



КРИПТОАРМ 5

Руководство программиста

ООО «ЦИФРОВЫЕ ТЕХНОЛОГИИ», © 2014

**РУКОВОДСТВО ПРОГРАММИСТА**  
**КРИПТОАРМ**  
**(версия 5.2)**

## ОГЛАВЛЕНИЕ

Введение .....	3
Раздел 1. Описание COM DigtCrypto.....	3
1.1. Свойства и методы .....	3
1.2. Примеры использования.....	4
1.2.1. Работа с криптопровайдерами .....	4
1.2.2. Работа с сертификатами и хранилищем сертификатов .....	5
1.2.3. Управление запросами .....	9
1.2.4. Клиент УЦ. Управление PKI запросами.....	15
1.2.5. Работа с профилями.....	25
1.2.6. Работа с подписями .....	30
1.2.7. Выполнение криптографических операций.....	34
1.2.8. Работа с версиями DigtCrypto .....	46
1.2.9. Журналирование событий .....	46
1.3. Интерфейсы DigtCrypto .....	46
1.3.1. Константы и перечисления .....	46
1.3.2. Объекты и интерфейсы.....	54
1.3.3. События.....	327
Раздел 2. Описание COM DigtTSPClient .....	330
2.1. Свойства и методы .....	330
2.2. Примеры использования.....	330
2.2.1. Создание и сохранение в хранилище нового TSP профиля .....	330
2.2.2. Получение штампа времени и его свойств.....	331
2.2.3. Получение подписи со включенным в нее штампом времени.....	333
2.3. Интерфейсы DigtTSPClient.....	334
2.3.1. Константы и перечисления .....	334
2.3.2. Объекты и интерфейсы.....	335
Раздел 3. Описание COM DigtOCSPClient.....	369
3.1. Свойства и методы .....	369
3.2. Примеры использования.....	369
3.2.1. Создание и сохранение OCSP профиля .....	369
3.2.2. Получение OCSP ответа и его свойства .....	370
3.3. Интерфейсы DigtOCSPClient .....	371
3.3.1. Константы и перечисления .....	371
3.3.2. Объекты и интерфейсы.....	372
Раздел 4. Описание COM- и NPAPI-библиотеки DigtCAPICOM .....	398
4.1. Инициализация DigtCAPICOM.....	398
4.2. Общие правила вызова API плагина .....	399
4.3. Специфика работы с датами.....	399
4.4. Перечисления и объекты.....	400
4.5. Примеры использования.....	404
4.5.1. Создание подписи, совмещенной с документом .....	404
4.5.2. Проверка подписи, совмещенной с документом .....	404
Раздел 5. Структура хранения профилей в реестре Windows .....	405

## ВВЕДЕНИЕ

При разработке нового программного решения ИТ-компаниям необходимо учитывать важность обеспечения данным решением безопасности информационного пространства. Практически во всех составляющих современного бизнеса - электронный документооборот, интернет-приложения, бухгалтерия и финансы, ERP и CAD/CAM/CAE-системы – стоит вопрос внедрения разработок в области криптографической защиты информации.

Встраивание COM-компонента DigtCrypto в программу или программный комплекс позволит решить задачи комплексной защиты данных. Компонент предназначен для предоставления удобного интерфейса к криптографическим функциям операционных систем MS Windows в целях защиты личной и корпоративной информации. DigtCrypto представляет собой библиотеку криптографических функций, созданную в соответствии со стандартами компонентной модели компании Microsoft.

## РАЗДЕЛ 1. ОПИСАНИЕ COM DIGTCRYPTO

### 1.1. СВОЙСТВА И МЕТОДЫ

Интерфейс	Описание
Certificate	Описывает сертификат
Certificates	Описывает коллекцию сертификатов
CertificateStore	Описывает хранилище сертификатов
CRL	Описывает список отзыва сертификатов
CRLEntry	Описывает параметры отозванного сертификата
CRLs	Описывает коллекцию списков отзыва сертификатов
CryptoProvider	Описывает криптопровайдер
CryptoProviders	Описывает коллекцию криптопровайдеров, установленных в системе
CTL	Описывает список доверенных сертификатов
CTLs	Описывает коллекцию списка доверенных сертификатов
CTLBuilder	Позволяет создать СДС
CTLEntry	Описывает свойства доверенного сертификата
CTLEntries	Описывает элемент списка доверенных сертификатов
CTLTemplate	Описывает шаблон СДС
Dictionary	Описывает коллекцию OID-ов заданного типа
DictionaryStore	Описывает хранилище словарей OID-ов
HashedData	Позволяет вычислить хэш.
OID	Описывает OID
OIDs	Описывает коллекцию OIDов
PKCS7Message	Описывает сообщение формата PKCS7 (подпись, шифрованные данные)
PKIProfile	Описывает профиль PKI запроса
PKIRequest	Описывает работу с PKI запросами
EncryptedMessage	Предоставляет свойства и методы для шифрования/расшифрования блоков данных на одном сессионном ключе
PKIRequestStore	Описывает хранилище PKI запросов
PolicyProfile	Описывает политику использования сертификатов
Profile	Описывает профиль (настройки для выполнения криптографических операций)
Profiles	Описывает коллекцию профилей
ProfileStore	Описывает хранилище профилей
RequestCertificate	Описывает запрос на сертификат
RequestTemplate	Описывает шаблон запроса на сертификат
ServiceProfile	Описывает настройки доступа к Веб-сервису (для загрузки профилей из централизованного хранилища на сервере Веб-сервиса)
Signature	Описывает подпись
Signatures	Описывает коллекцию подписей
Util	Описывает получение версии ПО "КриптоАРМ"
WebServiceProfile	Описывает профиль настроек для соединения с веб-сервисом.
<a href="#">Tsl</a>	Предоставляет свойства и методы для работы со списком аккредитованных УЦ

## 1.2. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

### 1.2.1. РАБОТА С КРИПТОПРОВАЙДЕРАМИ

#### Подключение отчуждаемого носителя

Подключение отчуждаемого носителя производится с помощью метода `ConnectRemovableKeyCarrierWizard` объекта `Cryptoproviders`.

```
Option explicit
'Объявление переменных
Dim oCryptoProviders
Set oCryptoProviders = CreateObject("DigtCrypto.CryptoProviders")

'Вызов мастера подключения отчуждаемого носителя
oCryptoProviders.ConnectRemovableKeyCarrierWizard()
```

#### Получение списка поддерживаемых криптопровайдеров

Чтобы получить список поддерживаемых криптопровайдеров, следует использовать метод `SupportedProviders` объекта `Cryptoproviders`.

```
Option explicit
'Объявление переменных
Dim oCryptoProviders
Dim oCryptoProviders1
Dim count
Dim index
Dim sProviderName

Set oCryptoProviders = CreateObject("DigtCrypto.CryptoProviders")
'Получение коллекции поддерживаемых криптопровайдеров
Set oCryptoProviders1= oCryptoProviders.SupportedProviders

'Получим количество поддерживаемых DigtCrypto криптопровайдеров, установленных
  в системе
count = oCryptoProviders1.Count
WScript.Echo "Количество поддерживаемых криптопровайдеров, установленных в
  системе: " & count

For index = 0 To count-1
    sProviderName=sProviderName + oCryptoProviders1.Item(index).Name +
Chr(13)
Next
MsgBox sProviderName, "Список поддерживаемых криптопровайдеров"
```

#### Получение списка системных криптопровайдеров

Чтобы получить список системных криптопровайдеров, следует использовать метод `SystemProviders` объекта `Cryptoproviders`.

```
Option explicit

'Объявление переменных
Dim oCryptoProviders
Dim oCryptoProviders1
Dim count
Dim index
Dim sProviderNam

'Создадим коллекцию доступных криптопровайдеров
Set oCryptoProviders = CreateObject("DigtCrypto.CryptoProviders")
```

```
'Получим коллекцию системных криптопровайдеров
Set oCryptoProviders1= oCryptoProviders.SystemProviders

'Получим количество системных криптопровайдеров в коллекции
count = oCryptoProviders1.Count
WScript.Echo "Количество системных криптопровайдеров: " & count

' Просмотрим наименования криптопровайдеров, установленных в системе
For index = 0 To count-1
    sProviderName=sProviderName + oCryptoProviders1.Item(index).Name +
    Chr(13)
Next
MsgBox sProviderName, , "Список системных криптопровайдеров"
```

## 1.2.2. РАБОТА С СЕРТИФИКАТАМИ И ХРАНИЛИЩЕМ СЕРТИФИКАТОВ

### Импорт, экспорт и сохранение сертификата

Чтобы произвести экспорт сертификата в строку, следует использовать метод `Export` объекта `Certificate`, чтобы произвести импорт сертификата, следует использовать метод `Import` объекта `Certificate`. Для сохранения сертификата используется метод `Save`.

```
Option Explicit

'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

const CERTIFICATE_PATH = "certificate.cer"

Dim oCertificate
Dim szData_Der

Set oCertificate = CreateObject( "DigtCrypto.Certificate" )

'Загружаем сертификат из файла
oCertificate.Load CERTIFICATE_PATH

'Экспортируем сертификат в DER формате
szData_Der = oCertificate.Export (DER_TYPE)

'Вызываем форму просмотра свойств сертификата
oCertificate.Import CStr(szData_Der)

'Сохраняем сертификат
oCertificate.Save "Cert.cer", BASE64_TYPE, 0

Set oCertificate = Nothing
```

### Получение свойств сертификата

Для получения свойств сертификата необходимо предварительно получить объект `Certificate`, загрузив его из файла или выбрав из хранилища. В данном примере используется выбор сертификата из хранилища.

```
Option Explicit

Const BASE64_TYPE = 0
Const DER_TYPE = 1

Const SYSTEM_STORE_MY = 1
```

```
Const SYSTEM_STORE_CA = 2
Const SYSTEM_STORE_ROOT = 4
Const SYSTEM_STORE_ADDRESS_BOOK = 16

Const LOCAL_MACHINE_STORE = 0
Const CURRENT_USER_STORE = 1

Dim oCertStore, oCerts, oCert
Dim sMess

Set oCertStore = CreateObject("DigtCrypto.CertificateStore")

if true then ' Чтобы выбрать сертификат из хранилища, замените true на false
    ' Вызов формы просмотра хранилища
    Set oCerts = oCertStore.Display(SYSTEM_STORE_MY or SYSTEM_STORE_ADDRESS_BOOK)
else
    ' Откроем хранилище личных сертификатов ("my") текущего пользователя
    oCertStore.Open CURRENT_USER_STORE, "my"
    Set oCerts = oCertStore.Store
end if

' Извлечение первого сертификата из полученного списка (список никак не отсортирован!)
Set oCert = oCerts.Item(0)

' Просмотрим сертификат
oCert.Display

' Получим свойства сертификатов
sMess = sMess + "SerialNumber: " + CStr(oCert.SerialNumber) + Chr(13)
sMess = sMess + "IssuerName: " + CStr(oCert.IssuerName) + Chr(13)
sMess = sMess + "SubjectName: " + CStr(oCert.SubjectName) + Chr(13)
sMess = sMess + "validFrom: " + CStr(oCert.validFrom) + Chr(13)
sMess = sMess + "validTo: " + CStr(oCert.validTo) + Chr(13)
sMess = sMess + "EKU: " + CStr(oCert.EKU) + Chr(13)
sMess = sMess + "ID: " + CStr(oCert.ID) + Chr(13)
sMess = sMess + "KU: " + CStr(oCert.KU) + Chr(13)
sMess = sMess + "ProviderName: " + CStr(oCert.ProviderName) + Chr(13) '
Свойство применимо только для сертификатов, имеющих привязку к контейнеру
sMess = sMess + "ContainerName: " + CStr(oCert.ContainerName) + Chr(13) '
Свойство применимо только для сертификатов, имеющих привязку к контейнеру
sMess = sMess + "CertContext: " + CStr(oCert.CertContext) + Chr(13)
sMess = sMess + "ThumbPrint: " + CStr(oCert.ThumbPrint) + Chr(13)
sMess = sMess + "PublicKeyAlg: " + CStr(oCert.PublicKeyAlg) + Chr(13)
sMess = sMess + "PublicKey(BASE64 TYPE): " +
CStr(oCert.PublicKey(BASE64_TYPE)) + Chr(13)
MsgBox sMess, , "Свойства сертификата"

' Очистим использованные переменные
Set oCert = Nothing
Set oCerts = Nothing
Set oCertStore = Nothing
```

### Установка свойств сертификата

#### Пример установки CryptoAPI-контекста сертификата

```
Option Explicit
Dim oCert, oCert1, sCertContext
Const CERTIFICATE_PATH = "certificate.cer"
Set oCert1 = CreateObject("DigtCrypto.Certificate")
```

```
'Загрузим сертификат из которого получим контекст
oCert1.Load CERTIFICATE_PATH
'Получение свойства CertContext
sCertContext = oCert1.CertContext
'Создадим новый объект Certificate
Set oCert = CreateObject ("DigtCrypto.Certificate")
'Установим ему контекст сертификата
oCert.CertContext = sCertContext
MsgBox CStr(oCert.CertContext)
```

### Пример установки серийного номера и издателя сертификата

```
Option Explicit
Dim oCert, sCertContext
Const CERTIFICATE_PATH = "certificate.cer"
Set oCert1 = CreateObject("DigtCrypto.Certificate")
'Создадим новый объект Certificate
Set oCert = CreateObject ("DigtCrypto.Certificate")
'Установим ему новые свойства сертификата
oCert.SerialNumber = "SerialNumber"
oCert.IssuerName = "CN=ANB, E=anb@mail.ru"
```

## Сравнение сертификатов

Сравнение сертификата производится с помощью метода `isEqual`, в качестве параметра которому передается указатель на объект с сертификатом для сравнения.

```
Option Explicit
Dim ErrorInfo
Dim oCertStore, oCerts, oCert, oCert1
Dim ResBool

Const BASE64_TYPE = 0
Const DER_TYPE = 1
Const CERTIFICATE_PATH = "certificate.cer"
Const CURRENT_USER_STORE = 1
Const LOCAL_MACHINE_STORE = 0

'Выберем первый сертификат из хранилища личных сертификатов
'Открываем хранилище для выбора сертификата
oCertStore.Open CURRENT_USER_STORE, "my"

'Вызов формы просмотра хранилища и получение коллекции сертификатов
Set oCerts = oCertStore.Display (1)

'Получение сертификата из коллекции
Set oCert = oCerts.Item (0)

'Получим второй сертификат загрузкой из файла
Set oCertStore = CreateObject("DigtCrypto.CertificateStore")

'Загрузим сертификат для сравнения
Set oCert1 = CreateObject("DigtCrypto.Certificate")
oCert1.Load CERTIFICATE_PATH

'Посмотрим его
oCert1.Display

'Сравнение сертификатов
ResBool = oCert.isEqual (oCert1)
If ResBool = true then
    MsgBox "Сертификаты равны"
```

```
else
    MsgBox "Сертификаты не равны"
end if

oCertStore.Close
```

### Проверка статуса сертификата

Проверка статуса сертификата производится с помощью метода IsValid с параметром POLICY\_TYPE, который определяет тип применяемой политики использования сертификатов.

```
Option Explicit

Dim oCertStore
Dim oCerts
Dim oCert
Dim status

Const CURRENT_USER_STORE = 1
Const POLICY_TYPE_NONE = 0 'Нет политики использования сертификатов
Const VS_CORRECT = 1
Const VS_UNSUFFICIENT_INFO = 2
Const VS_UNCORRECT = 3
Const VS_INVALID_CERTIFICATE_BLOB = 4
Const VS_CERTIFICATE_TIME_EXPIRIED = 5
Const VS_CERTIFICATE_NO_CHAIN = 6
Const VS_CERTIFICATE_CRL_UPDATING_ERROR = 7
Const VS_LOCAL_CRL_NOT_FOUND = 8
Const VS_CRL_TIME_EXPIRIED = 9
Const VS_CERTIFICATE_IN_CRL = 10
Const VS_CERTIFICATE_IN_LOCAL_CRL = 12
Const VS_CERTIFICATE_CORRECT_BY_LOCAL_CRL = 12
Const VS_CERTIFICATE_USING_RESTRICTED = 13

Set oCertStore = CreateObject("DigtCrypto.CertificateStore")
Set oCerts = CreateObject("DigtCrypto.Certificates")
Set oCert = CreateObject("DigtCrypto.Certificate")
' Открываем хранилище
oCertStore.Open CURRENT_USER_STORE, "my"

' Вызов формы просмотра хранилища и получение сертификата
Set oCerts = oCertStore.Display (1)
Set oCert = oCerts.Item (0)

status = -1
status = oCert.IsValid(POLICY_TYPE_NONE) 'Проверим статус сертификата
Select Case status
    Case VS_CORRECT
        MsgBox "Status: " & "Корректен "
    Case VS_UNSUFFICIENT_INFO
        MsgBox "Status: " & "Статус неизвестен"
    Case VS_UNCORRECT
        MsgBox "Status: " & "Некорректен"
    Case VS_INVALID_CERTIFICATE_BLOB
        MsgBox "Status: " & "Недействительный блок сертификата"
    Case VS_CERTIFICATE_TIME_EXPIRIED
        MsgBox "Status: " & "Время действия сертификата истекло или еще не наступило"
    Case VS_CERTIFICATE_NO_CHAIN
        MsgBox "Status: " & "Невозможно построить цепочку сертификации"
    Case VS_CERTIFICATE_CRL_UPDATING_ERROR
        MsgBox "Status: " & "Ошибка обновления сертификата"
    Case VS_LOCAL_CRL_NOT_FOUND
```



```
MsgBox "Status: " & "Не найден локальный СОС"
Case VS_CRL_TIME_EXPIRIED
MsgBox "Status: " & "Истекло время действия СОС"
Case VS_CERTIFICATE_IN_CRL
MsgBox "Status: " & "Сертификат находится в СОС"
Case VS_CERTIFICATE_IN_LOCAL_CRL
MsgBox "Status: " & "Сертификат находится в локальном СОС"
Case VS_CERTIFICATE_CORRECT_BY_LOCAL_CRL
MsgBox "Status: " & "Сертификат действителен по локальному СОС"
Case VS_CERTIFICATE_USING_RESTRICTED
MsgBox "Status: " & "Использование сертификата ограничено"

End Select
oCertStore.Close
```

### 1.2.3. УПРАВЛЕНИЕ ЗАПРОСАМИ

#### Создание, отправка запроса на сертификат

Создание запроса на сертификат (заполнение шаблона в интерактивном режиме)

```
Option Explicit

Dim oRequestCertificate
Set oRequestCertificate = CreateObject( "DigtCrypto.Request" )

'Вызов мастера создания запроса на сертификат
oRequestCertificate.Display()

set oRequestCertificate = nothing
```

#### Отправка запроса в Удостоверяющий центр

```
Option Explicit

const CA_ADDRESS = "172.17.2.72"
Dim oRequest, sNewCert
Dim oRequestTemplate
Dim szData, sResult
Set oRequest = CreateObject("DigtCrypto.Request")

'Заполним шаблон запроса на сертификат
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mari-el.ru"
oRequestTemplate.L = "Город"
oRequestTemplate.O = "Компания"
oRequestTemplate.OU = "Отдел"
oRequestTemplate.S = "Регион"
oRequestTemplate.ExtendedKeyUsage =
"<keyPurposeId>1.3.6.1.5.5.7.3.1</keyPurposeId>"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true
oRequestTemplate.ShowSendRequestWindow = true
oRequestTemplate.MarkKeysExportable = true
oRequestTemplate.KeyLength = 512
oRequestTemplate.HashAlgorithm = "1.3.14.3.2.26"

'Получим шаблон запроса на сертификат
oRequest.Template = oRequestTemplate
```

```
'Сгенерируем запрос
oRequest.Generate

set oRequestTemplate=nothing

'Получим параметры шаблона
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
Set oRequestTemplate = oRequest.Template

sResult = ""
sResult = sResult + "CN = " + CStr(oRequestTemplate.CN) + Chr(13)
sResult = sResult + "C = " + CStr(oRequestTemplate.C) + Chr(13)
sResult = sResult + "E = " + CStr(oRequestTemplate.E) + Chr(13)
sResult = sResult + "L = " + CStr(oRequestTemplate.L) + Chr(13)
sResult = sResult + "O = " + CStr(oRequestTemplate.O) + Chr(13)
sResult = sResult + "OU = " + CStr(oRequestTemplate.OU) + Chr(13)
sResult = sResult + "S = " + CStr(oRequestTemplate.S) + Chr(13)
sResult = sResult + "ExtendedKeyUsage = " +
CStr(oRequestTemplate.ExtendedKeyUsage) + Chr(13)
sResult = sResult + "CryptoProvider = " +
CStr(oRequestTemplate.CryptoProvider)+ Chr(13)
sResult = sResult + "KeyUsage = " + CStr(oRequestTemplate.KeyUsage)+ Chr(13)
sResult = sResult + "MarkKeysExportable = " +
CStr(oRequestTemplate.MarkKeysExportable)+ Chr(13)
sResult = sResult + "KeyLength = " + CStr(oRequestTemplate.KeyLength)+
Chr(13)
sResult = sResult + "HashAlgorithm = " + CStr(oRequestTemplate.HashAlgorithm)

MsgBox sResult, , "Параметры шаблона запроса на сертификат"

'Загрузим сертификат корневого ЦС, где будет обрабатываться запрос
Dim oIssuerCertificate
Set oIssuerCertificate = CreateObject("DigtCrypto.Certificate")
oIssuerCertificate.Load "root.cer"

'Отправим запрос на обработку в ЦС
sNewCert = oRequest.Send (CA_ADDRESS, oIssuerCertificate, 1)

'Проверим код статуса обработки запроса. Если статус не равен 5, то обработка
прошла успешно
If oRequest.CADisposition <> 5 then
    Set oCert = CreateObject ("DigtCrypto.Certificate")
    oCert.Import CStr(sNewCert)
    oCert.Display
else
    MsgBox "УЦ настроен на отложенную обработку запроса."
end if
```

### Создание запроса по шаблону

Существует возможность настраивать шаблон создания запроса на получение сертификата с помощью групповой политики.

#### ФАЙЛ ШАБЛОНА

Шаблон задает набор атрибутов Отличительного имени владельца сертификата, которые будут отображаться на шаге ввода идентификационной информации, при создании запроса на сертификат. При создании шаблона важно помнить, что порядок следования атрибутов, установленный в шаблоне, определяет и порядок атрибутов в запросе на сертификат. Также шаблон позволяет скрывать поля, устанавливать для атрибутов значения по умолчанию и устанавливать поля недоступными для изменений.

```

<?xml version="1.0" encoding="windows-1251" ?>
<CertificateRequestTemplate>
  <Subject>
    <DirectoryName>
      <RDNSSequence>
        <RDNEntry readonly="false" mandatory="true" hidden="true">
          <!--OID>2.5.4.3</OID-->
          <OID>CN</OID>
          <Name>Имя владельца</Name> <!-- Название поля -->
          <!--Value>Нечаев Андрей Евгеньевич</Value--> <!--
Значение, устанавливаемое по умолчанию -->
          </RDNEntry>
          <RDNEntry hidden="true">
            <OID>2.5.4.11</OID>
            <Name>Подразделение</Name>
            <Value>ИТ</Value>
          </RDNEntry>
          <RDNEntry readonly="true">
            <OID>2.5.4.10</OID>
            <Name>Организация</Name>
            <Value>Цифровые технологии</Value>
          </RDNEntry>
          <RDNEntry>
            <OID>2.5.4.8</OID>
            <Name>Регион</Name>
            <Value></Value>
          </RDNEntry>
          <RDNEntry>
            <OID>2.5.4.7</OID>
            <Name>Город</Name>
            <Value>Moscow</Value>
          </RDNEntry>
          <RDNEntry>
            <OID>2.5.4.6</OID>
            <Name>Страна</Name>
            <Value>RU</Value>
            <Values><Value>EN</Value><Value>RU</Value></Values>
          </RDNEntry>
          <RDNEntry>
            <OID>1.2.840.113549.1.9.1</OID>
            <Name>E-mail</Name>
            <Value>ane@trusted.ru</Value>
          </RDNEntry>
        </RDNSSequence>
      </DirectoryName>
    </Subject>
    <Extensions>
      <Extension>
        <OID>2.5.29.15</OID>
        <Name>Использование ключа</Name>
        <Critical>True</Critical>
        <Value>
          <KeyUsage>
            <DigitalSignature />
            <NonRepudiation />
            <kuKeyEncipherment />
            <DataEncipherment />
            <KeyAgreement />
            <KeyCertSign />
            <CRLSign />
          </KeyUsage>
        </Value>
      </Extension>
    </Extensions>
  </Subject>
</CertificateRequestTemplate>

```

```
</Extension>
<Extension>
  <OID>2.5.29.37</OID>
  <Name>Назначение сертификата</Name>
  <Value>
    <ExtendedKeyUsage>

<KeyPurposeId>1.3.6.1.5.5.7.3.4</KeyPurposeId>

<KeyPurposeId>1.3.6.1.5.5.7.3.2</KeyPurposeId>
    </ExtendedKeyUsage>
  </Value>
</Extension>
</Extensions>

</CertificateRequestTemplate>
```

### ПРИМЕР СОЗДАНИЯ ЗАПРОСА НА СЕРТИФИКАТ С ИСПОЛЬЗОВАНИЕМ ШАБЛОНА

```
Option Explicit
Dim oRequestCertificate, oRequestTemplate

Set oRequestCertificate = CreateObject( "DigtCrypto.Request" )
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
'Загрузка файла шаблона запроса на сертификат из XML файла
oRequestTemplate.LoadXMLTemplate ("certrequest.xml")
'Вызов мастера создания запроса на сертификат
oRequestCertificate.Template = oRequestTemplate
'Вызов мастера генерации запроса на сертификат.
'Параметры идентификационной информации и параметры ключа заполняются данными
из шаблона.
oRequestCertificate.Display()
set oRequestCertificate = nothing
set oRequestTemplate = nothing
```

### Проверка статуса запроса

Если политика УЦ предусматривает отложенную обработку запроса, то в качестве ответа при отправке запроса на обработку (Send) УЦ возвращает идентификатор полученного запроса.

В дальнейшем можно проверить состояние обработки запроса с помощью методов Retry или RetrievePending объекта Requestno.

Метод Retry позволяет получить статус обработки по идентификатору запроса:

```
Option Explicit
const ROOT_ISSUER = "CN=УЦ 2001,OU=Unit,O=Digt,L=Yoshkar-Ola,S=Марий
эл,C=RU,E=ca@mail.ru,"
const CA_ADDRESS = "172.17.2.72"
Dim oRequest, sNewCert, sNewCert1, oCert
Set oRequest = CreateObject("DigtCrypto.Request")

'Создаем шаблон запроса на сертификат
Dim oRequestTemplate
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mari-el.ru"
oRequestTemplate.ExtendedKeyUsage = "1.3.6.1.5.5.7.3.1"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true
```

```
'Получим шаблон запроса на сертификат
oRequest.Template = oRequestTemplate
'Сгенерируем запрос
oRequest.Generate

'Загрузим сертификат корневого ЦС
Dim oIssuerCertificate
Set oIssuerCertificate = CreateObject("DigtCrypto.Certificate")
oIssuerCertificate.Load "root.cer"

'Отправим запрос на обработку в ЦС (ЦС настроен на отложенную обработку за-
проса)
'В качестве ответа получим идентификатор запроса
sNewCert = oRequest.Send (CA_ADDRESS, oIssuerCertificate, 1)

'Здесь должна быть произведена ручная обработка запроса в ЦС
MsgBox "Идентификатор запроса: "+CStr(sNewCert)+ "Выпустите сертификат по
данному запросу в ЦС"

'Проверим статус запроса и получен выпущенный сертификат
sNewCert1 = oRequest.Retry (sNewCert, CA_ADDRESS, oIssuerCertificate, 1)

Set oCert = CreateObject ("DigtCrypto.Certificate")
oCert.Import CStr(sNewCert1)
oCert.Display
```

Метод `RetrievePending` позволяет получить статус обработки по запросу (шаблону сертификата):

```
Option Explicit
'Описание переменных и констант

Dim oCert, oCerts, oCertStore, oCertTemplate, sNewCert1
Dim oRequest

Const CURRENT_USER_STORE = 1

'Откроем хранилище запросов и выберем оттуда запрос, статус которого хотим
проверить
Set oCertStore = CreateObject("DigtCrypto.CertificateStore")
oCertStore.Open CURRENT_USER_STORE, "request"
' Вызов формы просмотра хранилища
Set oCerts = oCertStore.Display (32, 0, 0)
Set oCertTemplate = oCerts.Item (0)

'Запросим статус обработки запроса в УЦ
Set oRequest = CreateObject("DigtCrypto.Request")
sNewCert1 = oRequest.RetrievePending (oCertTemplate, 1)

'Если запрос обработан, то посмотрим полученный сертификат
If oRequest.CADisposition = 3 then
    Set oCert = CreateObject ("DigtCrypto.Certificate")
    oCert.Import CStr(sNewCert1)
    oCert.Display
else
    MsgBox "Запрос не обработан"
end if
```

### Экспорт, импорт и сохранение запроса на сертификат

Чтобы произвести экспорт запроса на сертификат в строку, следует использовать метод Export объекта Request, чтобы произвести импорт запроса, следует использовать метод Import объекта Request. Для сохранения запроса используется метод Save.

```
Option Explicit

Const DER_TYPE = 1
Dim oRequest, oRequestImport
Dim szData

Set oRequest = CreateObject("DigtCrypto.Request")

'Создаем шаблон запроса на сертификат
Dim oRequestTemplate
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )

'Заполним шаблон запроса на сертификат
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mari-el.ru"
oRequestTemplate.ExtendedKeyUsage =
"<keyPurposeId>1.3.6.1.5.5.7.3.1</keyPurposeId>"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true

'Получим шаблон запроса на сертификат
oRequest.Template = oRequestTemplate

'Переходим к генерации запроса"
oRequest.Generate

'Экспортируем запрос на сертификат в строку
szData = oRequest.Export (DER_TYPE)

Set oRequest = nothing

'Импортируем запрос на сертификат из строки
Set oRequestImport = CreateObject( "DigtCrypto.Request" )
oRequestImport.Import (szData)

'Сохраним запрос на сертификат в формате DER
oRequestImport.Save "CertificateReq_der.cer", DER_TYPE
Set oRequestImport = nothing
```

### Создание самоподписанного сертификата

Для создания самоподписанного сертификата используется метод CreateSelfSignedCertificate интерфейса Request. Данный метод производит вызов мастера создания самоподписанного сертификата.

```
Option Explicit

Dim oRequest, oCert
Set oRequest = CreateObject("DigtCrypto.Request")
Set oCert = CreateObject("DigtCrypto.Certificate")

'Создаем шаблон запроса на сертификат
Dim oRequestTemplate
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
```

```
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mari-el.ru"
oRequestTemplate.ExtendedKeyUsage =
"<keyPurposeId>1.3.6.1.5.5.7.3.1</keyPurposeId>"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true

'Получим шаблон запроса на сертификат
oRequest.Template = oRequestTemplat

'Сгенерируем самоподписанный сертификат
Set oCert = oRequest.CreateSelfSignedCertificate

oCert.Display
```

#### 1.2.4. КЛИЕНТ УЦ. УПРАВЛЕНИЕ PKI ЗАПРОСАМИ

##### Вызов мастера сбора данных для создания PKI запроса

Ниже приведен пример создания PKI запроса, сохранения данного запроса в файл, отправки по e-mail и непосредственно в Удостоверяющий центр.

