

## **V7Crypt – расширение платформы 1С:Предприятие 7.7**

<a href="#">Основные функции:</a> .....	2
<a href="#">Установка расширения:</a> .....	3
<a href="#">Описание:</a> .....	4
<a href="#">Защита конфигурации:</a> .....	5
<a href="#">Формирование шаблона лицензий:</a> .....	5
<a href="#">Сроки лицензий:</a> .....	5
<a href="#">Режимы защиты конфигурации:</a> .....	6
<a href="#">Формирование задания для кодирования:</a> .....	6
<a href="#">Выполнение кодирования:</a> .....	7
<a href="#">Привязка защищенной конфигурации или внешней обработки к рабочим станциям:</a> .....	7
<a href="#">Расширение blang:</a> .....	8
<a href="#">Множественные глобальные модули (МГМ):</a> .....	8
<a href="#">События МГМ:</a> .....	8
<a href="#">Обработка ожидания в МГМ:</a> .....	9
<a href="#">Собственные контексты модулей (внешние общие модули):</a> .....	9
<a href="#">Команда Sleep</a> .....	9
<a href="#">Макроопределение @autoheader</a> .....	9
<a href="#">Дополнительные события объекта Документ - Менеджер объекта</a> .....	10
<a href="#">Расширение V7CryptLib</a> .....	12

## **Основные функции:**

- 1. Полноценная поддержка нескольких глобальных модулей.**
- 2. Дополнительные события объекта Документ (менеджер объекта).**
- 3. Защита (компиляция и шифрование) конфигурации и внешних обработок:**

### **Стандартная лицензия:**

- полная защита конфигурации и внешних обработок, входящих в поставку, с запрещением выполнения не подписанных сторонних внешних обработок;
- ограничение работы защищенной конфигурации сроком один месяц;
- возможность привязки защищенной конфигурации к рабочим станциям.

### **Расширенная лицензия (при наличии файла регистрации):**

- полная защита конфигурации и внешних обработок, входящих в поставку, с возможностью выполнения не подписанных сторонних внешних обработок;
- защита внешних обработок;
- защита отдельных модулей конфигурации;
- ограничение работы защищенной конфигурации на любой срок, или без ограничений.
- возможность привязки защищенной внешней обработки к рабочим станциям.

## **4. Реализация основных алгоритмов шифрования и кодирования, в виде расширения платформы:**

- шифрование: RSA, RC5;
- кодирование: Base64;
- хеширование: MD5;
- сжатие: ZIP;
- вспомогательные: процедура сравнения двух защищенных обработок; получение уникального идентификатора рабочей станции; определение защищен ли модуль на этапе выполнения.

## **5. Макрокоманда @autoheader**

## **6. Команда Sleep (Пауза)**

## **Установка расширения:**

После установки программы на компьютер, необходимо установить расширение V7Crypt в каталоги платформ 1С: Предприятия 7.7. Для облегчения задачи, в поставку включена программа установщик. Пункт меню: «V7Crypt – Установка».

**Для установки расширения**, нажмите кнопку «Инсталлировать» и укажите путь к исполняемому файлу 1С. Никаких изменений в файлах платформы не производится!

- в каталог bin, платформы, копируется файл V7Crypt.dll
- в каталог bin, платформы, копируется файл V7Crypt.als
- стандартный файл blang.dll переименовывается в 1c\_blang.dll
- в каталог bin, платформы, копируется файл blang.dll
- в каталог bin, платформы, копируется файл v7events.dll

Если операция прошла успешно, в списке появится путь к этой платформе.

**Для деинсталляции расширения**, выделите в списке необходимую платформу и нажмите «Деинсталлировать». В случае полной деинсталляции программы, расширение из всех платформ списка удалится автоматически.

**Для обновления расширения**, необходимо установить новую версию программы в тот же каталог и запустить программу установщик. Обновление компоненты в платформах будет произведено автоматически. В случае если автоматическое обновление всех платформ списка нежелательно, можно обновить отдельные платформы, выборочной, повторной инсталляцией (без предварительной деинсталляции).

## Описание:

Программа V7Crypt – состоит из двух функциональных частей: кодирующей и исполняющей.

**Кодирующая часть - V7Crypt.dll**, реализована в виде внешней компоненты предназначенной для компиляции и кодирования модулей 1С. Для управления данной компонентой в комплект поставки включены внешние обработки:

<Каталог установки>\Crypt\V7CryptCont.ert – формирует ключи и шаблоны лицензий в виде контейнера;

<Каталог установки>\Crypt\V7CryptWork.ert – формирует задание для выполнения, т.е. набор требуемых действий по кодированию модулей;

<Каталог установки>\Crypt\V7Crypt.ert – выполняет кодирование на основе задания и шаблона лицензии.

