGrantedByLicense  
( unsigned __int32 features );
```

Отключает отдельные биты маски прав в процессе работы приложения.

Аргумент влияет только на включенные права. Следует использовать аргумент -1 для отключения всех включенных прав. Отключенные права не могут быть включены до следующего запуска приложения.

```
unsigned __int32 __stdcall PSA_Uninitialize ();
```

Деинициализирует ядро защиты перед выгрузкой protect.dll или перед выходом из программы.

```
unsigned __int32 __stdcall PSA_GetLicenseInformation ( unsigned  
__int32 *version, unsigned __int32 *type, bool *nonCommercial );
```

Функция определяет специальную информацию о текущей лицензии.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключи: LicenseVersion, LicenseType, IsNotCommercialLicense, тип ключей – REG_DWORD.

```
unsigned __int32 __stdcall PSA_IsLicenseExpired ( bool  
*isLicenseExpired, bool updateRunTimeData );
```

Функция проверяет, истек ли срок действия лицензии.

При использовании режима упрощенной инициализации значение берется из системного реестра, ключ IsLicenseExpired, тип ключа – REG_DWORD.

3.4.5 Интеграция SDK в разрабатываемое приложение

1. Включите PsaApi.h (PscApi.h) в исходные тексты. PsConfig.h включается из PsaApi.h (PscApi.h).
2. В настройках редактора связей компоновщика добавьте к входным файлам компоновщика protect.lib.
3. Скопируйте protect.dll и protect.exe в каталог, из которого будет запускаться защищаемое приложение.
4. Добавьте вызов необходимых функций API защиты в защищаемое приложение.
5. Скомпилируйте защищаемое приложение.

Примечания

1. Файлы с именем protect на самом деле будут иметь имя, заданное в настройках проекта.
2. Если в разрабатываемом приложении не используется ни одна из API-функций защиты, компоновщик произведет оптимизацию и не подключит protect.dll к приложению. Это часто является нежелательным, т.к. код защиты, позволяющий моделировать ее поведение, исполняется в DllMain protect.dll. Решением этой проблемы является использование специального метода, не имеющего функционала – функции void PSA_DummyFunction(). Ее вызов можно встроить в любом месте приложения для исключения оптимизации при компоновке в случае отсутствия в приложении вызовов API-функций защиты.
3. Для версий Visual C++ ранее 2005 может возникнуть проблема несовместимости файла protect.lib с проектом. В этом случае необходимо создать собственный файл protect.lib. Самый простой способ сделать это – создать собственную DLL, которая имела бы пустые функции с именами, совпадающими с именами функций StarForce API. Создание protect.lib путём написания DEF-файла с последующим использованием утилиты LIB.EXE не рекомендуется, так как это может привести к проблемам с декорированием имён.
4. Для интеграции SDK с приложением на Delphi скопируйте PsaApi.pas, PscApi.pas, PsConfig.pas, PsConstants.pas в директорию с проектом, а в проекте в разделе **uses** укажите PsaApi и PscApi. Если библиотека защиты называется не protect.dll, в файлах PsaApi.pas и PscApi.pas все вхождения строки ‘protect.dll’ замените на действительное имя библиотеки защиты.

Моделирование различных сценариев поведения защищенного программного продукта с помощью SDK

По умолчанию при запуске приложения, в которое встроена protect.dll, внутри основной функции этой библиотеки производится полноценная инициализация защиты с проверкой диска, активацией и т.п.

Поскольку при интенсивной отладке приложения это может быть неудобно, предусмотрен *режим упрощенной инициализации* protect.dll. Для этого в разделе системного реестра, указанном в разделе **Базовые параметры** в полях **Настройка путей ключа в реестре**, необходимо создать ключ Debug со значением QuickInitializationMode типа REG_DWORD. При QuickInitializationMode=0 производится полная инициализация защиты, а при QuickInitializationMode=1 включается режим упрощенной инициализации.

Если режим упрощенной инициализации включен, все API-функции защиты, которые должны возвращать какие-либо результаты, берут эти значения из реестра (из ключа Debug). Расшифровка соответствующих значений дана в справке по каждой конкретной API-функции.

Защита приложения

Защита приложения с помощью SDK производится как обычно, однако имя отладочной версии библиотеки защиты, указанной при генерации SDK (protect.dll по умолчанию), не должно изменяться. При необходимости изменить имя сгенерируйте SDK повторно, указав новое имя для библиотеки защиты, и произведите соответствующую модификацию защищенного приложения.

Ограничение функциональности защищенного программного продукта в зависимости от лицензии

Система StarForce предоставляет возможность ограничивать функциональность защищенного программного обеспечения в зависимости от вида лицензии на основе независимо отключаемых возможностей. Эта возможность реализована через дополнительное поле в активационном ключе, содержащее описание функциональности, предоставляемой данным продуктом. Это поле называется кодом прав. Каждый бит кода прав соотносится с набором функций защищенного приложения, который можно включить или выключить. Содержание кода прав для текущей лицензии можно получить с помощью API-функции защиты PSA_GetFeaturesGrantedByLicense. Данная функция является основным методом поддержки ограничения функционала защищаемого приложения.

В PsApi.h заданы константы для обозначения независимых битов, соответствующие по именам обозначениям, используемым при защите и генерации лицензий:

```
#define PSA_GrantedFeature0 0x00000001
#define PSA_GrantedFeature1 0x00000002
...
#define PSA_GrantedFeature31 0x80000000
```

Примечание. В проекте защиты поддерживается только нижние пять битов прав, соответствующих значениям констант 0x1 – 0x10.

В случае необходимости, можно дать константам более осмысленные имена. Так, в

программе можно определять:

```
#define PrintFeatureGranted      PSA_ GrantedFeature2
...
unsigned __int32 features;
if( PSA_GetFeaturesGrantedByLicense( &features ) != PSC_STATUS_SUCCESS )
{
    // Process Error
}
if( features & PrintFeatureGranted)
{
    PrintDocument();
}
```

Само по себе использование API не предоставляет защиты от взлома (если только сами вызовы API не размещены в защищенных функциях, что не всегда реализуемо). Для улучшения качества защиты от взлома можно задавать зависимость защищенных функций от набора прав, имеющихся в данной лицензии (при защите функции указывается набор битов; в таком случае для нормальной работы функции требуется, чтобы в текущей лицензии был выставлен хотя бы один из них). Если набор прав в лицензии не удовлетворяет требованиям защищенной функции, она не выполняется и возвращает 0 в вызывающую функцию. При этом рекомендуется писать программы таким образом, чтобы защищенные функции вызывались только в случае, когда их вызов разрешен текущей лицензией, например:

```
#define PrintFeatureGranted      PSA_ GrantedFeature2
...
unsigned __int32 features;
if( PSA_GetFeaturesGrantedByLicense( &features ) != PSC_STATUS_SUCCESS )
{
    // Processes Error
}
if( features & PrintFeatureGranted )
{
    SFLB_PrintDocument(); // Защ. функция с требованием PSA_
GrantedFeature2
}
```

Для проверки того, что защищенные функции всегда вызываются в нужном месте, в отладочной версии API защиты имеется возможность сделать автоматические проверки при некорректном вызове защищенных функций. Для этого в начале функции, которую планируется защищать, следует сделать проверку прав:

```
void SFLB_PrintDocument()
{
    PSA_CheckFeaturesGrantedByLicense( PrintRightGranted );
    // Код функции
}
```

Функция `PSA_CheckFeaturesGrantedByLicense` проверяет текущие права, сравнивая аргумент и результат вызова `PSA_GetFeaturesGrantedByLicense`, и в случае несовпадения выдает сообщение об ошибке:

```

unsigned __int32 __stdcall PSA_CheckFeaturesGrantedByLicense(
    unsigned __int32 features )
{
    uint32 currentFeatures;
    uint32 result = PSA_GetFeaturesGrantedByLicense( &currentFeatures );
    if( result != PSC_STATUS_SUCCESS )
    {
        return result;
    }
    if( ( features & currentFeatures ) == 0 )
    {
        MessageBox( HWND_DESKTOP, "Error", "Invalid rights", MB_OK );
    }
    return PSC_STATUS_SUCCESS;
}

```

Вместо вызова `PSA_CheckFeaturesGrantedByLicense` можно написать пользовательскую функцию проверки (которая, например, вместо выдачи сообщения писала бы результат проверки в файл). При этом необходимо помнить, что при использовании в процессе отладки `PSA_CheckFeaturesGrantedByLicense` ее вызов будет автоматически удален в защищенной версии, в случае же использования пользовательской функции ее вызов необходимо удалять вручную.

Для изменения функционала в процессе работы приложения можно использовать функцию `PSA_DisableFeaturesGrantedByLicense (int features)`;

Данная функция позволяет отключать отдельные биты маски прав во время работы приложения. Отключенные права не могут быть включены до следующего запуска приложения. Аргумент функции влияет только на разрешенные права. Чтобы отключить все разрешенные права, передайте -1 в качестве аргумента. Эта функция использует `PSA_GetTimeToLicenseExpiration` для выяснения периода до истечения лицензии. По истечении этого времени функция не завершает работу приложения, а отключает права в соответствии с истекшей лицензией.

Внимание! В SDK не реализован стандартный способ обработки деления на ноль (вывод соответствующего исключения). В обычном случае в программах не используется деление на 0. Тем не менее, поскольку в SDK нет проверки деления на ноль, эта проверка должна осуществляться в коде функции, если в данном месте возможно деление на 0. Если по тем или иным причинам Вам необходимо, чтобы было выдано соответствующее исключение, его также нужно создавать в коде функции.

3.4.6 Пример использования SDK

Для иллюстрации возможностей SDK приведен пример использования StarForce API для контроля целостности неизменяемых частей защищённого модуля в памяти и неизменяемых файлов.

```

//=====
//SelfCheckSample.cpp
//-----

```



```
//Description:

// To use this test, compile it with the LIB file from StarForce SDK,
// add a SelfTestSample.txt file to a container, select 'Use digital
// signature instead of file' command, and perform protection
//=====
#include <stdio.h>
#include <windows.h>
#include "PsaApi.h"

// Entry point
void main()
{
    unsigned __int32 status;
    bool checkResultOk;

    // Check memory (1st time)
    printf( "Performing self-check of read-only memory...\n" );
    status = PSA_CheckProtectedModulesReadOnlyMem( &checkResultOk );
    if( status != PSC_STATUS_SUCCESS )
    {
        printf( "PSA_CheckProtectedModulesReadOnlyMem failed with
error code 0x%08X\n. Terminating application." );
        return;
    }
    if( checkResultOk )
    {
        printf( "Done. Check passed. Read-only memory is valid.\n" );
    }
    else
    {
        printf( "Done. Check failed. Read-only memory is invalid.
Terminating application.\n" );
        return;
    }
    printf( "\n" );

    // Optional modification of memory
    char option[ 1 ];
    printf( "Do you want to modify read-only memory and check again (y/
n)?" );
    scanf( "%1s", option );
    if( option[ 0 ] == 'y' || option[ 0 ] == 'Y' )
    {
        printf( "Modifying read-only memory...\n" );
        char *readOnlyData = "Read only data";
        DWORD oldProtect;
        if( !VirtualProtect( readOnlyData, sizeof( char ),
PAGE_EXECUTE_READWRITE, &oldProtect ) )
        {
            printf( "VirtualProtect failed with error code %d\n",
GetLastError() );
            return;
        }
        readOnlyData[ 0 ] = 0;
        printf( "Done.\n" );
    }
}
```

```

        // Check memory (2nd time)
        printf( "Performing self-check of read-only memory...\n" );
        status = PSA_CheckProtectedModulesReadOnlyMem
( &checkResultOk );
        if( status != PSC_STATUS_SUCCESS )
        {
            printf( "PSA_CheckProtectedModulesReadOnlyMem failed
with error code 0x%08X\n. Terminating application." );
            return;
        }
        if( checkResultOk )
        {
            printf( "Done. Check passed. Read-only memory is valid.
\n" );
        }
        else
        {
            printf( "Done. Check failed. Read-only memory is
invalid. Terminating application.\n" );
            return;
        }
    }
    else
    {
        printf( "Modification skipped.\n" );
    }
    printf( "\n" );

    // Reading data from file
    printf( "Performing self-check of read-only files...\n" );
    FILE *f;
    if( fopen_s( &f, "SelfCheckSample.txt", "rt" ) != 0 )
    {
        printf( "Unable to open file SelfCheckSample.txt\n" );
        return;
    }
    char currentChar;
    printf( "The content of file SelfCheckSample is:\n" );
    while( ( currentChar = fgetc( f ) ) != EOF )
    {
        printf( "%c", currentChar );
    }
    printf( "\n" );
    int isValid;
    status = PSA_FsVerifyFileSignature( L"SelfCheckSample.txt", &isValid
);
    if( status != PSC_STATUS_SUCCESS )
    {
        printf( "PSA_FsVerifyFileSignature failed with error code 0x%
08X\n. Terminating application." );
        return;
    }
    if( isValid )
    {
        printf( "The file SelfCheckSample.txt was not modified.\n" );
    }

```