##### СОЗДАНИЕ, ОТПРАВКА, СОХРАНЕНИЕ ЗАПРОСА В ИНТЕРАКТИВНОМ (GUI) РЕЖИМЕ.

В данном примере выполняется вызов мастера сбора данных для создания PKI запроса и последующая генерация запроса. В зависимости от данных, собранных мастером, производится отправка запроса по электронной почте, в УЦ или сохранение запроса в файл.

```
Option Explicit
'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2

'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4

' REQUEST_SENDING_RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6

'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1

' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0
Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore
Dim GenResult, SendResult, SendEmailResult
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
'Вызов мастера сбора данных для создания PKI запроса
```

```
oPKIProfile.CollectData
'Устанавливаем заполненный профиль для создания PKI запроса
oPKIRequest.PKIProfile = oPKIProfile
'Генерация запроса на приостановление сертификата
GenResult = oPKIRequest.Generate
MsgBox GenResult

'Если в профиле установлено сохранение запроса в файл - сохранение запроса в
файл с установленным именем
If oPKIProfile.SaveRequest then
oPKIRequest.Save oPKIProfile.RequestFilename

'Если в профиле установлена отправка запроса по эл. почте - выполнение от-
правки
If oPKIProfile.SendRequestByEmail then
    SendEmailResult = oPKIRequest.SendEmail
    MsgBox SendEmailResult, , "Результат отправки запроса по e-mail"
end if
'Если в профиле установлена отправка запроса в УЦ - выполнение отправки
If oPKIProfile.SendRequest then
    SendResult = oPKIRequest.Send
    MsgBox SendResult, , "Результат отправки запроса в УЦ"
end if
```

### Создание, сохранение и отправка PKI запроса в Silent режиме

Ниже приведен пример создания PKI запроса, сохранения данного запроса в файл, отправки по e-mail и непосредственно в Удостоверяющий центр в неинтерактивном (Silent) режиме.

```
Option Explicit
'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
'REVOCATION_REASON
const REVOCATION_REASON_UNSPECIFIED = 0
const REVOCATION_REASON_KEY_COMPROMISE = 1
const REVOCATION_REASON_CA_COMPROMISE = 2
const REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3
const REVOCATION_REASON_SUPERSEDED = 4
const REVOCATION_REASON_Cease_of_operation = 5
const REVOCATION_REASON_CERTIFICATE_HOLD = 6
'REQUEST_SENDING_RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6
'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1
```



```
' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0
Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
Set oCertStore = CreateObject ("DigtCrypto.CertificateStore")
'Для выполнения приостановления сертификата необходимо использовать
3 сертификата:
'приостанавливаемый сертификат, сертификат подписи запроса, сертификат
для установки ssl соединения.
'В данном примере в роли этих сертификатов будет использоваться
один и тот же сертификат.
'Примечание: данный сертификат должен иметь назначение "Аутентификация
клиента".
'Для выбора сертификата открываем форму выбора сертификата из личного
хранилища
'Открытие хранилища
oCertStore.Open CURRENT_USER_STORE, "my"
' Вызов формы просмотра хранилища, получение коллекции с единственным
выбранным сертификатом
Set oCerts = oCertStore.Display (1)
'Получение сертификата из коллекции и установка его в профиль
в качестве приостанавливаемого сертификата, сертификата подписи,
сертификата установки ssl соединения
Set oSSLCertificate = oCerts.Item (0)
Set oOperationCertificate = oCerts.Item (0)
Set oSignCertificate = oCerts.Item (0)
oPKIProfile.OperationCertificate = oOperationCertificate
oPKIProfile.SignCertificate = oSignCertificate
oPKIProfile.SignatureCertificatePin = "1"
'Установка типа запроса - приостановление сертификата, Типа
- КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_SUSPENDING
oPKIProfile.CAType = CA_TYPE_CRYPTOPRO
oPKIProfile.Comment = "Приостановление"
oPKIProfile.RevocationReason = REVOCATION_REASON_CERTIFICATE_HOLD
oPKIProfile.CertificateHoldDuration = "1-0-0-0-0-0"
'Установка параметров сохранения запроса
oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = BASE64
oPKIProfile.RequestFilename = "Hold.req"
'Установка параметров отправки запроса в КриптоПРО УЦ
oPKIProfile.SendRequest = true
oPKIProfile.ServiceURL = "https://ca-server/ra/ra.wsdl"
oPKIProfile.SSLCertificate = oSSLCertificate
'Установка параметров отправки запроса по E-Mail
oPKIProfile.SendRequestByEmail = true
oPKIProfile.EmailMessage = "Примите запрос на приостановление"
oPKIProfile.EmailRecipient = "someone@ca.ru"
oPKIProfile.EmailSubject = "Запрос на приостановление"
'Установка заполненного профиля для создания запроса
oPKIRequest.PKIProfile = oPKIProfile
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate
MsgBox GenResult
'Если в профиле установлено сохранение запроса в файл - сохранение запроса в
файл с установленным именем
If oPKIProfile.SaveRequest then oPKIRequest.Save oPKIProfile.RequestFilename,
, oPKIProfile.RequestFilenameFormat

'Если в профиле установлена отправка запроса по эл. почте - выполнение от-
правки
```

```
If oPKIProfile.SendRequestByEmail then
    SendEmailResult = oPKIRequest.SendEmail
    MsgBox SendEmailResult, , "Результат отправки запроса по e-mail"
end if

'Если в профиле установлена отправка запроса в УЦ - выполнение отправки
If oPKIProfile.SendRequest then
    SendResult = oPKIRequest.Send
    MsgBox SendResult, , "Результат отправки запроса в УЦ"
end if
```

### Создание запроса на отзыв сертификата в silent режиме

Ниже приведен пример создания запроса на отзыв, сохранения данного запроса в файл в неинтерактивном (Silent) режиме.

```
Option Explicit

'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
'REVOCATION_REASON
const REVOCATION_REASON_UNSPECIFIED = 0
const REVOCATION_REASON_KEY_COMPROMISE = 1
const REVOCATION_REASON_CA_COMPROMISE = 2
const REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3
const REVOCATION_REASON_SUPERSEDED = 4
const REVOCATION_REASON_Cease_of_operation = 5
const REVOCATION_REASON_CERTIFICATE_HOLD = 6
'REQUEST_SENDING_RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6
'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1
const XML = 2
' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0

Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
Set oCertStore = CreateObject ("DigtCrypto.CertificateStore")

'Для создания запроса на отзыв сертификата необходимо использовать 2 сертификата:
'отзываемый сертификат, сертификат подписи запроса.
'В данном примере в роли этих сертификатов будет использоваться один и тот же
```

сертификат.

```
'Для выбора сертификата открываем форму выбора сертификата из личного храни-
лища
'Открытие хранилища
oCertStore.Open CURRENT_USER_STORE, "my"
'Вызов формы просмотра хранилища, получение коллекции с единственным выбран-
ным сертификатом
Set oCerts = oCertStore.Display (1)

'Получение сертификата из коллекции и установка его в профиль в качестве при-
останавливаемого сертификата, сертификата подписи.
Set oOperationCertificate = oCerts.Item (0)
Set oSignCertificate = oCerts.Item (0)
oPKIProfile.OperationCertificate = oOperationCertificate
oPKIProfile.SignCertificate = oSignCertificate
oPKIProfile.SignatureCertificatePin = "1"

'Установка типа запроса - приостановление сертификата, Типа УЦ - КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_SUSPENDING
oPKIProfile.CAType = CA_TYPE_CRYPTOPRO
oPKIProfile.Comment = "Отзыв"
oPKIProfile.RevocationReason = REVOCATION_REASON_KEY_COMPROMISE
'Установка параметров сохранения запроса
'oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = DER
oPKIProfile.RequestFilename = "Revoke.req"

oPKIRequest.PKIProfile = oPKIProfile

'Генерация запроса
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate
MsgBox GenResult

'Сохранение запроса в файл с установленным именем
oPKIRequest.Save oPKIProfile.RequestFilename,
oPKIProfile.RequestFilenameFormat
```

### Создание запроса на возобновление сертификата в silent режиме

Ниже приведен пример создания запроса на возобновление, сохранения данного за-  
проса в файл в неинтерактивном (Silent) режиме.

```
Option Explicit

'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
const REQUEST_TYPE_UPDATE_CERT = 8
const REQUEST_TYPE_UPDATE_KEY = 16
const REQUEST_TYPE_CERT_PKCS10 = 32
const REQUEST_TYPE_SELF_SIGNED_CERT = 64
const REQUEST_TYPE_INFO = &H40000000

'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
```

```
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
'REVOCATION_REASON
const REVOCATION_REASON_UNSPECIFIED = 0
const REVOCATION_REASON_KEY_COMPROMISE = 1
const REVOCATION_REASON_CA_COMPROMISE = 2
const REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3
const REVOCATION_REASON_SUPERSEDED = 4
const REVOCATION_REASON_Cease_OF_OPERATION = 5
const REVOCATION_REASON_CERTIFICATE_HOLD = 6
' REQUEST_SENDING_RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6
'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1
const XML = 2
' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0

Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
Set oCertStore = CreateObject ("DigtCrypto.CertificateStore")

'Для создания запроса на отзыв сертификата необходимо использовать 2 сертификата:
'отзываемый сертификат, сертификат подписи запроса.
'В данном примере в роли этих сертификатов будет использоваться один и тот же сертификат.

'Для выбора сертификата открываем форму выбора сертификата из личного хранилища
'Открытие хранилища
oCertStore.Open CURRENT_USER_STORE, "my"
' Вызов формы просмотра хранилища, получение коллекции с единственным выбранным сертификатом
Set oCerts = oCertStore.Display (1)

'Получение сертификата из коллекции и установка его в профиль в качестве приостанавливаемого сертификата, сертификата подписи.
Set oOperationCertificate = oCerts.Item (0)
Set oSignCertificate = oCerts.Item (0)
oPKIProfile.OperationCertificate = oOperationCertificate
oPKIProfile.SignCertificate = oSignCertificate
oPKIProfile.SignatureCertificatePin = "1"

'Установка типа запроса - приостановление сертификата, Типа УЦ - КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_RESUMING
oPKIProfile.CAType = CA_TYPE_CRYPTOPRO
oPKIProfile.Comment = "Возобновление"
'Установка параметров сохранения запроса
oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = DER
oPKIProfile.RequestFilename = "Revoke.req"

oPKIRequest.PKIProfile = oPKIProfile

'Генерация запроса
```

```
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate
MsgBox GenResult

'Сохранение запроса в файл с установленным именем
oPKIRequest.Save oPKIProfile.RequestFilename,
oPKIProfile.RequestFilenameFormat
```

### Создание запроса на обновление ключа сертификата в silent режиме

Ниже приведен пример создания запроса на обновление ключа сертификата для КриптоПро УЦ, сохранения данного запроса в файл в не интерактивном (Silent) режиме.

Отличие запроса на обновление ключа сертификата предназначенного для КриптоПро УЦ от запроса для Microsoft CA состоит в том, что в случае КриптоПро УЦ запрос должен быть подписан. Запрос на обновление ключа в Microsoft CA ничем не отличается от создания запроса на сертификат.

```
Option Explicit

'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
const REQUEST_TYPE_UPDATE_CERT = 8
const REQUEST_TYPE_UPDATE_KEY = 16
const REQUEST_TYPE_CERT_PKCS10 = 32
const REQUEST_TYPE_SELF_SIGNED_CERT = 64
const REQUEST_TYPE_INFO = &H40000000
'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
'REVOCATION_REASON
const REVOCATION_REASON_UNSPECIFIED = 0
const REVOCATION_REASON_KEY_COMPROMISE = 1
const REVOCATION_REASON_CA_COMPROMISE = 2
const REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3
const REVOCATION_REASON_SUPERSEDED = 4
const REVOCATION_REASON_Cease_of_operation = 5
const REVOCATION_REASON_CERTIFICATE_HOLD = 6
' REQUEST SENDING RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6
'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1
const XML = 2
' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0

Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore, oRequestTemplate
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
```

```
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
Set oCertStore = CreateObject ("DigtCrypto.CertificateStore")

'Для выполнения обновления ключа сертификата в КриптоПро УЦ необходимо ис-
пользовать 2 сертификата:
'сертификат, ключи которого обновляются, сертификат подписи запроса.
'В данном примере в роли этих сертификатов будет использоваться один и тот же
сертификат.
'Для выбора сертификата открываем форму выбора сертификата из личного храни-
лища
'Открытие хранилища
oCertStore.Open CURRENT_USER_STORE, "my"

' Вызов формы просмотра хранилища, получение коллекции с единственным выбран-
ным сертификатом
Set oCerts = oCertStore.Display (1)
'Получение сертификата из коллекции и установка его в профиль в качестве при-
останавливаемого сертификата, сертификата подписи, сертификата установки ssl
соединения
Set oOperationCertificate = oCerts.Item (0)
Set oSignCertificate = oCerts.Item (0)

'Заполнение шаблона запроса на сертификат
'Получение данных из выбранного сертификата
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
oRequestTemplate.Certificate = oOperationCertificate
oRequestTemplate.ExtendedKeyUsage = oOperationCertificate.EKU
oRequestTemplate.CryptoProvider = oOperationCertificate.ProviderName
oRequestTemplate.KeyUsage = oOperationCertificate.KU
oRequestTemplate.CreateNewKeySet = true
oRequestTemplate.MarkKeysExportable = true
'Необходимо задавать уникальное имя контейнера
oRequestTemplate.Keyset = "NewContainer"

'Установим шаблон запроса на сертификат в профиль для формирования PKI запро-
са
oPKIProfile.RequestTemplate = oRequestTemplate
oPKIProfile.SignCertificate = oSignCertificate
oPKIProfile.SignatureCertificatePin = "1"

'Установка типа запроса - приостановление сертификата, Типа УЦ - КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_UPDATE_KEY
oPKIProfile.CAType = CA_TYPE_CRYPTOPRO

'Установка параметров сохранения запроса
oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = DER
oPKIProfile.RequestFilename = "key_update.req"

'Установка профиля создания PKI запроса
oPKIRequest.PKIProfile = oPKIProfile

'Генерация запроса
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate

'Сохранение запроса на обновление ключа сертификата
oPKIRequest.Save oPKIProfile.RequestFilename,
oPKIProfile.RequestFilenameFormat
```

**Создание запроса на сертификат в silent режиме**

Ниже приведен пример создания запроса на сертификат для КриптоПро УЦ, сохранения данного запроса в файл в неинтерактивном (Silent) режиме.

Примечание: Запрос на первый сертификат для КриптоПро УЦ не подписывается.

Option Explicit

```
'CA_TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
const REQUEST_TYPE_UPDATE_CERT = 8
const REQUEST_TYPE_UPDATE_KEY = 16
const REQUEST_TYPE_CERT_PKCS10 = 32
const REQUEST_TYPE_SELF_SIGNED_CERT = 64
const REQUEST_TYPE_INFO = &H40000000
'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
'REVOCATION_REASON
const REVOCATION_REASON_UNSPECIFIED = 0
const REVOCATION_REASON_KEY_COMPROMISE = 1
const REVOCATION_REASON_CA_COMPROMISE = 2
const REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3
const REVOCATION_REASON_SUPERSEDED = 4
const REVOCATION_REASON_Cease_of_operation = 5
const REVOCATION_REASON_CERTIFICATE_HOLD = 6
' REQUEST_SENDING_RESULT
const REQUEST_SENDING_SUCCESS = 1
const REQUEST_SENDING_REQUEST_MISSING = 2
const REQUEST_SENDING_REQUEST_SOAP_MISSING = 3
const REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4
const REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5
const REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6
'PROFILEEXITFORMAT
const BASE64 = 0
const DER = 1
const XML = 2
' Тип хранилищ
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0
```

```
Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore, oRequestTemplate
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")
```

```
'Заполнение шаблона запроса на сертификат
'Получение данных из выбранного сертификата
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mail.ru"
oRequestTemplate.L = "Город"
oRequestTemplate.O = "Компания"
oRequestTemplate.OU = "Отдел"
oRequestTemplate.S = "Регион"
```

```
oRequestTemplate.ExtendedKeyUsage =
"<keyPurposeId>1.3.6.1.5.5.7.3.1</keyPurposeId>"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true
oRequestTemplate.ShowSendRequestWindow = true
oRequestTemplate.MarkKeysExportable = true
oRequestTemplate.KeyLength = 512
'Необходимо задавать уникальное имя контейнера вручную или вовсе не задавать
его, тогда имя будет сгенерировано автоматически
oRequestTemplate.Keyset = "NewContainer"

'Установим шаблон запроса на сертификат в профиль для формирования PKI за-
са
oPKIProfile.RequestTemplate = oRequestTemplate

'Установка типа запроса - приостановление сертификата, Типа УЦ - КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_CERT_PKCS10
oPKIProfile.CAType = CA_TYPE_CRYPTOPRO

'Установка параметров сохранения запроса
oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = DER
oPKIProfile.RequestFilename = "cert_req"

'Установка профиля создания PKI запроса
oPKIRequest.PKIProfile = oPKIProfile

'Генерация запроса
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate

'Сохранение запроса на сертификат
oPKIRequest.Save oPKIProfile.RequestFilename,
oPKIProfile.RequestFilenameFormat
```

### Создание самоподписанный сертификат в silent режиме

Ниже приведен пример создания самоподписанного сертификата, сохранения его в файл в неинтерактивном (Silent) режиме.

```
Option Explicit

'CA TYPE
const CA_TYPE_CMS = 1
const CA_TYPE_CRYPTOPRO = 2
'REQUEST_TYPE
const REQUEST_TYPE_REVOCATION = 1
const REQUEST_TYPE_SUSPENDING = 2
const REQUEST_TYPE_RESUMING = 4
const REQUEST_TYPE_UPDATE_CERT = 8
const REQUEST_TYPE_UPDATE_KEY = 16
const REQUEST_TYPE_CERT_PKCS10 = 32
const REQUEST_TYPE_SELF_SIGNED_CERT = 64
const REQUEST_TYPE_INFO = &H40000000
'REQUEST_GENERATION_RESULT
const REQUEST_GENERATION_SUCCESS = 1
const REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2
const REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3
const REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4
''PROFILEEXITFORMAT
const BASE64 = 0
```



```
const DER = 1
const XML = 2

Dim oPKIProfile, oPKIRequest, oOperationCertificate, oSignCertificate,
oSSLCertificate, oCerts, oCertStore, oRequestTemplate
Set oPKIProfile = CreateObject ("DigtCrypto.PKIProfile")
Set oPKIRequest = CreateObject ("DigtCrypto.PKIRequest")

'Заполнение шаблона запроса на сертификат
'Получение данных из выбранного сертификата
Set oRequestTemplate = CreateObject( "DigtCrypto.CRequestTemplate" )
oRequestTemplate.CN = "Петр Иванов"
oRequestTemplate.C = "RU"
oRequestTemplate.E = "pivanov@mail.ru"
oRequestTemplate.L = "Город"
oRequestTemplate.O = "Компания"
oRequestTemplate.OU = "Отдел"
oRequestTemplate.S = "Регион"
oRequestTemplate.ExtendedKeyUsage =
"<keyPurposeId>1.3.6.1.5.5.7.3.1</keyPurposeId>"
oRequestTemplate.CryptoProvider = "Microsoft Base Cryptographic Provider
v1.0"
oRequestTemplate.KeyUsage = 1
oRequestTemplate.CreateNewKeySet = true
oRequestTemplate.ShowSendRequestWindow = true
oRequestTemplate.MarkKeysExportable = true
oRequestTemplate.KeyLength = 512
'Необходимо задавать уникальное имя контейнера вручную или вовсе не задавать
его, тогда имя будет сгенерировано автоматически
'oRequestTemplate.Keyset = "NewContainer"

'Установим шаблон запроса на сертификат в профиль для формирования PKI запро-
са
oPKIProfile.RequestTemplate = oRequestTemplate

'Установка типа запроса - приостановление сертификата, Типа УЦ - КриптоПРО УЦ
oPKIProfile.RequestType = REQUEST_TYPE_SELF_SIGNED_CERT

'Установка параметров сохранения
oPKIProfile.SaveRequest = true
oPKIProfile.RequestFilenameFormat = DER
oPKIProfile.RequestFilename = "selfsigned_cert.cer"

'Установка профиля создания PKI запроса
oPKIRequest.PKIProfile = oPKIProfile

'Генерация запроса
Dim GenResult, SendEmailResult, SendResult
GenResult = oPKIRequest.Generate

'Сохранение сертификата
oPKIRequest.Save oPKIProfile.RequestFilename,
oPKIProfile.RequestFilenameFormat
```

### 1.2.5. РАБОТА С ПРОФИЛЯМИ

#### Создание нового профиля и установка его в хранилище

Данный пример позволяет создать новый профиль, заполнить его параметры и со-хранить новый профиль в хранилище профилей.

```
Option Explicit
const CERT_FOR_ENCRYPT = 0
```

```
const CERT_FOR_DECRYPT = 1
const CERT_FOR_SIGN = 2
const BASE64 = 0
const DER = 0
const REGISTRY_STORE = 0
const CURRENT_USER_STORE = 1
const LOCAL_MACHINE_STORE = 0
' enum CERTIFICATE_VERIFY_LEVEL
const CERTIFICATE_VERIFY_LOCAL_CRL = 1
const CERTIFICATE_VERIFY_ONLINE_CRL = 2
'const CERTIFICATE_VERIFY_OCSP = 3 ' reserved
'enum SILENT_LEVEL
const SILENT_LEVEL_SERVER = 1
const SILENT_LEVEL_REQUIRED = 2
const SILENT_LEVEL_WINDOWS = 3
const SILENT_LEVEL_INTERACTIVE = 4
' CERTIFICATE_PUPROSE
const CERTIFICATE_FOR_NEW_SIGNATURE = 1
const CERTIFICATE_FOR_SIGNATURE_VERIFYING = 2
const CERTIFICATE_FOR_ENCIPHER = 3
const CERTIFICATE_FOR_DECIPHER = 4
' CERTIFICATE_WORKABILITY
const CERTIFICATE_ALLOWED = 1
const CERTIFICATE_RESTRICTED = 2
const CERTIFICATE_UNKNOWN = 3

const CERTIFICATE_PATH = "certificate.cer"

'Устанавливаю значения параметров профиля, сохраняю, затем считываю и сравни-
ваю

Dim objCertificate, oOIDs, oOID, oPolicyProfile
Dim lWorkability, i
dim sResult
Dim objProfile
Set objProfile = CreateObject("DigtCrypto.Profile")
'Здесь установим параметры профиля
objProfile.Provider = "Provider"
objProfile.HashAlg = "Hashalg"
objProfile.EncAlg = "EncAlg"
objProfile.SignFinalWindow = false
objProfile.SignStartWindow = false
objProfile.SignFormatWindow = false
objProfile.EncryptFinalWindow = true
objProfile.EncryptStartWindow = true
objProfile.EncryptFormatWindow = true
objProfile.SignIncludeBase64Headers = true
objProfile.EncryptIncludeBase64Headers = true
objProfile.SignExitFormat = DER
objProfile.EncryptExitFormat = DER
objProfile.UseCertificateForEncrypt = true
objProfile.Detach = true
objProfile.DefaultPath = "def/path"
objProfile.IncludeSignatureTime = true
objProfile.ShowStatusReport = true
objProfile.EncryptToSenderAddress = true
objProfile.Resource = "Resource"
objProfile.SignPropertiesWindow = true
objProfile.SignCertificateWindow = true
objProfile.EncryptPropertiesWindow = true
objProfile.EncryptRecipientsWindow = true
objProfile.Name = "Test profile"
objProfile.Comment = "Comment"
objProfile.SilentLevel = SILENT_LEVEL_SERVER
```

```
'Установим политику использования сертификатов

'Создадим новый OID
Set oOID = CreateObject ("DigtCrypto.OID")
oOID.FriendlyName = "Аутентификация пользователя"
oOID.Value = "1.3.6.1.5.5.7.3.2"
MsgBox "FriendlyName = " + CStr(oOID.FriendlyName) + Chr(13) + "Value = " +
CStr(oOID.Value), , "Установленное значение"

'Добавим этот OID в коллекцию OIDs
Set oOIDs = CreateObject ("DigtCrypto.OIDs")
oOIDs.Add oOID

'Заполним политику SignatureRequiredOIDs
Set oPolicyProfile = CreateObject ("DigtCrypto.PolicyProfile")
oPolicyProfile.SignatureRequiredOIDs = oOIDs
Set oOIDs = nothing

'Заполним политику использования сертификата в профиле
objProfile.Policy = oPolicyProfile

dim oCertificate
Set oCertificate = CreateObject ("DigtCrypto.Certificate")
Dim oRecipients
Set oRecipients = CreateObject ("DigtCrypto.Certificates")
dim oCertificateStore
Set oCertificateStore = CreateObject ("DigtCrypto.CertificateStore")

'Установим сертификаты подписи в профиль (сертификаты шифрования и расшиф-
рования устанавливаются таким же образом, с соответствующими параметрами)
oCertificate.Load CERTIFICATE_PATH
'Проверим, удовлетворяет ли сертификат политике
lWorkability = -1
lWorkability = objProfile.CheckCertificate (oCertificate, CERTIFI-
CATE_FOR_NEW_SIGNATURE)
If lWorkability = 1 then
    MsgBox oCertificate.SerialNumber
    objProfile.SetCertificate CERT_FOR_SIGN, "1", oCertificate
    oRecipients.Add oCertificate ' формирую коллекцию сертификатов получа-
телей
else
    MsgBox "Сертификат не подходит по политике использования"
end if

'Установим коллекцию сертификатов получателей в профиль
objProfile.Recipients = oRecipients

'Установим коллекцию сертификатов для проверки статуса
objProfile.SetVerifiedCertificates CERTIFICATE_VERIFY_ONLINE_CRL ,
oRecipients

'Запомним идентификатор нашего нового профиля
dim sProfileID : sProfileID = objProfile.ID

'Сохраним созданный профиль в хранилище
Dim objProfiles
Set objProfiles = CreateObject ("DigtCrypto.Profiles")
Dim objProfileStore
Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")

objProfileStore.Open(REGISTRY_STORE) ' Открываем хранилище профилей в реестре
компьютера
```

```
Set objProfiles = objProfileStore.Store ' Получаем коллекцию профилей из хранилища
objProfiles.Add (objProfile) ' Добавляет профиль в коллекцию профилей
objProfileStore.Store = objProfiles ' Устанавливаем коллекцию профилей в хранилище
objProfileStore.Save ' Сохраняем хранилище профилей
objProfileStore.Close ' Закрываем хранилище

set objProfile = nothing
set objProfileStore = nothing
set objProfiles = nothing
```

### Получение параметров профиля

Данный пример описывает получение некоторых параметров профиля по умолчанию.

```
Option Explicit
const REGISTRY_STORE = 0
'CERT_TYPE
const CERT_FOR_SIGN = 2

Dim objCertificate
Dim lWorkability, i
dim sResult
Dim objProfile

Dim objProfiles
Set objProfiles = CreateObject ("DigtCrypto.Profiles")
Dim objProfileStore
Set objProfileStore = CreateObject ("DigtCrypto.ProfileStore")
objProfileStore.Open (REGISTRY_STORE)
Set objProfiles = objProfileStore.Store

'Получим профиль по умолчанию
Set objProfile = objProfiles.DefaultProfile

'получим некоторые параметры профиля
GetProfileProps = ""
GetProfileProps = GetProfileProps + "ID=" + objProfile.ID + Chr(13)
GetProfileProps = GetProfileProps + "SignStartWindow=" + CStr(objProfile.SignStartWindow) + Chr(13)
GetProfileProps = GetProfileProps + "Name=" + objProfile.Name + Chr(13)
GetProfileProps = GetProfileProps + "Comment=" + objProfile.Comment + Chr(13)
GetProfileProps = GetProfileProps + "SilentLevel=" + CSTR(objProfile.SilentLevel) + Chr(13)

'Получение сертификата выполнения операции на примере сертификата подписи
set objCertificate = objProfile.GetCertificate (CERT_FOR_SIGN)
GetProfileProps = GetProfileProps + "Сертификат подписи=" + objCertificate.SerialNumber + Chr(13)
'Получение пин кода сертификата подписи
objProfile.GetPin (CERT_FOR_SIGN)
GetProfileProps = GetProfileProps + "Пин сертификата подписи=" + objProfile.GetPin (CERT_FOR_SIGN) + Chr(13)

'Получение коллекции сертификатов на примере получателей шифрованного сообщения
dim objCertificates
set objCertificates = CreateObject ("DigtCrypto.Certificates")
Set objCertificates = objProfile.Recipients
For i = 0 to (objCertificates.Count-1) 'Число сертификатов получателей
Set oCertificate = objCertificates.Item (i)
```

```
GetProfileProps = GetProfileProps + "Сертификат получателя " + CStr(i+1) +  
"=" + oCertificate.SerialNumber + Chr(13)  
Next  
MsgBox GetProfileProps  
  
Set objProfile = nothing  
set objProfileStore = nothing  
set objProfiles = nothing
```

### Удаление профиля

Компонент удаляет выбранный пользователем профиль в списке из хранилища профилей.

```
Option Explicit  
const REGISTRY_STORE = 0  
  
Dim objProfileStore  
Dim objProfiles  
Dim objProfile, objProfileFind  
Dim sErrorMessage  
  
Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")  
Set objProfiles = CreateObject("DigtCrypto.Profiles")  
Set objProfile = CreateObject("DigtCrypto.Profile")  
objProfile.Name = "Добавленный"  
  
objProfileStore.Open(REGISTRY_STORE)  
  
'Метод [propget] Store. Получаем коллекцию профилей  
Set objProfiles = objProfileStore.Store  
objProfiles.Remove 3  
objProfileStore.Store = objProfiles  
objProfileStore.Save  
objProfileStore.Close  
Set objProfiles = nothing
```

### Пример вызова менеджера профилей

```
Option Explicit  
  
const REGISTRY_STORE = 0  
  
Dim oProfileStore  
Set oProfileStore = CreateObject( "DigtCrypto.ProfileStore" )  
oProfileStore.Open( REGISTRY_STORE )  
  
oProfileStore.Display()
```

### Создание копии профиля

Чтобы получить копию текущего профиля, следует использовать метод Clone объекта Profile.

```
Dim objProfile, objProfileClone  
Set objProfile = CreateObject("DigtCrypto.Profile")  
objProfile.Name = "Тестовый профиль"  
Set objProfileClone = objProfile.Clone ()  
WScript.Echo "Имя клонированного профиля: "&objProfileClone.Name&Chr(13)&_
```

```
"Имя исходного профиля :  
"&objProfile.Name+Chr(13)
```

### Загрузка хранилища профилей с Веб-сервиса

Операция доступна в если лицензия КриптоАРМ включает использование модуля Клиент Веб-сервиса. При этом необходимо, чтобы пользователь, выполняющий загрузку профилей с сервера Веб-сервиса, был зарегистрирован в Сервисе.

```
Option explicit  
'Объявление переменных  
Const LOG_LEVEL_DISABLED = 0 'Логирование отключено  
Const LOG_LEVEL_ERROR = 1 'Ошибки  
Const LOG_LEVEL_WARN = 2 'Предупреждения  
Const LOG_LEVEL_INFO = 3 'Информация  
Const LOG_LEVEL_DEBUG = 4 'Отладка  
Const LOG_LEVEL_TRACE = 5 'Трассировка  
Const strPattern = "log4cplus.appender.toDSL.layout.ConversionPattern=%d  
[%5t] [%p] [%c] [%x] - %m%n"  
Dim oUtil , sLoggerID, strLoggerStatement  
Set oUtil = CreateObject ("DigtCrypto.Util")  
sLoggerID = oUtil.StartupLogger (strPattern, LOG_LEVEL_TRACE, 500*1024, "DSS"  
, 0)  
MsgBox sLoggerID, , "sLoggerID"  
strLoggerStatement = oUtil.GetLoggerStatements (sLoggerID )  
MsgBox strLoggerStatement  
oUtil.ShutdownLogger sLoggerID
```

### 1.2.6. РАБОТА С ПОДПИСЯМИ

#### Проверка статуса подписи

Чтобы проверить статус подписи необходимо воспользоваться методом Verify объекта Signature.

```
Option Explicit  
const CERT_AND_SIGN = 0  
const SIGN_ONLY = 1  
Dim i  
Dim oPKCS7Message  
Dim oSignatures  
Dim oSignature  
Const DT_SIGNED_DATA = 2  
  
Set oPKCS7Message = CreateObject( "DigtCrypto.PKCS7Message" )  
oPKCS7Message.Load DT_SIGNED_DATA, "test.txt.pem", ""  
'Получим коллекцию ЭЦП  
Set oSignatures = oPKCS7Message.Signatures  
  
'oPKCS7Message.Display  
'Получим количество подписей в коллекции  
Dim n : n = oSignatures.Count  
'MsgBox n, , "Количество подписей"  
For i= 0 to n-1  
    'Получим указатель на элемент коллекции подписей  
    Set oSignature = oSignatures.Item (i)  
    dim Status  
    Status = oSignature.Verify (SIGN_ONLY, "test.txt")  
    MsgBox "Status: " & CStr (Status)  
next
```

**Получение свойств подписи**

```
' VBScript source code
Option Explicit
Dim oPKCS7Message
Dim i, sResult
Dim oSignatures
Dim oSignature, oSign
Dim oCertificate

Const DT_SIGNED_DATA = 2

Set oPKCS7Message = CreateObject( "DigtCrypto.PKCS7Message" )
'Загрузка присоединенной подписи
oPKCS7Message.Load DT_SIGNED_DATA, "test.txt.pem", ""

'Получим коллекцию ЭЦП
Set oSignatures = oPKCS7Message.Signatures
'Получим количество подписей
Dim n : n = oSignatures.Count
dim oSigns
Set oSigns = CreateObject ("DigtCrypto.Signatures")

for i = 0 to n-1
    'Получим свойства для каждой подписи
    Set oSignature = oSignatures.Item (i)
    'Просмотрим подпись
    oSignature.Display
    'Получаем индекс подписчика
    Dim uLongSignerIndex : uLongSignerIndex = oSignature.SignerIndex
    sResult = sResult + "Индекс подписчика  " + CStr(uLongSignerIndex) +
Chr(13)

    'Получаем комментарий подписи
    Dim sComments: sComments = oSignature.Comments
    sResult = sResult + "Комментарий подписи  " + CStr(sComments) +
Chr(13)

    'Получаем время подписи
    Dim sSigningTime: sSigningTime = oSignature.SigningTime
    sResult = sResult + "Время подписи  " + CStr(sSigningTime) + Chr(13)

    'Получаем идентификатор ресурса подписи
    Dim sResource : sResource = oSignature.Resource
    sResult = sResult + "Идентификатор ресурса подписи  " +
CStr(sResource) + Chr(13)

    'Получаем хэш алгоритм подписи
    Dim sHashAlg : sHashAlg = oSignature.HashAlg
    sResult = sResult + "Хэш алгоритм подписи  " + CStr(sHashAlg) +
Chr(13)

    'Получаем алгоритм подписи ЭЦП
    Dim sHashEncAlg : sHashEncAlg = oSignature.HashEncAlg
    sResult = sResult + "Алгоритм подписи ЭЦП  " + CStr(sHashEncAlg) +
Chr(13)

    'Получаем тип эцп: 1-отсоединенная, 0-присоединенная
    Dim sDetached : sDetached = oSignature.Detached
    sResult = sResult + "Тип эцп  " + CStr(sDetached) + Chr(13)

    'Получаем номер версии протокола CMS
    Dim lCMSVersion : lCMSVersion = oSignature.CMSVersion
    sResult = sResult + "Номер версии протокола CMS  " + CStr(lCMSVersion)
+ Chr(13)
```

```
'Получаем Тип Содержимого сообщения
Dim sContentType: sContentType = oSignature.ContentType
sResult = sResult + "Тип содержимого сообщения  " + CStr(sContentType)
+ Chr(13)
MsgBox sResult, , "Свойства подписи"

'Получаем указатель на сертификат подписи
Set oCertificate = oSignature.Certificate
oCertificate.Display
sResult = ""

next
```