Обработки оставлены в открытом виде, чтобы служить примером работы с компонентой.

**Исполняющая часть - Blang.dll**, реализована в виде обертки (wrapper) для родной библиотеки 1С Blang.dll, при установке в платформу библиотека 1С переименовывается в 1С\_blang.dll, а обертка копируется на ее место, расширяя функционал платформы.

## **Защита конфигурации:**

### **Формирование шаблона лицензий:**

1. Запустите обработку Crypt\V7CryptCont.ert и создайте новый контейнер. При формировании контейнера будет также сформирован секретный ключ для операций с данным контейнером, сохраните его в надежном месте.
2. Добавьте в контейнер новый ключ.
3. Добавьте в контейнер шаблон лицензии. В шаблоне укажите ключ, который лицензия будет использовать, период действия лицензии, а также ее тип. В данный момент создается только шаблон. Лицензия будет создана в момент кодирования на основании этого шаблона и файла регистрации. Если файла регистрации нет, то на лицензию будут наложены ограничения, а именно:

- срок лицензии будет 1 месяц;

- доступен режим только полной защиты конфигурации.

Файл регистрации нужен только на одном компьютере, на том, который выполняет кодирование. На клиентских машинах файл регистрации не нужен.

Порядок получения регистрационного файла описан в документе «Регистрация».

### **Сроки лицензий:**

Срок периода лицензии считается с момента защиты модуля в месяцах, включая текущий. Иначе говоря, если в текущем месяце, например 25-го числа, защитить модуль с периодом действия 1 месяц, то фактически модуль проработает не 30 дней, а до 1-го числа следующего месяца, т.е. 5-6 дней. По истечении этого срока Вам нужно будет предоставить обновление, например, со сроком действия еще на месяц. С другой стороны, если Вы внесете изменения в модуль, защитите его, и предоставите обновление, то тем самым можете неявно продлить срок его использования.

### **Режимы защиты конфигурации:**

1. **Полная защита конфигурации** - защищает всю конфигурацию единой лицензией, что делает невозможным выполнения внешних обработок, не подписанных той же лицензией. Удобно, когда необходимо полностью защитить программу.
2. **Защита модулей** - защита каждого модуля отдельной лицензией, что позволяет защищать как отдельные обработки, так и конфигурации, но с возможностью выполнения внешних, не подписанных обработок.
3. **Выборочная защита** - позволяет защищать как отдельные обработки, так и конфигурации, при этом доступна выборочная защита отдельных частей конфигурации. Возможно выполнение внешних, не подписанных обработок.

### **Формирование задания для кодирования:**

1. Запустите обработку Crypt\V7CryptWork.ert и создайте новое задание. Укажите контейнер лицензии, шаблон лицензии, который будете использовать. А также путь к объектам, которые необходимо защитить, и путь для помещения защищенных файлов.
2. По умолчанию будет создан типовой список объектов задания:
  - копировать рекурсивно Источник\\* => Приемник\\*
  - кодировать Источник\1cv7.md => Приемник\1cv7.md
  - кодировать рекурсивно Источник\\*.ert => Приемник\\*.ert
  - исключить Источник\NEW\_STRU\\*Список объектов задания обрабатывается по типу вытесняющего стека. Если копировать и исключать ни чего не надо, то можно указать только действия для объектов кодирования.
3. Сохраните задание.

### **Выполнение кодирования:**

Запустите обработку `Crypt\V7Crypt.ert`, выберите задание, нажмите сформировать. Модули компилируются самой IC, bin-коды шифруются и сохраняются. Соответственно, запуск процедур кодирования необходимо выполнять именно в той базе, которую защищаете. Иначе контексты компиляции будут другие, и защищенная программа будет работать не корректно, или совсем не будет работать.

Если в модуле возникнет синтаксическая ошибка при компиляции, то такой модуль будет защищен, но в нем будет присутствовать исходный текст!

### **Привязка защищенной конфигурации или внешней обработки к рабочим станциям:**

Для обеспечения привязки Вашей программы к рабочим станциям можно воспользоваться способом, показанным в примере: `example\Привязка.ert`. Защита базируется на криптостойкости алгоритма шифрования RSA. Поэтому, обязательно используйте только Ваши собственные наборы ключей (набор ключей, например, можно сформировать обработкой `example\V7CryptLib.ert`). Способ формирования кода ответа, описан в примере: `example\КодОтвета.ert`.