```
    }  
    else  
    {  
        printf( "The file SelfCheckSample.txt was modified.\n" );  
    }  
}
```

3.5 Дополнительные возможности защиты

3.5.1 Настройка GUI

Внимание! Сообщения **GUI** появляются на компьютере конечного пользователя в одном из нижеперечисленных случаев:

1. При установке драйвера;
2. При возникновении ошибок.

3.5.1.1 Назначение модифицируемого GUI

Модифицируемый GUI позволяет менять в широких пределах интерфейс и логику поведения защищённого приложения при проверке, активации и т.п. Вот некоторые типовые задачи, которые пользователь может решить, применяя модификацию GUI:

- Модификация внешнего вида окон, в том числе в соответствии с дизайном своего приложения (актуально для игр); использование собственной библиотеки для отображения диалоговых окон.
- Добавление дополнительных окон с разъяснениями и предупреждениями.
- Изменение процедуры активации (добавление дополнительных полей, перенаправление на свой сайт, запрет ручной активации и т.п.)
- Изменение процедуры обращения в техподдержку, изменение формата отчёта об ошибках.
- Изменение логики работы приложения при использовании пробного периода (в т.ч. при его истечении).

3.5.1.2 Рекомендации по написанию индивидуальных GUI-проектов

Для написания индивидуального GUI-проекта:

1. В разделе **SDK** выставите флажок в поле «Включить настройки пользовательского интерфейса».
2. Проведите защиту приложения.
3. Сгенерируйте SDK. В составе SDK в папке GUI будет находиться файл `UserInterfaceCustomizable.vcproj`.
4. Откройте с помощью VisualStudio файл `UserInterfaceCustomizable.vcproj`. При открытии файла откажитесь от использования SourceControl-системы. Особое внимание следует уделять файлу `Config.h`: при редактировании проекта и повторных защитах он может меняться, поэтому в этих случаях необходимо повторно генерировать SDK и заменять `Config.h` в своём проекте на новый.
5. Скомпилируйте проект и сохраните его. В результате компиляции получится файл

protect.gui.

6. Скопируйте protect.gui в папку, где расположена protect.dll, и запустите защищённое приложение. Убедитесь в его работоспособности.
7. Внесите модификации в проект и повторите шаги 5 и 6.

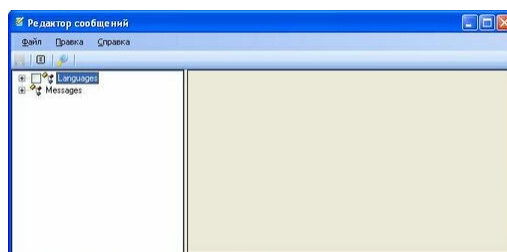
3.5.2 Редактор сообщений

Внимание! Сообщения GUI появляются на компьютере конечного пользователя в одном из нижеперечисленных случаев:

1. При установке драйвера;
2. При возникновении ошибок.

3.5.2.1 Интерфейс приложения

Внешний вид редактора показан на [рисунке ниже](#).



Редактор сообщений

Интерфейс редактора состоит из следующих элементов: главное меню, панель инструментов, дерево сообщений/языков и соответствующее ему контекстное меню (только для языков); рабочее поле в правой панели – для отображения информации о языках и сообщениях.

3.5.2.2 Команды меню

Меню Файл

Команда	Назначение команды
Сохранить	Сохранение изменений
Выход	Закрытие редактора сообщений

Редактор сообщений. Меню Файл

Меню Правка

Команда	Назначение команды
Найти	Поиск по сообщениям
Добавить язык	Добавление языка




Редактор сообщений. Меню Правка

Меню Справка

Команда	Назначение команды
О программе	Вывод на экран информации о программе

Редактор сообщений. Меню Справка

3.5.2.3 Кнопки панели команд

Пиктограмма	Назначение
	Сохранение изменений
	Добавление языка
	Поиск по сообщениям

Кнопки панели команд

3.5.2.4 Использование быстрых клавиш

Комбинация клавиш	Назначение
Ctrl+S	Сохранение изменений
Ctrl+F	Поиск по сообщениям
Ctrl+Z	Отмена; только для полей ввода

Редактор сообщений. Быстрые клавиши

3.5.2.5 Дерево сообщений и языков

Структура

В дереве сообщений и языков поддерживается пять типов узлов (см. [рисунок](#)):



1. **Languages**- список языков;
2. **Language** – язык;
3. **Messages** - структура сообщений и групп сообщений;
4. **Message** – сообщение;
5. Группа сообщений.



Дерево сообщений и языков


Цветовое выделение

Для иконок дерева приняты следующие принципы цветового выделения:

1. После названия языка в скобках указывается количество сообщений, отмеченных флажком **Todo**, то есть сообщений, которые либо не переведены на данный язык, либо их перевод устарел по смыслу. Если количество равно 0, то иконка рядом с названием серая , если же есть непереведенные сообщения, то красная , что является предупреждением о том, что работа не закончена.
2. Тот же принцип используется для сообщений: если сообщение переведено на все языки, то иконка серая, иначе красная.

3.5.2.6 Функции редактора

Сохранение изменений

Сохранение производится командой **Сохранить** (или при нажатии на кнопку ) . После закрытия редактора сообщений необходимо сохранить файл проекта. Таким образом, все изменения сообщений сохраняются в файле проекта.


Загрузка сохраненных сообщений

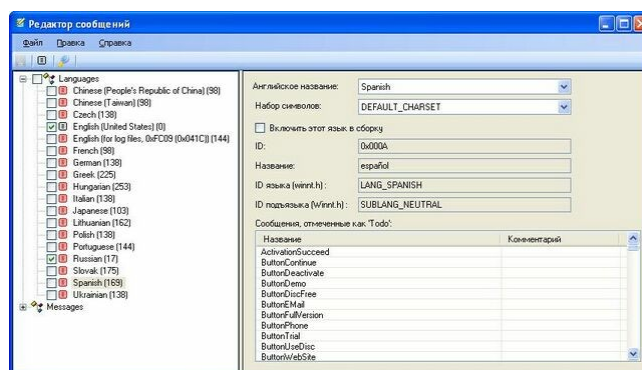
Загрузка сообщений производится автоматически из файла проекта после запуска редактора.

Выход из программы

Если одно из полей содержит ошибку, то при выходе будет предложено выйти без сохранения или продолжить редактирование, так как сохранение документов с ошибками не допускается.

Добавление, редактирование, удаление языков

Чтобы добавить новый язык, нажмите на кнопку  или в контекстном меню или в меню **Правка** выполните команду **Добавить язык**. В конце дерева языков появится новый язык, в правой части окна – поля для ввода информации о языке (см. [ниже](#)).



Параметры языка

При добавлении языка автоматически добавляется первый неиспользованный язык из полного списка. После этого в выпадающем списке в поле **Английское название** можно выбрать требуемый язык. При изменении языка не разрешено дублирование.

В поле **Char set** рекомендуется использовать установленное по умолчанию значение

DEFAULT_CHARSET.

Для включения выбранного языка в MSG-файл с сообщениями поставьте флажок в поле **Включить этот язык в сборку**.

В полях **ID**, **Название**, **ID языка** и **ID подязыка** указана справочная информация по выбранному языку.

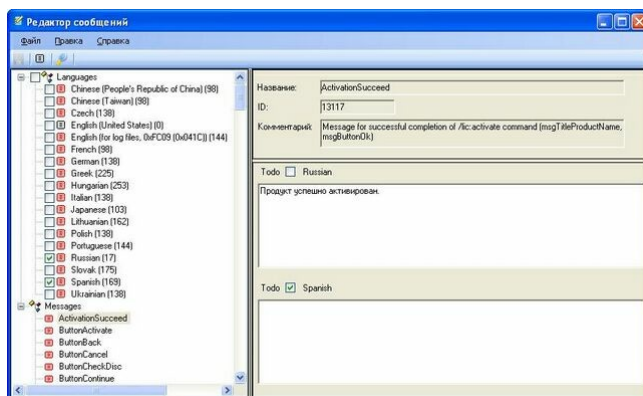
В таблице **Сообщения**, **отмеченные как 'Todo'** перечислены названия сообщений, подлежащие переводу на данный язык, и комментарии к ним. Для перехода к редактированию сообщения из списка достаточно двойного нажатия на него в списке.

Редактирование сообщений

Чтобы отредактировать текст сообщения:


1. Отметьте его в дереве в области узла **Messages**;
2. Поставьте флажок около нужных языков в дереве, после чего в рабочем поле сообщения появятся поля для редактирования текста сообщения на выбранных языках (см. [рисунок](#)).

После окончания редактирования текста снимите флажок в поле **Todo**: отсутствие флажка означает, что при выборе данного языка в окне защиты данное сообщение будет отображаться на этом языке, а не на языке по умолчанию (английском).



Редактирование сообщения

Поиск

Окно поиска появляется при нажатии на кнопку  или выполнении команды **Найти** в меню **Правка**. Окно поиска также можно вызвать сочетанием клавиш **Ctrl+F**.

Поиск проводится только по сообщениям, а именно:

- в названиях сообщений,
- в ID сообщений,
- в комментариях к сообщениям,
- в тексте сообщений.

Отображение результатов поиска:

1. В строке выделяется только первое вхождение, поиск последующих вхождений в одной строке не проводится.

2. Если язык, в переводе на который найдено вхождение, скрыт, то в дереве у этого языка автоматически ставится флажок, и язык отображается. При этом те языки, которые были включены ранее, не скрываются, даже если искомый текст в них не найден.
3. Одновременно выделяются найденные вхождения во всех строках.

Последовательность работы с функцией поиска:

1. При первом открытии окна поиска кнопка **Найти сначала** недоступна, она становится активной только после ввода текста в поле **Найти**.
2. Если текст найден, надпись на кнопке меняется на **Найти Далее**. При нажатии на эту кнопку поиск начинается со следующего сообщения в дереве (с учетом вложенности).
3. Если текст не найден, появляется соответствующее сообщение. Поиск сначала не проводится.
4. При выделении произвольного узла в дереве поиск будет начинаться с указанного узла (**Найти сначала**) или со следующего (**Найти Далее**).
5. Если текст был найден, но значение поля **Найти** было изменено, надпись на кнопке **Найти Далее** меняется на **Найти сначала**, и поиск начинается с выделенного узла дерева.

Внимание! Окно поиска не сохраняет запрос после закрытия.

6. Может быть открыто только одно окно поиска. При открытии окна поиска кнопка **Поиск** на панели инструментов становится неактивной, сочетание клавиш **Ctrl+F** также не работает. При закрытии окна поиска кнопка вновь становится доступной.

4 Подготовка к выпуску и выпуск продукта

4.1 Изменение настроек системы защиты

Ряд опций, заданных в файле проекта (т.е. при выполнении процесса защиты), можно переопределять путем записи их новых значений в системный реестр на компьютере конечного пользователя. В этом случае система защиты будет брать значения из реестра, и только если там их нет, будут использоваться значения, заданные в процессе защиты. Запись значений в системный реестр обычно производится на стадии инсталляции продукта на компьютере конечного пользователя при помощи соответствующих операций, включенных в инсталляционные скрипты.

Для каждого защищенного приложения в системном реестре на компьютере конечного пользователя создается раздел с полным именем, заданным в Protection Studio в разделе **Базовые параметры** (см. [Раздел "Базовые параметры"](#)).

По умолчанию полное имя раздела: HKEY_CURRENT_USER/SOFTWARE/<имя компании>/<имя проекта>/Keys (в имени раздела могут использоваться только следующие символы: латинский алфавит, цифры, \ () { } [] . : подчёркивание, пробел, дефис). В этом разделе (например, в процессе инсталляции приложения) может создаваться подраздел Settings, где может переопределяться ряд параметров защиты:

1. Параметры с данными текстового типа переопределяются величинами с типом REG_SZ в соответствии с нижеприведенной таблицей:

Раздел Protection Studio	Наименование параметра в Protection Studio	Ключ в реестре	Имя величины
Параметры пользовательского интерфейса	Поле «Сообщения защиты»: Служба поддержки/Сайт	Protection\Gui\Support	Page
Параметры пользовательского интерфейса	Поле «Сообщения защиты»: Служба поддержки/E-mail	Protection\Gui\Support	Email
Параметры пользовательского интерфейса	Поле «Сообщения защиты»: Служба поддержки/Телефон	Protection\Gui\Support	Phone
Проект	Название компании	Protection\Gui\About	Company
Проект	Название продукта	Protection\Gui\About	Product

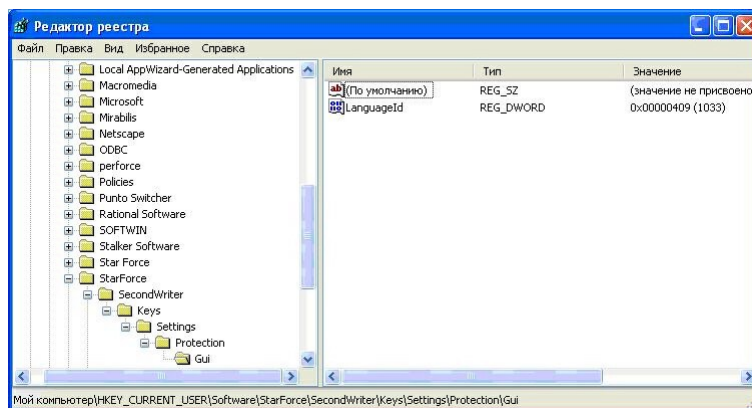
Параметры текстового типа

2. Для переопределения языка интерфейса задается ключ Protection\Gui с величиной LanguageId типа REG_DWORD.

Варианты идентификаторов языка соответствуют значениям стандартного типа LANGID системы Windows. Существующие варианты языка интерфейса указаны в разделе **Параметры пользовательского интерфейса** в качестве значений опции **Опции GUI|Язык** (см. [рисунок](#)). Если указанный язык не поддерживается, выбор языка производится согласно установкам в разделе **Язык и региональные стандарты ОС**.

Точный ID языка можно узнать в **Редакторе сообщений** (см. [рисунок](#), поле ID).

Пример записи в реестре ключа для английского языка интерфейса показан на [рисунке ниже](#) (редактор реестра можно вызвать из меню: **Пуск/Start|Выполнить/Run|regedit**).



Запись в реестре ключа, определяющего язык интерфейса

4.2 Дополнительная возможность инсталляции защищенного продукта

В процессе работы сервиса **Защита файлов** (см. [Защита файлов](#)) создается исполняемый файл интерфейса защиты `protect.exe` (или файл с тем именем, на которое "protect" было изменено в настройках файла проекта). Начало работы с этим файлом может осуществляться одним из двух способов:

1. Непосредственный вызов файла защиты. После этого появляется окно "Protection System" с перечнем действий, которые может выполнить этот файл. Наиболее полный вариант представлен на рисунке ниже.



Диалоговое окно файла защиты `protect.exe`

Если в проекте не используется драйвер, то группы **Driver management commands** в окне не будет.

2. Запуск файла защиты из командной строки с одним из ключей.

Это делает его весьма полезным при решении задач инсталляции/деинсталляции защищенного продукта на компьютере конечного пользователя.

Ниже приводится перечень ключей, которые могут быть использованы для запуска `protect.exe` в сценариях инсталляции/деинсталляции, и их соответствие командам, представленным в диалоговом окне.

Ключ	Команда	Назначение
<code>/drv: check [/nogui]</code>	Check driver status	Производит проверку, установлен ли драйвер защиты. Код возврата – 0, если драйвер не установлен, в противном случае не 0.
–	Download and install driver update	Загружает с сайта обновление и обновляет драйвер.
<code>/drv: install [/nogui] [/forcereboot]</code>	Install driver	Устанавливает драйвер защиты.
<code>/drv: uninstall [/nogui]</code>	Remove driver if it is not used by other installed applications	Деинсталлирует драйвер защиты в том случае, если он не используется другими установленными защищенными приложениями. Если драйвер используется другими защищенными приложениями, то он остается в системе.

Ключ	Команда	Назначение
/drv: remove [/nogui]	Remove driver completely	Деинсталлирует драйвер защиты независимо от наличия в системе приложений, использующих его.
/? или /help	–	Показывает окно со справкой по используемым ключам.
/eventlog: register	Register event log	Создание раздела в реестре для записи Event Log защищенного приложения-сервиса.

Команды и ключи запуска файла защиты protect.exe

Помимо ключей основных операций в командной строке могут также использоваться два дополнительных ключа:

Ключ	Назначение
/nogui	Если этот ключ не задан, сообщения исполняемого файла защиты выводятся в стандартном интерфейсе защиты. В противном случае вся информация содержится только в коде возврата.
/forcereboot	Задаёт принудительную перезагрузку компьютера после установки.

Дополнительные ключи запуска protect.exe

Значения кодов выхода и кодов ошибок показаны в следующей таблице:

Тип ошибки	Код возврата	Описание
DRVMAN_SUCCESS	0	Инсталляция/деинсталляция драйвера прошла успешно.
DRVMAN_ERROR	1	Во время инсталляции/деинсталляции драйвера возникла непредвиденная ошибка – для получения более подробной информации об ошибке необходимо вызвать GetLastError().
DRVMAN_NEED_ADMIN	3	Ошибка. Инсталляция/деинсталляция драйвера требует наличия прав администратора для данного компьютера.
DRVMAN_USAGE_CONFLICT	4	Ошибка. Для инсталляции/деинсталляции нужно закрыть все активные приложения и вызвать функцию еще раз.
DRVMAN_DEBUGGER_DETECTED	5	Ошибка. Обнаружен отладчик, продолжение инсталляции невозможно.
DRVMAN_INCOMPATIBLE_OS	6	Ошибка. Система защиты несовместима с данной операционной системой.
DRVMAN_SAFE_MODE	7	Ошибка. Windows загружена в режиме Safe Mode. Продолжение инсталляции невозможно.
DRVMAN_COMPATIBILITY_MODE	8	Ошибка. Попытка запустить приложение в режиме совместимости (Compatibility mode).

Типы ошибок и коды возврата

5 Сопровождение продукта

5.1 Драйвер защиты

Для каждого продукта, защищаемого с применением драйвера, в процессе защиты создается собственный драйвер. Цифровая подпись драйвера производится сборщиком защиты, однако пользователь (издатель) может переподписать их своей цифровой подписью.

Решение о применении драйвера защиты принимается пользователем на этапе согласования проекта защиты со службой технической поддержки StarForce и не может быть изменено пользователем без обращения в службу поддержки.

Возможность использования драйвера защиты устанавливается на этапе создания рабочего пространства – проекта защиты, и отображается в поле **Драйвер** в разделе **Базовые параметры** (см. [Раздел "Базовые параметры"](#)). Это поле может иметь следующие значения:

Значения поля Драйвер	Доступные опции защиты
Включено	Антиэмуляция включена. Также можно использовать защиту файлов данных (создание контейнеров).
Отключено	Защита файлов данных в данном проекте не поддерживаются.

Поле "Драйвер"

5.1.1 Установка драйвера защиты на компьютере конечного пользователя

Если установка драйвера защиты не предусмотрена в инсталляционном сценарии, то при первом запуске приложения пользователю предлагается подтвердить установку драйвера. С появлением этого предложения связаны две опции GUI: **Вопрос об инсталляции драйвера** и **Информация в окне вопроса об инсталляции драйвера** (см. [Раздел "Параметры пользовательского интерфейса"](#)).

Чтобы установка драйвера защиты производилась в процессе инсталляции защищенного программного продукта, следует использовать исполняемый файл интерфейса защиты (см. [Дополнительная возможность инсталляции защищенного продукта](#)). При запуске его в виде отдельного процесса с различными ключами (например, из инсталляционного скрипта) можно выполнять функции установки и удаления драйвера защиты.

Для удобства конечных пользователей и в целях повышения совместимости защищенных продуктов, рекомендуется предоставление конечным пользователям следующих материалов (или ссылок на них в сети Интернет):

http://www.star-force.ru/faq/for_users.php

По этой ссылке размещена более подробная информация о драйвере защиты StarForce, а также представлены разработанные специалистами компании утилиты для обновления и удаления драйвера защиты вручную и информация по их применению.

5.2 Обновление защищенного продукта. Защита обновленных версий

Обновлениями могут быть любые файлы, которые либо должны заменить файлы предыдущей версии продукта, либо быть добавленными в предыдущую версию.

Для защиты обновлений выполните следующие действия:

1. Загрузите файл проекта защиты, использовавшийся для защиты той версии продукта, для которой предназначено обновление.
2. В разделе **Файлы** добавьте новые исполняемые файлы, замените старые версии файлов новыми и измените при необходимости пути неизменившихся файлов. Если у обновленных исполняемых файлов сохранились прежние пути, то при защите автоматически будут использоваться эти обновленные файлы.
3. Если необходимо, настройте защиту функций модулей и другие настройки защиты.
4. Создайте контейнер(ы) с именем, отличным от тех, которые уже были использованы для контейнеров данного продукта, и добавьте в него (них) только новые и изменившиеся файлы данных. В окне «Менеджер контейнеров» (см. [рисунок](#)) уберите флажок около названия старых контейнеров.
5. Выполните защиту файлов (запустите сервис **Защита файлов**).

Внимание! Для контейнеров, уже содержащихся в открытом проекте (т.е. тех, которые были защищены в предыдущей версии) никаких модификаций настроек защиты производить **нельзя**.

После окончания работы сервиса защиты будут получены следующие (обновленные) файлы:

- модули защиты (по умолчанию PROTECT.DLL, PROTECT.EXE);
- файл PROTECT.X86/X64;
- защищенные исполняемые файлы программного продукта, которые используют обновленные данные;
- старые и новые контейнеры.

Для того чтобы приложение, установленное на компьютере конечного пользователя, использовало обновленные данные, поместите новые контейнеры в соответствующую папку приложения (при этом старые контейнеры должны остаться на прежнем месте; в противном случае приложении не запустится). При наличии нескольких контейнеров для защищенного программного продукта более высоким приоритетом будет обладать созданный позже. Если один и тот же файл данных имеется в нескольких контейнерах, то его чтение будет производиться только из контейнера с более высоким приоритетом.

Новый контейнер данных может свободно распространяться, например, через сайт разработчика программного продукта (при этом старые контейнеры распространять не требуется).

6 Диагностика. Устранение ошибок

6.1 Диагностика и устранение ошибок при установке защиты

Все действия по защите отображаются на вкладке **Журнал**.

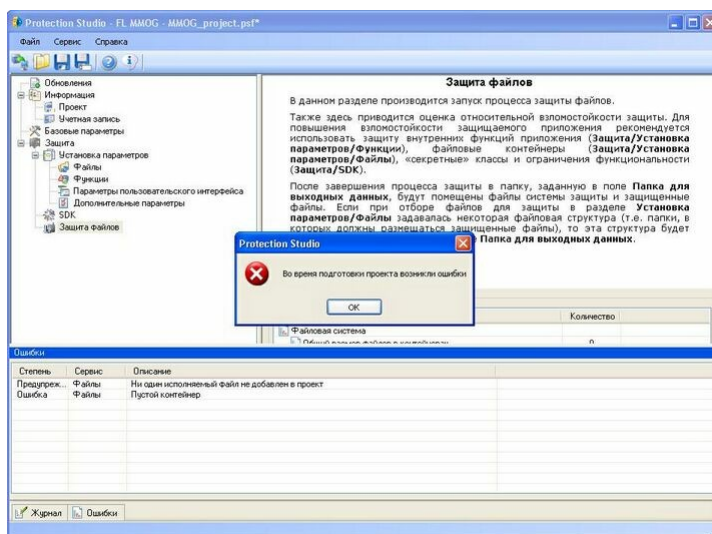
В отдельных случаях ошибочных действий при установке параметров защиты Protection Studio сразу сигнализирует об этом. Так, например, при неверном задании имени папки, из которой должен браться файл для защиты, имя файла и название папки будут записаны в рабочем поле раздела **Файлы** красным цветом, что означает ошибку. Однако не для всех данных проверка правильности производится непосредственно при вводе.

При запуске сервиса сначала обязательно идет проверка заданных для него параметров. Если все верно, сервис начинает работу, а в нижней части окна PS появляется закладка с именем сервиса, содержащая информацию о ходе его выполнения. Кроме того, ход работы сервиса отображается на вкладке **Журнал**.

Ошибки, возникающие при исполнении сервиса, в зависимости от типа выдаются и описываются либо на вкладке сервиса, либо на вкладке **Ошибки**. Вкладка **Ошибки** содержит только ошибки задания параметров в Protection Studio перед запуском сервиса. По степени серьезности они делятся на *Предупреждения* и собственно *Ошибки*. *Предупреждение* не препятствует запуску и работе сервиса, тогда как с *Ошибкой* его выполнение невозможно.

Если при обращении к сервису обнаружены ошибки в настройках, то сервис не начинает работу, на экране появляется информационное окно с сообщением: "Во время подготовки проекта возникли ошибки", а во вкладке **Ошибки** выводятся указания на степень серьезности ошибки и раздел, где допущена ошибка, а также ее описание. В этом случае закладка запускаемого сервиса не появляется.

Так, например, при попытке защитить проект в отсутствие исполняемого файла и при наличии пустого контейнера процесс защиты не запускается и, следовательно, не появляется закладка **Защита** (см. [Защита файлов](#)), как показано на [рисунке ниже](#).



Ошибки при защите

В случае успешного запуска сервиса в соответствующем ему всплывающем окне отражается один сеанс его работы. Если необходимо снова начать выполнение этого сервиса, сначала закрыть его вкладку. Информация обо всех действиях, производимых за полный сеанс работы пользователя с Protection Studio, сохраняется в окне **Журнал**.

При возникновении проблем с установкой защиты может оказаться полезной дополнительная отладочная информация, генерация которой включается в окне "Опции" (см. [Команды меню](#)).

В [таблице ниже](#) показаны ошибки, наиболее часто встречающиеся при установке защиты и соединении с сервером:

Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
Connection to the remote host has been closed according to mutual agreement.	Сбой в сети или на сервере	Повторить попытку соединения с сервером, иначе - обратиться в службу технической поддержки.
Failed to compose output module <имя модуля>. Try to change extra data handling options.	В процессе защиты размер исполняемой части модуля увеличился, в результате чего оверлей не был помещен по оригинальному адресу, что требуется при установленном для опции Размещение оверлеев значении: "Сохранять оверлей в защищенном файле на той же позиции".	Проверить, что файл не упакован и не защищен другими системами защиты, или обратиться в службу технической поддержки для выполнения такой проверки и настройки проекта защиты под данный файл.
File is already protected by Protection System	В качестве файла, подлежащего защите, указан уже защищенный модуль.	Обновить путь до данного исполняемого файла, чтобы он указывал на оригинальную (незащищенную) версию.