### Получение коллекции заверяющих подписей

Получить коллекцию заверяющих подписей можно с помощью метода Cosignature объекта Signature.

```
' VBScript source code
Option Explicit
Dim oPKCS7Message
Dim i, n1
Dim oSignatures
Dim oSignature, oSign
Dim oCertificate

Const DT_SIGNED_DATA = 2

Set oPKCS7Message = CreateObject( "DigtCrypto.PKCS7Message" )
'Загрузка присоединенной подписи
oPKCS7Message.Load DT_SIGNED_DATA, "test.txt.pem", ""

'Получим коллекцию ЭЦП
Set oSignatures = oPKCS7Message.Signatures
'Получим количество подписей
Dim n : n = oSignatures.Count
dim oSigns
Set oSigns = CreateObject ("DigtCrypto.Signatures")

for i = 0 to n-1
    'Получим свойства для каждой подписи
    Set oSignature = oSignatures.Item (i)

    'Получим подписи коллекцию встречных (заверяющих) подписей
    set oSigns = oSignature.Cosignature
    n1 = n1 + "Количество встречных подписей у подписи № " + CStr(i+1) +
    "= " + CStr(oSigns.Count) + Chr(13)
    MsgBox CStr(n1)
next
```

### Подпись содержимого формы

В данном примере производится подпись содержимого текстового поля формы:

```
<html>
<head>
<title>Signing test</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<script language="JavaScript">
<!--
function OnOK()
```



```
{
    var form = document.form;

    var str = form.text_to_sign.value;
    if (str == "")
    {
        alert("No text found!");
        return;
    }

    var signed_text = SignText(str);
    if (signed_text == "")
    {
        alert("No signature found!");
        return;
    }

    alert(signed_text);

    //form.signature.value = signed_text;
    //form.submit();
}

function SignText(text)
{
    var retStr = "";

    try
    {
        var oCertificate = new ActiveXObject("DigtCrypto.Certificate");
        var oPKCS7Message = new ActiveXObject("DigtCrypto.PKCS7Message");
        var oProfile = new ActiveXObject("DigtCrypto.Profile");

        if (text != "")
        {
            var DT_PLAIN_DATA = 0;
            var DT_SIGNED_DATA = 2;
            var BASE64_TYPE = 0;
            var SIGN_WIZARD_TYPE = 1;
            var SILENT_LEVEL_REQUIRED = 3;

            var isDetachedSign = true; // generate detached (true) or attached
            (false) signature

            oProfile.SilentLevel = SILENT_LEVEL_REQUIRED;
            oProfile.DisableInputFilesWindow = true;
            if ( oProfile.CollectData(SIGN_WIZARD_TYPE) ) // it returns false
when user cancel wizard
            {
                oProfile.SignIncludeBase64Headers = false;
                oProfile.Detach = isDetachedSign;

                //if ( oProfile.CheckData(SIGN_WIZARD_TYPE) ) // may be skipped if
is called after .CollectData()
                //{
                // throw "Not enough data (code: " +
oProfile.CheckData(SIGN_WIZARD_TYPE) + ")!";
                //}

                oPKCS7Message.Import(DT_PLAIN_DATA, text); // loading of data

                oPKCS7Message.Profile = oProfile;
                oPKCS7Message.Sign(); // signing
            }
        }
    }
}
```

```

        retStr = oPKCS7Message.Export(DT_SIGNED_DATA, BASE64_TYPE); // ex-
porting of signed document
    }
}
catch(e)
{
    alert("Exception caught: " + ((typeof e == "object") ? e.description
: e));
}

return retStr;
}
//-->
</script>
</head>
<body class="popup" onload="window.focus();">
<form name="form" method="post" action="sign_form.asp">
    <input type="hidden" name="signature" value="">
    <table border="0" cellspacing="0" cellpadding="0" class="popup">
        <tr><td>Enter a text to sign</td></tr>
        <tr><td>
            <input type="text" name="text_to_sign" value="" size="40">
            <input type="button" name="btnOK" value="Submit" OnClick="OnOK()">
        </td></tr>
    </table>
</form>
</body>
</html>

```

### 1.2.7. ВЫПОЛНЕНИЕ КРИПТОГРАФИЧЕСКИХ ОПЕРАЦИЙ

#### Создание подписи

Создание подписи производится с помощью метода Sign. Параметры подписи метод получает из профиля, поэтому для выполнения операции необходимым условием является наличие установленного у объекта PKCS7Message свойства Profile.

```

Option explicit
' в хранилище должен быть профиль с заполненными параметрами ЭЦП

'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0

'enum DATATYPE (тип данных)
const PLAIN_DATA = 0
const SIGNED_DATA = 2

'enum WIZARD_TYPE and RESULTTYPE
const SIGN_WIZARD_TYPE = 1

'enum CHECKING_WIZARD
const ALL_OK = 0

const PLAIN_DATA_FILE = "test.txt"
const OUTPUT_DATA_FILE = "test.txt.sign"

dim CheckResult

```

```

Dim objProfile, objProfileStore, objProfiles
Dim oPKCS7Message

'Получим профиль по умолчанию или создадим новый, если его нет

Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый
    профиль
end if

'Приступим к получению подписи, используя данные, полученные из профиля
objProfile.CollectData SIGN_WIZARD_TYPE 'Запустим мастер подписи для сбора
данных
CheckResult = objProfile.CheckData (SIGN_WIZARD_TYPE) 'Проверим, все ли дан-
ные собраны
If CheckResult = ALL_OK then
    Set oPKCS7Message = CreateObject ("DigtCrypto.PKCS7Message")
    oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
    oPKCS7Message.Load PLAIN_DATA, CStr (PLAIN_DATA_FILE), "" 'Загрузим
исходные данные
    oPKCS7Message.Sign 'Подпишем данные, используя параметры подписи из
    профиля
    oPKCS7Message.Save SIGNED_DATA, BASE64_TYPE, OUTPUT_DATA_FILE 'Сохраним
    данные
    set oPKCS7Message = nothing
else
    MsgBox "Профиль некорректно заполнен"
end if

```

### Добавление подписи

Создание параллельной подписи (добавление подписи) производится с помощью метода Sign, выполняемого на уже подписанных данных. Параметры подписи метод получает из профиля, поэтому для выполнения операции необходимым условием является наличие установленного у объекта PKCS7Message свойства Profile.

```

Option explicit
'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0

'enum DATATYPE (тип данных)
const PLAIN_DATA = 0
const SIGNED_DATA = 2

'enum WIZARD_TYPE and RESULTTYPE
const ADD_SIGN_WIZARD_TYPE = 2

'enum CHECKING_WIZARD
const ALL_OK = 0
const SIGN_DATA_FILE = "test.txt.sign"
const OUTPUT_DATA_FILE = "test.txt.addsign"
Dim CheckResult
Dim objProfile, objProfileStore, objProfiles, oSigners, oFirstSigner,

```

```
oPKCS7Message
Dim sHashAlg, sOriginalHashAlg

'Получим профиль по умолчанию или создадим новый, если его нет
Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый профиль
end if

'загрузим файл присоединенной подписи
Set oPKCS7Message = CreateObject("DigtCrypto.PKCS7Message")
oPKCS7Message.Load SIGNED_DATA, CStr(SIGN_DATA_FILE), ""

'Получим коллекцию подписей
Set oSigners = oPKCS7Message.Signers
If oSigners.Count > 0 then
    'Установим профиль с настройками
    oPKCS7Message.Profile = objProfile
    objProfile.CollectData ADD_SIGN_WIZARD_TYPE
    CheckResult = -10
    CheckResult = objProfile.CheckData (ADD_SIGN_WIZARD_TYPE)
    If CheckResult = ALL_OK then
        'Добавим подпись
        oPKCS7Message.Sign
        'Сохраним данные
        oPKCS7Message.Save SIGNED_DATA, BASE64_TYPE, OUTPUT_DATA_FILE
        set oPKCS7Message = nothing
    else
        MsgBox "Профиль некорректно заполнен"
    end if
else
    MsgBox "Нет подписчиков в сообщении"
end if
```

### Заверение подписи

Создание встречной подписи (заверение подписи) производится с помощью метода `Cosign` объекта `PKCS7Message`, которому в качестве параметра передается индекс заверяемой подписи. Метод выполняемого на уже подписанных данных. Параметры соподписи метод получает из профиля, поэтому для выполнения операции необходимым условием является наличие установленного у объекта `PKCS7Message` свойства `Profile`.

```
Option explicit
' в хранилище должен быть профиль с заполненными параметрами ЭЦП

'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0

'enum DATATYPE (тип данных)
const PLAIN_DATA = 0
const SIGNED_DATA = 2

'enum WIZARD_TYPE and RESULTTYPE
const COSIGN_WIZARD_TYPE = 4
```

```
'enum CHECKING_WIZARD
const ALL_OK = 0

const SIGN_DATA_FILE = "test.txt.sign"
const OUTPUT_DATA_FILE = "test.txt.cosign"

dim CheckResult
Dim objProfile, objProfileStore, objProfiles, oSignatures, oSignature,
oFirstSigner, oPKCS7Message
Dim sHashAlg, sOriginalHashAlg

'Получим профиль по умолчанию или создадим новый, если его нет

Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый
    профиль
end if
'загрузим файл присоединенной подписи
Set oPKCS7Message = CreateObject("DigtCrypto.PKCS7Message")
oPKCS7Message.Load SIGNED_DATA, CStr(SIGN_DATA_FILE), ""
'Получим коллекцию подписей
Set oSignatures = oPKCS7Message.Signatures
If oSignatures.Count > 0 then
    oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
    objProfile.CollectData COSIGN_WIZARD_TYPE
    CheckResult = -10
    CheckResult = objProfile.CheckData (COSIGN_WIZARD_TYPE)
    If CheckResult = ALL_OK then
        oPKCS7Message.Profile = objProfile
        'Заверим подпись порядковым индексом 0
        oPKCS7Message.Cosign 0
        oPKCS7Message.Save SIGNED_DATA, BASE64_TYPE, OUTPUT_DATA_FILE
    'Сохраним данные
        set oPKCS7Message = nothing
    else
        MsgBox "Профиль некорректно заполнен"
    end if
else
    MsgBox "Нет подписчиков в сообщении"
end if
```

### Открытие окна управления PKCS7 данными

Открытие окна управления подписанными данными или окна управления шифрованными данными производится с помощью метода Display объекта PKCS7Message. Для открытия окон управления необходимо, чтобы у объекта PKCS7Message было установлено свойство Profile.

```
Option explicit
' в хранилище должен быть профиль с заполненными параметрами ЭЦП

'enum DATATYPE (тип данных)
const DT_PLAIN_DATA = 0
const DT_SIGNED_DATA = 2
const DT_ENVELOPED_DATA = 3
```

```
'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0

const SINGED_DATA_FILE = "test.txt.sign"
const ENCRYPTED_DATA_FILE = "test.txt.encrypt"

Dim oPKCS7Message, objProfileStore, objProfiles, objProfile

Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый
    профиль
end if
'загрузим файл подписи
Set oPKCS7Message = CreateObject("DigtCrypto.PKCS7Message")
oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
oPKCS7Message.Load DT_SIGNED_DATA, SINGED_DATA_FILE, ""
oPKCS7Message.Display

'Загрузим файл зашифрованных данных
Set oPKCS7Message = Nothing
Set oPKCS7Message = CreateObject("DigtCrypto.PKCS7Message")
oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
oPKCS7Message.Load DT_ENVELOPED_DATA, ENCRYPTED_DATA_FILE, ""
oPKCS7Message.Display
```

## Шифрование данных

Шифрование данных производится с помощью метода Encrypt объекта PKCS7Message. Параметры шифрования метод получает из профиля, поэтому для выполнения операции необходимым условием является наличие установленного у объекта PKCS7Message свойства Profile.

```
Option explicit
' в хранилище должен быть профиль с заполненными параметрами ЭЦП

'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0

'enum DATATYPE (тип данных)
const DT_PLAIN_DATA = 0
const DT_ENVELOPED_DATA = 3

'enum WIZARD_TYPE and RESULTTYPE
const ENCRYPT_WIZARD_TYPE = 64

'enum CHECKING_WIZARD
const ALL_OK = 0

const PLAIN_DATA_FILE = "test.txt"
const OUTPUT_DATA_FILE = "test.txt.encrypt"
```

```
dim CheckResult
Dim objProfile, objProfileStore, objProfiles
Dim oPKCS7Message

'Получим профиль по умолчанию или создадим новый, если его нет

Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый
    профиль
end if

'Приступим к получению подписи, используя данные, полученные из профиля
objProfile.CollectData ENCRYPT_WIZARD_TYPE
CheckResult = objProfile.CheckData (ENCRYPT_WIZARD_TYPE) 'Проверим, все ли
данные собраны
If CheckResult = ALL_OK then
    Set oPKCS7Message = CreateObject ("DigtCrypto.PKCS7Message")
    oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
    oPKCS7Message.Load DT_PLAIN_DATA, PLAIN_DATA_FILE, "" 'Загрузим исходные
данные
    oPKCS7Message.Encrypt
    oPKCS7Message.Save DT_ENVELOPED_DATA, BASE64_TYPE, OUTPUT_DATA_FILE
'Сохраним данные
    set oPKCS7Message = nothing
else
    MsgBox "Профиль некорректно заполнен"
end if
```

### Шифрование блоков данных на одном сессионном ключе

Шифрование блоков данных на одном сессионном ключе производится с помощью методов объекта EncryptedMessage. Данная операция производится только в неинтерактивном режиме и параметры для шифрования необходимо задавать в профиле.

**ШИФРОВАНИЕ ДАННЫХ, ЗАГРУЖЕННЫХ ИЗ ОБЛАСТИ ПАМЯТИ ПРОИЗВОДИТСЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ OPEN, EXPORT.**

**ПРИМЕР 1. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОВМЕЩЕН С ПЕРВЫМ БЛОКОМ ШИФРОВАННЫХ ДАННЫХ**

Шифрование с сохранением первой порции (заголовка) шифрованного сообщения в один файл с первым зашифрованным блоком данных.

```
Option Explicit

'enum FORMAT
Dim strStroka
Dim file, fso

const BASE64= 0
const DER = 1

const CERT_FOR_ENCRYPT = 0
const CERT_FOR_DECRYPT = 1
const CERT_FOR_SIGN = 2
```

```
Const DT_PLAIN_TEXT = 0

Set fso = CreateObject( "Scripting.FileSystemObject" )

'Открытие хранилища профилей и получение профиля по умолчанию
'В профиле должны быть установлены необходимые для выполнения шифрования дан-
ные
Dim oProfileStore : Set oProfileStore =
CreateObject("DigtCrypto.ProfileStore")
Dim oProfile : Set oProfile = CreateObject("DigtCrypto.Profile")
oProfileStore.Open 0

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")

'Установим профиль, с использованием которого будем производить шифрование
oMsg.Profile= oProfileStore.Store.DefaultProfile

'Инициализация объекта
oMsg.Open DT_PLAIN_TEXT

'Импорт в объект первого блока данных
oMsg.Import "Первый блок данных"

'Сохранение зашифрованного первого блока данных вместе с первой порцией (за-
головком) зашифрованного сообщения в файл
strStroka = oMsg.Export()
Set file = fso.CreateTextFile("File_attached_header.enc", True)
file.write (strStroka)
file.close

'Импортируем второй блок данных
oMsg.Import "Второй блок данных"

'Сохранение зашифрованного второго блока данных в файл
strStroka = oMsg.Export()

Set file = fso.CreateTextFile("File_attached_header2.enc", True)
file.write (strStroka)
file.close
```

## ПРИМЕР 2. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОХРАНЕН ОТДЕЛЬНО ОТ ПЕРВОГО БЛОКА ШИФРОВАННЫХ ДАННЫХ

Шифрование с сохранением первой порции (заголовка) зашифрованного сообщения в отдельный файл от первого зашифрованного блока данных.

```
Option Explicit

'enum FORMAT
const BASE64= 0
const DER = 1

const CERT_FOR_ENCRYPT = 0
const CERT_FOR_DECRYPT = 1
const CERT_FOR_SIGN = 2
Const DT_PLAIN_TEXT = 0

Dim strStroka
Dim file, fso
Set fso = CreateObject( "Scripting.FileSystemObject" )

'Открытие хранилища профилей и получение профиля по умолчанию
```



```
'В профиле должны быть установлены необходимые для выполнения шифрования дан-
ные
Dim oProfileStore : Set oProfileStore =
CreateObject("DigtCrypto.ProfileStore")
oProfileStore.Open 0

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")

'Установим профиль, с использованием которого будем производить шифрование
oMsg.Profile= oProfileStore.Store.DefaultProfile

'Инициализация объекта
oMsg.Open DT_PLAIN_TEXT
'Сохранение первой порцией (заголовком) зашифрованного сообщения в файл
strStroka = oMsg.Export()
Set file = fso.CreateTextFile("File_header.enc", True)
file.write (strStroka)
file.close

'Импорт в объект первого блока данных
oMsg.Import "Первый блок данных"

'Сохранение зашифрованного первого блока данных в файл
strStroka = oMsg.Export()
Set file = fso.CreateTextFile("File.enc", True)
file.write (strStroka)
file.close

'Импорт в объект второго блока данных
oMsg.Import "Второй блок данных"

'Сохранение зашифрованного первого блока данных в файл
strStroka = oMsg.Export()
Set file = fso.CreateTextFile("File1.enc", True)
file.write (strStroka)
file.close
```

**ШИФРОВАНИЕ ДАННЫХ, ЗАГРУЖЕННЫХ ИЗ ФАЙЛА ПРОИЗВОДИТСЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ OPEN, LOAD.**

### **ПРИМЕР 3. ЗАГОЛОВOK ШИФРОВАННОГО СООБЩЕНИЯ СОВМЕЩЕН С ПЕРВЫМ БЛОКОМ ШИФРОВАННЫХ ДАННЫХ**

Шифрование с сохранением первой порции (заголовка) зашифрованного сообщения в один файл с первым зашифрованным блоком данных.

```
Option Explicit
Const DT_PLAIN_TEXT = 0

Dim oProfileStore : Set oProfileStore =
CreateObject("DigtCrypto.ProfileStore")
Открытие хранилища профилей и получение профиля по умолчанию
'В профиле по умолчанию должны быть установлены необходимые для выполнения
шифрования данные
oProfileStore.Open 0

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
'Установим профиль, с использованием которого будем производить шифрование
oMsg.Profile = oProfileStore.Store.DefaultProfile

'Инициализация объекта
oMsg.Open DT_PLAIN_TEXT
```

```
'Загрузка первого файла с исходными данными
oMsg.Load "data1.in"
'Сохранение шифрованного первого файла с исходными данными вместе с заголовком
шифрованного сообщения
oMsg.Save "header&data1.enc"

'Загрузка второго файла с исходными данными
oMsg.Load "data2.in"
'Сохранение шифрованного первого файла с исходными данными вместе с заголовком
шифрованного сообщения
oMsg.Save "data2.enc"
```

#### **ПРИМЕР 4. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОХРАНЕН ОТДЕЛЬНО ОТ ПЕРВОГО БЛОКА ШИФРОВАННЫХ ДАННЫХ**

Шифрование с сохранением первой порции (заголовка) шифрованного сообщения в отдельный файл от первого зашифрованного блока данных.

```
Option Explicit
Const DT_PLAIN_TEXT = 0

Dim oProfileStore : Set oProfileStore =
CreateObject("DigtCrypto.ProfileStore")
Открытие хранилища профилей и получение профиля по умолчанию
'В профиле по умолчанию должны быть установлены необходимые для выполнения
шифрования данные
oProfileStore.Open 0

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
'Установим профиль, с использованием которого будем производить шифрование
oMsg.Profile = oProfileStore.Store.DefaultProfile

'Инициализация объекта
oMsg.Open DT_PLAIN_TEXT
'Сохранение заголовка шифрованного сообщения в отдельный файл
oMsg.Save "header.enc"
'Загрузка первого файла с исходными данными
oMsg.Load "data1.in"
'Сохранение шифрованного первого файла с исходными данными
oMsg.Save "data1.enc"
'Загрузка второго файла с исходными данными
oMsg.Load "data2.in"
'Сохранение шифрованного первого файла с исходными данными
oMsg.Save "data2.enc"
```

### **Расшифрование данных**

Расшифрование данных производится с помощью метода Decrypt, выполняемого на зашифрованных данных. Параметры расшифрования метод получает из профиля, поэтому для выполнения операции необходимым условием является наличие установленного у объекта PKCS7Message свойства Profile.

```
Option explicit
' в хранилище должен быть профиль с заполненными параметрами ЭЦП

'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1

'enum PROFILESTORETYPE (профили)
const REGISTRY_STORE = 0
```

```
'enum DATATYPE (тип данных)
const DT_PLAIN_DATA = 0
const DT_ENVELOPED_DATA = 3

'enum WIZARD_TYPE and RESULTTYPE
const DECRYPT_WIZARD_TYPE = 1024

'enum CHECKING_WIZARD
const ALL_OK = 0

const INPUT_DATA_FILE = "test.txt.encrypt"
const OUTPUT_DATA_FILE = "test.txt.decrypted"

dim CheckResult
Dim objProfile, objProfileStore, objProfiles
Dim oPKCS7Message

'Получим профиль по умолчанию или создадим новый, если его нет

Set objProfileStore = CreateObject("DigtCrypto.ProfileStore")
objProfileStore.Open REGISTRY_STORE 'Открываем хранилище профилей
Set objProfiles = objProfileStore.Store 'Получаем коллекцию профилей
If objProfiles.Count > 0 then
    Set objProfile = objProfiles.DefaultProfile 'Получим профиль по умолчанию
else
    Set objProfile = CreateObject("DigtCrypto.Profile") 'Создадим новый
    профиль
end if

'Приступим к получению подписи, используя данные, полученные из профиля
objProfile.CollectData DECRYPT_WIZARD_TYPE
CheckResult = objProfile.CheckData (DECRYPT_WIZARD_TYPE) 'Проверим, все ли
данные собраны
If CheckResult = ALL_OK then
    Set oPKCS7Message = CreateObject ("DigtCrypto.PKCS7Message")
    oPKCS7Message.Profile = objProfile 'Установим профиль с настройками
    oPKCS7Message.Load DT_ENVELOPED_DATA, INPUT_DATA_FILE, "" 'Загрузим
исходные данные
    oPKCS7Message.Decrypt
    oPKCS7Message.Save DT_PLAIN_DATA, BASE64_TYPE, OUTPUT_DATA_FILE 'Сохраним
данные
    set oPKCS7Message = nothing
else
    MsgBox "Профиль некорректно заполнен"
end if
```

### Расшифрование блоков данных на одном сессионном ключе

Расшифрование блоков данных на одном сессионном ключе производится с помощью методов объекта EncryptedMessage. Данная операция производится только в неинтерактивном режиме.

**РАСШИФРОВАНИЕ ДАННЫХ, ЗАГРУЖЕННЫХ ИЗ ОБЛАСТИ ПАМЯТИ ПРОИЗВОДИТСЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ OPEN, EXPORT.**

#### ПРИМЕР 1. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОВМЕЩЕН С ПЕРВЫМ БЛОКОМ ШИФРОВАННЫХ ДАННЫХ

Расшифрование с блоков шифрованных данных. В первом блоке содержится заголовок шифрованного сообщения и первый шифрованный блок данных.

```
Option Explicit
const CERT_FOR_ENCRYPT = 0
const CERT_FOR_DECRYPT = 1
const CERT_FOR_SIGN = 2

Dim strStroka, fso, file1, strStroka1, file

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
Set fso = CreateObject( "Scripting.FileSystemObject" )

'Инициализация объекта
oMsg.Open 6, "file_attached_header.enc"
'Экспортируем расшифрованные данные в строку
strStroka1 = oMsg.Export()

'Сохраним первый блок расшифрованных данных в файл
Set file = fso.CreateTextFile("File.data", True)
file.write (strStroka1)
file.close
strStroka1 = ""

'Считаем в строку второй блок зашифрованных данных
Set file1 = fso.OpenTextFile("file_attached_header2.enc", 1)
strStroka = file1.ReadAll
file1.Close()

'Импортируем в объект второй блок данных
oMsg.Import strStroka
'Экспортируем расшифрованные данные в строку
strStroka1 = oMsg.Export()

'Сохраним расшифрованных данные в файл
Set file = fso.CreateTextFile("File2.data", True)
file.write (strStroka1)
file.close
```

## **ПРИМЕР 2. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОХРАНЕН ОТДЕЛЬНО ОТ ПЕРВОГО БЛОКА ШИФРОВАННЫХ ДАННЫХ**

Расшифрование шифрованных блоков данных в случае, когда заголовок шифрованного сообщения сохранен отдельно от данных.

```
Option Explicit
const CERT_FOR_ENCRYPT = 0
const CERT_FOR_DECRYPT = 1
const CERT_FOR_SIGN = 2

Dim strStroka, fso, file1, strStroka1, file

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
Set fso = CreateObject( "Scripting.FileSystemObject" )

'Инициализация объекта с загрузкой заголовка шифрованного сообщения
oMsg.Open 6, "file_header.enc"

'Считаем в строку первый блок зашифрованных данных
Set file1 = fso.OpenTextFile("file_attached_header2.enc", 1)
strStroka = file1.ReadAll
file1.Close()

'Импортируем в объект первый блок данных
oMsg.Import strStroka
'Экспортируем расшифрованные данные в строку
strStroka1 = oMsg.Export()
```

```
'Сохраним первый блок расшифрованных данных в файл
Set file = fso.CreateTextFile("File.data", True)
file.write (strStroka1)
file.close
strStroka1 = ""

'Считаем в строку второй блок зашифрованных данных
Set file1 = fso.OpenTextFile("file_attached_header2.enc", 1)
strStroka = file1.ReadAll
file1.Close()

'Импортируем в объект второй блок данных
oMsg.Import strStroka
'Экспортируем расшифрованные данные в строку
strStroka1 = oMsg.Export()

'Сохраним расшифрованных данные в файл
Set file = fso.CreateTextFile("File2.data", True)
file.write (strStroka1)
file.close
```

**РАСШИФРОВАНИЕ ДАННЫХ, ЗАГРУЖЕННЫХ ИЗ ФАЙЛА ПРОИЗВОДИТСЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ OPEN, LOAD.**

### **ПРИМЕР 3. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОВМЕЩЕН С ПЕРВЫМ БЛОКОМ ШИФРОВАННЫХ ДАННЫХ**

Расшифрование данных в случае, когда первая порция (заголовок) шифрованного сообщения сохранен в один файл с первым зашифрованным блоком данных.

```
Option Explicit

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
'Инициализация объекта
oMsg.Open 6, "header&data1.enc"
'Сохраняем первый расшифрованный блок данных в файл
oMsg.Save "data1_dec.out"
'Загрузка второго файла с шифрованными данными
oMsg.Load "data2.enc"
'Сохранение первого расшифрованного блока данных в файл
oMsg.Save "data2_dec.out"
```

### **ПРИМЕР 4. ЗАГОЛОВОК ШИФРОВАННОГО СООБЩЕНИЯ СОХРАНЕН ОТДЕЛЬНО ОТ ПЕРВОГО БЛОКА ШИФРОВАННЫХ ДАННЫХ**

Расшифрование данных в случае, когда первая порция (заголовок) шифрованного сообщения сохранен отдельно от первого зашифрованного блока данных.

```
Option Explicit

Dim oMsg : Set oMsg = CreateObject("DigtCrypto.EncryptedMessage")
'Инициализация объекта
oMsg.Open 6, "header.enc"
Загрузка первого файла с шифрованными данными
oMsg.Load "data1.enc"
'Сохранение первого расшифрованного блока данных в файл
oMsg.Save "data1_dec.out"
```

## 1.2.8. РАБОТА С ВЕРСИЯМИ DIGTCRYPTO

### Получение версии DigtCrypto

Получение версии DigtCrypto производится с помощью метода Version объекта Util.

```
Option explicit
'Объявление переменных
Dim sVersion, oUtil

Set oUtil = CreateObject ("DigtCrypto.Util")
'Получение версии библиотеки DigtCrypto
sVersion = oDigtCryptoUtil.Version
MsgBox sVersion, , "DigtCrypto's version"
'Установка лицензии
oUtil.License = "TD4TF-GKHFP-FKTAT-VPKDJ-HGCFD-RWJMF-XWTPT"
'Получение лицензии
sLicense = oUtil.License
MsgBox sLicense
```

## 1.2.9. ЖУРНАЛИРОВАНИЕ СОБЫТИЙ

### Подключение журналирования

Подключение журналирования производится с помощью методов StartupLogger, GetLoggerStatements и ShutdownLogger объекта Util.

```
Option explicit
'Объявление переменных
Const LOG_LEVEL_DISABLED = 0 'Логирование отключено
Const LOG_LEVEL_ERROR = 1 'Ошибки
Const LOG_LEVEL_WARN = 2 'Предупреждения
Const LOG_LEVEL_INFO = 3 'Информация
Const LOG_LEVEL_DEBUG = 4 'Отладка
Const LOG_LEVEL_TRACE = 5 'Трассировка
Const strPattern = "log4cplus.appender.toDSL.layout.ConversionPattern=%d [%5t] [%p] [%c] [%x] - %m%n"
Dim oUtil, sLoggerID, strLoggerStatement
Set oUtil = CreateObject ("DigtCrypto.Util")
sLoggerID = oUtil.StartupLogger (strPattern, LOG_LEVEL_TRACE, 500*1024, "DSS", 0) ' Запуск журналирования
strLoggerStatement = oUtil.GetLoggerStatements (sLoggerID) ' Получение списка зажурналированных событий
MsgBox strLoggerStatement
oUtil.ShutdownLogger sLoggerID
```

## 1.3. ИНТЕРФЕЙСЫ DIGTCRYPTO

### 1.3.1. КОНСТАНТЫ И ПЕРЕЧИСЛЕНИЯ

Константа	Описание
CA_TYPE	тип Центра Сертификации
CERTIFICATE_PUPROSE	Тип сертификата (по назначению)
CERTIFICATE_VERIFY_LEVEL	Уровень проверки
CERTIFICATE_WORKABILITY	Применимость сертификата для выполнения операции в зависимости от политики назначений сертификата
CERTTYPE	Тип сертификата
CHECKING_RESULT	Статус завершения проверки полноты заполнения параметров профиля для выполнения операции
CRL_UPDATE_FLAGS	Флаги обновления СОС
DATATYPE	Тип данных
EXTENTION_FLAG	Флаг расширения сертификата
FORMAT	Формат данных

KEY_SPEC	Использование ключа
LOG_LEVEL	Тип событий для логирования
MATCH_CRITERIA	Критерий совпадения
POLICY_TYPE	Тип политики использования сертификатов
PROFILEEXITFORMAT	Формат выходного файла, устанавливаемый в профиле
PROFILESTORETYPE	Тип хранилища настроек
PKIREQUEST_STATUS	Статус PKI запроса (обработан/не обработан)
PKIREQUEST_STATUS_TYPE	Тип проверки статуса PKI запроса
REQUEST_GENERATION_RESULT	Статус генерации PKI запроса
REQUEST_SENDING_RESULT	Статус PKI отправки запроса
REQUEST_TYPE	Тип PKI запроса
REVOCATION_REASON	Причина отзыва сертификата
SEARCH_CRITERIA	Критерий поиска
SERVICE_AUTH_TYPE	Тип аутентификации для доступа к Веб-сервису
SIGNATURE_TYPE	Тип подписи
SILENT_LEVEL	Уровень интерактивности приложения
STORETYPE	Тип хранилища сертификатов
SYSTEM_STORE_LOCATION	Типы хранилищ сертификатов
TIME_STAMP_TYPE	Тип штампа времени
VERIFYFLAG	Степень проверки подписи
VERIFYSTATUS	Результат проверки корректности сертификата
WIZARD_TYPE	Тип мастера операции, для которой проверяется полнота заполнения параметров профиля

## CA\_TYPE

Типы Центров Сертификации

Значение	Описание
CA_TYPE_CMS = 1	ЦС, поддерживающий CMS формат электронных документов
CA_TYPE_CRYPTOPRO = 2	ЦС, разработанный кампанией КриптоПРО
CA_TYPE_WEB_SERVICE = 4	Использование модуля Веб-сервис в качестве Центра регистрации

## CERTIFICATE\_PURPOSE

Определяет предназначение сертификата (при заполнении параметров выполнения операции в настройках).