## **Расширение blang:**

### **Множественные глобальные модули (МГМ):**

Теперь Глобальных модулей может быть столько - сколько Вам необходимо. Помимо основного, глобальным модулем считается обработка конфигурации (не внешняя), название которой начинается с «гм» или «gm» (регистр букв не важен).

Например: Обработка.гмМойМодуль

### **События МГМ:**

Начиная с версии 1.0.0.7, множественные глобальные модули являются полноценными. В них транслируются все глобальные события основного глобального модуля. А также для них может быть назначена собственная обработка ожидания (тестовая функция).

Обратите внимание на то, что порядок, в котором модули расположены в дереве конфигурации, важен. Он определяет последовательность обработки модулей. Первый в списке – основной глобальный модуль.

Глобальные модули инициализируются в прямом порядке. Как следствие, вышестоящие модули не могут (не должны) вызывать процедуры нижестоящих.

Если в нескольких МГМ модулях объявлена одинаковая процедура, то при вызове из вне, будет выполнена процедура модуля последнего по порядку (своеобразное переопределение). Будьте внимательны с таким переопределением, так как при вызове изнутри, модуль будет использовать собственную процедуру. Во избежание путаницы с одинаковыми названиями параметров и процедур, их вызов можно производить, явно указывая имя модуля, например: гмМойМодуль.МояПроцедура(); Также способ применим для вызова переопределенной процедуры «предка».

Обработка событий происходит в обратном порядке. Т.е самый последний по порядку модуль получает событие, если процедура для получения события в нем не определена или после ее выполнения статус возврата не будет сброшен в ноль, то управление перейдет к следующему модулю, вверх по списку, пока не достигнет основного глобального модуля. Обратите внимание, если по мере обработки события, один из модулей установит статус возврата в ноль, обработка события прервется и вышестоящие модули не получат управление.



## **Обработка ожидания в МГМ:**

Для назначения МГМ модулю собственной обработки ожидания в платформу введена команда «**ОбработкаОжиданияМГМ**» или «**IdleProcessingMGM**». Команда, по сути, является аналогом стандартной функции «ОбработкаОжидания», но в первом параметре принимает имя модуля МГМ, для которого устанавливается обработчик, второй параметр имя процедуры обработчика, третий интервал в секундах.

## **Собственные контексты модулей (внешние общие модули)**

Есть ряд задач, при которых желательно использовать общие методы, но они не настолько глобальные, для того чтобы их размещать в глобальном модуле. В классическом программировании 1С, для подобных случаев обычно создается обработка с необходимыми функциями, а ее вызов и обмен параметрами выполняется через передачу параметра формы.

Начиная с версии 1.0.1.4, в платформу, добавлена команда **СоздатьКонтекст (CreateContext)**. Параметр команды - «Обработка.имя\_обработки» или путь к внешней обработке, результат возврата – контекст модуля обработки. В контексте модуля доступны переменные и методы модуля, отмеченные как «Экспорт». Обработка используется только как хранилище текста модуля, поэтому форма и макеты таблиц не будут доступны.

## **Команда Sleep**

Существует множество вариантов реализации данной команды в виде внешних компонент. Но, как кажется автору, в составе расширения blang ей самое место.

Команда «**Sleep**» или «**Пауза**» - позволяет приостановить выполнение программы на некоторое время без загрузки процессора (аналог API функции). В качестве параметра – интервал в микросекундах (1/1000 сек).

## **Макроопределение @autoheader**

Компилятор 1С требует объявление функции до ее первого использования, для этого в языке предусмотрена специальная инструкция «Далее». Макрокоманда @autoheader позволяет отказаться от такого предварительного определения. Для использования возможности, укажите в первой строке модуля: @autoheader

## Дополнительные события объекта Документ - Менеджер объекта

События в 1С - место слабое. Например, нет события ПриЗаписи в режиме объекта, а только в режиме формы. Что не позволяет полноценно обеспечивать целостность информации в базе данных для более-менее сложных случаев. Примером такого случая может стать ситуация при которой необходимо при проведении (удалении и т.д.) документа произвести изменение (или проведение) документов логически с ним связанных. Или как довольно частая необходимость сформировать документом проводки на разные даты (например, созданием дополнительной операции на другую дату). Для таких ситуаций, скажем прямо, событий маловато.