Function: <имя> can not be converted to pcode. Possible reasons: function contains shared blocks, does not have convertible instructions.	Функция не может быть защищена, возможно, потому, что она содержит блоки, к которым есть обращения из других функций, или она не содержит инструкций, на которые можно установить защиту.	Отключить защиту для данной функции и выбрать для защиты другую функцию; или использовать pdb-файл.
Данная операция не может быть выполнена из-за превышения соответствующей квоты проекта: <число>. Обратитесь в службу поддержки (support@star-force.com) для увеличения квоты.	Превышен предел выполнения (квоты) данной операции для данного проекта.	Направить запрос менеджеру по продажам StarForce об увеличении квоты на данную операцию для данного проекта.
Настройки проекта не могут быть извлечены с сервера защиты. Соответствующая схема не найдена. Обратитесь в службу поддержки	Изменение версии или схемы проекта.	Обратиться в службу технической поддержки для обновления версии или схемы проекта.

Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
(support@star-force.com)		
Не найдено активных проектов защиты для подключения. Свяжитесь с отделом продаж для продления/ разблокирования требуемых проектов и затем попробуйте снова.	Окончание срока действия или блокирование всех проектов защиты, подключенных к данной учетной записи.	Связаться с отделом продаж для продления/ разблокирования требуемых проектов и затем снова попробовать подсоединиться к серверу защиты.
Сервер в данный момент занят. Превышено количество подключенных пользователей. Повторите попытку позже или подключитесь к другому серверу защиты.	Занятость сервера из-за предельного количества подключенных пользователей.	Повторить попытку соединения с сервером позже или подключиться к другому серверу защиты.
Срок действия данного рабочего пространства истек. Для продления срока его действия обратитесь в отдел продаж.	Время действия данного рабочего пространства пользователя на сервере защиты StarForce истекло.	Направить в отдел продаж запрос на продление срока действия данного рабочего пространства.

Ошибки при установке защиты

Особенности поведения системы защиты

- При попытке программно переместить файл на диске на место файла, помещенного в контейнер, вместо сообщения об ошибке ERROR_ALREADY_EXISTS (183) система защиты выдает сообщение ERROR_ACCESS_DENIED (5).
- При попытке программного создания новой директории с именем директории, уже существующей в контейнере, вместо сообщения об ошибке ERROR_ALREADY_EXISTS (183) система защиты возвращает сообщение ERROR_ACCESS_DENIED (5).
- При попытке создания новой поддиректории внутри директории, уже существующей в контейнере, вместо сообщения об ошибке ERROR_DISK_FULL (112) возвращается сообщение об ошибке ERROR_PATH_NOT_FOUND(3).

Если возникают трудности с устранением ошибок при установке защиты, обратитесь в Службу технической поддержки StarForce (см. [Техподдержка](#)).

6.2 Диагностика и устранение ошибок при запуске защищенного приложения

В таблице ниже представлены ошибки, наиболее часто встречающиеся во время запуска защищенного приложения, с рекомендациями по их исправлению, а также ID соответствующих сообщений в Редакторе сообщений.

ID	Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
MessageB	Зафиксирована попытка	Попытка запустить	Закрыть программный

ID	Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
adProcess Owner	запуска приложения из запрещенного программного продукта. Закройте программный продукт и перезапустите приложение.	приложение из запрещенного программного продукта.	продукт и перезапустить приложение.
MessageCanNotRun UnderDebugger	Это приложение не может быть запущено под отладчиком. Деактивируйте все активные отладчики и запустите приложение снова.	Работает отладчик(и).	Деактивировать активные отладчики и запустить приложение.
MessageDataImport	Невозможно запустить приложение. Модуль %1!s! содержит защищенные импорты данных. Ошибка установки защиты.	При загрузке обнаружены импорты данных.	Обратиться с отчетом об ошибке в службу технической поддержки.
MessageDriverOpen Failed	Ошибка при доступе к драйверу. Код ошибки: %1!d!.	Некорректная работа с драйвером защиты.	Обратиться с отчетом об ошибке в службу технической поддержки.
MessageDriversAre NotInstalled	Драйвер не установлен на вашем компьютере или установлен неполностью.	На компьютере пользователя совсем не установлен или установлен неполностью драйвер системы защиты.	Полностью установить драйвер.
MessageDriversUpdateNetworkError	Подключение к серверу обновления невозможно. Проверьте подключение к Интернет и попробуйте снова. Код ошибки: %1!d!. Чтобы получить отчет об ошибке для последующей передачи его в службу технической поддержки продукта, нажмите на соответствующую кнопку.	Сервер обновлений не отвечает.	Нажать на кнопку "Повторить" для для повторной активации приложения.
MessageExpiredWithNoBinding	Срок действия приложения истек.	Срок действия приложения истек.	Обратиться в службу технической поддержки для приобретения нового продукта.
MessageFileSystemInitFailed	Ошибка при доступе к данным приложения. Переустановите приложение и попробуйте снова.	Сбой в системе.	Переустановить приложение.
MessageInternalError	Внутренняя ошибка (код ошибки: 0x%1!X!). Закройте все приложения и попробуйте снова. Если ошибка останется, нажмите на кнопку " Отчет об ошибке " и вышлите отчет в службу технической поддержки продукта.	Сбой в системе.	Закрыть все открытые приложения и перезапустить программу.

ID	Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
MessageInvalidDll	Загрузка файла динамической библиотеки %1!s! невозможна.	Некоторые dll-файлы не запускаются на компьютере пользователя.	Обратиться с отчетом об ошибке в службу технической поддержки.
MessageInvalidTimeSettings	Невозможно запустить приложение в связи с некорректными установками времени в системе. Проверьте установки времени и перезапустите приложение. Если ошибка повторится, нажмите на кнопку "Отчет об ошибке" и вышлите отчет в службу технической поддержки продукта.	Неверные установки времени в системе.	Проверить установки времени и перезапустить приложение. При возникновении ошибок нажать на "Отчет об ошибке" и выслать отчет об ошибке в службу поддержки продукта.
MessageModuleInfected	Модуль %1!s! программы был изменен. Возможно, он поврежден или заражен вирусом. Проверьте ваш компьютер на наличие вирусов и переустановите приложение.	Один из защищенных модулей приложения или библиотека защиты повреждены или заражены вирусом.	Проверить компьютер на наличие вирусов и переустановить приложение.
MessageMultipleInstancesConflict	Приложение уже запущено. Невозможно запустить вторую копию приложения.	Попытка запустить вторую копию приложения.	Использовать запущенное приложение или закрыть и перезапустить его.
MessageNeedAdministrator	Для установки или удаления драйвера необходимо иметь права Администратора. Чтобы получить отчет об ошибке для последующей передачи его в службу технической поддержки продукта, нажмите на соответствующую кнопку.	Для установки/удаления драйвера необходимо иметь права администратора.	Убедиться в наличии прав администратора и попробовать снова.
MessageNeedCloseAllPrograms	Обнаружен конфликт между драйвером и его версией, использующейся другой программой. Закройте все окна и перезапустите приложение.	Несовместимость драйвера защиты с драйверами, используемыми другой программой.	Перезапустить приложение, закрыв предварительно все лишние программы.
MessagePcEmulatorDetected	Была обнаружена попытка использования системного эмулятора. Пожалуйста, закройте все эмулирующие программы и попробуйте снова.	На компьютере пользователя запущена программа-эмулятор.	Закрыть эмулирующие программы и запустить приложение снова.
MessageRegistryAccessError	Ошибка при доступе или попытке внесения изменений в системный реестр. Удостоверьтесь в наличии права на запись в системный реестр. Если ошибка повторится, нажмите на кнопку "Отчет об ошибке" и вышлите	Нет прав доступа в системный реестр.	Проверить наличие прав на запись в системный реестр и повторить попытку.

ID	Текст сообщения об ошибке	Возможная причина	Рекомендации по устранению ошибки
	отчет в службу технической поддержки продукта.		
MessageRemoteSessionDetected	Запуск приложения на удаленном компьютере невозможен. Запустите приложение локально, с вашего компьютера.	Версия защиты не поддерживает запуск приложения на удаленном компьютере.	Запустить приложение с локального компьютера.
MessageSafeMode	Windows загружена в режиме защиты от сбоев (Safe Mode). Запуск программы невозможен. Перезагрузите компьютер в обычном режиме. Если ошибка повторится, нажмите на кнопку "Отчет об ошибке" и вышлите отчет в службу технической поддержки продукта.	Windows работает в режиме защиты от сбоев.	Перезагрузить компьютер в обычном режиме и повторить попытку.
MessageShellExecuteDoesNotWork	Возможно, на вашем компьютере установлена программа Quick View Plus, что приводит к некорректной работе системы защиты. Для решения проблемы запустите Quick View Plus и отключите следующую опцию: View → Configure Quick View Plus → Applications → Windows Explorer → View Unregistered File Types.	На компьютере установлена программа Quick View Plus.	Отключить опцию: View → Configure Quick View Plus → Applications → Windows Explorer → View Unregistered File Types и повторить попытку.
MessageUnsupportedOperatingSystem	Ваша операционная система не соответствует системным требованиям продукта.	Операционная система пользователя не соответствует системным требованиям продукта.	Переустановить систему. Уточнить версию в службе технической поддержки.
MessageUnsupportedProcessor	Для обеспечения корректной работы приложения требуется процессор не ниже Pentium (tm) или любой другой, совместимый с ним. Чтобы получить отчет об ошибке для последующей передачи его в службу технической поддержки продукта, нажмите на соответствующую кнопку.	Используемый процессор не поддерживает приложение.	Использовать компьютер с другим процессором.

Ошибки при запуске защищенного приложения

При определении настроек защиты можно влиять на количество этих сообщений и объем помещаемой в них информации. Для этого служит ряд опций раздела **Параметры пользовательского интерфейса: Вопрос об установке драйвера, Информация в окне об установке драйвера, Информация в окне сообщений об ошибках, Показывать индикатор состояния** и др. (см. [рисунок](#)).

Кроме того, есть возможность задать запись информации в системный журнал событий, воспользовавшись файлом защиты `protect.exe`. Для этого надо выставить флажок в поле **Register event log** диалогового окна файла (см. [рисунок](#)).

Со всеми вопросами конечному пользователю предлагается обращаться в Службу поддержки продукта, контактная информация которой указывается в соответствующем поле ввода в разделе **Параметры пользовательского интерфейса** (см. [рисунок](#)).

6.3 Рекомендации по размещению файлов библиотек защиты

При возникновении проблем с запуском защищенного приложения рекомендуется изменить расположение файлов `PROTECT.DLL` и `PROTECT.EXE`. Варианты размещения этих файлов в зависимости от структуры приложения описаны ниже.

Размещение `PROTECT.DLL` и `PROTECT.EXE`

Если защищаемые исполняемые файлы размещены в установленном приложении не в одной папке, возникает вопрос о поиске библиотеки защиты `PROTECT.DLL` и модуля `GUI PROTECT.EXE`. Действительно, т.к. каждый защищённый модуль импортирует функции из `PROTECT.DLL`, эта библиотека должна загрузиться в память при загрузке защищённого модуля. Файл `PROTECT.EXE` должен быть размещён так, чтобы `PROTECT.DLL` смогла его найти.

Рассмотрим пример. Пусть защищаются 2 файла: `protected1.exe` и `protected2.dll`, размещённые следующим образом:

```
ROOT_FOLDER
  SUBFOLDER1
    protected1.exe
  SUBFOLDER2
    protected2.dll
```

Возникает вопрос: как расположить `PROTECT.DLL` и `PROTECT.EXE` после защиты для нормальной работы приложения?

Protection Studio по умолчанию размещает по одной копии `PROTECT.DLL` в каждой папке, содержащей хотя бы один защищённый модуль, и пару `PROTECT.DLL` + `PROTECT.EXE` в корневой папке. Благодаря такому размещению нужные файлы находятся системой почти всегда, но имеется некоторая избыточность. Кроме того, в некоторых случаях (которые нельзя определить на этапе защиты) при таком размещении возникают проблемы с запуском приложения.

Это приводит к тому, что иногда всё же приходится заниматься размещением этих файлов вручную.

Ниже будут описаны соображения, которые следует принимать во внимание при размещении `PROTECT.DLL` и `PROTECT.EXE`, а также случай, когда стандартное размещение файлов приводит к неработоспособному приложению (с указанием средств борьбы с этим).

Первое, что следует принимать во внимание при размещении `PROTECT.DLL`, – это

алгоритм поиска DLL в Windows по умолчанию. Сначала будем считать, что приложение не переопределяет этот порядок. Переопределение порядка поиска DLL используется при поддержке сложных плагинов, содержащих несколько DLL. Большинство приложений стандартный порядок поиска не переопределяет.

Порядок по умолчанию поиска DLL в Windows таков:

- 1) папка, в которой находится EXE-модуль, инициировавший процесс;
- 2) системная директория (обычно C:\Windows\System32);
- 3) директория windows (обычно C:\Windows);
- 4) текущая папка;
- 5) папки, указанные в переменной окружения PATH

Примечание. В версиях Windows до XP SP2 "текущая папка" стоит на втором месте в списке.

Исходя из пункта 1), корректным будет размещение PROTECT.DLL с EXE, который идентифицирует процесс. Например, если в приведённом выше примере предположить, что protected1.exe – это единственный EXE в приложении и больше PROTECT.EXE никто не загружает, то PROTECT.DLL и PROTECT.EXE корректно разместить следующим образом:

```
ROOT_FOLDER
  SUBFOLDER1
    protected1.exe
    protect.dll
    protect.exe
  SUBFOLDER2
    protected2.dll
```

Если же защищаемое приложение имеет несколько EXE-файлов, размещённых в разных папках, PROTECT.DLL приходится размещать в каждой из них, причём вне зависимости от того, защищённый в папке EXE или незащищённый:

```
ROOT_FOLDER
  SUBFOLDER1
    protected1.exe
    protect.dll
    protect.exe
  SUBFOLDER2
    protected2.dll
  SUBFOLDER3
    protected3.exe
    protect.dll
    protect.exe
  SUBFOLDER4
    not_protected4.exe
    protect.dll
```

```
protect.exe
```

Учитывая то, что структура папок указывается на этапе защиты в Protection Studio, PROTECT.DLL всегда может найти ROOT_FOLDER. Это избавляет нас от хранения PROTECT.EXE в нескольких экземплярах: его можно просто вынести в ROOT_FOLDER. Рядом с PROTECT.EXE, однако, приходится класть копию PROTECT.DLL для того, чтобы работали функции PROTECT.EXE при запуске его без параметров (активация, установка драйвера и т.п.). Структура папок получается следующая:

```
ROOT_FOLDER
  protect.dll
  protect.exe
  SUBFOLDER1
    protected1.exe
    protect.dll
  SUBFOLDER2
    protected2.dll
  SUBFOLDER3
    protected3.exe
    protect.dll
  SUBFOLDER4
    not_protected4.exe
    protect.dll
```

Если известно, что защищённое приложение всегда будет запускаться при указании какой-то определённой текущей папки, можно оставить только одну копию PROTECT.DLL + PROTECT.EXE, разместив её в этой папке:

```
ROOT_FOLDER
  CURRENT_FOLDER_FOR_ALL_RUNS_FOR_ALL_EXES
    protect.dll
    protect.exe
  SUBFOLDER1
    protected1.exe
  SUBFOLDER2
    protected2.dll
  SUBFOLDER3
    protected3.exe
  SUBFOLDER4
    not_protected4.exe
```

Если в приложении защищены только DLL, PROTECT.DLL необходимо размещать не рядом с DLL, а рядом с EXE (в данном случае незащищённым):

```
ROOT_FOLDER
  SUBFOLDER1
    protected1.dll
  SUBFOLDER2
```

```
not_protected2.exe
protect.dll
protect.exe
```

Теперь рассмотрим случай, когда стандартный порядок поиска DLL переопределяется программистом. Такое переопределение осуществляется использованием для загрузки DLL функции LoadLibraryEx (порядок поиска DLL, указанных в таблице импортов, изменить нельзя; поэтому если все защищённые DLL загружаются только через таблицы импортов, будет всегда использоваться стандартный порядок).

Переопределённый порядок отличается от стандартного только тем, что вместо папки, в которой находится EXE-модуль, инициировавший процесс (т.е. пункт 1) из списка выше), используется папка, в которой находится DLL, загружаемая функцией LoadLibraryEx. Такое изменение алгоритма позволяет плагину, состоящему из нескольких DLL, корректно найти все эти DLL вне зависимости от расположения вызывающего EXE (при использовании стандартного алгоритма поиска дополнительные DLL плагина просто бы не нашлись).

Таким образом, при защите плагинов, которые загружаются с помощью LoadLibraryEx, корректно разместить PROTECT.DLL рядом с основной загружаемой DLL плагина:

```
ROOT_FOLDER
  PLUGINS
    protected1.dll
    protect.exe
```

Тем не менее, здесь возможен ряд коллизий:

Во-первых, плагин не обязательно загружается функцией LoadLibraryEx (программист, использующий плагин, мог просто не предусмотреть сложного плагина). Тогда при размещении PROTECT.DLL рядом с плагином она может не найтись.

Во-вторых, если защищён и плагин, и использующее его приложение, возникает ещё более сложная ситуация. Пусть, например, структура папок такова:

```
ROOT_FOLDER
  protected1.exe (загружает protected2.dll с помощью LoadLibraryEx)
  PLUGINS
    protected2.dll
```

Тогда если PROTECT.DLL разместить только рядом с protected1.exe, его не найдёт protected2.dll; если только рядом с protected2.dll – его не найдёт protected1.exe; если и там и там, то он загрузится в память дважды (т.к. он будет загружаться из разных папок), что приведёт либо к двойному прохождению проверки, либо к ошибке MULTIPLE_INSTANCES_CONFLICT.

В подобных сложных случаях универсального рецепта размещения дать нельзя. Самый простой способ – требование запуска приложения всегда из одной определённой текущей папки (куда и следует положить PROTECT.DLL). Также можно рассмотреть возможность модификации переменной окружения PATH.

Вспомогательные файлы, такие, как PROTECT.X86, PROTECT.X64, PROTECT.MSG, PROTECT.GUI, следует размещать рядом с PROTECT.EXE. Обычно достаточно одной копии в корне проекта.

7 Глоссарий

Термин	Пояснение
Анализ	Исследование защищённого приложения путём дизассемблирования или пошаговой трассировки с целью выявления алгоритмов работы его отдельных функций.
Версия StarForce	Вариант продукта, образующий совокупность определённых функциональных возможностей. Версия защиты определяется комплексом версии Protection Studio и версией серверных сервисов, используемых при защите приложений.
Взлом	Метод, призванный отвязать защищенное приложение от физического носителя лицензии и от ядра защиты за счет выделения и нейтрализации блока защиты из кода защищенных модулей приложения путем дизассемблирования и пошаговой отладки. Также для полной работоспособности приложения без объекта привязки при взломе восстанавливают оригинальные части кода защищенных модулей.
Драйвер защиты	Исполняемый модуль, являющийся частью системы защиты от промышленного пиратства. Драйвер защиты использует низкоуровневые средства Windows для предотвращения попыток взлома и эмуляции.
Защита	Совокупность действий, направленных на противодействие нелегальному использованию защищаемых продуктов.
Защита внутренних функций	Процесс, при котором из модуля частично извлекается код защищаемой функции и в преобразованном виде помещается в ядро защиты. Наиболее эффективный способ повышения уровня защиты приложения.
Защита данных приложения	Процесс, при котором защищаемые файлы данных помещаются в специальный файловый контейнер в зашифрованном виде. Защита файлов данных возможна только при защите исполняемого модуля, осуществляющего доступ к этим файлам.
Защита импортируемых функций	Извлечение из защищаемого модуля ссылок на импортируемые функции и помещение их в ядро защиты. Есть как автоматическая защита импортов (входит в шифрование модулей), так и дополнительная защита импортов, призванная незначительно повысить уровень защиты в случаях, когда нет доступа к коду защищаемого приложения и защита внутренних функций не возможна.
Защита модулей	Преобразование модулей защищаемого приложения с целью обеспечения защиты от взлома и анализа. Защита модулей может включать в себя один или несколько следующих способов противодействия взлому и анализу: 1. Защита внутренних функций 2. Защита импортируемых функций 3. Шифрование модулей 4. Защита данных приложения.
Защищаемое приложение	Часть защищаемого продукта, включающая в себя файлы, непосредственно подлежащие защите.
Защищаемый продукт	Продукт, подлежащий защите от несанкционированного использования с помощью продуктов StarForce.
Квота	Ограничение на количество выполнений определенных действий по защите продукта (защита файлов и т.д.). Квоты действуют в рамках рабочего пространства и доступны только пользователям этого пространства, но они могут быть распределены при создании подпространств между другими пользователями.

Термин	Пояснение
Конечный пользователь. End-user	Пользователь продуктов, защищенных системой защиты.
Контейнер проектов	Рабочее пространство, предоставляющее пользователям, назначенным в него, возможность создания проектов защиты StarForce.
Метод защиты	Совокупность действий по преобразованию защищаемого приложения с целью обеспечения его защиты от несанкционированного использования.
Модуль	Относительно самостоятельная часть некоторой системы, выполняющая свою, строго определенную функцию.
Пользователь системы защиты. User	Юридическое или физическое лицо, обладающее правом на использование системы защиты. Он же клиент, заказчик.
Пользовательский интерфейс защищенного приложения (GUI)	Часть системы защиты, необходимая для обеспечения диалога с пользователем защищенного приложения. GUI защиты представлен набором типовых диалоговых окон, служащих для вывода информации об ошибках и т.д.
Привязка	Метод защиты, основанный на внедрении в защищаемое приложение прямой, либо косвенной связи с физическим носителем лицензии.
Проект защиты StarForce	Рабочее пространство, задающее совокупность настроек защиты определенного защищаемого продукта.
Рабочее пространство	Совокупность информационных объектов, доступных пользователю системы защиты StarForce. Виды рабочих пространств: контейнер проектов, проект.
Роль	Набор статических прав, определяющий спектр возможных действий пользователя в системе относительно конкретного рабочего пространства. Одному пользователю может быть назначено любое количество уникальных ролей в одном рабочем пространстве, в этом случае права, разрешённые каждой из ролей, суммируются.
Система защиты StarForce Коротко: Система защиты	Совокупность технических средств, осуществляющих защиту программных продуктов. Установка защиты производится с помощью программного комплекса Protection Studio.
Срок действия проекта защиты StarForce	Срок, исчисляемый с момента открытия соответствующего проекта StarForce, в течение которого на безвозмездной основе осуществляется защита новых версий соответствующего защищаемого продукта.
Уровень защиты	Комплексный критерий, определяющий степень устойчивости защищённого приложения к взлому, анализу и эмуляции.
Учетная запись	Набор параметров, однозначно идентифицирующих пользователя в системе StarForce: имя пользователя (логин) и пароль. Учетная запись дает право её обладателю осуществлять определенные действия по защите продуктов в рамках рабочих пространств, закрепленных за ней, и в соответствии с установленной ролью учетной записи в проекте. Учетная запись также содержит дополнительные сведения о пользователе. Например, email учетной записи используется для рассылки сообщений об определенных событиях, связанных с учетной записью.
Файл проекта защиты	Файл, содержащий настройки защиты конкретной версии защищаемого приложения для определенного проекта защиты.
Ядро защиты	Компонент системы защиты, реализующий средства защиты от взлома, анализа и эмуляции в процессе работы защищенного приложения.
FL MMOG	Продукт FL MMOG обеспечивает защиту игровых серверов от анализа, модификации и возможности запуска на неавторизованных площадках, защиту кода игры от анализа и взлома, защиту от запуска и выполнения чит-программ

Термин	Пояснение
	и ботов, защиту трафика между сервером MMOG и клиентом
Protection Studio (PS)	Программный комплекс, предназначенный для установки защиты на приложения.
SDK защиты. Software Development Kit	SDK защиты – это набор средств для комплексной интеграции защиты в защищаемый продукт.

Глоссарий

8 Техподдержка

В случае возникновения вопросов или трудностей при работе с программой Protection Studio и системой защиты в целом, обратитесь в службу технической поддержки компании StarForce:

- электронная почта: support@star-force.com
- телефон: +7 (495) 967 14 53

Предметный указатель

S

SDK
 функции API защиты 4, 48, 51

Д

Драйвер защиты 16, 25, 70, 82

З

Защита файлов 16, 20, 21, 25, 43
 защита функций 32, 82

П

Параметры защиты 15, 21
Привязка 24, 82