Значение	Описание
CERTIFICATE_FOR_NEW_SIGNATURE = 1	Сертификат для новой подписи
CERTIFICATE_FOR_SIGNATURE_VERIFYING = 2	Сертификат для подписи и соподписи
CERTIFICATE_FOR_ENCIPHER = 3	Сертификат для шифрования
CERTIFICATE_FOR_DECIPHER = 4	Сертификат для расшифрования

## CERTIFICATE\_VERIFY\_LEVEL

Уровень проверки статуса сертификата.

Значение	Описание
CERTIFICATE_VERIFY_CLEAR = 0	Очищает сертификат из всех уровней проверки. Может быть использовано только при вызове свойства SetVerifiedCertificate
CERTIFICATE_VERIFY_LOCAL_CRL = 1	Проверка сертификата по локальному СОС
CERTIFICATE_VERIFY_ONLINE_CRL = 2	Проверка сертификата по СОС, предварительно полученному из УЦ
CERTIFICATE_VERIFY_REV_PROV = 4	Проверка сертификат с помощью Revocation Provider
CERTIFICATE_VERIFY_OCSP = 8	Проверка сертификат в службе OCSP
CERTIFICATE_VERIFY_NO_REVOCATION_CHECK = 16	Без проверки сертификата на отозванность. Используется, когда необходимо осуществить проверку сертификата только по политике или лицензии.

## CERTIFICATE\_WORKABILITY

Определяет применимость сертификата для выполнения операции в зависимости от политики назначения сертификата.

Значение	Описание
CERTIFICATE_ALLOWED = 1	Сертификат разрешен для использования в операции
CERTIFICATE_RESTRICTED = 2	Сертификат ограничен для использования в операции
CERTIFICATE_UNKNOWN = 3	Неизвестна применимость сертификата в операции

## CERTTYPE

Тип сертификата в зависимости от выполняемой операции.

Значение	Описание
CERT_FOR_ENCRYPT = 0	Сертификат для шифрования
CERT_FOR_DECRYPT = 1	Сертификат для расшифрования
CERT_FOR_SIGN = 2	Сертификат для подписи
CERT_SENDER = 3	Сертификат отправителя зашифрованного сообщения

## CHECKING\_RESULT

Статус завершения проверки полноты заполнения параметров профиля для выполнения операции.

Значение	Описание
ALL_OK = 0	OK
E_SIGNATURE_MISSING_CERTIFICATE = 1	Отсутствует сертификат подписи
E_ENCRYPTION_MISSING_CERTIFICATE = 2	Отсутствует сертификат шифрования
E_ENCRYPTION_MISSING_CRYPTOPROVIDER = 3	Отсутствует криптопровайдер для шифрования
E_ENCRYPTION_MISSING_ALGORITHM = 4	Отсутствует алгоритм для шифрования
E_ENCRYPTION_MISSING_RECIPIENTS_LIST = 5	Не заполнен список сертификатов получателей
NO_DEFAULT_PROFILE = 6	Не установлены настройки по умолчанию
NO_DECRYPT_CERTIFICATE_IN_PROFILE = 7	Отсутствует сертификат расшифрования в настройках
NO_DECRYPT_CERTIFICATE_IN_STORE = 8	Отсутствует сертификат расшифрования в хранилище
DIFFERENT_CERT_TYPES_FOR_ENCRYPTING = 9	Сертификаты получателей отличаются по типу от криптопровайдера шифрования
E_SIGNATURE_INVALID_CERTIFICATE = 10	Сертификат подписи недействителен
E_SIGNATURE_INVALID_HASH_ALG = 11	Отсутствует или некорректный алгоритм хэширования подписи

## CRL\_UPDATE\_FLAGS

Определяет степень проверки подписи.

Значение	Описание
CRL_UPDATE_STORE_LOCAL_MACHINE = 1	Будет обновлен СОС в машинном хранилище
CRL_UPDATE_NOT_INSTALL = 2	СОС будет получен, но не будет установлен
CRL_UPDATE_NOT_VERIFY = 4	СОС не будет проверен



## DATA\_TYPE

Тип данных, используемый при работе с криптографическими сообщениями.

Значение	Описание
DT_AUTO_DETECT = -1	Тип криптографического сообщения определяется автоматически
DT_PLAIN_DATA = 0	Исходные (подписываемые или шифруемые) данные
DT_SIGNED_DATA = 2	Подписанное сообщение (криптографическое сообщение, содержащее результат формирования ЭЦП)
DT_ENVELOPED_DATA = 3	Шифрованное сообщение, содержащее зашифрованные данные и информацию о получателях зашифрованных данных (криптографическое сообщение, содержащее результат шифрования).
DT_ENCRYPTED_DATA = 6	Зашифрованные данные (блок зашифрованных данных без информации о получателях зашифрованных данных)

## FORMAT

Формат данных

Значение	Описание
BASE64_TYPE = 0	Формат данных в кодировке BASE4
DER_TYPE = 1	Формат данных без дополнительных преобразований
XML_TYPE = 2	Данные в формате XML
HEX_TYPE = 3	Данные в шестнадцатичном формате
UNICODE_STRING_TYPE = 4	Данные в формате UNICODE

## KEY\_SPEC

Тип использования ключа.

Значение	Описание
KEY_SPEC_AT_KEYEXCHANGE = 1	Область использования ключа: шифрование. Необходим для включения следующих битов в расширении "Использование ключа" сертификата: Key Encipherment, Data Encipherment и Key Agreement.
KEY_SPEC_AT_SIGNATURE = 2	Область использования ключа: подпись. Необходим для включения следующих битов в расширении "Использование ключа" сертификата: Digital Signature и Non-Repudiation.
KEY_SPEC_KEYEXCHANGE_AND_SIGNATURE = 3	Область использования ключа: подпись и шифрование. Позволяет включать любые биты расширения "Использование ключа" сертификата.

## LOG\_LEVEL

Определяет тип событий для логирования.

Значение	Описание
LOG_LEVEL_DISABLED = 0	Логирование отключено
LOG_LEVEL_ERROR = 1	Ошибки
LOG_LEVEL_WARN = 2	Предупреждения
LOG_LEVEL_INFO = 3	Информация
LOG_LEVEL_DEBUG = 4	Отладка
LOG_LEVEL_TRACE = 5	Трассировка

## MATCH\_CRITERIA

Критерий совпадения (при поиске в справочнике назначений (OID-ов)).

Значение	Описание
MATCH_CRITERIA_EQUAL = 1	Полное соответствие
MATCH_CRITERIA_BEGIN = 2	Начинается с
MATCH_CRITERIA_SUBSTR = 3	Содержит
MATCH_CRITERIA_END = 4	Заканчивается на

## POLICY\_TYPE

Определяет тип политики использования сертификатов

Значение	Описание
POLICY_TYPE_NONE = 0	Нет политики использования сертификатов
POLICY_TYPE_SIGNATURE = 1	Политика назначений для сертификатов подписи
POLICY_TYPE_ENCRYPT = 2	Политика назначений для сертификатов шифрования

## PROFILEEXITFORMAT

Формат выходного файла, устанавливаемый в профиле.

Формат выходного файла, устанавливаемый в Значение	Описание
BASE64 = 0	Формат выходного файла BASE64
DER = 1	Формат выходного файла DER

## PROFILESTORETYPE

Тип хранилища настроек

Значение	Описание
REGISTRY_STORE = 0	Тип хранилища настроек - Реестр
XML_STORE = 1	Тип хранилища настроек - XML файл

## RDN\_ENTRY\_FLAG

Описывает флаги, применяемые при подучении/отображении атрибутов относительного отличительного имени (RDN) сертификата

Константа	Описание
RDN_ENTRY_READONLY = 1	Отображать поле RDN только для чтения
RDN_ENTRY_MANDATORY = 2	Устанавливать поле RDN обязательным для заполнения
RDN_ENTRY_HIDDEN = 4	Скрывать поле RDN
RDN_ENTRY_INCLUDE_IF_EMPTY = 8	Используется при получении значения RDN. Значение поля включается в RDN, даже если оно пусто.

## REFRESH\_STATUS\_TYPE

Описывает результат отправки PKI запроса в УЦ.

Константа	Описание
REFRESH_STATUS_BY_CRL = 1	Проверка статуса запроса по СОС
REFRESH_STATUS_BY_CA = 2	Проверка статуса запроса в УЦ

## PKIREQUEST\_STATUS

Описывает статус обработки PKI запроса.

Константа	Описание
PKIREQUEST_CREATED = 1	Создан
REQUEST_PENDING = 2	В ожидании обработки
REQUEST_DENIED = 4	Отклонен
PKIREQUEST_PROCESSED = 8	Обработан

## REQUEST\_GENERATION\_RESULT

Описывает результат генерации PKI запроса.

Константа	Описание
REQUEST_GENERATION_SUCCESS = 1	Успешная генерация
REQUEST_GENERATION_OPER_CERT_NOT_FOUND = 2	Не найден сертификат, над которым производится операция
REQUEST_GENERATION_OPER_CERT_SN_NOT_FOUND = 3	Не найден серийный номер сертификата, над которым производится операция
REQUEST_GENERATION_SIGN_REQUEST_FAILED = 4	Ошибка подписи PKI запроса

## REQUEST\_SENDING\_RESULT

Описывает возможные результаты отправки PKI запроса.

Константа	Описание
REQUEST_SENDING_SUCCESS = 1	Операция отправки запроса прошла успешно
REQUEST_SENDING_REQUEST_MISSING = 2	Отсутствует запрос для отправки
REQUEST_SENDING_REQUEST_SOAP_MISSING = 3	Отсутствует объект для генерации SOAP-запросов
REQUEST_SENDING_REQUEST_SOAP_INIT_ERROR = 4	Произошла ошибка при инициализации SOAP-объекта
REQUEST_SENDING_REQUEST_SUBMIT_ERROR = 5	Произошла ошибка при постановке запроса в очередь
REQUEST_SENDING_REQUEST_MISSING_PROXY_PARAMS = 6	Произошла ошибка при постановке запроса в очередь
REQUEST_SENDING_REQUEST_ADDRESS_MISSING = 7	Отсутствует адрес отправки
REQUEST_SENDING_REQUEST_FILE_ERROR = 8	Ошибка при работе с файлом
REQUEST_SENDING_REQUEST_ERROR = 9	Ошибка отправки

## REQUEST\_TYPE

Описывает типы PKI запросов.

Константа	Описание
REQUEST_TYPE_REVOCATION = 1	Запрос на отзыв сертификата
REQUEST_TYPE_SUSPENDING = 2	Запрос на приостановление действия сертификата
REQUEST_TYPE_RESUMING = 4	Запрос на возобновление действия сертификата
REQUEST_TYPE_UPDATE_KEY = 16	Запрос на обновление ключа сертификата
REQUEST_TYPE_CERT_PKCS10 = 32	Запрос на сертификат в формате PKCS10
REQUEST_TYPE_SELF_SIGNED_CERT = 64	Запрос на самоподписанный сертификат

## REVOCATION\_REASON

Описывает причину отзыва сертификата.

Константа	Описание
REVOCATION_REASON_UNHOLD = -1	Возобновление сертификата
REVOCATION_REASON_UNSPECIFIED = 0	Причина не указана
REVOCATION_REASON_KEY_COMPROMISE = 1	Компрометация ключа
REVOCATION_REASON_CA_COMPROMISE = 2	Компрометация ЦС
REVOCATION_REASON_CHANGE_OF_AFFILIATION = 3	Изменение принадлежности
REVOCATION_REASON_SUPERSEDED = 4	Сертификат заменен
REVOCATION_REASON_Cease_OF_OPERATION = 5	Прекращение работы

REVOCATION\_REASON\_CERTIFICATE\_HOLD = 6

Приостановление действия сертификата

## SEARCH\_CRITERIA

Критерий поиска идентификатора (OID) использования сертификата.

Значение	Описание
SEARCH_CRITERIA_BY_FRIENDLY_NAME = 1	Поиск по имени
SEARCH_CRITERIA_BY_VALUE = 2	Поиск по значению

## SERVICE\_AUTH\_TYPE

Значение	Описание
SERVICE_AUTH_TYPE_ANONYMOUS = 0	Анонимный доступ
SERVICE_AUTH_TYPE_BASIC = 1	Обычная аутентификация
SERVICE_AUTH_TYPE_DIGEST = 2	Краткая проверка подлинности
SERVICE_AUTH_TYPE_NTLM = 3	Встроенная проверка подлинности Windows
SERVICE_AUTH_TYPE_NEGOTIATE = 4	Встроенная проверка подлинности Windows или Kerberos

## SIGNATURE\_TYPE

Описывает типы подписи

Константа	Описание
SIGNATURE_TYPE_UNKNOWN = 0	Неизвестный тип подписи
SIGNATURE_TYPE_BASIC = 1	Обычная подпись
SIGNATURE_TYPE_CADES_X_LONG_TYPE_1 = 96	Улучшенная проверка данных.

## SILENT\_LEVEL

Уровень интерактивности выполнения криптографических операций.

Значение	Описание
SILENT_LEVEL_SERVER = 1	Серверный уровень. Отключено отображение всех окон (в т.ч. и CSP)
SILENT_LEVEL_WINDOWS_ONLY = 2	Показываются окна, отмеченные в настройках интерфейса пользователя (в т.ч. и CSP)
SILENT_LEVEL_REQUIRED = 3	Показываются окна, отмеченные в настройках интерфейса пользователя, а также те, которые необходимы для сбора информации, достаточной для выполнения операции (а также все окна CSP)
SILENT_LEVEL_INTERACTIVE = 4	Отображаются все окна, все зависимости от настроек интерфейса пользователя

## STORETYPE

Определяет тип хранилища сертификатов.

Значение	Описание
STORE_TYPE_UNKNOWN = 0xffffffff	Тип хранилища не определен
STORE_TYPE_AUTO_DETECT = 0xffffffff	Автоматическое определение типа хранилища
LOCAL_MACHINE_STORE = 0	Хранилище компьютера
CURRENT_USER_STORE = 1	Хранилище текущего пользователя
STORE_TYPE_SYSTEM = 4	Системное хранилище
STORE_TYPE_MEMORY = 8	Хранилище - память
STORE_TYPE_PKCS7 = 16	Хранилище - файл формата PKCS #7
STORE_TYPE_SST = 32	Хранилище сериализованных сертификатов (SST)
STORE_TYPE_PFX = 64	Хранилище - файл обмена личной информацией PKCS #12 (.PFX, .P12).

**SYSTEM\_STORE\_LOCATION**

Битовая маска хранилищ сертификатов, предназначенных для отображения (закладками).

Значение	Описание
SYSTEM_STORE_MY = 1,	Личное хранилище сертификатов
SYSTEM_STORE_CA = 2,	Промежуточные Центры Сертификации
SYSTEM_STORE_ROOT = 4,	Доверенные Корневые Центры Сертификации
SYSTEM_STORE_TRUST = 8,	Доверенные издатели сертификатов
SYSTEM_STORE_ADDRESS_BOOK = 16,	Сертификаты других пользователей
SYSTEM_STORE_REQUEST = 32,	Хранилище запросов
SYSTEM_STORE_CRL_CA = 512	Хранилище списков отзыва сертификатов (COC)

**TIME\_STAMP\_TYPE**

Описывает типы штампов времени, включаемых в подпись

Константа	Описание
TIME_STAMP_TYPE_NONE = 0	Нет штампов в подписи
TIME_STAMP_TYPE_CONTENT = 1	Бит штампа на подписываемые данные (подписанный атрибут)
TIME_STAMP_TYPE_SIGNATURE = 2	Бит штампа на подпись (в CAdES-X-Long)

**VERIFYFLAG**

Определяет степень проверки подписи.

Значение	Описание
CERT_AND_SIGN = 0	Проверка подписи и сертификата
SIGN_ONLY = 1	Проверка только подписи

**VERIFYSTATUS**

Определяет результат проверки корректности сертификата.

Значение	Описание
VS_CORRECT = 1	Корректен
VS_UNINSUFFICIENT_INFO = 2	Статус неизвестен
VS_INCORRECT = 3	Некорректен
VS_INVALID_CERTIFICATE_BLOB = 4	Недействительный блок сертификата
VS_CERTIFICATE_TIME_EXPIRED = 5	Время действия сертификата истекло или еще не наступило
VS_CERTIFICATE_NO_CHAIN = 6	Невозможно построить цепочку сертификации
VS_CERTIFICATE_CRL_UPDATING_ERROR = 7	Ошибка обновления СОС
VS_LOCAL_CRL_NOT_FOUND = 8	Не найден локальный СОС
VS_CRL_TIME_EXPIRED = 9	Истекло время действия СОС
VS_CERTIFICATE_IN_CRL = 10	Сертификат находится в СОС
VS_CERTIFICATE_IN_LOCAL_CRL = 11	Сертификат находится в локальном СОС
VS_CERTIFICATE_CORRECT_BY_LOCAL_CRL = 12	Сертификат действителен по локальному СОС
VS_CERTIFICATE_USING_RESTRICTED = 13	Использование сертификата ограничено
VS_CERTIFICATE_RESTRICTED_BY_LICENSE = 14	Использование сертификата запрещено лицензией
VS_REVOCATION_STATUS_UNKNOWN = 15	Статус отзыва по Revocation Pdovider неизвестен (неверная лицензия или др.)
VS_REVOCATION_OCSP_ERROR = 16	Не удалось проверить статус сертификата по OCSP
VS_CADES_SUCCESS = VS_CORRECT	Улучшенная подпись (CAdES) корректна
VS_CADES_INVALID_REFS_AND_VALUES = 101	Улучшенная подпись (CAdES) содержит неверную информацию о подлинности
VS_CADES_SIGNER_NOT_FOUND = 102	Не найдена информация о подписчике в улучшенной подписи (CAdES)
VS_CADES_NO_VALID_SIGNATURE_STAMP = 103	Улучшенная подпись (CAdES) содержит неверный штамп времени на подпись

VS_CADES_REFS_AND_VALUES_NO_MATCH = 104	Несоответствие информации о подлинности улучшенной подписи (CADES)
VS_CADES_NO_CHAIN = 105	Ошибка построения пути сертификации для сертификата подписи в улучшенной подписи (CADES)
VS_CADES_END_CERT_REVOCATION = 106	Сертификат подписчика улучшенной подписи (CADES) недействителен
VS_CADES_CHAIN_CERT_REVOCATION = 107	Недействителен один или несколько сертификатов цепочки (для сертификата подписи в улучшенной подписи (CADES))
VS_CADES_BAD_SIGNATURE = 108	Улучшенная подпись (CADES) математически некорректна
VS_CADES_NO_VALID_CADES_C_TIMESTAMP = 109	Улучшенная подпись (CADES) содержит неверный штамп времени на доказательства подлинности
VS_TSL_OLD_VERSION = 112	Установлена старая версия списка аккредитованных УЦ (TSL)
VS_TSL_FILE_NOT_FOUND = 113	Не найден список аккредитованных УЦ (TSL)
VS_TSL_FILE_NOT_LOAD_FROM_SERVER = 114	Не удалось загрузить список аккредитованных УЦ с сервера.
VS_CERTIFICATE_NOT_QUALIFIED = 115	Сертификат не является квалифицированным сертификатом ключа проверки электронной подписи

## WIZARD\_TYPE

Описывает типы мастеров выполнения криптографических операций.

Константа	Описание
SIGN_WIZARD_TYPE = 1	Мастер подписи
ADD_SIGN_WIZARD_TYPE = 2	Мастер добавления подписи, вызываемый из окна управления подписью
COSIGN_WIZARD_TYPE = 4	Мастер заверения подписи, вызываемый из окна управления подписью
CONNECT_KEYCARRIER_WIZARD_TYPE = 8	Мастер подключения отчуждаемого носителя
ENCRYPT_WIZARD_TYPE = 64	Мастер шифрования
DECRYPT_WIZARD_TYPE = 1024	Мастер расшифрования
VERIFY_SIGNATURE_WIZARD_TYPE = 2048	Мастер проверки подписи
DECRYPT_VERIFY_SIGNATURE_WIZARD_TYPE = 4096	Мастер расшифрования и проверки подписи
DROP_SIGNATURE_WIZARD_TYPE = 8192	Мастер снятия и проверки подписи
VIEW_DOCUMENT_WIZARD_TYPE = 16384	Мастер просмотра документа, содержащегося в шифрованном файле или файле присоединенной подписи
OPEN MANAGERS_WIZARD_TYPE = 32768	Мастер открытия менеджера PKCS7 сообщения
ADD_SIGN_WIZARD_TYPE_WITH_FILES = 65536	Мастер добавления подписи, вызываемый независимо от окна управления подписью

### 1.3.2. ОБЪЕКТЫ И ИНТЕРФЕЙСЫ

#### Требования

Клиент: требуется Windows 8/7/XP/2000 Professional.

Сервер: требуется Windows Server 2008/2003/Windows 2000 Server SP4.

CERTIFICATE (GUID: 243CF830-9C2C-4E47-9EF9-2F9FADFCD3B1)

#### Методы

Метод	Описание
<a href="#">Display</a>	Метод Display позволяет показать форму сертификата
<a href="#">Export</a>	Метод Export позволяет осуществлять экспорт сертификата в строку
<a href="#">Import</a>	Метод Import позволяет осуществлять импорт сертификата из строки
<a href="#">isEqual</a>	Метод позволяет сравнить пару сертификатов

<a href="#">IsValid</a>	Метод позволяет проверить сертификат. В случае отсутствия профиля (свойство Profile) выполняется построение цепочки сертификации до доверенного УЦ и проверка по локальному СОС. Если профиль установлен, то уровень проверки сертификата (локальный СОС или получение из УЦ) берется из него
<a href="#">Load</a>	Метод позволяет загрузить данные сертификата с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.
<a href="#">Save</a>	Метод позволяет сохранять сертификат по указанному идентификатору ресурса (URI).

**Свойства**

Свойство	Описание
<a href="#">AuthorityKeyId</a>	Уникальный идентификатор, связанный с ключевой парой сертификата. Служит для того, чтобы можно было отличить сертификаты [издателя] имеющие одинаковые DN и разные ключи.ключи.
<a href="#">CertContext</a>	Свойство позволяет получить/установить контекст сертификата
<a href="#">CertificationPath</a>	Свойство позволяет получить путь сертификации (цепочку сертификатов) данного сертификата
<a href="#">ContainerName</a>	Свойство позволяет получить/установить имя контейнера сертификата (если сертификат имеет связку с закрытым ключом)
<a href="#">EKU</a>	Свойство позволяет получить список идентификаторов расширенного использования открытого ключа сертификата в виде XML строки
<a href="#">ID</a>	Свойство позволяет получить уникальный идентификатор сертификата – хэш от отличительного имени издателя и серийного номера сертификата по алгоритму MD5
<a href="#">IssuerName</a>	Свойство позволяет получить/установить отличительное имя издателя сертификата
<a href="#">KeySpec</a>	Свойство позволяет получить/установить тип ключа сертификата
<a href="#">KU</a>	Свойство позволяет получить доступ к информации поля «Использование ключа» сертификата
<a href="#">Policies</a>	Свойство позволяет получить список идентификаторов политик сертификата.
<a href="#">Profile</a>	Свойство позволяет получить/установить профиль с настройками
<a href="#">ProviderName</a>	Свойство позволяет получить/установить имя криптопровайдера, который управляет контейнером сертификата
<a href="#">ProviderType</a>	Свойство позволяет получить/установить тип криптопровайдера, который управляет контейнером сертификата
<a href="#">PublicKey</a>	Свойство позволяет получить значение открытого ключа в форматах DER или BASE64
<a href="#">PublicKeyAlg</a>	Свойство позволяет получить идентификатор алгоритма открытого ключа
<a href="#">SerialNumber</a>	Свойство позволяет получить/установить серийный номер сертификата
<a href="#">SubjectKeyId</a>	Уникальный идентификатор, связанный с ключевой парой данного сертификата.
<a href="#">SubjectName</a>	Свойство позволяет получить отличительное имя владельца сертификата
<a href="#">ThumbPrint</a>	Свойство позволяет получить значение хэша всего сертификата по алгоритму SHA-1
<a href="#">validFrom</a>	Свойство позволяет получить дату и время начала действия сертификата
<a href="#">validTo</a>	Свойство позволяет получить дату и время окончания действия сертификата

**ICERTIFICATE::DISPLAY [C++]****CERTIFICATE::DISPLAY [VBS]**

[C++]

Метод Display позволяет показать форму сертификата.

HRESULT Display(

```
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

[VBS]

Метод Display позволяет показать форму сертификата.

```
object.Display()
```

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

## ICERTIFICATE::EXPORT [C++]

### CERTIFICATE::EXPORT [VBS]

[C++]

Метод Export позволяет осуществлять экспорт сертификата в строку.

```
HRESULT Export(  
    FORMAT fFormat  
    VARIANT_BOOL bBase64Headers,  
    BSTR* pbstrCert  
);
```

### Параметры

fFormat [in] Формат загружаемого сертификата (DER, Base64). Допустимыми значениями являются значения FORMAT.

bAddBase64Headers [in, optional, defaultvalue(0)] Обрамление заголовками (только для формата Base64).

pbstrCert [out, retval] Приемник экспортируемых из компонента данных.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен интерфейс ICertificate, например из коллекции сертификатов.

[VBS]

Метод Export позволяет осуществлять экспорт сертификата в строку.

```
object.Export(fFormat, bAddBase64Headers)
```

### Параметры

fFormat Формат загружаемого сертификата (DER, Base64). Допустимыми значениями являются значения FORMAT.

bAddBase64Headers Обрамление заголовками (только для формата Base64).

### Возвращаемое значение

Строка с экспортируемыми из компонента данными.



## **Примечания**

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

**ICertificate::Import [C++]****Certificate::Import [VBS]**

[C++]

Метод Import позволяет осуществлять импорт сертификата из строки.

```
HRESULT Import(  
    BSTR bstrCert  
);
```

**Параметры**

bstrCert [in] Источник импортируемых данных, расположенный в памяти.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет осуществлять импорт сертификата из строки.

```
object.Import(bstrCert)
```

**Параметры**

bstrCert Источник импортируемых данных, расположенный в памяти.

**ICertificate::IsEqual [C++]****Certificate::IsEqual [VBS]**

[C++]

Метод **IsEqual** позволяет сравнить пару сертификатов.

```
HRESULT IsEqual(  
    ICertificate* pCertificate,  
    VARIANT_BOOL bIssuerAndSerialOnly,  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

pCertificate [in] Сертификат, который необходимо сравнить с сертификатом, загруженным ранее в данный объект.

bIssuerAndSerialOnly [in, optional, defaultvalue(-1)] Сравнивать только издателя и серийный номер, иначе будут сравниваться контексты.

pVal [out, retval] Результат сравнения.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**Примечания**

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

[VBS]

Метод **IsEqual** позволяет сравнить пару сертификатов.

```
object.IsEqual(pCertificate, bIssuerAndSerialOnly)
```

### Параметры

pCertificate Сертификат, который необходимо сравнить с сертификатом, загруженным ранее в данный объект.

bIssuerAndSerialOnly [in, optional, defaultvalue(-1)] Сравнивать только издателя и серийный номер, иначе будут сравниваться контексты.

### Возвращаемое значение

Результат сравнения.

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

## ICERTIFICATE::ISVALID [C++]

## CERTIFICATE::ISVALID [VBS]

[C++]

Метод isValid позволяет проверить сертификат. Выполняется: построение цепочки сертификации до доверенного УЦ, проверка сертификата по СОС.

```
HRESULT isValid(  
    POLICY_TYPE enmFilterType  
    VERIFYSTATUS* vsStatus  
);
```

### Параметры

enmFilterType [out, retval] Тип политики, учитываемый при проверке статуса сертификатов. Допустимыми значениями являются значения POLICY\_TYPE.

vsStatus [out, retval] Статус сертификата. Допустимыми значениями являются значения VERIFYSTATUS.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

[VBS]

Метод isValid позволяет проверить сертификат. Выполняется: построение цепочки сертификации до доверенного УЦ, проверка сертификата по СОС.

```
object.isValid
```

### Возвращаемое значение

Статус сертификата. Допустимыми значениями являются значения VERIFYSTATUS.

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

**ICERTIFICATE::LOAD [C++]****CERTIFICATE::LOAD [VBS]****[C++]**

Метод Load позволяет загрузить данные сертификата с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

```
HRESULT Load(  
    BSTR bstrURI  
);
```

**Параметры**

bstrURI [in] URI (имя файла), откуда необходимо загрузить сертификат.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Load позволяет загрузить данные сертификата с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

```
object.Load(bstrURI)
```

**Параметры**

bstrURI  
URI (имя файла), откуда необходимо загрузить сертификат.

**ICERTIFICATE::SAVE [C++]****CERTIFICATE::SAVE [VBS]****[C++]**

Метод Save позволяет сохранять сертификат по указанному идентификатору ресурса (URI).

```
HRESULT Save(  
    BSTR bstrURI,  
    FORMAT fFormat  
    VARIANT_BOOL bBase64Headers  
);
```

**Параметры**

bstrURI [in] URI (имя файла), куда необходимо сохранить сертификат  
fFormat [in] Указывает формат сохранения данных сертификата (DER или Base64). Допустимыми значениями являются значения FORMAT.  
bAddBase64Headers [in, optional, defaultvalue(-1)] Обрамление заголовками (только для формата Base64).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Save позволяет сохранять сертификат по указанному идентификатору ресурса (URI).

```
object.Save(bstrURI, fFormat, bAddBase64Headers)
```

## Параметры

bstrURI URI (имя файла), куда необходимо сохранить сертификат.  
fFormat Указывает формат сохранения данных сертификата. Допустимыми значениями являются значения FORMAT.  
bAddBase64Headers Обрамление заголовками (только для формата Base64).

## ICERTIFICATE::AUTHORITYKEYID [C++]

## CERTIFICATE::AUTHORITYKEYID [VBS]

[C++]

Свойство AuthorityKeyId - это уникальный идентификатор, связанный с ключевой парой сертификата. Служит для того, чтобы можно было отличить сертификаты [издателя] имеющие одинаковые DN и разные ключи.ключи.

```
HRESULT get_AuthorityKeyId(  
    FORMAT frmtFormat  
    VARIANT* pVal  
);
```

## Параметры

frmtFormat [in, defaultvalue(HEX\_TYPE)] Формат выходных данных. Может принимать значения из набора FORMAT.  
pVal [out, retval] Возвращаемое значение - указатель на буфер с данными.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство AuthorityKeyId - это уникальный идентификатор, связанный с ключевой парой сертификата. Служит для того, чтобы можно было отличить сертификаты [издателя] имеющие одинаковые DN и разные ключи.ключи.

```
object.AuthorityKeyId( frmtFormat )
```

## Параметры

frmtFormat Формат выходных данных. Может принимать значения из набора FORMAT.

## Возвращаемые значения

Буфер с данными.

## ICERTIFICATE::CERTCONTEXT [C++]

## CERTIFICATE::CERTCONTEXT [VBS]

[C++]

Свойство CertContext позволяет получить/установить CryptoAPI-контекст сертификата. В случае установки нового значения вручную, автоматически инициализируются свойства IssuerName и SerialNumber.

```
HRESULT get_CertContext(  
    long* pVal  
);  
HRESULT put_CertContext(  
    long* pVal  
);
```

## Параметры

pVal [out, retval] Указатель на CryptoAPI-контекст сертификата.  
pVal [in] Указатель на CryptoAPI-контекст сертификата

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CertContext позволяет получить/установить CryptoAPI-контекст сертификата. В случае установки нового значения вручную, автоматически инициализируются свойства IssuerName и SerialNumber.

```
object.CertContext [= Value]
```

## Параметры

Value Указатель на CryptoAPI-контекст сертификата.

## ICERTIFICATE::CERTIFICATIONPATH [C++]

## CERTIFICATE::CERTIFICATIONPATH [VBS]

[C++]

Свойство CertificationPath позволяет получить путь сертификации (цепочку сертификатов) данного сертификата.

```
HRESULT get_CertificationPath(  
    LONG lIndex  
    ICertificates** pVal  
);
```

## Параметры

lIndex [in, defaultvalue(0)] - порядковый номер построенной цепочки (может быть найдено несколько путей сертификата, полученных с использованием нескольких кросс-сертификатов). По умолчанию = 0. В текущей версии не учитывается (зарезервировано на будущее).  
pVal [out, retval] - Указатель на указатель на коллекцию сертификатов ICertificates, содержащую цепочку сертификатов.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство **CertificationPath** позволяет получить путь сертификации (цепочку сертификатов) данного сертификата.

```
object.CertificationPath [lIndex]
```

## Параметры

lIndex  
[in, defaultvalue(0)] - порядковый номер построенной цепочки (может быть найдено несколько путей сертификата, полученных с использованием нескольких кросс-сертификатов). По умолчанию = 0. В текущей версии не учитывается (зарезервировано на будущее).

## Возвращаемые значения

Возвращает указатель на интерфейс ICertificates.

**ICERTIFICATE::CONTAINERNAME [C++]****CERTIFICATE::CONTAINERNAME [VBS]****[C++]**

Свойство ContainerName позволяет получить/установить имя контейнера сертификата. Получение имени криптопровайдера возможно, только если сертификат имеет связь с закрытым ключом.

```
HRESULT put_ContainerName(  
    BSTR pVal  
);  
HRESULT get_ContainerName(  
    BSTR* pVal  
);
```

**Параметры**

pVal [in] Имя контейнера.  
pVal [out, retval] Имя контейнера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ContainerName позволяет получить/установить имя контейнера сертификата. Получение имени криптопровайдера возможно только если сертификат имеет связь с закрытым ключом.

```
object.ContainerName[=pVal]
```

**Параметры**

Имя контейнера.