И так в чем суть. Помимо стандартных предопределений: «ОбработкаПроведения» и «ОбработкаУдаленияПроведения», в модуле документа теперь можно определить дополнительные процедуры для обработки событий:

**"ПриЗаписи" или "OnWrite";**

**"ПередПроведением" или "BeforePosting";**

**"ПослеПроведения" или "AfterPosting";**

**"ПередУдалениемПроведения" или "BeforeUnposting";**

**"ПослеУдаленияПроведения" или "AfterUnposting";**

**"ПередУдалением" или "BeforeDelete";**

**"ПослеУдаления" или "AfterDelete";**

**"ПередОтменойУдаления" или "BeforeUnDelete";**

**"ПослеОтменыУдаления" или "AfterUnDelete".**

Процедуры не имеют входящих аргументов и параметра возврата. В процедурах: «ПриЗаписи», а также с префиксом «Перед», можно отменить выполняемое действие, установив статус возврата в ноль, при этом будет брошено стандартное исключение языка 1С. Процедуры, с префиксом «После», действие отменить не могут и вызываются только как уведомление.

Немаловажный факт, что события с префиксами «Перед» и «После» выполняются вне рамок транзакции побуждающего действия, т.е. не блокируют базу данных.

События «Перед» и «После» выполняются в рамках одного контекста модуля документа, т.е. в событии «Перед» можно выбрать данные (например, проводки) и

сохранить их в переменной модуля документа, а в событии «После», сравнить эти данные с результатом после выполнения действия, и произвести необходимые операции.

Например, в событии «ПередУдалениемПроведения» можно выбрать и сохранить в переменную модуля документа список проводок, которые будут удалены. А в событии «ПослеУдаленияПроведения» просмотреть список удаленных проводок и выполнить проведение зависимых документов.

Обратите внимание, что стандартное предопределение модуля документа - «ОбработкаУдаленияПроведения», будет выполнено в собственном контексте, т.е. не сможет обратиться к значениям переменных установленных в событии «ПередУдалениемПроведения». Это связано с тем, что 1С создает для этого события собственный контекст, к которому пока не удалось подобраться.

С предопределением «ОбработкаПроведения» таких проблем нет. Можно, к примеру, выполнить всю черную работу (например, выбрать итоги) в функции «ПередПроведением», а в самой обработке проведения просто создать проводки (что снизит время блокировки базы данных). Также в событии «ПередПроведением» можно посмотреть первоначальные проводки документа, если он перепроводится (полезный подход для организации динамических последовательностей).

И еще один момент, не допускайте рекурсивного срабатывания событий. Это может привести к зависанию системы. Пример такой рекурсии: При проведении документа №1, проводим документ №2, который проводит документ №1.

Дополнительные события модуля документа транслируются в дополнительные события глобального модуля (и в МГМ соответственно) в виде процедур с одним аргументом - контекстом документа:

**"ПриЗаписиДокумента" или "OnWriteDocument";**

**"ПередПроведениемДокумента" или "BeforePostingDocument";**

**"ПослеПроведенияДокумента" или "AfterPostingDocument";**

**"ПередУдалениемПроведенияДокумента" или "BeforeUnpostingDocument";**

**"ПослеУдаленияПроведенияДокумента" или "AfterUnpostingDocument";**

**"ПередУдалениемДокумента" или "BeforeDeleteDocument";**

**"ПослеУдаленияДокумента" или "AfterDeleteDocument";**

**"ПередОтменойУдаленияДокумента" или "BeforeUnDeleteDocument";**

**"ПослеОтменыУдаленияДокумента" или "AfterUnDeleteDocument".**

## Расширение V7CryptLib

В платформу добавлен новый объект «V7CryptLib». Платформа считает его своей частью, поэтому для его использования достаточно создать его как любой другой объект: `crypt=СоздатьОбъект(«V7CryptLib»);`

Объект реализует следующий набор функций:

- Генерация RSA и RC5 ключей
- Шифрование по методу RSA и RC5
- Кодирование методом Base64
- Хеширование методом MD5
- Сжатие методом Zip
- Сравнение защищенных обработок
- Получение идентификатора рабочей станции.
- Определение защищен ли модуль, на этапе выполнения.

Подробнее с методами данного объекта можно ознакомиться через «Синтаксис-помощник» конфигулятора.

Пример работы с данным объектом реализован в примере: <Каталог установки>\example\V7CryptLib.ert