**Возвращаемое значение**

Имя контейнера.

**ICERTIFICATE::EKU [C++]****CERTIFICATE::EKU [VBS]****[C++]**

Свойство ECU позволяет получить список идентификаторов расширенного использования открытого ключа сертификата в виде XML строки.

```
HRESULT get_EKU(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] Строка, содержащая список объектных идентификаторов расширенного использования открытого ключа в XML виде.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ECU позволяет получить список идентификаторов расширенного использования открытого ключа сертификата в виде XML строки.

`object.EKU`**Возвращаемое значение**

Строка, содержащая список объектных идентификаторов расширенного использования открытого ключа в XML виде.

**ICERTIFICATE::ID [C++]****CERTIFICATE::ID [VBS]**

[C++]

Свойство ID позволяет получить уникальный идентификатор сертификата – хэш от отличительного имени издателя и серийного номера сертификата по алгоритму MD5.

```
HRESULT get_ID(  
    VARIANT* pVal  
);
```

**Параметры**

pVal [out, retval] Указатель на возвращаемое значение. Идентификатор сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ID позволяет получить уникальный идентификатор сертификата – хэш от отличительного имени издателя и серийного номера сертификата по алгоритму MD5.

`object.ID`**Возвращаемое значение**

Идентификатор сертификата.

**ICERTIFICATE::ISSUERNAME [C++]****CERTIFICATE::ISSUERNAME [VBS]**

[C++]

Свойство IssuerName позволяет получить/установить отличительное имя издателя сертификата.

```
HRESULT get_IssuerName(  
    BSTR* pVal  
);  
HRESULT put_IssuerName(  
    BSTR pVal  
);
```



### Параметры

pVal [out, retval] Отличительное имя в виде строки.  
pVal [in] Отличительное имя в виде строки.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство IssuerName позволяет получить/установить отличительное имя издателя сертификата.

```
object.IssuerName [= pVal]
```

### Параметры

pVal Отличительное имя в виде строки.

## ICERTIFICATE::KEYSPEC [C++]

## CERTIFICATE::KEYSPEC [VBS]

[C++]

Свойство KeySpec позволяет получить/установить тип ключа сертификата. Получение типа ключа возможно только если сертификат имеет связь с закрытым ключом.

```
HRESULT put_KeySpec(  
    Long pVal,  
);  
HRESULT get_KeySpec(  
    Long* pVal,  
);
```

### Параметры

pVal [in] - устанавливаемый тип ключа: 1- Шифрование; 2- Подпись; 3- Шифрование и подпись.

pVal [out, retval] - возвращаемый тип ключа: 1- Шифрование, Шифрование и подпись; 2- Подпись.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство KeySpec позволяет получить/установить тип ключа сертификата. Получение типа ключа возможно только если сертификат имеет связь с закрытым ключом.

```
object.KeySpec [=pVal]
```

### Параметры

Устанавливаемый тип ключа: 1 - Шифрование; 2- Подпись; 3- Шифрование и подпись.

### Возвращаемое значение

Возвращаемый тип провайдера: 1 - Шифрование, Шифрование и подпись; 2- Подпись.

**ICERTIFICATE::KU [C++]****CERTIFICATE::KU [VBS]****[C++]**

Свойство KU позволяет получить доступ к информации поля «Использование ключа» сертификата.

```
HRESULT get_KU(  
    VARIANT* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение использования ключа. Представляет собой сумму битов:

Значение	Описание
CERT_DIGITAL_SIGNATURE_KEY_USAGE = 0x80	Цифровая подпись
CERT_NON_REPUDIATION_KEY_USAGE = 0x40	Неотрекаемость
CERT_KEY_ENCRYPTMENT_KEY_USAGE = 0x20	Шифрование ключа
CERT_DATA_ENCRYPTMENT_KEY_USAGE = 0x10	Шифрование данных
CERT_KEY_AGREEMENT_KEY_USAGE = 0x08	Согласование ключей
CERT_KEY_CERT_SIGN_KEY_USAGE = 0x04	Подпись ключа сертификата
CERT_CRL_SIGN_KEY_USAGE = 0x02	Автономное подписание списков отзыва
CERT_ENCRYPT_ONLY_KEY_USAGE = 0x01	Только шифрование

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство KU позволяет получить доступ к информации поля «Использование ключа» сертификата.

```
object.KU
```

**Возвращаемое значение**

Сумму битов, определяющих назначение ключа.

Значение	Описание
CERT_DIGITAL_SIGNATURE_KEY_USAGE = &H0x80	Цифровая подпись
CERT_NON_REPUDIATION_KEY_USAGE = &H0x40	Неотрекаемость
CERT_KEY_ENCRYPTMENT_KEY_USAGE = &H0x20	Шифрование ключа
CERT_DATA_ENCRYPTMENT_KEY_USAGE = &H0x10	Шифрование данных
CERT_KEY_AGREEMENT_KEY_USAGE = &H0x08	Согласование ключей
CERT_KEY_CERT_SIGN_KEY_USAGE = &H0x04	Подпись ключа сертификата
CERT_CRL_SIGN_KEY_USAGE = &H0x02	Автономное подписание списков отзыва
CERT_ENCRYPT_ONLY_KEY_USAGE = &H0x01	Только шифрование

**ICERTIFICATE::POLICIES [C++]****CERTIFICATE::POLICIES [VBS]**

[C++]

Свойство Policies позволяет получить список идентификаторов политик сертификата (OID).

```
HRESULT get_Policies(  
    IOIDs** pVal  
);
```

**Параметры**

pVal [out, retval] - полученная коллекция политик сертификата (OID).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Policies позволяет получить список идентификаторов политик сертификата (OID).

```
object.Policies
```

**Возвращаемое значение**

Коллекция, содержащая список объектных идентификаторов политик сертификата (OID).

**ICERTIFICATE::PROFILE [C++]****CERTIFICATE::PROFILE [VBS]**

[C++]

Свойство Profile позволяет получить/установить профиль с настройками.

```
HRESULT get_Profile(  
    IProfile** pVal,  
);  
HRESULT put_Profile(  
    IProfile* newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на интерфейс IProfile.  
newVal [in] - указатель на интерфейс IProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет получить/установить профиль с настройками.

```
object.Profile [= Value]
```

**Параметры**

Value - указатель на интерфейс Profile.

**ICERTIFICATE::PROVIDERNAME [C++]****CERTIFICATE::PROVIDERNAME [VBS]**

[C++]

Свойство ProviderName позволяет получить/установить имя криптопровайдера, который управляет контейнером сертификата.

```
HRESULT put_ProviderName(  
    BSTR pVal,  
);  
HRESULT get_ProviderName(  
    BSTR* pVal  
);
```

**Параметры**

pVal [in] - строка, содержащая имя криптопровайдера.  
pVal [out, retval] - имя криптопровайдера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProviderName позволяет получить/установить имя криптопровайдера, который управляет контейнером сертификата.

```
object.ProviderName [=pVal]
```

**Параметры**

Устанавливаемое значение - строка, содержащая имя криптопровайдера.

**Возвращаемое значение**

Возвращаемое значение - имя криптопровайдера.

**ICERTIFICATE::PROVIDERTYPE [C++]****CERTIFICATE::PROVIDERTYPE [VBS]**

[C++]

Свойство ProviderType позволяет получить/установить тип криптопровайдера, управляющего ключевым контейнером сертификата. Получение типа криптопровайдера возможно, только если сертификат имеет связь с закрытым ключом.

```
HRESULT put_ProviderType(  
    Long pVal,  
);  
HRESULT get_ProviderType(  
    Long* pVal,  
);
```

**Параметры**

pVal [in] - устанавливаемый номер типа провайдера.  
pVal [out, retval] - возвращаемый номер типа провайдера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ProviderType позволяет установить имя криптопровайдера, управляющего ключевым контейнером сертификата. Получение типа криптопровайдера возможно, только если сертификат имеет связку с закрытым ключом.

```
object.ProviderType [=pVal]
```

**Параметры**

Устанавливаемый номер типа провайдера.

**Возвращаемое значение**

Возвращаемый номер типа провайдера.

**ICERTIFICATE::PUBLICKEY [C++]****CERTIFICATE::PUBLICKEY [VBS]****[C++]**

Свойство PublicKey позволяет получить значение открытого ключа в форматах DER или BASE64.

```
HRESULT get_PublicKey(  
    FORMAT frmtFormat,  
    BSTR* pVal  
);
```

**Параметры**

frmtFormat [in] - форматDER или BASE64. Допустимыми значениями являются значения FORMAT.

pVal [out, retval] - строка со значением открытого ключа в форматах DER или BASE64.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство PublicKey позволяет получить значение открытого ключа в форматах DER или BASE64.

```
object.PublicKey (frmtFormat)
```

**Параметры**

frmtFormat - форматDER или BASE64. Допустимыми значениями являются значения FORMAT.

**Возвращаемое значение**

Строка со значением открытого ключа в форматах DER или BASE64.

**ICERTIFICATE::PUBLICKEYALG [C++]****CERTIFICATE::PUBLICKEYALG [VBS]****[C++]**

Свойство PublicKeyAlg позволяет получить идентификатор алгоритма открытого ключа.

```
HRESULT get_PublicKeyAlg(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - идентификатор алгоритма.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PublicKeyAlg позволяет получить идентификатор алгоритма открытого ключа.

object.PublicKeyAlg

**Возвращаемое значение**

Идентификатор алгоритма.

**ICERTIFICATE::SERIALNUMBER [C++]****CERTIFICATE::SERIALNUMBER [VBS]**

[C++]

Свойство SerialNumber позволяет получить/установить серийный номер сертификата.

```
HRESULT get_SerialNumber(  
    BSTR* pVal  
);  
HRESULT put_SerialNumber(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - серийный номер в виде строки.  
pVal [in] - серийный номер в виде строки.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SerialNumber позволяет получить/установить серийный номер сертификата.

object.SerialNumber [= pVal]

**Параметры**

pVal - серийный номер в виде строки.

**ICERTIFICATE::SIGNATUREALG [C++]****CERTIFICATE::SIGNATUREALG [VBS]**

[C++]

Свойство SignatureAlg позволяет получить алгоритм подписи сертификата.

```
HRESULT get_SignatureAlg(  
    BSTR* pVal,  
);
```

**Параметры**

pVal [out, retval] - возвращает строку с OID-ом алгоритма подписи сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureAlg позволяет получить алгоритм подписи сертификата.

object.SignatureAlg

### Параметры

Value - возвращает строку с OID-ом алгоритма подписи сертификата.

## ICERTIFICATE::SUBJECTKEYID [C++]

### CERTIFICATE::SUBJECTKEYID [VBS]

[C++]

Свойство SubjectKeyId - это уникальный идентификатор, связанный с ключевой парой данного сертификата.

```
HRESULT get_SubjectKeyId(  
    FORMAT frmtFormat ,  
    VARIANT* pVal  
);
```

### Параметры

frmtFormat [in, defaultvalue(HEX\_TYPE)] - формат выходных данных. Может принимать значения из набора FORMAT.  
pVal [out, retval] - возвращаемое значение - указатель на буфер с данными.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SubjectKeyId - это уникальный идентификатор, связанный с ключевой парой данного сертификата.

object.SubjectKeyId frmtFormat

### Параметры

frmtFormat - формат выходных данных. Может принимать значения из набора FORMAT.

### Возвращаемые значения

Буфер с данными.

## ICERTIFICATE::SUBJECTNAME [C++]

### CERTIFICATE::SUBJECTNAME [VBS]

[C++]

Свойство SubjectName позволяет получить/установить отличительное имя владельца сертификата.

```
HRESULT get_SubjectName (
```

```
BSTR* pVal  
);  
  
HRESULT put_SubjectName(  
    BSTR pVal  
);
```

#### Параметры

pVal [in] - отличительное имя в виде строки.  
pVal [out, retval] - отличительное имя в виде строки.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SubjectName позволяет получить/установить отличительное имя владельца сертификата.

object.SubjectName [pVal]

#### Параметры

pVal - отличительное имя в виде строки.

### ICERTIFICATE::THUMBPRINT [C++]

### CERTIFICATE::THUMBPRINT [VBS]

[C++]

Свойство ThumbPrint позволяет получить значение хэша всего сертификата по алгоритму SHA-1.

```
HRESULT get_ThumbPrint(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] - значение хэша.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ThumbPrint позволяет получить значение хэша всего сертификата по алгоритму SHA-1.

object.ThumbPrint

#### Возвращаемое значение

Значение хэша.

### ICERTIFICATE::VALIDFROM [C++]

### CERTIFICATE::VALIDFROM [VBS]

[C++]

Свойство validFrom позволяет получить дату и время начала действия сертификата.

```
HRESULT get_validFrom(  
    DATE* pVal
```



```
);
```

### Параметры

pVal [out, retval] - дата и время начала срока действия сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство validFrom позволяет получить дату и время начала действия сертификата.

```
object.validFrom
```

### Возвращаемое значение

Дата и время начала срока действия сертификата.

## ICERTIFICATE::VALIDTO [C++]

## CERTIFICATE::VALIDTO [VBS]

[C++]

Свойство validTo получить дату и время окончания действия сертификата.

```
HRESULT get_validTo(  
    DATE* pVal  
);
```

### Параметры

pVal [out, retval] - дата и время окончания срока действия сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство validTo получить дату и время окончания действия сертификата.

```
object.validTo
```

### Возвращаемое значение

Дата и время окончания срока действия сертификата.

## CERTIFICATES (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

### Методы

Метод	Описание
<a href="#">Add</a>	Метод позволяет добавлять сертификаты в коллекцию
<a href="#">Remove</a>	Метод позволяет удалять сертификаты из коллекции

### Свойства

Свойство	Описание
<a href="#">Certificate</a>	Свойство позволяет получить сертификат из коллекции
<a href="#">Count</a>	Свойство позволяет получить количество объектов типа Certificate в коллекции
<a href="#">Item</a>	Свойство позволяет получить указатель на объект типа Certificate

**ICERTIFICATES::ADD [C++]****CERTIFICATES::ADD [VBS]****[C++]**

Метод Add позволяет добавлять сертификаты в коллекцию.

```
HRESULT Add(  
    ICertificate* addVal  
);
```

**Параметры**

addVal [in] - указатель на интерфейс типа ICertificate.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**Примечания**

Вызов метода возможен после того, как получен интерфейс ICertificate, например из коллекции сертификатов.

**[VBS]**

Метод Add позволяет добавлять сертификаты в коллекцию.

```
object.Add(addVal)
```

**Параметры**

addVal Указатель на объект типа Certificate.

**Примечания**

Вызов метода возможен после того, как получен объект Certificate, например из коллекции сертификатов.

**ICERTIFICATES::REMOVE [C++]****CERTIFICATES::REMOVE [VBS]****[C++]**

Метод Remove позволяет удалять сертификаты из коллекции.

```
HRESULT Remove(  
    long lIndex  
);
```

**Параметры**

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1]

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Remove позволяет удалять сертификаты из коллекции.

```
object.Remove(lIndex)
```

**Параметры**

lIndex Номер элемента в коллекции минус 1 [0 .. Count-1]

**ICERTIFICATES::CERTIFICATE [C++]****CERTIFICATES::CERTIFICATE [VBS]**

[C++]

Свойство Certificate позволяет получить сертификат из коллекции.

```
HRESULT get_Certificate(  
    VARIANT vtIssuerName,  
    VARIANT vtSerialNumber,  
    VARIANT* pCertificate  
);
```

**Параметры**

vtIssuerName [in] - отличительное имя издателя искомого сертификата (полный DN).  
vtSerialNumber [in] - серийный номер искомого сертификата.  
pCertificate [out, retval] - в случае успешного поиска содержит указатель на ICertificate, в противном случае - пустую ссылку.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Certificate позволяет получить сертификат из коллекции.

```
object.Certificate (vtIssuerName, vtSerialNumber)
```

**Параметры**

vtIssuerName - отличительное имя издателя искомого сертификата (полный DN).  
vtSerialNumber - серийный номер искомого сертификата.

**Возвращаемое значение**

В случае успешного поиска содержит указатель на Certificate, в противном случае - пустую ссылку ("Nothing").

**ICERTIFICATES::COUNT [C++]****CERTIFICATES::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество объектов типа Certificate в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

**Параметры**

pVal [out, retval] - количество сертификатов в коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество объектов типа Certificate в коллекции.

```
object.Count
```

**Возвращаемое значение**

Количество сертификатов в коллекции.

**ICERTIFICATES::ITEM [C++]****CERTIFICATES::ITEM [VBS]**

[C++]

Свойство Item позволяет получить указатель на интерфейс типа ICertificate.

```
HRESULT get_Item(  
    long lIndex,  
    ICertificate** pVal  
);
```

**Параметры**

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1]  
pVal [out, retval] - объект-приемник.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить указатель на объект типа Certificate.

```
object.Item (lIndex)
```

**Параметры**

lIndex Номер элемента в коллекции минус 1 [0 .. Count-1]

**Возвращаемое значение**

Объект-приемник.

**CERTIFICATESTORE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
Add	Метод позволяет добавить сертификат в хранилище
AddCRL	Метод позволяет добавить СОС в хранилище
AddCTL	Метод позволяет добавить СДС в открытое ранее хранилище.
Close	Метод позволяет закрыть открытое хранилище
Display	Метод позволяет отобразить форму просмотра хранилища сертификатов
DownloadCACertificates	Метод позволяет загрузить сертификаты из УЦ в хранилище Других пользователей
Export	Метод позволяет экспортировать сертификат из хранилища в строку
Import	Метод позволяет импортировать сертификат из строки в хранилище
Load	Метод позволяет загрузить сертификат(ы) из файла в хранилище
Open	Метод позволяет открыть хранилище сертификатов
Remove	Метод позволяет удалить сертификат из хранилища
RemoveCRL	Метод позволяет удалить СОС из хранилища
RemoveCTL	Метод позволяет удалить СДС из хранилища

RunImportWizard	Метод позволяет вызвать мастер импорта сертификата или СОС в хранилище
Save	Метод позволяет сохранить загруженное хранилище сертификатов в файл

### Свойства

Свойство	Описание
Certificate	Свойство позволяет получить сертификат из коллекции
CRLs	Свойство позволяет получить доступ к коллекции СОС открытого хранилища
CTLs	Свойство позволяет получить коллекцию СДС из открытого хранилища
Store	Свойство позволяет получить доступ к коллекции сертификатов открытого хранилища
Type	Свойство позволяет получить доступ к коллекции сертификатов открытого хранилища
Profile	Свойство позволяет получить/установить профиль с настройками

### ICERTIFICATESTORE::ADD [C++]

### CERTIFICATESTORE::ADD [VBS]

[C++]

Метод Add позволяет добавить сертификат в открытое ранее хранилище.

```
HRESULT Add(  
    ICertificate* pCertificate,  
    VARIANT_BOOL bByRequest  
);
```

### Параметры

pCertificate [in] – устанавливаемый сертификат.  
bByRequest [in, optional, defaultvalue(0)] – флаг поиска шаблона сертификата в хранилище запросов для получения имени контейнера ключевой пары.(True/False). В случае передачи значения TRUE можно установить личный сертификат с привязкой к контейнеру КП, информация о котором берется из шаблона сертификата. Шаблон сертификата ищется в хранилище запросов по значению открытого ключа и после успешной установки сертификата удаляется из хранилища.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен интерфейс ICertificate, например, импортом из файла.

[VBS]

Метод Add позволяет добавить сертификат в открытое ранее хранилище.

```
object.Add(pCertificate, bByRequest)
```

### Параметры

pCertificate – устанавливаемый сертификат.  
bByRequest – флаг поиска шаблона сертификата в хранилище запросов для получения имени контейнера ключевой пары.(True/False). В случае передачи значения TRUE можно установить личный сертификат с привязкой к контейнеру КП, информация о котором берется из шаблона сертификата. Шаблон сертификата ищется в хранилище запросов по значению открытого ключа и после успешной установки сертификата удаляется из хранилища.

### Примечания

Вызов метода возможен после того, как получен объект Certificate, например, импортом из файла.

## ICERTIFICATESTORE::ADDCTL [C++]

## CERTIFICATESTORE::ADDCTL [VBS]

[C++]

Метод AddCRL позволяет добавить СОС в открытое ранее хранилище.

```
HRESULT AddCRL(  
    ICRL* pCRL,  
);
```

### Параметры

pCRL [in] - устанавливаемый СОС.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен интерфейс ICRL, например, импортом из файла.

[VBS]

Метод AddCRL позволяет добавить СОС в открытое ранее хранилище.

```
object.AddCRL(pCertificate)
```

### Параметры

pCertificate - устанавливаемый СОС.

### Примечания

Вызов метода возможен после того, как получен объект CRL, например, импортом из файла.

## ICERTIFICATESTORE::ADDCTL [C++]

## CERTIFICATESTORE::ADDCTL [VBS]

[C++]

Метод AddCTL позволяет добавить СДС в открытое ранее хранилище.

```
HRESULT AddCTL(  
    ICTL* pCTL,  
);
```

### Параметры

pCTL [in] - устанавливаемый СДС.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен интерфейс ICTL, например, импортом из файла.

**[VBS]**

Метод AddCTL позволяет добавить СДС в открытое ранее хранилище.

```
object.AddCTL(pCTL)
```

**Параметры**

pCTL – устанавливаемый СДС.

**Примечания**

Вызов метода возможен после того, как получен объект CTL, например, импортом из файла.

**ICERTIFICATESTORE::CLOSE [C++]****CERTIFICATESTORE::CLOSE [VBS]****[C++]**

Метод Close позволяет закрыть открытое хранилище.

```
HRESULT Close(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Close позволяет закрыть открытое хранилище.

```
object.Close()
```

**ICERTIFICATESTORE::DISPLAY [C++]****CERTIFICATESTORE::DISPLAY [VBS]****[C++]**

Метод Display позволяет отобразить форму просмотра хранилища сертификатов.

```
HRESULT Display(  
    long IStoreMask,  
    POLICY_TYPE enmFilterType  
    VARIANT_BOOL bMultiSelect  
    ICertificates** pVal  
);
```

**Параметры**

IStoreMask [in] – битовая маска хранилищ, предназначенных для отображения (закладками). Возможные значения из SYSTEM\_STORE\_LOCATION.  
enmFilterType [in, optional, defaultvalue(0)] – тип политики для проверки статуса отображаемых сертификатов. Возможные значения из POLICY\_TYPE.  
bMultiSelect [in, optional, defaultvalue(0)] – флаг включения режима выбора нескольких сертификатов.  
pVal [out, retval] – коллекция выбранных сертификатов, которая в случае отказа от выбора не содержит элементов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет отобразить форму просмотра хранилища сертификатов.

```
object.Display(IStoreMask, enmFilterType, bMultiSelect)
```

#### Параметры

IStoreMask - битовая маска хранилищ, предназначенных для отображения (закладками). Возможные значения из SYSTEM\_STORE\_LOCATION.  
enmFilterType [optional, defaultvalue(0)] - тип политики для проверки статуса отображаемых сертификатов. Возможные значения из POLICY\_TYPE.  
bMultiSelect [optional, defaultvalue(0)] - флаг включения режима выбора нескольких сертификатов.

#### Возвращаемое значение

Объект типа Certificates, содержащий коллекцию выбранных сертификатов, в случае отказа от выбора содержит пустую коллекцию.

### ICERTIFICATESTORE::DOWNLOADCACERTIFICATES [C++]

### CERTIFICATESTORE::DOWNLOADCACERTIFICATES [VBS]

[C++]

Метод DownloadCACertificates загружает сертификаты пользователей УЦ в хранилище Других пользователей.

```
HRESULT DownloadCACertificates(  
    CA_TYPE enmTypeCA,  
    ICertificates** ppCerts  
);
```

#### Параметры

enmTypeCA [in] - тип УЦ CA\_TYPE  
ppCerts [out, retval] - возвращаемое значение - коллекция сертификатов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод DownloadCACertificates загружает сертификаты пользователей УЦ в хранилище Других пользователей.

```
object.DownloadCACertificates(enmTypeCA)
```

#### Параметры

enmTypeCA - тип УЦ CA\_TYPE

#### Возвращаемое значение

Возвращаемое значение - коллекция сертификатов.

### ICERTIFICATESTORE::EXPORT [C++]

### CERTIFICATESTORE::EXPORT [VBS]

[C++]

Метод Export позволяет экспортировать хранилище в строку.



```
HRESULT Export(  
    STORETYPE stType,  
    FORMAT fFormat,  
    BSTR* pVal  
);
```

#### Параметры

stType [in] - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE.  
fFormat [in] - формат выходного файла. Допустимыми значениями являются значения FORMAT.  
pVal [out, retval] - строка с экспортированными данными.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать хранилище в строку.

```
object.Export(stType, fFormat)
```

#### Параметры

stType - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE.  
fFormat - формат выходного файла. Допустимыми значениями являются значения FORMAT.

#### Возвращаемое значение

Возвращаемое значение - строка с экспортированными данными.

### ICERTIFICATESTORE::IMPORT [C++]

### CERTIFICATESTORE::IMPORT [VBS]

[C++]

Метод Import позволяет импортировать сертификат из строки в хранилище.

```
HRESULT Import(  
    STORETYPE stType,  
    BSTR bstrSource  
);
```

#### Параметры

stType [in] - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_AUTO\_DETECT, STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE.  
bstrSource [in] - источник импортируемых данных, расположенный в памяти.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать сертификат из строки в хранилище.

```
object.Import(stType, bstrSource)
```

#### Параметры

stType - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_AUTO\_DETECT, STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE  
bstrSource - источник импортируемых данных, расположенный в памяти.

**Возвращаемое значение**

Возвращаемое значение - коллекция сертификатов.

**ICERTIFICATESTORE::LOAD [C++]****CERTIFICATESTORE::LOAD [VBS]**

[C++]

Метод Load загружает сертификат(ы) из файла в хранилище.

```
HRESULT Load(  
    STORETYPE stType,  
    BSTR bstrSourceURI  
);
```

**Параметры**

stType [in] - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_AUTO\_DETECT, STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE  
bstrSourceURI [in] - путь к файлу с сертификатом.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load загружает сертификат(ы) из файла в хранилище.

```
object.Load(stType, bstrSourceURI)
```

**Параметры**

stType - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_AUTO\_DETECT, STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE  
bstrSourceURI - путь к файлу с сертификатом.

**Возвращаемое значение**

Возвращаемое значение - коллекция сертификатов.

**ICERTIFICATESTORE::OPEN [C++]****CERTIFICATESTORE::OPEN [VBS]**

[C++]

Метод Open позволяет открыть хранилище сертификатов.

```
HRESULT Open(  
    STORETYPE stType,  
    BSTR bstrName  
);
```

**Параметры**

stType [in] - тип хранилища. Допустимыми значениями являются значения LOCAL\_MACHINE\_STORE, CURRENT\_USER\_STORE, STORE\_TYPE\_MEMORY из STORETYPE.  
bstrName [in] - название хранилища: "my", "root", "AddressBook", "trust", "ca", "request" (не зависит от регистра).

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет открыть хранилище сертификатов.

```
object.Open(stType, bstrName)
```

#### Параметры

stType - тип хранилища. Допустимыми значениями являются значения LOCAL\_MACHINE\_STORE, CURRENT\_USER\_STORE, STORE\_TYPE\_MEMORY из STORETYPE.  
bstrName - название хранилища: "my", "root", "AddressBook", "trust", "ca", "request" (не зависит от регистра).

### ICERTIFICATESTORE::REMOVE [C++]

#### CERTIFICATESTORE::REMOVE [VBS]

[C++]

Метод Remove позволяет удалить сертификат из хранилища.

```
HRESULT Remove(  
    ICertificate* pCertificate  
);
```

#### Параметры

pCertificate [in] - сертификат, который необходимо удалить.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить сертификат из хранилища.

```
object.Remove(pCertificate)
```

#### Параметры

pCertificate - сертификат, который необходимо удалить.

**ICERTIFICATESTORE::REMOVECRL [C++]****CERTIFICATESTORE::REMOVECRL [VBS]**

[C++]

Метод RemoveCRL позволяет удалить СОС из хранилища.

```
HRESULT RemoveCRL(  
    ICRL* pCRL  
);
```

**Параметры**

pCRL [in] – СОС, который необходимо удалить.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RemoveCRL позволяет удалить СОС из хранилища.

```
object.RemoveCRL(pCRL)
```

**Параметры**

pCertificate СОС, который необходимо удалить.

**ICERTIFICATESTORE::REMOVECTL [C++]****CERTIFICATESTORE::REMOVECTL [VBS]**

[C++]

Метод RemoveCTL позволяет удалить СДС из хранилища.

```
HRESULT RemoveCTL(  
    ICTL* pCTL  
);
```

**Параметры**

pCTL [in] – СДС, который необходимо удалить.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RemoveCTL позволяет удалить СДС из хранилища.

```
object.RemoveCTL(pCTL)
```

**Параметры**

pCTL СДС, который необходимо удалить.

**ICERTIFICATESTORE::RUNIMPORTWIZARD [C++]****CERTIFICATESTORE::RUNIMPORTWIZARD [VBS]**

[C++]

Метод RunImportWizard позволяет вызвать мастер импорта сертификата или СОС в хранилище.

```
HRESULT RunImportWizard(  
    long lFlags  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

lFlags [in, optional, defaultvalue(0)] – флаг может быть установлен равным любой комбинации нижеперечисленных флагов:

Значение	Описание
CRYPTUI_WIZ_IMPORT_NO_CHANGE_DEST_STORE = 0x00010000	Пользователь не будет иметь возможности изменять тип хранилища, куда будет установлен сертификат, на соответствующей странице мастера
CRYPTUI_WIZ_IMPORT_ALLOW_CERT = 0x00020000	Мастер будет позволять только импорт сертификатов
CRYPTUI_WIZ_IMPORT_ALLOW_CRL = 0x00040000	Мастер будет позволять только импорт СОС
CRYPTUI_WIZ_IMPORT_ALLOW_CTL = 0x00080000	Мастер будет позволять только импорт списков доверенных сертификатов
WIZ_IMPORT_SET_PRIVATE_CERT = 0x00000100	Мастер будет вызван с выставленным флагом установки личного сертификата
WIZ_IMPORT_NO_CHANGE_PRIVATE_CERT = 0x00000200	Мастер будет вызван с выставленным и недоступным для изменения флагом установки личного сертификата
WIZ_IMPORT_SET_CERT_INTO_CONTAINER = 0x00000400	Мастер будет вызван с выставленным флагом установки личного сертификата в контейнер
WIZ_IMPORT_NO_CHANGE_CERT_INTO_CONTAINER = 0x00000800	Мастер будет вызван с выставленным и недоступным для изменения флагом установки личного сертификата в контейнер
WIZ_IMPORT_USE_OPENED_STORE = 0x00001000	Будет запущен стандартный Windows мастер импорта сертификатов/СОС
CRYPTUI_WIZ_IMPORT_ENGINE = 0x80000000	Будет запущен стандартный Windows мастер импорта сертификатов/СОС

pVal [out, retval] – возвращает флаг успешного или неуспешного импорта.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RunImportWizard позволяет вызвать мастер импорта сертификата или СОС в хранилище.

```
object.RunImportWizard(lFlags)
```

**Параметры**

lFlags

Флаг может быть установлен равным любой комбинации нижеперечисленных флагов:

Значение	Описание
CRYPTUI_WIZ_IMPORT_NO_CHANGE_DEST_STORE = 0x00010000	Пользователь не будет иметь возможности изменять тип хранилища, куда будет установлен сертификат, на соответствующей странице мастера
CRYPTUI_WIZ_IMPORT_ALLOW_CERT = 0x00020000	Мастер будет позволять только импорт сертификатов
CRYPTUI_WIZ_IMPORT_ALLOW_CRL = 0x00040000	Мастер будет позволять только импорт СОС
CRYPTUI_WIZ_IMPORT_ALLOW_CTL = 0x00080000	Мастер будет позволять только импорт списков доверенных сертификатов
WIZ_IMPORT_SET_PRIVATE_CERT = 0x00000100	Мастер будет вызван с выставленным флагом установки личного сертификата
WIZ_IMPORT_NO_CHANGE_PRIVATE_CERT = 0x00000200	Мастер будет вызван с выставленным и недоступным для изменения флагом установки личного сертификата
WIZ_IMPORT_SET_CERT_INTO_CONTAINER = 0x00000400	Мастер будет вызван с выставленным флагом установки личного сертификата в контейнер
WIZ_IMPORT_NO_CHANGE_CERT_INTO_CONTAINER = 0x00000800	Мастер будет вызван с выставленным и недоступным для изменения флагом установки личного сертификата в контейнер
WIZ_IMPORT_USE_OPENED_STORE = 0x00001000	Будет запущен стандартный Windows мастер импорта сертификатов/СОС
CRYPTUI_WIZ_IMPORT_ENGINE = 0x80000000	Будет запущен стандартный Windows мастер импорта сертификатов/СОС

**Возвращаемые значения**

Возвращает флаг успешного или неуспешного импорта.

**ICERTIFICATESTORE::SAVE [C++]****CERTIFICATESTORE::SAVE [VBS]**

[C++]

Метод Save позволяет сохранить хранилище сертификатов в файл.

```
HRESULT Save(
    STORETYPE stType,
    FORMAT fFormat,
    BSTR* bstrDestURI
);
```

**Параметры**

stType [in] - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE.  
fFormat [in] - формат выходного файла. Допустимыми значениями являются значения FORMAT.  
bstrDestURI [in] - URI (имя файла), куда необходимо сохранить хранилище сертификатов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить хранилище сертификатов в файл.

```
object.Save(stType, fFormat, bstrDestURI)
```

#### Параметры

stType - тип хранилища. Допустимыми значениями являются значения STORE\_TYPE\_PKCS7, STORE\_TYPE\_SST, STORE\_TYPE\_PFX из STORETYPE.  
fFormat - формат выходного файла. Допустимыми значениями являются значения FORMAT.  
bstrDestURI - URI (имя файла), куда необходимо сохранить хранилище сертификатов.

### ICERTIFICATESTORE::CERTIFICATE [C++]

### CERTIFICATESTORE::CERTIFICATE [VBS]

[C++]

Свойство Certificate позволяет получить сертификат из коллекции.

```
HRESULT get_Certificate(  
    VARIANT vtIssuerName,  
    VARIANT vtSerialNumber,  
    VARIANT* pCertificate  
);
```

#### Параметры

vtIssuerName [in] - отличительное имя издателя искомого сертификата (полный DN).  
vtSerialNumber [in] - серийный номер искомого сертификата.  
pCertificate [out, retval] - в случае успешного поиска содержит указатель на ICertificate, в противном случае - пустая ссылка).

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Certificate позволяет получить сертификат из коллекции.

```
object.Certificate (vtIssuerName, vtSerialNumber)
```

#### Параметры

vtIssuerName - отличительное имя издателя искомого сертификата (полный DN).  
vtSerialNumber - серийный номер искомого сертификата.

#### Возвращаемое значение

В случае успешного поиска содержит указатель на Certificate, в противном случае - "Nothing".

### ICERTIFICATESTORE::CRLs [C++]

### CERTIFICATESTORE::CRLs [VBS]

[C++]

Свойство CRLs позволяет получить коллекцию СОС из открытого хранилища.

```
HRESULT get_CRLs(  
    ICRLs** pVal  
);
```

### Параметры

pVal [out, retval] - указатель на возвращаемое значение свойства (коллекцию СОС открытого хранилища).

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### [VBS]

Свойство CRLs позволяет получить коллекцию СОС из открытого хранилища.

```
object.CRLs
```

### Возвращаемое значение

Указатель на возвращаемое значение свойства (коллекцию СОС открытого хранилища).

## ICERTIFICATESTORE::CTLs [C++]

### CERTIFICATESTORE::CTLs [VBS]

### [C++]

Свойство CTLs позволяет получить коллекцию СДС из открытого хранилища.

```
HRESULT get_CTLs(  
    ICTLs** pVal  
);
```

### Параметры

pVal [out, retval] - указатель на возвращаемое значение свойства (коллекцию СДС открытого хранилища).

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### [VBS]

Свойство CTLs позволяет получить коллекцию СДС из открытого хранилища.

```
object.CTLs
```

### Возвращаемые значения

Указатель на возвращаемое значение свойства (коллекцию СДС открытого хранилища).

## ICERTIFICATESTORE::PROFILE [C++]

### CERTIFICATESTORE::PROFILE [VBS]

### [C++]

Свойство Profile позволяет получить/установить профиль с настройками.

```
HRESULT get_Profile(  
    IProfile** pVal  
);
```

```
HRESULT put_Profile(  
    IProfile* pVal
```



```
);
```

### Параметры

pVal [in] - указатель на интерфейс IProfile.  
pVal [out, retval] - указатель на интерфейс IProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов свойства возможен после того, как получен интерфейс IProfile, например из коллекции профилей.

[VBS]

Свойство Profile позволяет получить/установить профиль с настройками.

```
object.Profile [=pVal]
```

### Параметры

pVal - указатель на объект IProfile.

### Примечания

Вызов свойства возможен после того, как получен объект IProfile, например из коллекции профилей.

## ICERTIFICATESTORE::STORE [C++]

## CERTIFICATESTORE::STORE [VBS]

[C++]

Свойство Store позволяет получить коллекцию сертификатов из открытого хранилища.

```
HRESULT get_Store(  
    ICertificates** pVal  
);
```

### Параметры

pVal [out, retval] - указатель на возвращаемое значение свойства (коллекцию сертификатов открытого хранилища).

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Store позволяет получить коллекцию сертификатов из открытого хранилища.

```
object.Store
```

### Возвращаемое значение

Указатель на возвращаемое значение свойства (коллекцию сертификатов открытого хранилища).

**ICERTIFICATESTORE::TYPE [C++]****CERTIFICATESTORE::TYPE [VBS]**

[C++]

Свойство Type позволяет получить тип хранилища сертификатов.

```
HRESULT get_Type(  
    STORETYPE* pVal  
);
```

**Параметры**

pVal [out, retval] - тип хранилища. Допустимыми значениями являются значения из STORETYPE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Type позволяет получить тип хранилища сертификатов.

```
object.Type(pVal)
```

**Возвращаемое значение**

Тип хранилища. Допустимыми значениями являются значения из STORETYPE.

**CRL (GUID: D6D6A8C4-2E5E-41DD-9432-361926D46B24)****Методы**

Метод	Описание
Display	Метод позволяет открыть форму просмотра СОС
Export	Метод позволяет произвести экспорт СОС в строку
FindCertificate	Метод позволяет найти сертификат в списке отзыва
Import	Метод позволяет осуществить импорт СОС из строки
IsValid	Метод позволяет проверить статус СОС
Load	Метод позволяет загрузить данные СОС с указанного идентификатора ресурса (URI).
Save	Метод позволяет сохранить данные СОС по указанному идентификатору ресурса (URI)

**Свойства**

Свойство	Описание
CRLContext	Свойство позволяет получить/установить CryptoAPI-контекст СОС.
IssuerName	Свойство позволяет получить имя издателя СОС
NextUpdate	Свойство позволяет получить дату следующей публикации СОС
SignerCertificate	Свойство позволяет получить сертификат подписи под CRL из указанного хранилища.
ThisUpdate	Свойство позволяет получить дату текущей публикации СОС

**ICRL::DISPLAY [C++]****CRL::DISPLAY [VBS]**

[C++]

Метод Display отображает окно просмотра параметров СОС.

```
HRESULT Display(  
);
```

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display отображает окно просмотра параметров СОС.

```
object.Display()
```

#### ICRL::EXPORT [C++]

#### CRL::EXPORT [VBS]

[C++]

Метод Export позволяет произвести экспорт СОС в строку.

```
HRESULT Export(  
    FORMAT fFormat  
    VARIANT_BOOL bBase64Headers  
    BSTR* pVal  
);
```

#### Параметры

fFormat [in] – формат загружаемого СОС (DER, Base64). Допустимыми значениями являются значения FORMAT.  
bBase64Headers [in, defaultvalue(0)] – обрамление заголовками (только для формата Base64).  
pVal [out, retval] Строка – приемник экспортируемых из компонента данных.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

#### Примечания

Вызов метода возможен после того, как получен интерфейс ICRL, например из коллекции СОС

[VBS]

Метод Export позволяет произвести экспорт СОС в строку.

```
object.Export(fFormat, bAddBase64Headers)
```

#### Параметры

fFormat – формат загружаемого СОС (DER, Base64). Допустимыми значениями являются значения FORMAT.  
bAddBase64Headers – обрамление заголовками (только для формата Base64).

#### Возвращаемое значение

Строка с экспортируемыми из компонента данными.

#### Примечания

Вызов метода возможен после того, как получен объект CRL, например из коллекции СОС.

#### ICRL::FINDCERTIFICATE[C++]

## CRL::FINDCERTIFICATE[VBS]

[C++]

Метод FindCertificate позволяет произвести поиск сертификата в СОС.

```
HRESULT FindCertificate(  
    VARIANT varCertificate  
    ICRLentry** pVal  
);
```

### Параметры

varCertificate [in] - интерфейс ICertificate.  
pVal [out, retval] - в случае успешного поиска содержит указатель на интерфейс ICRLentry, в противном случае - пустую ссылку.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод FindCertificate позволяет произвести поиск сертификата в СОС.

```
object.FindCertificate(varCertificate)
```

### Параметры

varCertificate - интерфейс ICertificate.

### Возвращаемое значение

В случае успешного поиска содержит указатель на интерфейс ICRLentry, в противном случае - "Nothing".

## ICRL::IMPORT [C++]

## CRL::IMPORT [VBS]

[C++]

Метод Import позволяет осуществлять импорт СОС из строки.

```
HRESULT Import(  
    BSTR bstrCRL  
);
```

### Параметры

bstrCRL [in] - источник импортируемых данных, расположенный в памяти.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет осуществлять импорт СОС из строки.

```
object.Import(bstrCRL)
```

### Параметры

bstrCRL - источник импортируемых данных, расположенный в памяти.

**ICRL::LOAD [C++]****CRL::LOAD [VBS]**

[C++]

Метод Load позволяет загрузить данные СОС с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

HRESULT Load(

```
BSTR bstrURI  
);
```

**Параметры**

bstrURI [in] – URI (имя файла), откуда необходимо загрузить СОС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить данные СОС с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

```
object.Load(bstrURI)
```

**Параметры**

bstrURI – URI (имя файла), откуда необходимо загрузить СОС.

**ICRL::SAVE [C++]****CRL::SAVE [VBS]**

[C++]

Метод Save позволяет сохранять СОС по указанному идентификатору ресурса (URI).

```
HRESULT Save(  
    BSTR bstrURI,  
    FORMAT fFormat,  
    VARIANT_BOOL bBase64Headers  
);
```

**Параметры**

bstrURI [in] – URI (имя файла), куда необходимо сохранить СОС.

fFormat [in] – указывает формат сохранения данных СОС (DER или Base64). Допустимыми значениями являются значения FORMAT.

bAddBase64Headers [in, defaultvalue(-1)] – обрамление заголовками (только для формата Base64).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранять СОС по указанному идентификатору ресурса (URI).

```
object.Save(bstrURI, fFormat, bAddBase64Headers)
```

**Параметры**

bstrURI - URI (имя файла), куда необходимо сохранить СОС.  
fFormat - указывает формат сохранения данных СОС. Допустимыми значениями являются значения FORMAT.  
bAddBase64Headers - обрамление заголовками (только для формата Base64).

## ICRL::CRLCONTEXT [C++]

## CRL::CRLCONTEXT [VBS]

[C++]

Свойство CRLContext позволяет получить/установить CryptoAPI-контекст СОС.

HRESULT get\_CRLContext(

```
    long* pVal
);
HRESULT put_CRLContext(
    long* pVal
);
```

### Параметры

pVal [out, retval] - указатель на CryptoAPI-контекст СОС.  
pVal [in] - указатель на CryptoAPI-контекст СОС

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CRLContext позволяет получить/установить CryptoAPI-контекст СОС.

```
object.CRLContext [= Value]
```

### Параметры

Value - указатель на CryptoAPI-контекст СОС.

## ICRL::ISSUERNAME [C++]

## CRL::ISSUERNAME [VBS]

[C++]

Свойство IssuerName позволяет получить отличительное имя издателя СОС.

```
HRESULT get_IssuerName(
    BSTR* pVal
);
```

### Параметры

pVal [out, retval] - отличительное имя в виде строки.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство IssuerName позволяет получить/установить отличительное имя издателя СОС.

```
object.IssuerName
```

**Возвращаемое значение**

Отличительное имя в виде строки.

**ICRL::NEXTUPDATE [C++]****CRL::NEXTUPDATE [VBS]**

[C++]

Свойство NextUpdate позволяет получить дату и время следующего обновления СОС.

```
HRESULT get_NextUpdate(  
    DATE* pVal  
);
```

**Параметры**

pVal [out, retval] - дата и время следующего обновления СОС

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство NextUpdate позволяет получить дату и время следующего обновления СОС.

object.NextUpdate

**Возвращаемое значение**

Дата и время следующего обновления СОС.

**ICRL::SIGNERCERTIFICATE [C++]****CRL::SIGNERCERTIFICATE [VBS]**

[C++]

Свойство SignerCertificate позволяет получить сертификат подписи под CRL из указанного хранилища.

```
HRESULT get_SignerCertificate(  
    VARIANT vtStore  
    ICertificates** pVal  
);
```

**Параметры**

vtStore [in, optional] - хранилище сертификатов. Не обязательный параметр. Если его не указывать, для поиска будет использовано хранилище корневых сертификатов.

pVal [out, retval] - указатель на указатель на сертификат подписи.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignerCertificate позволяет получить сертификат подписи под CRL из указанного хранилища.

object.SignerCertificate [vtStore]

## Параметры

vtStore [in, optional] – хранилище сертификатов. Не обязательный параметр. Если его не указывать, для поиска будет использовано хранилище корневых сертификатов.

## Возвращаемые значения

Сертификат подписи

## ICRL::THISUPDATE [C++]

## CRL::THISUPDATE [VBS]

[C++]

Свойство ThisUpdate позволяет получить дату и время выпуска текущего СОС.

```
HRESULT get_ThisUpdate(  
    DATE* pVal  
);
```

## Параметры

pVal [out, retval] – дата и время выпуска текущего СОС

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ThisUpdate позволяет получить дату и время выпуска текущего СОС.

object.ThisUpdate

## Возвращаемое значение

Дата и время выпуска текущего СОС.

## CRLENTY (GUID: D6D6A8C4-2E5E-41DD-9432-361926D46B24)

## Свойства

Свойство	Описание
ReasonCode	Свойство позволяет получить код причины отзыва сертификата
RevocationDate	Свойство позволяет получить дату отзыва сертификата
SerialNumber	Свойство позволяет получить серийный номер отозванного сертификата

## ICRLENTY::REASONCODE [C++]

## CRLENTY::REASONCODE [VBS]

[C++]

Свойство ReasonCode получить код причины отзыва сертификата.

```
HRESULT get_ReasonCode(  
    VARIANT* pVal  
);
```

## Параметры



pVal [out, retval] - возвращает код причины отзыва сертификата. Возможны следующие значения кода причины отзыва:

Значение	Описание
unspecified = 0	Причина отзыва не указана
keyCompromise = 1	Компрометация ключа
cACompromise = 2	Компрометация ЦС
affiliationChanged = 3	Изменение принадлежности
superseded = 4	Сертификат заменен
cessationOfOperation = 5	Прекращение работы
certificateHold = 6	Сертификат приостановлен

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ReasonCode получить код причины отзыва сертификата.

```
object.ReasonCode
```

### Возвращаемое значение

Код причины отзыва сертификата. Возможны следующие значения кода причины отзыва:

Значение	Описание
unspecified = 0	Причина отзыва не указана
keyCompromise = 1	Компрометация ключа
CACompromise = 2	Компрометация ЦС
affiliationChanged = 3	Изменение принадлежности
superseded = 4	Сертификат заменен
cessationOfOperation = 5	Прекращение работы
certificateHold = 6	Сертификат приостановлен

### ICRLENTY::REVOCATIONDATE [C++]

### CRLENTY::REVOCATIONDATE [VBS]

[C++]

Свойство RevocationDate получить дату отзыва сертификата.

```
HRESULT get_RevocationDate(  
    VARIANT* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает дату отзыва сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RevocationDate получить дату отзыва сертификата.

```
object.RevocationDate
```

#### Возвращаемое значение

Дату отзыва сертификата.

ICRLENTY::SERIALNUMBER [C++]

CRLENTRY::SERIALNUMBER [VBS]

[C++]

Свойство RevocationDate получить серийный номер отозванного сертификата.

```
HRESULT get_SerialNumber(  
    VARIANT* pVal  
);
```

#### Параметры

pVal [out, retval] - серийный номер отозванного сертификата.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SerialNumber получить серийный номер отозванного сертификата.

```
object.SerialNumber
```

#### Возвращаемое значение

Серийный номер отозванного сертификата

CRLs (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

#### Методы

Метод	Описание
Add	Метод позволяет добавлять СОС в коллекцию
Remove	Метод позволяет удалять СОС из коллекции

#### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество объектов типа CRL в коллекции
Item	Свойство позволяет получить указатель на объект типа CRL

ICRLs::ADD [C++]

CRLs::ADD [VBS]

[C++]

Метод Add позволяет добавлять СОС в коллекцию.

```
HRESULT Add(  
    ICRL* addVal  
);
```

#### Параметры

addVal [in] – указатель на интерфейс типа ICRL.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

#### Примечания

Вызов метода возможен после того, как получен интерфейс ICRL, например из коллекции СОС.

[VBS]

Метод Add позволяет добавлять СОС в коллекцию.

```
object.Add(addVal)
```

#### Параметры

addVal – указатель на объект типа CRL.

#### Примечания

Вызов метода возможен после того, как получен объект CRL, например из коллекции СОС.

### ICRL::REMOVE [C++]

### CRL::REMOVE [VBS]

[C++]

Метод Remove позволяет удалять СОС из коллекции.

```
HRESULT Remove(  
    long lIndex  
);
```

#### Параметры

lIndex [in] – номер элемента в коллекции минус 1 [0 .. Count-1]

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалять СОС из коллекции.

```
object.Remove(lIndex)
```

#### Параметры

lIndex – номер элемента в коллекции минус 1 [0 .. Count-1]

### ICRL::COUNT [C++]

### CRL::COUNT [VBS]

[C++]

Свойство Count позволяет получить количество объектов типа СОС в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - количество СОС в коллекции.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество объектов типа СОС в коллекции.

```
object.Count
```

#### Возвращаемое значение

Количество СОС в коллекции.

### ICRLs::ITEM [C++]

### CRLs::ITEM [VBS]

[C++]

Свойство Item позволяет получить указатель на интерфейс типа ICRL.

```
HRESULT get_Item(  
    long lIndex,  
    ICRL** pVal  
);
```

#### Параметры

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1]

pVal [out, retval] - объект-приемник.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить указатель на объект типа CRL.

```
object.Item (lIndex)
```

#### Параметры

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1]

#### Возвращаемое значение

Объект-приемник.

### CRYPTOProvider (GUID: D6D6A8C4-2E5E-41DD-9432-361926D46B24)

#### Методы

Метод	Описание
-------	----------

deleteContainer	Метод позволяет удалить контейнер по имени из хранилища контейнеров крипто-провайдера
Display	Метод позволяет отобразить окно с информацией о криптопровайдере

### Свойства

Свойство	Описание
Name	Свойство позволяет получить имя криптопровайдера

## ICRYPTOProvider::DELETECONTAINER [C++]

## CRYPTOProvider::DELETECONTAINER [VBS]

[C++]

Метод deleteContainer позволяет удалить контейнер по имени из хранилища контейнеров крипто-провайдера.

```
HRESULT deleteContainer(  
    BSTR bstrContainerName  
);
```

### Параметры

bstrContainerName [in] - имя контейнера, который необходимо удалить.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечание

Перед удалением контейнера нужно получить указатель на объект ICryptoProvider

[VBS]

Метод deleteContainer позволяет удалить контейнер по имени из хранилища контейнеров крипто-провайдера.

```
object.deleteContainer(bstrContainerName)
```

### Параметры

bstrContainerName - имя контейнера, который необходимо удалить.

### Примечание

Перед удалением контейнера нужно получить указатель на объект CryptoProvider

## ICRYPTOProvider::DISPLAY [C++]

## CRYPTOProvider::DISPLAY [VBS]

[C++]

Метод Display позволяет отобразить окно с информацией о криптопровайдере.

```
HRESULT Display(  
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет отобразить окно с информацией о криптопровайдере.

```
object.Display()
```

**ICRYPTOPROVIDER::NAME [C++]**

**CRYPTOPROVIDER::NAME [VBS]**

[C++]

Свойство Name позволяет получить имя криптопровайдера.

```
HRESULT get_Name(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] - строковое значение имени криптопровайдера.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Name позволяет получить имя криптопровайдера.

```
object.Name
```

#### Возвращаемое значение

Строковое значение имени криптопровайдера.

### CRYPTOPROVIDERS (GUID: 32F50510-2813-4974-B0B2-98B2F2FCF327)

При создании объекта CryptoProviders создается коллекция криптопровайдеров, доступных для работы.

Список доступных для работы DigtCrypto криптопровайдеров может быть ограничен из-за нескольких причин:

наличия или отсутствия лицензии на ПО «КриптоАРМ» (наличие лицензии влияет на функциональные возможности ПО «КриптоАРМ»), установленных в системе криптопровайдеров, поддерживаемых ПО «КриптоАРМ» криптопровайдеров (см. список поддерживаемых криптопровайдеров)

При отсутствии лицензии (версия «КриптоАРМ Старт») DigtCrypto может работать только с несколькими стандартными предустановленными в системе Windows криптопровайдерами:

- Microsoft Base Cryptographic Provider v1.0
- Microsoft Enhanced Cryptographic Provider v1.0
- Microsoft Strong Cryptographic Provider
- Microsoft Base DSS Cryptographic Provider

Доступ к остальным поддерживаемым провайдерам обеспечивается лицензией.

#### Методы

Метод	Описание
Display	Метод позволяет отобразить окно со списком криптопровайдеров

ConnectRemovableKeyCarrierWizard

Метод позволяет запустить мастер подключения отчуждаемого носителя

**Свойства**

Свойство	Описание
Count	Свойство позволяет получить количество элементов в коллекции
Item	Свойство позволяет получить элемент коллекции криптопровайдеров по индексу
Provider	Свойство позволяет получить криптопровайдер по его строковому имени
SupportedProviders	Свойство позволяет получить коллекцию криптопровайдеров, поддерживаемых ПО "КриптоАРМ" и установленных в системе
SystemProviders	Свойство позволяет получить коллекцию установленных в системе криптопровайдеров

**ICRYPTOPROVIDERS::DISPLAY [C++]****CRYPTOPROVIDERS::DISPLAY [VBS]**

[C++]

Метод Display позволяет отобразить окно со списком установленных в системе криптопровайдеров.

```
HRESULT Display(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет отобразить окно со списком установленных в системе криптопровайдеров.

```
object.Display()
```

**ICRYPTOPROVIDERS::CONNECTREMOVABLEKEYCARRIERWIZARD [C++]****CRYPTOPROVIDERS::CONNECTREMOVABLEKEYCARRIERWIZARD [VBS]**

[C++]

Метод ConnectRemovableKeyCarrierWizard позволяет запустить мастер подключения отчуждаемого носителя.

```
HRESULT ConnectRemovableKeyCarrierWizard(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод ConnectRemovableKeyCarrierWizard позволяет запустить мастер подключения отчуждаемого носителя.

```
object.ConnectRemovableKeyCarrierWizard()
```

**ICRYPTOPROVIDERS::COUNT [C++]****CRYPTOPROVIDERS::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество элементов в коллекции криптопровайдеров.

```
HRESULT get_Count(  
    long* pVal  
);
```

**Параметры**

pVal [out, retval] - числовое значение количества элементов в коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество элементов в коллекции криптопровайдеров.

```
object.Count
```

**Возвращаемое значение**

Числовое значение количества элементов.

**ICRYPTOPROVIDERS::ITEM [C++]****CRYPTOPROVIDERS::ITEM [VBS]**

[C++]

Свойство Item позволяет получить элемент коллекции криптопровайдеров по индексу.

```
HRESULT get_Item(  
    long lIndex,  
    ICryptoProvider** pVal  
);
```

**Параметры**

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].  
pVal [out, retval] - указатель на интерфейс ICryptoProvider.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент коллекции криптопровайдеров по индексу.

```
object.Item ( lIndex)
```

**Параметры**

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1].

**Возвращаемое значение**

Указатель на интерфейс CryptoProvider.

**ICRYPTOPROVIDERS::PROVIDER [C++]**



## CRYPTOPROVIDERS::PROVIDER [VBS]

[C++]

Свойство Provider позволяет получить криптопровайдер по его строковому имени.

```
HRESULT get_Provider(  
    BSTR bstrName,  
    VARIANT* pProvider  
);
```

### Параметры

bstrName [in] – строка с именем криптопровайдера.  
pProvider [out, retval] – в случае успешного поиска содержит указатель на ICryptoProvider, в противном случае – пустую ссылку.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Provider позволяет получить криптопровайдер по его строковому имени.

```
object.Provider (bstrName)
```

### Параметры

bstrName – строка с именем криптопровайдера.

### Возвращаемое значение

В случае успешного поиска содержит указатель на CryptoProvider, в противном случае – Nothing".

## ICRYPTOPROVIDERS::SUPPORTEDPROVIDERS [C++]

## CRYPTOPROVIDERS::SUPPORTEDPROVIDERS [VBS]

[C++]

Свойство SupportedProviders позволяет получить коллекцию криптопровайдеров, поддерживаемых ПО «КриптоАРМ» и установленных в системе.

```
HRESULT get_SupportedProviders(  
    long* pVal  
);
```

### Параметры

pVal [out, retval] – коллекция криптопровайдеров, поддерживаемых ПО «КриптоАРМ» и установленных в системе.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечание

При наличии полной действительной лицензии на ПО «КриптоАРМ» данная коллекция криптопровайдеров будет совпадать с коллекцией доступных криптопровайдеров.

[VBS]

Свойство SupportedProviders позволяет получить коллекцию криптопровайдеров, поддерживаемых ПО «КриптоАРМ» и установленных в системе.

`object.SupportedProviders`**Возвращаемое значение**

Коллекция криптопровайдеров, поддерживаемых ПО «КриптоАРМ» и установленных в системе.

**Примечание**

При наличии полной действительной лицензии на ПО «КриптоАРМ» данная коллекция криптопровайдеров будет совпадать с коллекцией доступных криптопровайдеров.

**ICRYPTOPROVIDERS::SYSTEMPROVIDERS [C++]****CRYPTOPROVIDERS::SYSTEMPROVIDERS [VBS]**

[C++]

Свойство SystemProviders позволяет получить коллекцию установленных в системе криптопровайдеров.

```
HRESULT get_SystemProviders(  
    ICryptoProviders** pVal  
);
```

**Параметры**

pVal [out, retval] – коллекция установленных в системе криптопровайдеров.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SystemProviders позволяет получить коллекцию установленных в системе криптопровайдеров.

`object.SystemProviders`**Возвращаемое значение**

Коллекция установленных в системе криптопровайдеров.

**CTL (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)****Методы**

Метод	Описание
Import	Метод позволяет осуществлять импорт СДС из строки
FindSubject	Метод позволяет находить по сертификату соответствующие ему элементы СДС.
Load	Метод позволяет загрузить данные СДС с указанного идентификатора ресурса (URI)
Export	Метод позволяет произвести экспорт СДС в строку
Save	Метод позволяет сохранять СДС по указанному идентификатору ресурса (URI)

**Свойства**

Свойство	Описание
CTLContext	Свойство позволяет получить/установить CryptoAPI-контекст СДС
Entries	Свойство позволяет получить коллекцию элементов СДС
Issuer	

ListIdentifier	Свойство позволяет получить идентификатор списка доверенных сертификатов
SequenceNumber	Содержит число, увеличивающееся при каждом обновлении СДС
SignerCertificate	
SubjectAlgorithm	
ThisUpdate	Свойство позволяет получить дату и время выпуска текущего СДС
NextUpdate	Свойство позволяет получить дату и время следующего обновления СДС
UsageIdentifiers	Свойство определяет возможные назначения сертификатов списка. Возвращает коллекцию объектных идентификаторов
Version	Свойство позволяет узнать версию СДС

## ICTL::IMPORT [C++]

## CTL::IMPORT [VBS]

[C++]

Метод Import осуществлять импорт СДС из строки.

```
HRESULT Import(  
    BSTR bstrCRL  
);
```

### Параметры

bstrCRL [in] – источник импортируемых данных, расположенный в памяти.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import осуществлять импорт СДС из строки.

```
object.Import(bstrCTL)
```

### Параметры

bstrCTL – буфер (строка) с содержимым CTL.

## ICTL::FINDSUBJECT [C++]

## CTL::FINDSUBJECT [VBS]

[C++]

Метод FindSubject позволяет находить по сертификату соответствующие ему элементы СДС.

```
HRESULT FindSubject(  
    ICertificate* pSrchCert,  
    ICTLEntry** pFoundedEntry  
);
```

### Параметры

pSrchCert [in] – указатель на сертификат, который будет отыскиваться.

pFoundedEntry [out,retval] – указатель на коллекцию соответствующих сертификату элементов СДС.

### Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод FindSubject позволяет находить по сертификату соответствующие ему элементы СДС.

```
object.FindSubject pSrchCert
```

#### Параметры

pSrchCert - сертификат, который будет отыскиваться.

#### Возвращаемое значения

Коллекция соответствующих сертификату элементов СДС.

### ICTL::LOAD [C++]

#### CTL::LOAD [VBS]

[C++]

Метод Load позволяет загрузить данные СДС с указанного идентификатора ресурса (URI).

```
HRESULT Load(  
    BSTR bstrURI  
);
```

#### Параметры

bstrURI [in] - URI (имя файла), откуда необходимо загрузить СДС.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод **Load** позволяет загрузить данные СДС с указанного идентификатора ресурса (URI)

```
object.Load(bstrURI)
```

#### Параметры

bstrURI - URI (имя файла), откуда необходимо загрузить СДС.

### ICTL::EXPORT [C++]

#### CTL::EXPORT [VBS]

[C++]

Метод Export позволяет произвести экспорт СДС в строку.

```
HRESULT Export(  
    FORMAT fFormat  
    VARIANT_BOOL bBase64Headers  
    BSTR* pVal  
);
```

#### Параметры

fFormat [in] - формат загружаемого СДС (DER, Base64). Допустимыми значениями являются значения FORMAT.

bBase64Headers [in, defaultvalue(0)] - обрамление заголовками (только для формата Base64).

pVal [out, retval] - строка - приемник экспортируемых из компонента данных.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечания

Вызов метода возможен после того, как получен интерфейс ICRL, например из коллекции СОС

[VBS]

Метод Export позволяет произвести экспорт СДС в строку.

object.Export(fFormat, bAddBase64Headers)

### Параметры

fFormat - устанавливает формат экспортируемых данных - der или Base64. принимает значения из FORMAT.

bAddBase64Headers - устанавливает, необходимо ли использовать служебные заголовки. Действительно только для base64.

### Возвращаемые значения

Возвращает строку содержащую в себе данные СДС.

## ICTL::SAVE [C++]

## CTL::SAVE [VBS]

[C++]

Метод Save позволяет сохранять СДС по указанному идентификатору ресурса (URI).

```
HRESULT Save(  
    BSTR bstrURI,  
    FORMAT fFormat,  
    VARIANT_BOOL bBase64Headers  
);
```

### Параметры

bstrURI [in] - URI (имя файла), куда необходимо сохранить СДС.

fFormat [in] - указывает формат сохранения данных СДС (DER или Base64). Допустимыми значениями являются значения FORMAT.

bAddBase64Headers [in, defaultvalue(-1)] - обрамление заголовками (только для формата Base64).

### Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранять СДС по указанному идентификатору ресурса (URI).

object.Save(bstrURI, fFormat, bAddBase64Headers)

### Параметры

bstrURI - URI (имя файла), куда необходимо сохранить СДС.

fFormat - указывает формат сохранения данных СДС. Допустимыми значениями являются значения FORMAT.

bAddBase64Headers - устанавливает, необходимо ли использовать служебные заголовки. Действительно только для base64.

**ICTL::CTLCONTEXT [C++]****CTL::CTLCONTEXT [VBS]**

[C++]

Свойство CTLContext позволяет получить/установить CryptoAPI-контекст СДС.

```
HRESULT get_CTLContext (
    long* pVal
);
HRESULT put_CTLContext (
    long* pVal
);
```

**Параметры**

pVal [out, retval] - указатель на CryptoAPI-контекст СДС.  
pVal [in] - указатель на CryptoAPI-контекст СДС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CTLContext позволяет получить/установить CryptoAPI-контекст СДС.

```
object.CTLContext[= Value]
```

**Параметры**

Value - указатель на контекст СДС.

**Возвращаемые значения**

Контекст СДС.

**ICTL::ENTRIES [C++]****CTL::ENTRIES [VBS]**

[C++]

Свойство Entries позволяет получить коллекцию элементов СДС.

```
HRESULT get_Entries (
    ICTLEntries** pVal
);
```

**Параметры**

pVal [out, retval] - представляет коллекцию элементов СДС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Entries позволяет получить коллекцию элементов СДС.

```
object.Entries
```

**Возвращаемые значения**

Коллекция элементов СДС.

**ICTL::ListIdentifier [C++]****CTL::ListIdentifier [VBS]****[C++]**

Свойство ListIdentifier позволяет получить идентификатор списка доверенных сертификатов.

```
HRESULT get_ListIdentifier(  
    FORMAT frmtFormat,  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - значение идентификатора списка.

pVal [in] - указывает формат сохранения идентификатора (DER или Base64).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ListIdentifier позволяет получить идентификатор списка доверенных сертификатов.

```
object.ListIdentifier[frmtFormat]
```

**Параметры**

frmtFormat - устанавливает формат получаемого идентификатора.

**Возвращаемые значения**

Буфер с идентификатором списка. Обычно строка в формате Unicode.

**ICTL::SEQUENCENUMBER [C++]****CTL::SEQUENCENUMBER [VBS]****[C++]**

Свойство SequenceNumber содержит число, увеличивающееся при каждом обновлении СДС.

```
HRESULT get_SequenceNumber(  
    FORMAT frmtFormat,  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - значение номера в последовательности обновлений, в указанном формате.

pVal [in] - указывает формат сохранения данных (DER или Base64).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство SequenceNumber содержит число, увеличивающееся при каждом обновлении СДС.

```
object.SequenceNumber[frmtFormat]
```

**Параметры**

frmtFormat - устанавливает формат получаемых данных.

**Возвращаемые значения**

Буфер содержащий номер обновления СДС.

**ICTL::THISUPDATE [C++]****CTL::THISUPDATE [VBS]**

[C++]

Свойство ThisUpdate позволяет получить дату и время выпуска текущего СДС.

```
HRESULT get_ThisUpdate(  
    DATE* pVal  
);
```

**Параметры**

pVal [out, retval] - дата и время выпуска текущего СДС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ThisUpdate позволяет получить дату и время выпуска текущего СДС.

object.ThisUpdate

**Возвращаемые значения**

Дата и время выпуска текущего СДС.

**ICTL::NEXTUPDATE [C++]****CTL::NEXTUPDATE [VBS]**

[C++]

Свойство NextUpdate позволяет получить дату и время следующего обновления СДС.

```
HRESULT get_NextUpdate(  
    DATE* pVal  
);
```

**Параметры**

pVal [out, retval] - дата и время следующего обновления СДС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство NextUpdate позволяет получить дату и время следующего обновления СДС.

object.NextUpdate

**Возвращаемые значения**

Дата и время следующего обновления СДС.

**ICTL::USAGEIDENTIFIERS [C++]**



**CTL::USAGEIDENTIFIERS [VBS]****[C++]**

Свойство UsageIdentifiers определяет возможные назначения сертификатов списка. Возвращает коллекцию объектных идентификаторов.

```
HRESULT get_UsageIdentifiers(  
    IOIDs** pVal  
);
```

**Параметры**

pVal [out, retval] - представляет коллекцию OID-ов идентификаторов использования.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство UsageIdentifiers определяет возможные назначения сертификатов списка. Возвращает коллекцию объектных идентификаторов.

```
object.UsageIdentifiers
```

**Возвращаемые значения**

Указатель на объект IOIDs - коллекцию объектных идентификаторов.

**ICTL::USAGEIDENTIFIERS [C++]****CTL::USAGEIDENTIFIERS [VBS]****[C++]**

Свойство UsageIdentifiers определяет возможные назначения сертификатов списка. Возвращает коллекцию объектных идентификаторов.

```
HRESULT get UsageIdentifiers(  
    IOIDs** pVal  
);
```

**Параметры**

pVal [out, retval] - представляет коллекцию OID-ов идентификаторов использования.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство UsageIdentifiers определяет возможные назначения сертификатов списка. Возвращает коллекцию объектных идентификаторов.

```
object.UsageIdentifiers
```

**Возвращаемые значения**

Указатель на объект IOIDs - коллекцию объектных идентификаторов.

**CTLs (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)****Методы**

Метод	Описание
Add	Метод позволяет добавлять СДС в коллекцию
Remove	Метод позволяет удалять СДС из коллекции

**Свойства**

Свойство	Описание
Count	Свойство позволяет получить количество объектов типа СДС в коллекции
Item	Свойство позволяет получить указатель на интерфейс типа ICTL

**ICTLs::Add [C++]****CTLs::Add [VBS]****[C++]**

Метод Add позволяет добавлять СДС в коллекцию.

```
HRESULT Add(  
    ICTL* addVal  
);
```

**Параметры**

addVal [in] - представляет СДС, который будет добавлен в коллекцию.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**Примечания**

Вызов метода возможен после того, как получен интерфейс ICTL, например из коллекции СОС.

**[VBS]**

Метод Add позволяет добавлять СДС в коллекцию.

```
object.Add(addVal)
```

**Параметры**

addVal - объект типа ICTL, представляющий добавляемый СДС.

**Примечания**

Вызов метода возможен после того, как получен объект CTL, например из коллекции СДС.

**ICTLs::Remove [C++]****CTLs::Remove [VBS]****[C++]**

Метод Remove позволяет удалять СДС из коллекции.

```
HRESULT Remove (
```

```
long lIndex  
);
```

### Параметры

lIndex [in] - номер элемента, который будет удален.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалять СДС из коллекции.

```
object.Remove(lIndex)
```

### Параметры

lIndex - устанавливает номер удаляемого элемента в коллекции.

## ICTLs::COUNT [C++]

## CTLs::COUNT [VBS]

[C++]

Свойство Count позволяет получить количество объектов типа СДС в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

### Параметры

pVal [in] - количество элементов в коллекции.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество объектов типа СДС в коллекции.

```
object.Count
```

### Возвращаемое значение

Количество СДС в коллекции.

## ICTLs::ITEM [C++]

## CTLs::ITEM [VBS]

[C++]

Свойство Item позволяет получить указатель на интерфейс типа ICTL.

```
HRESULT get_Item(  
    long lIndex,  
    ICTL** pVal  
);
```

### Параметры

lIndex [in] - номер возвращаемого элемента.

pVal [out, retval] - представляет СДС, возвращаемый из коллекции.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить указатель на интерфейс типа ICTL.

object.Item (lIndex)

### Параметры

lIndex - устанавливает номер получаемого элемента коллекции.

### Возвращаемое значение

Объект ICTL, представляющий получаемый элемент коллекции.

## CTLBUILDER (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

### Методы

Метод	Описание
Generate	Метод выполняет генерацию CTL из установленного шаблона

### Свойства

Свойство	Описание
Template	Свойство позволяет записывать шаблон, по которому генерится CTL

### ICTLBUILDER::GENERATE [C++]

### CTLBUILDER::GENERATE [VBS]

[C++]

Метод Generate выполняет генерацию CTL из установленного шаблона.

```
HRESULT Generate(  
    ICTL** pVal  
);
```

### Параметры

pVal [out, retval] - возвращает сгенерированный CTL.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Generate выполняет генерацию CTL из установленного шаблона.

object.Generate

### Возвращаемое значение

Сгенерированный CTL.

### ICTLBUILDER::TEMPLATE [C++]

**CTLBUILDER::TEMPLATE [VBS]**

[C++]

Свойство Template позволяет записывать шаблон, по которому генерится CTL.

```
HRESULT get_Template(  
    ICTLTemplate** pVal  
);  
HRESULT put_Comment(  
    ICTLTemplate* newVal  
);
```

**Параметры**

pVal [out, retval] - шаблон, содержащийся в свойстве.  
pVal [in] - шаблон, который будет установлен в свойство.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Template позволяет записывать шаблон, по которому генерится CTL.

```
object.Template [= pVal]
```

**Параметры**

pVal - устанавливаемый шаблон, по которому будет генерироваться CTL.

**Возвращаемое значение**

Установленный шаблон, по которому будет генерироваться CTL.

**CTLENTY (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)****Методы**

Метод	Описание
FillFromCertificate	Метод позволяет заполнить поля объекта (элемента СДС) в соответствии со значениями полей передаваемого сертификата.
RetrieveCertificate	Метод возвращает сертификат, соответствующий данному элементу списка доверенных сертификатов

**Свойства**

Свойство	Описание
SubjectIdentifier	Свойство позволяет получить уникальный идентификатор элемента списка. Это может быть хэш или любая уникальная последовательность байтов.

**ICTLENTY::FILLFROMCERTIFICATE [C++]****CTLENTY::FILLFROMCERTIFICATE [VBS]**

[C++]

Метод FillFromCertificate позволяет заполнить поля объекта (элемента СДС) в соответствии со значениями полей передаваемого сертификата.

```
HRESULT FillFromCertificate(  
    ICertificate* pVal  
);
```

### Параметры

pVal [in] - указатель на сертификат, в соответствии с которым будут заполняться поля элемента СДС.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод FillFromCertificate позволяет заполнить поля объекта в соответствии со значениями полей передаваемого сертификата.

```
object.FillFromCertificate pVal
```

### Параметры

pVal - сертификат, в соответствии с которым будут заполняться поля элемента СДС.

### ICTENTRY::RETRIEVECERTIFICATE [C++]

### CTENTRY::RETRIEVECERTIFICATE [VBS]

[C++]

Метод RetrieveCertificate возвращает сертификат, соответствующий данному элементу списка доверенных сертификатов.

```
HRESULT RetrieveCertificate(  
    ICertificateStore* pIStore  
    ICertificate** pICert  
);
```

### Параметры

pIStore [in] - устанавливает хранилище, в котором будет осуществлен поиск запрашиваемого сертификата.

pICert [out, retval] - возвращает указатель на интерфейс ICertificate, представляющий запрашиваемый сертификат.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RetrieveCertificate возвращает сертификат, соответствующий данному элементу списка доверенных сертификатов.

```
object.RetrieveCertificate [pIStore]
```

### Параметры

pIStore - устанавливает хранилище сертификатов, в котором будет осуществлен поиск запрашиваемого сертификата.

### Возвращаемое значение

Возвращает объект представляющий искомый сертификат.

### ICTENTRY::SUBJECTIDENTIFIER [C++]

### CTENTRY::SUBJECTIDENTIFIER [VBS]

[C++]

Свойство SubjectIdentifier позволяет получить уникальный идентификатор элемента списка. Это может быть хэш или любая уникальная последовательность байтов.

```
HRESULT get_SubjectIdentifier(  
    FORMAT frmtFormat,  
    BSTR* pVal  
);
```

#### Параметры

frmtFormat [in] - выходной формат. Допустимыми значениями являются следующие значения:  
BASE64\_TYPE = 0 Формат используемой кодировки при сохранении BASE64  
HEX\_TYPE = 3 Формат отображающий строку с шестнадцатеричными кодами байтов данных  
pVal [out, retval] - строка, принимающая значение идентификатора.

[VBS]

Свойство SubjectIdentifier позволяет получить уникальный идентификатор элемента списка. Это может быть хэш или любая уникальная последовательность байтов.

```
object.SubjectIdentifier [pVal]
```

#### Параметры

frmtFormat - устанавливает формат возвращаемого значения.  
Допустимыми значениями являются следующие значения:  
BASE64\_TYPE = 0 Формат используемой кодировки при сохранении BASE64  
HEX\_TYPE = 3 Формат отображающий строку с шестнадцатеричными кодами байтов данных

#### Возвращаемое значение

Строка содержащая идентификатор списка.

### CTLENTRIES

#### Методы

Метод	Описание
Add	Метод позволяет добавить элемент СДС в коллекцию
FillFromCertificates	Метод позволяет заполнить коллекцию элементов СДС в соответствии с передаваемыми сертификатами.
Remove	Метод Remove позволяет удалить из коллекции элемент с указанным индексом
RetrieveCertificates	Метод позволяет получить сертификаты, соответствующие этому элементу.

#### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество элементов в коллекции
Item	Свойство позволяет получить элемент коллекции

#### ICTLENTRIES::ADD [C++]

#### CTLENTRIES::ADD [VBS]

[C++]

Метод Add позволяет добавить элемент СДС в коллекцию.

```
HRESULT Add(  
    ICTLEntry* addVal  
);
```

### Параметры

addVal [in] – представляет объект типа ICTLEntry, который будет добавлен в коллекцию.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить элемент СДС в коллекцию.

```
object.Add(addVal)
```

### Параметры

addVal – указатель на объект типа CTLEntry.

### Примечания

Вызов метода возможен после того, как получен объект CTL, например из коллекции СДС.

## ICTLENTRIES::FILLFROMCERTIFICATES [C++]

## CTLENTRIES::FILLFROMCERTIFICATES [VBS]

[C++]

Метод FillFromCertificates позволяет заполнить коллекцию элементов СДС в соответствии с передаваемыми сертификатами.

```
HRESULT FillFromCertificates(  
    ICertificates* pVal  
);
```

### Параметры

pVal [in] – указатель на коллекцию сертификатов, в соответствии с которыми будет заполняться коллекция элементов СДС.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод FillFromCertificates позволяет заполнить коллекцию элементов СДС в соответствии с передаваемыми сертификатами.

```
object.FillFromCertificates pVal
```

### Параметры

pVal – коллекция сертификатов, в соответствии с которыми будет заполняться коллекция элементов СДС.

## ICTLENTRIES::REMOVE [C++]

## CTLENTRIES::REMOVE [VBS]

[C++]



Метод Remove позволяет удалить из коллекции элемент с указанным индексом.

```
HRESULT Remove(  
    long lIndex  
);
```

#### Параметры

lIndex [in] - номер элемента коллекции, который будет удален.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить из коллекции элемент с указанным индексом.

```
object.Remove(lIndex)
```

#### Параметры

lIndex - устанавливает значение элемента списка, который будет удален.

### ICTLENTRIES::RETRIEVECERTIFICATES [C++]

### CTLENTRIES::RETRIEVECERTIFICATES [VBS]

[C++]

Метод RetrieveCertificates позволяет получить сертификаты, соответствующие этому элементу.

```
HRESULT RetrieveCertificates(  
    VARIANT varpICertificateStore ,  
    VARIANT_BOOL vbSkipAbsent,  
    ICertificates** pVal  
);
```

#### Параметры

varpICertificateStore [in, optional] - хранилище сертификатов в котором будет осуществляться поиск.  
vbSkipAbsent [in, defaultvalue(-1)] - указывает, надо ли пропускать отсутствующие сертификаты.  
pVal [out, retval] - коллекция сертификатов, содержащая найденные сертификаты.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RetrieveCertificates позволяет получить сертификаты, соответствующие этому элементу.

```
object.RetrieveCertificates [varpICertificateStore], [vbSkipAbsent]
```

#### Параметры

varpICertificateStore - хранилище сертификатов в котором будет осуществляться поиск.  
vbSkipAbsent - указывает, надо ли пропускать отсутствующие сертификаты.

#### Возвращаемые значения

Коллекция сертификатов, содержащая найденные сертификаты.

**ICTLENTRIES::COUNT [C++]****CTLENTRIES::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество элементов в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

**Параметры**

pVal [out, retval] - количество элементов в коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество элементов в коллекции.

```
object.Count
```

**Возвращаемое значение**

Количество элементов в коллекции.

**ICTLENTRIES::ITEM [C++]****CTLENTRIES::ITEM [VBS]**

[C++]

Свойство Item позволяет получить элемент коллекции.

```
HRESULT get_Item(  
    long lIndex,  
    ICTLEntries** pVal  
);
```

**Параметры**

lIndex [in] - номер возвращаемого элемента.

pVal [out] - объект ICTLEntry, представляющий возвращаемый элемент коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент коллекции.

```
object.Item (lIndex)
```

**Параметры**

lIndex - номер элемента списка, который будет возвращен.

**Возвращаемое значение**

Указатель на объект ICTLEntry, представляющий элемент коллекции.

**CTLTEMPLATE (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)****Методы**

Метод	Описание
CollectData	Метод вызывает мастер сбора данных, необходимых для создания CTL.
Import	Метод позволяет загрузить данные шаблона из памяти.
Load	Метод позволяет загрузить данные шаблона из файла.

**Свойства**

Свойство	Описание
AddToStore	Свойство позволяет получить/установить флаг установки создаваемого СДС в хранилище.
DestinationFilename	Свойство позволяет получить/установить имя файла, в который будет сохранен созданный СДС.
Entries	Свойство позволяет получить/установить коллекцию элементов списка доверенных сертификатов.
IncludeCertificates	Свойство позволяет получить/установить коллекцию сертификатов, которые должны быть включены в создаваемый СДС.
ListIdentifier	Свойство позволяет записывать шаблон, по которому генерится CTL.
NextUpdate	Свойство позволяет получить дату и время следующего обновления СДС.
SaveToFile	Свойство позволяет получить/установить флаг сохранения создаваемого CTL в файл.
SequenceNumber	Свойство позволяет получить/установить значение номера последовательности.
SignCTL	Свойство позволяет получить/установить флаг подписывания создаваемого CTL.
SignProfile	Свойство позволяет получить/установить профиль с настройками, с помощью которых будет подписан создаваемый CTL.
SubjectAlgorithm	Свойство позволяет получить/установить алгоритм, с помощью которого будут хэшироваться сертификаты для списка доверенных сертификатов (SubjectIdentifier в CTLEntry).
ThisUpdate	Свойство позволяет получить/установить дату и время выпуска текущего СДС.
UsagIdentifiers	Свойство позволяет получить/установить список назначений сертификатов списка.
UseNextUpdate	Свойство позволяет

**ICTLTEMPLATE::COLLECTDATA [C++]****CTLTEMPLATE::COLLECTDATA [VBS]****[C++]**

Метод CollectData вызывает мастер сбора данных, необходимых для создания CTL.

```
HRESULT CollectData(  
    VARIANT_BOOL * pVal,  
);
```

**Параметры**

pVal [out, retval] - индикатор завершения мастера. TRUE - завершен, FALSE - прерван.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CollectData вызывает мастер сбора данных, необходимых для создания CTL.

```
object.CollectData
```

#### Возвращаемые значения

Результат завершения мастера. TRUE – завершен, FALSE – прерван.

#### ICTLTEMPLATE::IMPORT [C++]

#### CTLTEMPLATE::IMPORT [VBS]

[C++]

Метод Import позволяет загрузить данные шаблона из памяти.

```
HRESULT Import(  
    BSTR bstrCtl,  
    FORMAT fFormat)  
;
```

#### Параметры

bstrCtl [in] – источник импортируемых данных, расположенный в памяти.  
fFormat [in, defaultvalue(DER\_TYPE)] – формат импортируемых из памяти данных. Допустимыми значениями являются значения FORMAT.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет загрузить данные шаблона из памяти.

```
object.Import(bstrCtl, fFormat)
```

#### Параметры

bstrCtl – источник импортируемых данных, расположенный в памяти.  
fFormat – формат импортируемых из памяти данных. Допустимыми значениями являются значения FORMAT.

#### ICTLTEMPLATE::LOAD [C++]

#### CTLTEMPLATE::LOAD [VBS]

[C++]

Метод Load позволяет загрузить данные шаблона из файла.

```
HRESULT Load(  
    [in] BSTR bstrURI ,  
    [in, defaultvalue(DER_TYPE)] FORMAT fFormat);  
;
```

#### Параметры

bstrURI [in] – путь к файлу, из которого будут считаны данные шаблона.  
fFormat [in, defaultvalue(DER\_TYPE)] – формат файла, из которого будут считаны данные шаблона. Допустимыми значениями являются значения FORMAT.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить данные шаблона из файла.

```
object.Load(bstrURI, fFormat)
```

**Параметры**

bstrURI - путь к файлу, из которого будут считаны данные шаблона.  
fFormat - формат файла, из которого будут считаны данные шаблона. Допустимыми значениями являются значения FORMAT.

**ICTLTEMPLATE::ADDTOSTORE [C++]****CTLTEMPLATE::ADDTOSTORE [VBS]**

[C++]

Свойство AddToStore позволяет получить/установить флаг установки создаваемого СДС в хранилище.

```
HRESULT get_AddToStore(  
    VARIANT* pVal  
);  
HRESULT put_AddToStore(  
    VARIANT newVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - будет производиться установка, FALSE - установка производиться не будет.  
newVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - будет производиться установка, FALSE - установка производиться не будет.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство AddToStore позволяет получить/установить флаг установки создаваемого СДС в хранилище.

```
object.Detach [= newVal]
```

**Параметры**

newVal - устанавливаемый шаблон, по которому будет генерироваться CTL.

**Возвращаемое значение**

Принимаемое значение TRUE или FALSE. TRUE - будет производиться установка, FALSE - установка производиться не будет.

**ICTLTEMPLATE::DESTINATIONFILENAME [C++]****CTLTEMPLATE::DESTINATIONFILENAME [VBS]**

[C++]

Свойство DestinationFilename позволяет получить/установить имя файла, в который будет сохранен созданный СДС.

```
HRESULT get_DestinationFilename(  
    BSTR* pVal  
);  
HRESULT put_DestinationFilename(  
    BSTR newVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку, содержащую имя файла.  
newVal [in] - строка, содержащая имя файла.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DestinationFilename позволяет получить/установить имя файла, в который будет сохранен созданный СДС.

```
object.DestinationFilename [= newVal]
```

#### Параметры

newVal - строка, содержащая имя файла.

### ICTLTEMPLATE::ENTRIES [C++]

### CTLTEMPLATE::ENTRIES [VBS]

[C++]

Свойство Entries позволяет получить/установить коллекцию элементов списка доверенных сертификатов.

```
HRESULT get_Entries(  
    ICTLEntries* pVal  
);  
HRESULT put_Entries(  
    ICTLEntries newVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - указатель на интерфейс ICTLEntries.  
newVal [in] - устанавливаемое значение - указатель на интерфейс ICTLEntries.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Entries позволяет получить/установить коллекцию элементов списка доверенных сертификатов.

```
object.Entries [= newVal]
```

#### Параметры

newVal - принимаемое значение - указатель на объект CTLEntries.

**Возвращаемое значение**

Возвращаемое значение - указатель на объект CTLEntries.

**ICTLTEMPLATE::INCLUDECERTIFICATES [C++]****CTLTEMPLATE::INCLUDECERTIFICATES [VBS]**

[C++]

Свойство IncludeCertificates позволяет получить/установить коллекцию сертификатов, которые должны быть включены в создаваемый СДС.

```
HRESULT get_IncludeCertificates(  
    ICertificates** pVal  
);  
HRESULT put_IncludeCertificates(  
    ICertificates* newVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - указатель на интерфейс ICertificates.  
newVal [in] - устанавливаемое значение - указатель на интерфейс ICertificates.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство IncludeCertificates позволяет получить/установить коллекцию сертификатов, которые должны быть включены в создаваемый СДС.

```
object.IncludeCertificates [= newVal]
```

**Параметры**

newVal  
Принимаемое значение - указатель на объект Certificates.

**Возвращаемое значение**

Возвращаемое значение - указатель на объект Certificates.

**ICTLTEMPLATE::LISTIDENTIFIER [C++]****CTLTEMPLATE::LISTIDENTIFIER [VBS]**

[C++]

Свойство ListIdentifier позволяет записывать шаблон, по которому генерится CTL.

```
HRESULT get_Template(  
    ICTLTemplate** pVal  
);  
HRESULT put_Comment(  
    ICTLTemplate* newVal  
);
```

**Параметры**

pVal [out, retval] - шаблон, содержащийся в свойстве.

pVal [in] - шаблон, который будет установлен в свойство.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Template позволяет записывать шаблон, по которому генерится CTL.

```
object.Template [= newVal]
```

### Параметры

newVal - устанавливаемый шаблон, по которому будет генерироваться CTL.

### Возвращаемое значение

Установленный шаблон, по которому будет генерироваться CTL.

## ICTLTEMPLATE::NEXTUPDATE [C++]

## CTLTEMPLATE::NEXTUPDATE [VBS]

[C++]

Свойство **NextUpdate** позволяет получить дату и время следующего обновления СДС.

```
HRESULT get_NextUpdate(  
    DATE* pVal  
);  
HRESULT put_NextUpdate(  
    DATE newVal  
);
```

### Параметры

pVal [out, retval] - дата и время следующего обновления СДС.  
newVal [in] - дата и время следующего обновления СДС.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство NextUpdate позволяет получить дату и время следующего обновления СДС.

```
object.NextUpdate [=newVal]
```

### Параметры

newVal - дата и время следующего обновления СДС.

## ICTLTEMPLATE::SAVETOFILE [C++]

## CTLTEMPLATE::SAVETOFILE [VBS]

[C++]

Свойство SaveToFile позволяет получить/установить флаг сохранения создаваемого CTL в файл.

```
HRESULT get_SaveToFile(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_SaveToFile(  
    VARIANT_BOOL newVal  
);
```



```
VARIANT_BOOL newVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - будет производиться сохранение, FALSE - сохранение производиться не будет.  
newVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - будет производиться сохранение, FALSE - сохранение производиться не будет.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SaveToFile позволяет получить/установить флаг сохранения создаваемого CTL в файл.

```
object.SaveToFile [= newVal]
```

### Параметры

newVal Принимаемое значение TRUE или FALSE. TRUE - будет производиться сохранение, FALSE - сохранение производиться не будет.

## ICTLTEMPLATE::SEQUENCENUMBER [C++]

## CTLTEMPLATE::SEQUENCENUMBER [VBS]

[C++]

Свойство SequenceNumber позволяет получить/установить значение номера последовательности.

```
HRESULT get_SequenceNumber(  
FORMAT frmtFormat,  
BSTR* pVal  
);  
HRESULT put_SequenceNumber(  
FORMAT frmtFormat,  
[in] BSTR newVal  
);
```

### Параметры

frmtFormat [in, defaultvalue(HEX\_TYPE)] - позволяет установить формат данных, в которых будет получено/установлено значение.  
pVal [out, retval] - указатель на буфер в памяти, в который сохранится значение номера последовательности.  
newVal [in] - указатель на буфер в памяти, в котором хранится значение номера последовательности.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SequenceNumber позволяет получить/установить значение номера последовательности.

```
object.SequenceNumber[(frmtFormat)] [= newVal]
```

### Параметры

frmtFormat - позволяет установить формат данных, в которых будет получено/установлено значение.  
newVal - указатель на буфер в памяти, в который сохранится значение номера последовательности.

**Возвращаемое значение**

Возвращаемое значение – указатель на буфер в памяти, в котором хранится значение номера последовательности.

**ICTLTEMPLATE::SIGNCTL [C++]****CTLTEMPLATE::SIGNCTL [VBS]**

[C++]

Свойство SignCTL позволяет получить/установить флаг подписывания создаваемого CTL.

```
HRESULT get_SignCTL(  
    VARIANT* pVal  
);  
HRESULT put_SignCTL(  
    VARIANT newVal  
);
```

**Параметры**

pVal [out, retval] – возвращаемое значение TRUE или FALSE. TRUE – создаваемый CTL будет подписанным, FALSE – создаваемый CTL будет не подписанным.  
newVal [in] – устанавливаемое значение TRUE или FALSE. TRUE – создаваемый CTL будет подписанным, FALSE – создаваемый CTL будет не подписанным.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignCTL позволяет получить/установить флаг подписывания создаваемого CTL.

```
object.SignCTL[= newVal]
```

**Параметры**

newVal – принимаемое значение TRUE или FALSE. TRUE – создаваемый CTL будет подписанным, FALSE – создаваемый CTL будет не подписанным.

**ICTLTEMPLATE::SIGNPROFILE [C++]****CTLTEMPLATE::SIGNPROFILE [VBS]**

[C++]

Свойство SignProfile позволяет получить/установить профиль с настройками, с помощью которых будет подписан создаваемый CTL.

```
HRESULT get_SignProfile(  
    IProfile** pVal  
);  
HRESULT put_SignProfile(  
    IProfile* newVal  
);
```

**Параметры**

pVal [out, retval] – возвращает интерфейс IProfile, описывающий установленный профиль.  
newVal [in] – указатель на интерфейс IProfile, описывающий устанавливаемый профиль.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignProfile позволяет получить/установить профиль с настройками, с помощью которых будет подписан создаваемый CTL.

```
object.Profile [= newVal]
```

**Параметры**

newVal - указатель на интерфейс Profile, описывающий профиль.

**ICTLTEMPLATE::SUBJECTALGORITHM [C++]****CTLTEMPLATE::SUBJECTALGORITHM [VBS]**

[C++]

Свойство SubjectAlgorithm позволяет получить/установить алгоритм, с помощью которого будут хэшироваться сертификаты для списка доверенных сертификатов (SubjectIdentifier в CTLEntry).

```
HRESULT put_SubjectAlgorithm(  
    BSTR* pVal  
);  
HRESULT get_SubjectAlgorithm(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - возвращает строку с OID-ом алгоритма.  
newVal [in] - строка с OID-ом алгоритма.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SubjectAlgorithm позволяет получить/установить алгоритм, с помощью которого будут хэшироваться сертификаты для списка доверенных сертификатов (SubjectIdentifier в CTLEntry).

```
object.SubjectAlgorithm[= newVal]
```

**Параметры**

newVal - строка с OID-ом алгоритма.

**Возвращаемое значение**

Установленный шаблон, по которому будет генерироваться CTL.

**ICTLTEMPLATE::THISUPDATE [C++]****CTLTEMPLATE::THISUPDATE [VBS]**

[C++]

Свойство ThisUpdate позволяет получить/установить дату и время выпуска текущего СДС.

```
HRESULT get_ThisUpdate(  
    DATE* pVal
```

```
);  
HRESULT put_ThisUpdate(  
    DATE newVal  
);
```

#### Параметры

pVal [out, retval] – дата и время выпуска текущего СДС  
newVal [in] – дата и время выпуска текущего СДС

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ThisUpdate позволяет получить/установить дату и время выпуска текущего СДС.

```
object.ThisUpdate [= newVal]
```

#### Параметры

newVal – дата и время выпуска текущего СДС

#### Возвращаемое значение

Дата и время выпуска текущего СДС.

### ICTLTEMPLATE::USAGEIDENTIFIERS [C++]

### CTLTEMPLATE::USAGEIDENTIFIERS [VBS]

[C++]

Свойство UsageIdentifiers позволяет получить/установить список назначений сертификатов создаваемого списка.

```
HRESULT put_UsageIdentifiers(  
    IOIDs* newVal  
);  
HRESULT get_UsageIdentifiers(  
    IOIDs** pVal  
);
```

#### Параметры

newVal [in] – указатель на интерфейс IOIDs.  
pVal [out, retval] – возвращает указатель на интерфейс IOIDs.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UsageIdentifiers позволяет получить/установить список назначений сертификатов списка.

```
object.UsageIdentifiers [= newVal]
```

#### Параметры

newVal – указатель на интерфейс IOIDs.

#### Возвращаемое значение

Указатель на интерфейс IOIDs.

**ICTLTEMPLATE::USENEXTUPDATE [C++]****CTLTEMPLATE::USENEXTUPDATE [VBS]****[C++]**

Свойство UseNextUpdate позволяет запустить/остановить флаг необходимости включения в CTL расширения NextUpdate.

```
HRESULT get_UseNextUpdate(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseNextUpdate(  
    VARIANT_BOOL newVal  
);
```

**Параметры**

newVal [in] TRUE - NextUpdate будет включен в создаваемый CTL, FALSE - NextUpdate не будет включен в создаваемый CTL.  
pVal [out, retval] TRUE - окна с отображением текущего статуса запущено, FALSE - остановлено.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство UseNextUpdate позволяет запустить/остановить флаг необходимости включения в CTL расширения NextUpdate.

```
object.UseNextUpdate [= newVal]
```

**Параметры**

newVal - TRUE - NextUpdate будет включен в создаваемый CTL, FALSE - NextUpdate не будет включен в создаваемый CTL.

**Возвращаемое значение**

Возвращает флаг необходимости включения в CTL расширения NextUpdate.

**DICTIONARY (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)****Свойства**

Свойство	Описание
Type	Свойство позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных

**IDictionary::Type [C++]****Dictionary::Type [VBS]****[C++]**

Свойство Type позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных.

```
HRESULT get_Type(  
    OUT DICTIONARY_TYPE* enumType-a  
);  
STDMETHOD put_Type(  

```

```
IN DICTIONARY_TYPE enumType-a  
);
```

### Параметры

enumType-a - позволяет получить значение типа словаря. Может принимать значения из DICTIONARY\_TYPE.

enumType - позволяет установить значение типа словаря. Может принимать значения из DICTIONARY\_TYPE.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Type позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных.

```
object.Type [=enumType]
```

### Параметры

enumType - позволяет установить значение типа словаря. Может принимать значения из DICTIONARY\_TYPE.

### Возвращаемые значения

Значение типа словаря. Может принимать значения из DICTIONARY\_TYPE.

## DICTIONARYSTORE (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

### Методы

Метод	Описание
Add	Метод позволяет добавить словарь в коллекцию
Clear	Метод позволяет очистить хранилище
Close	Метод позволяет закрыть хранилище
Open	Метод позволяет открыть хранилище
Save	Метод позволяет сохранить в хранилище текущую коллекцию словарей. Старые данные, находящиеся в хранилище, удаляются
Remove	Метод позволяет удалить из коллекции элемент (словарь) с указанным индексом

### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество словарей в хранилище
Item	Свойство позволяет получить элемент из хранилища, по заданному индексу

### IDictionaryStore::Add [C++]

### DictionaryStore::Add [VBS]

[C++]

Метод Add позволяет добавить словарь в коллекцию.

```
HRESULT Add(  
    IDictionary* pDict  
);
```

**Параметры**

pDict - представляет объект словаря, который будет добавлен в хранилище.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить словарь в коллекцию.

```
object.Add( pDict )
```

**Параметры**

pDict - представляет объект словаря, который будет добавлен в хранилище.

**IDictionaryStore::Clear [C++]****DictionaryStore::Clear [VBS]**

[C++]

Метод Clear позволяет очистить хранилище.

```
HRESULT Clear(  
    );
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clear позволяет добавить словарь в коллекцию.

```
object.Clear
```

**IDictionaryStore::Close [C++]****DictionaryStore::Close [VBS]**

[C++]

Метод Close позволяет закрыть хранилище.

```
HRESULT Close(  
    );
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Close позволяет закрыть хранилище.

```
object.Close
```

**IDictionaryStore::Open [C++]****DictionaryStore::Open [VBS]**

**[C++]**

Метод Open позволяет открыть хранилище.

```
HRESULT Open(  
PROFILESTORETYPE stStoreType,  
BSTR bstrPath  
);
```

**Параметры**

stStoreType [in] – тип хранилища словарей. Допустимыми значениями являются значения PROFILESTORETYPE.  
bstrPath [in, defaultvalue("")] – указывает имя xml-файла хранилища OID-ов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Open позволяет открыть хранилище.

```
object.Open(stStoreType,bstrPath)
```

**Параметры**

stStoreType – тип хранилища словарей. Допустимыми значениями являются значения PROFILESTORETYPE.  
bstrPath – указывает имя xml-файла хранилища OID-ов.

**IDictionaryStore::Save [C++]****DictionaryStore::Save [VBS]****[C++]**

Метод Save позволяет сохранить в хранилище текущую коллекцию словарей. Старые данные, находящиеся в хранилище, удаляются.

```
HRESULT Save(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Save позволяет сохранить в хранилище текущую коллекцию словарей. Старые данные, находящиеся в хранилище, удаляются.

```
object.Save
```

**IDictionaryStore::Remove [C++]****DictionaryStore::Remove [VBS]****[C++]**

Метод Remove позволяет удалить из коллекции элемент (словарь) с указанным индексом.

```
HRESULT Remove(  
LONG lIndex  
);
```



**Параметры**

`lIndex` - указывает, какой элемент из хранилища необходимо удалить.

**Возвращаемые значения**

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Метод `Remove` позволяет удалить из коллекции элемент (словарь) с указанным индексом.

```
object.Remove( lIndex )
```

**Параметры**

`lIndex` - указывает, какой элемент из хранилища необходимо удалить.

**IDictionaryStore::Count [C++]****DictionaryStore::Count [VBS]**

[C++]

Свойство `Count` позволяет получить количество словарей в хранилище.

```
HRESULT get_Count(  
    LONG* pVal  
);
```

**Параметры**

`pVal` - позволяет получить количество элементов в хранилище.

**Возвращаемые значения**

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `Count` позволяет получить количество словарей в хранилище.

```
object.Count
```

**Возвращаемые значения**

Количество элементов в хранилище.

**IDictionaryStore::Item [C++]****DictionaryStore::Item [VBS]**

[C++]

Свойство `Item` позволяет получить элемент из хранилища, по заданному индексу.

```
HRESULT get_Item(  
    long lIndex  
    IDictionary** pVal  
);
```

**Параметры**

`lIndex [in]` - номер элемента в хранилище минус 1 [0 .. Count-1].  
`pVal [out, retval]` - указатель на `IDictionary`.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент из хранилища, по заданному индексу.

```
object.Item (lIndex)
```

### Параметры

lIndex – номер элемента в хранилище минус 1 [0 .. Count-1].

### Возвращаемые значения

Указатель на Dictionary.

## ENCRYPTEDMESSAGE (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

Предоставляет свойства и методы для шифрования/расшифрования блоков данных на одном сессионном ключе.

### Методы

Свойства	Описание
Export	Метод позволяет экспортировать данные из объекта в область памяти (строку)
Import	Метод позволяет импортировать данные в объект
Load	Метод позволяет загружать последующие фрагменты данных в объект
Open	Метод позволяет инициализировать объект для его последующей обработки. В качестве объекта могут выступать: открытые данные, которые необходимо зашифровать; зашифрованные данные.
Save	Метод позволяет сохранять данные из интерфейса

### Свойства

Свойство	Описание
Profile	Свойство позволяет получить профиль для шифрования

## IENCRYPTEDMESSAGE::EXPORT [C++]

## ENCRYPTEDMESSAGE::EXPORT [VBS]

[C++]

Метод Export позволяет экспортировать данные из объекта в область памяти (строку).

```
HRESULT Export(  
    BSTR* pbstrDestination  
);
```

### Параметры

pbstrDestination [out, retval] – приемник экспортируемых из объекта данных.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать данные из объекта в область памяти (строку).

```
object.Export()
```

**Возвращаемое значение**

Строка – приемник экспортируемых из объекта данных.

**IENCRYPTEDMESSAGE::IMPORT [C++]****ENCRYPTEDMESSAGE::IMPORT [VBS]**

[C++]

Метод Import позволяет импортировать данные в объект из области памяти.

```
HRESULT Import(  
    BSTR bstrSource  
);
```

**Параметры**

bstrSource [in] – источник импортируемых данных, расположенный в памяти.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать данные в объект из области памяти.

```
object.Import(bstrSource)
```

**Параметры**

bstrSource – источник импортируемых данных, расположенный в памяти.

**IENCRYPTEDMESSAGE::LOAD [C++]****ENCRYPTEDMESSAGE::LOAD [VBS]**

[C++]

Метод Load позволяет загружать данные для их последующей обработки. Данными могут служить:

- открытые данные, которые необходимо зашифровать;
- зашифрованные данные.

```
HRESULT Load(  
    BSTR bstrSrcFileName  
);
```

**Параметры**

bstrSrcFileName [in] – URI (имя файла), откуда необходимо загрузить данные.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загружать данные для их последующей обработки. Данными могут служить:

- открытые данные, которые необходимо зашифровать;

– зашифрованные данные.

```
object.Load (bstrSrcFileName)
```

### Параметры

bstrSrcFileName – URI (имя файла), откуда необходимо загрузить данные.

## IENCRYPTEDMESSAGE::OPEN [C++]

## ENCRYPTEDMESSAGE::OPEN [VBS]

[C++]

Метод Open позволяет инициализировать объект для его последующего использования для шифрования или расшифрования.

В случае шифрования необходимо указать только тип DT\_PLAIN\_TEXT. При этом выходной параметр не обрабатывается. Если непосредственно после метода Open вызвать метод Save, то в отдельный файл будет сохранена первая порция (заголовок) шифрованного сообщения.

В случае расшифрования входными параметрами необходимо указать тип DT\_ENCRYPTED\_DATA, исходные данные - первую порцию (заголовок) шифрованного сообщения.

```
HRESULT Open(  
    DATA_TYPE enumDataType  
    BSTR bstrSource  
    ICertificate** ppDecryptingCertificate  
);
```

### Параметры

enumDataType [in] – тип загружаемых данных. Допустимыми значениями являются значения типа DATA\_TYPE: DT\_PLAIN\_TEXT – для шифрования, и DT\_ENCRYPTED\_DATA – для расшифрования.

bstrSource [in, optional, defaultvalue("")] – строка с исходными данными, содержащая первую порцию (заголовок) шифрованного сообщения. Параметр используется только в случае расшифрования.

ppDecryptingCertificate [out, retval] – сертификат расшифрования. Возвращается сертификат только в случае выполнения успешного расшифрования. Если сертификат расшифрования не найден или производилась операция шифрования, то параметр возвращает NULL.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет инициализировать объект для его последующего использования для шифрования или расшифрования.

В случае шифрования необходимо указать только тип DT\_PLAIN\_TEXT. При этом выходной параметр не обрабатывается. Если непосредственно после метода Open вызвать метод Save, то в отдельный файл будет сохранена первая порция (заголовок) шифрованного сообщения.

В случае расшифрования входными параметрами необходимо указать тип DT\_ENCRYPTED\_DATA, исходные данные - первую порцию (заголовок) шифрованного сообщения.

```
object.Open (enumDataType, bstrSource)
```

### Параметры

enumDataType - тип загружаемых данных. Допустимыми значениями являются значения типа DATA\_TYPE: DT\_PLAIN\_TEXT - для шифрования, и DT\_ENCRYPTED\_DATA - для расшифрования.

bstrSource - строка с исходными данными, содержащая первую порцию (заголовок) зашифрованного сообщения. Параметр используется только в случае расшифрования.

### Возвращаемые значения

Сертификат расшифрования. Возвращается сертификат только в случае выполнения успешного расшифрования. Если сертификат расшифрования не найден или производилась операция шифрования, то параметр возвращает Nothing.

## IENCRYPTEDMESSAGE::SAVE [C++]

### ENCRYPTEDMESSAGE::SAVE [VBS]

[C++]

Метод Save позволяет сохранять данные из интерфейса. Если непосредственно после метода Open вызвать метод Save, то в отдельный файл будет сохранена первая порция (заголовок) зашифрованного сообщения. Если метод Save вызвать после Load или Import, то заголовок будет сохранен в одном файле с первым зашифрованным блоком.

```
HRESULT Save(  
    BSTR bstrDestFileName  
);
```

### Параметры

bstrDestFileName [in] - имя файла, куда необходимо сохранить данные.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранять данные из интерфейса. Если непосредственно после метода Open вызвать метод Save, то в отдельный файл будет сохранена первая порция (заголовок) зашифрованного сообщения. Если метод Save вызвать после Load или Import, то заголовок будет сохранен в одном файле с первым зашифрованным блоком.

```
object.Save(bstrDestFileName)
```

### Параметры

Имя файла, куда необходимо сохранить данные.

## IENCRYPTEDMESSAGE::PROFILE [C++]

### ENCRYPTEDMESSAGE::PROFILE [VBS]

[C++]

Свойство Profile позволяет получить/установить профиль с настройками.

```
HRESULT get_Profile(  
    IProfile** pVal  
);  
HRESULT put_Profile(  
    IProfile* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает интерфейс IProfile, описывающий текущий профиль.

pVal [in] - указатель на интерфейс IProfile, описывающий текущий профиль.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет получить/установить профиль с настройками.

```
object.Profile [= pVal]
```

### Параметры

pVal - указатель на интерфейс Profile, описывающий текущий профиль.

## IDictionaryStore::Item [C++]

## DictionaryStore::Item [VBS]

[C++]

Свойство Item позволяет получить элемент из хранилища, по заданному индексу.

```
HRESULT get_Item(  
    long lIndex  
    IDictionary** pVal  
);
```

### Параметры

lIndex [in] - номер элемента в хранилище минус 1 [0 .. Count-1].

pVal [out, retval] - указатель на IDictionary.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент из хранилища, по заданному индексу.

```
object.Item (lIndex)
```

### Параметры

lIndex - номер элемента в хранилище минус 1 [0 .. Count-1].

### Возвращаемые значения

Указатель на Dictionary.

## OID (GUID: 6039A122-6BEA-41AA-9C01-AD71C757A07C)

### Свойства

Свойство	Описание
FriendlyName	Свойство позволяет получить/установить пользовательское имя OIda.
Value	Свойство позволяет получить/установить значение OIda.

### Методы

Метод	Описание
Display	Свойство позволяет открыть форму просмотра идентификатора (OID).

**IOID::DISPLAY [C++]****OID::DISPLAY [VBS]**

[C++]

Метод Display позволяет открыть форму просмотра/редактирования идентификатора (OID-a).

```
HRESULT Display(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает TRUE, если было произведено нажатие ОК, и FALSE, если Отмена.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет открыть форму просмотра/редактирования идентификатора (OID-a).

```
object.Display ()
```

**Возвращаемое значение**

Возвращает TRUE, если было произведено нажатие ОК, и FALSE, если Отмена.

**IOID::FRIENDLYNAME [C++]****OID::FRIENDLYNAME [VBS]**

[C++]

Свойство FriendlyName получить/установить пользовательское имя OIDa.

```
HRESULT get_FriendlyName(  
    BSTR* pVal  
);  
HRESULT put_FriendlyName(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строковое значение имени OIDa.  
newVal [in] - строковое значение имени OIDa.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство FriendlyName получить/установить пользовательское имя OIDa.

```
object.FriendlyName [= Value]
```

**Параметры**

Value - строковое значение имени OIDa.

## IOID::VALUE [C++]

## OID::VALUE [VBS]

[C++]

Свойство Value позволяет получить/установить значение OIDa.

```
HRESULT get_Value(  
    BSTR* pVal  
);  
HRESULT put_Value(  
    BSTR newVal  
);
```

### Параметры

pVal [out, retval] - указатель на строковое значение OIDa.  
newVal [in] - строковое значение OIDa.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Value позволяет получить/установить значение OIDa.

object.Value [= Value]

### Параметры

Value - строковое значение OIDa.

## OIDPARAM (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

### Свойства

Свойства	Описание
Flags	Свойство позволяет получить/установить способ отображения и получения атрибутов относительного отличительного имени (RDN)
OID	Свойство позволяет получить/установить указатель на интерфейс IOID
Value	Свойство позволяет получить/установить значение атрибута относительного отличительного имени (RDN)

## IOIDPARAM::FLAGS [C++]

## OIDPARAM::FLAGS [VBS]

[C++]

Свойство Flags позволяет получить/установить способ отображения и получения атрибутов относительного отличительного имени (RDN) Используется это свойство на данный момент только в контексте интерфейса IRDNEntry.

```
HRESULT put_Flags(  
    long newVal  
);  
HRESULT get_Flags(  
    long* pVal  
);
```



```
long* pVal  
);
```

### Параметры

pVal [in] – целое число. В контексте IRDNEntry принимает значение из RDN\_ENTRY\_FLAG.

pVal [out, retval] – указатель на целое число. В контексте IRDNEntry принимает значение из RDN\_ENTRY\_FLAG.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Flags позволяет получить/установить строковое значение атрибута RDN.

```
object.Flags[=pVal]
```

### Параметры

pVal – устанавливаемое значение – целое число. В контексте IRDNEntry принимает значение из RDN\_ENTRY\_FLAG.

### Возвращаемое значение

Указатель на целое число. В контексте IRDNEntry принимает значение из RDN\_ENTRY\_FLAG.

## IOIDPARAM::OID [C++]

### OIDPARAM::OID [VBS]

[C++]

Свойство OID позволяет получить/установить указатель на интерфейс IOID, определяющий идентификатор и дружественное имя атрибута RDN. При этом, при отображении полей RDN (на шаге установки идентификационной информации в мастере создания запроса) значение, указанное в свойстве Value интерфейса IOID будет отображаться на форме как названия полей, а значение свойства Value интерфейса IOIDParam будет отображаться значением атрибута.

```
HRESULT put_OID(  
    IOID* newVal  
);  
HRESULT get_OID(  
    IOID** pVal  
);
```

### Параметры

pVal [in] – указатель на интерфейс IOID.

pVal [out, retval] – указатель на указатель на интерфейс IOID.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OID позволяет получить/установить указатель на объект OID, определяющий идентификатор и дружественное имя атрибута RDN. При этом, при отображении полей RDN (на шаге установки идентификационной информации в мастере создания запроса) значение, указанное в свойстве Value объекта OID будет отображаться на форме как названия полей, а значение свойства Value объекта OIDParam будет отображаться значением атрибута.

```
object.OID()
```

### Параметры

pVal - устанавливаемое значение - указатель на объект OID.

### Возвращаемое значение

Указатель на указатель на объект OID.

## OIDPARAM::VALUE [C++]

## OIDPARAM::VALUE [VBS]

[C++]

Свойство Value позволяет получить/ установить значение атрибута RDN.

```
HRESULT put_Value(  
    VARIANT newVal  
);  
HRESULT get_Value(  
    VARIANT* pVal  
);
```

### Параметры

pVal [in] - строковое значение.

pVal [out, retval] - указатель на строку.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Value позволяет получить/ установить значение атрибута RDN.

```
object.Value[=pVal]
```

### Параметры

pVal - устанавливаемое значение - строковое значение.

### Возвращаемое значение

Указатель на строку.

## OIDPARAMS (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

### Методы

Свойства	Описание
Add	Метод позволяет добавить элемент OIDParam в коллекцию элементов
Remove	Свойство позволяет удалить элемент OIDParam из коллекции элементов

### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество элементов коллекции OIDParams
Item	Свойство позволяет получить/ установить элемент коллекции

**IOIDPARAMS::ADD [C++]****OIDPARAMS::ADD [VBS]**

[C++]

Метод Add позволяет добавить элемент OIDParam в коллекцию элементов.

```
HRESULT Add(  
    IOIDParam* pNewOIDParam  
);
```

**Параметры**

pVal [in] - указатель на интерфейс IOIDParam.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить элемент OIDParam в коллекцию элементов.

```
object.Add(pNewOIDParam)
```

**Параметры**

pNewOIDParam - указатель на объект OIDParam.

**IOIDPARAMS::REMOVE [C++]****OIDPARAMS::REMOVE [VBS]**

[C++]

Метод Remove позволяет удалить элемент OIDParam из коллекции элементов.

```
HRESULT Remove(  
    VARIANT varIndex  
);
```

**Параметры**

varIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить элемент OIDParam из коллекции элементов.

```
object.Remove(pNewOIDParam)
```

**Параметры**

varIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].

**IOIDPARAMS::COUNT [C++]****OIDPARAMS::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество элементов коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - количество элементов коллекции.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Count позволяет получить количество элементов коллекции.

```
object.Count(pNewOIDParam)
```

#### Возвращаемые значения

Количество элементов коллекции.

### IOIDPARAMS::ITEM [C++]

### OIDPARAMS::ITEM [VBS]

[C++]

Свойство Item позволяет получить/ установить элемент коллекции .

```
HRESULT get_Item(  
    VARIANT varIndex  
    IOIDParam* pVal  
);  
HRESULT put_Item(  
    VARIANT varIndex  
    IOIDParam** pVal  
);
```

#### Параметры

varIndex [in] - индекс элемента в коллекции. Индексом элемента может выступать как номер элемента в коллекции, так и ключ (OIDParam.OID.Value)  
pVal [in] - указатель на интерфейс IOIDParam, который будет добавляться в коллекцию. При этом, если в коллекции уже имеется элемент с номером varIndex, то переданный элемент OIDParam заменит в коллекции существующий. Иначе будет добавлен новый элемент в коллекции с указанным номером.  
pVal [out, retval] - указатель на указатель на интерфейс IOIDParam, получаемый по идентификатору элемента в коллекции.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Item позволяет получить/ установить элемент коллекции.

```
object.Item(varIndex, =pVal)
```

#### Параметры

varIndex - индекс элемента в коллекции. Индексом элемента может выступать как номер элемента в коллекции, так и ключ (OIDParam.OID.Value)  
pVal - устанавливаемое значение - указатель на объект OIDParam, который будет добавляться в коллекцию. При этом, если в коллекции уже имеется элемент с но-

мером varIndex, то переданный элемент OIDParam заменит в коллекции существующий. Иначе будет добавлен новый элемент в коллекции с указанным номером.

Возвращаемые значения

Указатель на указатель на объект OIDParam, получаемый по идентификатору элемента в коллекции.

OIDs (GUID: EA2B23CA-042E-4EA9-845F-E72013FCD245)

Методы

Метод	Описание
Add	Метод позволяет добавить элемент в коллекцию
Display	Метод позволяет открыть форму просмотра коллекции идентификаторов (OID)
Export	Метод позволяет произвести экспорт коллекции идентификаторов (OID-ов) в XML строку
Find	Метод позволяет найти все элементы коллекции, удовлетворяющие заданным критериям
Import	Метод позволяет произвести импорт коллекции идентификаторов (OID-ов) из XML строки
Load	Метод позволяет загрузить коллекцию идентификаторов (OID-ов) из файла в формате XML
Remove	Метод позволяет удалять элемент из коллекции
Save	Метод позволяет сохранить коллекцию идентификаторов (OID-ов) в файл в формате XML

Свойства

Свойство	Описание
Count	Свойство позволяет получить размер коллекции
Item	Свойство позволяет получить элемент коллекции

OIDs::Add [C++]

OIDs::Add [VBS]

[C++]

Метод Add позволяет добавить элемент в коллекцию.

```
HRESULT Add(  
    IOID* pNewOID,  
    VARIANT_BOOL vbReplaceExistingItem  
);
```

Параметры

pNewOID [in] - добавляемый элемент.  
vbReplaceExistingItem [in] - флаг замещения элемента с таким же Value.

Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить элемент в коллекцию.

```
object.Add(pNewOID, vbReplaceExistingItem)
```

## Параметры

pNewOID - добавляемый элемент.

vbReplaceExistingItem - флаг замещения элемента с таким же Value.

## OIDs::DISPLAY [C++]

## OIDs::DISPLAY [VBS]

[C++]

Метод Display позволяет открыть форму просмотра коллекции идентификаторов (OID).

```
HRESULT Display(  
    VARIANT_BOOL* pVal  
);
```

## Параметры

lIndex [out, retval] - возвращает TRUE, если в окне нажали кнопку «ОК» или «Применить» и FALSE, если была нажата «Отмена».

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет открыть форму просмотра коллекции идентификаторов (OID).

```
object.Display()
```

## Возвращаемое значение

Возвращает TRUE, если в окне нажали кнопку "OK" или "Применить" и FALSE, если была нажата "Отмена"

## OIDs::EXPORT [C++]

## OIDs::EXPORT [VBS]

[C++]

Метод Export позволяет произвести экспорт коллекции идентификаторов (OID-ов) в XML строку.

```
HRESULT Export(  
    BSTR* pVal  
);
```

## Параметры

pVal [out, retval] - XML строка, содержащая коллекцию идентификаторов.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет произвести экспорт коллекции идентификаторов (OID-ов) в XML строку.

```
object.Export()
```

## Возвращаемое значение

XML строка, содержащая коллекцию идентификаторов.

## OIDs::FIND [C++]

### OIDs::FIND [VBS]

[C++]

Метод Find позволяет найти все элементы коллекции, удовлетворяющие заданным критериям.

```
HRESULT Find(  
    BSTR bstrSearchCondition,  
    SEARCH_CRITERIA enmSearchCriteria,  
    MATCH_CRITERIA enmMatchCriteria,  
    IOIDs** pSearchData  
);
```

#### Параметры

bstrSearchCondition [in] – искомое значение.  
enmSearchCriteria [in] – место поиска. Допустимыми значениями являются значения SEARCH\_CRITERIA.  
enmMatchCriteria [in, optional, defaultvalue(1)] – критерий совпадения. Допустимыми значениями являются значения MATCH\_CRITERIA.  
pSearchData [out, retval] – результирующая коллекция.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Find позволяет найти все элементы коллекции, удовлетворяющие заданным критериям.

```
object.Find(bstrSearchCondition, enmSearchCriteria, enmMatchCriteria)
```

#### Параметры

bstrSearchCondition – искомое значение.  
enmSearchCriteria – место поиска. Допустимыми значениями являются значения SEARCH\_CRITERIA.  
enmMatchCriteria – критерий совпадения. Допустимыми значениями являются значения MATCH\_CRITERIA.

#### Возвращаемое значение

Результирующая коллекция.

## OIDs::IMPORT [C++]

### OIDs::IMPORT [VBS]

[C++]

Метод Import позволяет произвести импорт коллекции идентификаторов (OID-ов) из XML строки.

```
HRESULT Import(  
    BSTR bstrSource  
);
```

#### Параметры

bstrSource [in] – XML строка, содержащая коллекцию идентификаторов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет произвести импорт коллекции идентификаторов (OID-ов) из XML строки.

```
object.Import(bstrSource)
```

#### Параметры

bstrSource - XML строка, содержащая коллекцию идентификаторов.

### OIDs::Load [C++]

### OIDs::Load [VBS]

[C++]

Метод Load позволяет загрузить коллекцию идентификаторов (OID-ов) из файла в формате XML.

```
HRESULT Load(  
    BSTR bstrURI  
);
```

#### Параметры

bstrURI [in] - строка, содержащая URI адрес XML файла.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить коллекцию идентификаторов (OID-ов) из файла в формате XML.

```
object.Load(bstrURI)
```

#### Параметры

bstrURI - строка, содержащая URI адрес XML файла.

### OIDs::Remove [C++]

### OIDs::Remove [VBS]

[C++]

Метод Remove позволяет удалить элемент из коллекции.

```
HRESULT Remove(  
    long lIndex  
);
```

#### Параметры

lIndex [in] - порядковый номер элемента в коллекции минус 1 [0 .. Count-1].

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить элемент из коллекции.

```
object.Remove(lIndex)
```

#### Параметры

lIndex - порядковый номер элемента в коллекции минус 1 [0 .. Count-1].



**OIDs::SAVE [C++]****OIDs::SAVE [VBS]**

[C++]

Метод Save позволяет сохранить коллекцию идентификаторов (OID-ов) в файл в формате XML.

```
HRESULT Save(  
    BSTR bstrURI  
);
```

**Параметры**

bstrURI [in] - строка, содержащая URI адрес XML файла.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить коллекцию идентификаторов (OID-ов) в файл в формате XML.

```
object.Save(bstrURI)
```

**Параметры**

bstrURI - строка, содержащая URI адрес XML файла.

**IOIDs::COUNT [C++]****OIDs::COUNT [VBS]**

[C++]

Свойство Count позволяет получить размер коллекции.

```
HRESULT get_Count(  
    LONG* pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на размер коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить размер коллекции.

```
object.Count
```

**Возвращаемое значение**

Указатель на размер коллекции.

**IOIDs::ITEM [C++]****OIDs::ITEM [VBS]**

[C++]

Свойство `Item` позволяет получить элемент коллекции.

```
HRESULT get_Item(  
    LONG lZeroBasedIndex,  
    IOID** pVal  
);
```

#### Параметры

`lZeroBasedIndex` [in] - номер элемента в коллекции минус один [0 .. Count-1].  
`pVal` [out, retval] - указатель на элемент коллекции.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `Item` позволяет получить элемент коллекции.

```
object.Item [= lZeroBasedIndex, pVal]
```

#### Параметры

`lZeroBasedIndex` - номер элемента в коллекции минус один [0 .. Count-1].  
`pVal` - указатель на элемент коллекции.

## HASHEDDATA (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

### Методы

Метод	Описание
Hash	Метод позволяет загрузить данные из буфера для вычисления хэша
HashFile	Метод позволяет загрузить данные из файла и вычислить хэш

### Свойства

Свойство	Описание
Algorithm	Свойство позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных
ProviderName	Свойство позволяет получить/задать криптопровайдер, который будет использоваться при хэшировании данных
Value	Свойство позволяет получить вычисленное значение хэша данных

## IHASHEDDATA::HASH [C++]

## HASHEDDATA::HASH [VBS]

[C++]

Метод `Hash` позволяет загрузить данные из буфера для вычисления хэша.

```
HRESULT Hash(  
    BSTR bstrData  
);
```

#### Параметры

`bstrData`  
[in] Строка (буфер) с данными, хэш которых будет вычислен.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Hash позволяет загрузить данные из буфера для вычисления хэша.

```
object.Hash=[bstrData]
```

#### Параметры

bstrData - буфер с данными, хэш которых будет вычислен.

#### IHASHEDDATA::HASHFILE [C++]

#### HASHEDDATA::HASHFILE [VBS]

[C++]

Метод HashFile позволяет загрузить данные из файла и вычислить хэш.

```
HRESULT HashFile(  
    BSTR bstrFilename  
);
```

#### Параметры

bstrFilename [in] - имя файла, хэш данных которого будет вычислен.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Hash позволяет загрузить данные из файла и вычислить хэш.

```
object.Hash=[bstrFilename]
```

#### Параметры

bstrFilename - имя файла, хэш данных которого будет вычислен.

#### IHASHEDDATA::ALGORITHM [C++]

#### HASHEDDATA::ALGORITHM [VBS]

[C++]

Свойство Algorithm позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных.

```
HRESULT get_Algorithm(  
    VARIANT* pVal  
);  
HRESULT put_Algorithm(  
    VARIANT newVal  
);
```

#### Параметры

pVal - позволяет получить строку с OID-ом алгоритма хэширования.

newVal - позволяет установить строку со значением OID-а алгоритма хэширования.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Algorithm позволяет задать/получить алгоритм хэширования, который будет использоваться при хэшировании данных.

```
object.Algorithm=[newVal]
```

#### Параметры

newVal - позволяет установить строку со значением OID-а алгоритма хэширования.

#### Возвращаемые значения

Строка с OID-ом алгоритма хэширования.

### IHASHEDDATA::PROVIDERNAME [C++]

### HASHEDDATA::PROVIDERNAME [VBS]

[C++]

Свойство ProviderName позволяет получить/задать криптопровайдер, который будет использоваться при хэшировании данных.

```
STDMETHOD get_ProviderName(  
    BSTR* pVal  
);  
STDMETHOD put_ProviderName(  
    BSTR newVal  
);
```

#### Параметры

pVal - позволяет получить имя криптопровайдера, который будет использован при хэшировании данных.

newVal - позволяет установить имя криптопровайдера, который будет использован при хэшировании данных.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProviderName позволяет получить/задать криптопровайдер, который будет использоваться при хэшировании данных.

```
object.ProviderName[=newVal]
```

#### Параметры

newVal - позволяет задать имя криптопровайдера, который будет использован при хэшировании данных.

#### Возвращаемые значения

Имя криптопровайдера, который будет использован при хэшировании данных.

### IHASHEDDATA::VALUE [C++]

### HASHEDDATA::VALUE [VBS]

**[C++]**

Свойство Value позволяет получить вычисленное значение хэша данных.

```
HRESULT get_Value(  
    FORMAT fFormat,  
    BSTR* pVal  
);
```

**Параметры**

pVal - строка, содержащая хэш.

fFormat - позволяет установить формат кодирования возвращаемого значения. Принимает следующие значения из FORMAT:

DER\_TYPE

BASE64\_TYPE

При передаче значения DER\_TYPE данные будут возвращены в исходном виде.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство Value позволяет получить вычисленное значение хэша данных.

```
object.Value[fFormat]
```

**Параметры**

fFormat - позволяет установить формат кодирования возвращаемого значения. Принимает следующие значения из FORMAT:

DER\_TYPE

BASE64\_TYPE - при передаче значения DER\_TYPE данные будут возвращены в исходном виде.

**Возвращаемые значения**

Строка, содержащая хэш.

**PKCS7MESSAGE (GUID: 0440DABC-BFFB-4AF6-9AF9-06FD52B0B5B8)****Методы**

Метод	Описание
Cosign	Метод позволяет добавлять встречную подпись к подписи, которая загружена в компонент. Встречная подпись формируется на основе информации из текущего профиля
Decrypt	Метод позволяет расшифровывать загруженные данные. Необходимые данные берутся из профиля
Display	Метод отображает загруженное сообщение с возможностью его редактирования
Encrypt	Метод позволяет шифровать загруженные данные (если они были загружены как PLAIN_DATA). Необходимые для операции данные берутся из профиля
Export	Метод позволяет экспортировать данные из компонента в область памяти
Import	Метод позволяет импортировать данные в компонент из области памяти
JoinSignature	Метод позволяет объединить отделенную подпись с исходными данными.
Load	Метод позволяет загружать данные для их последующей обработки. Данными могут служить: открытые данные, которые необходимо зашифровать; зашифрованные данные; подписанные данные (отделенная или совмещенная подпись)

Save	Метод позволяет сохранять данные из интерфейса
Sign	Метод позволяет подписать загруженные данные. Данные, необходимые для создания подписи, берутся из профиля
SplitSignature	Метод позволяет отделить подпись от исходных данных
GetContentOID	Метод позволяет получить OID контента из PKCS-7/9 сообщений. Метод не изменяет состояние COM-объекта, для этой цели следует использовать Load или Import.

### Свойства

Свойство	Описание
ContentType	Свойство позволяет получить описание типа загруженных в интерфейс данных
Profile	Свойство позволяет установить/получить указатель на интерфейс профиля IProfile
Recipients	Свойство позволяет установить коллекцию сертификатов получателей шифрованного сообщения
Signatures	Свойство позволяет получить коллекцию подписей загруженного ранее сообщения
Signers	Свойство позволяет получить коллекцию сертификатов подписчиков
ShowProgress	Свойство позволяет запустить/остановить окно с отображением текущего статуса ("процента") выполнения операции, а также получить текущее состояние (запущен или остановлен).

### IPKCS7MESSAGE::COSIGN [C++]

### PKCS7MESSAGE::COSIGN [VBS]

#### [C++]

Метод Cosign позволяет добавлять встречную подпись к подписи, которая загружена в компонент. Встречная подпись формируется на основе информации из текущего профиля.

Параметры профиля, необходимые для выполнения операции добавления подписи: Comment - комментарии к подписи Detach - флаг формирования отдельной подписи HashAlg - идентификатор алгоритма хеширования, используемого для создания подписи IncludeSignatureTime - флаг включения времени создания подписи IncludeTimeStampToken - флаг включения в подпись штампа времени SignIncludeBase64Headers - флаг добавления Base64-заголовков в выходном подписанном файле SignDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции подписи SignExitFormat - выходной формат подписанного сообщения TSPProfile или TSPProfileID - параметры, позволяющие установить профиль Службы Штампов Времени. Данные параметры необходимы, если IncludeTimeStampToken = True. Resource - значение идентификатора (URI) подписываемого файла.

Для выполнения операции добавления подписи необходимо установить в профиль сертификат подписи. Установка сертификата в профиль производится с помощью метода SetCertificate.

```
HRESULT Cosign(  
    long lIndex  
);
```

#### Параметры

`lIndex [in]` - порядковый номер подписи, которую необходимо заверить.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Cosign позволяет добавлять встречную подпись к подписи, которая загружена в компонент. Встречная подпись формируется на основе информации из текущего профиля.

Параметры профиля, необходимые для выполнения операции заверения подписи: Comment - комментарии к подписи Detach - флаг формирования отдельной подписи HashAlg - идентификатор алгоритма хеширования, используемого для создании подписи IncludeSignatureTime - флаг включения времени создания подписи. При выполнении операции заверения подписи необходимо установить IncludeSignatureTime = True. IncludeTimeStampToken - флаг включения в подпись штампа времени SignIncludeBase64Headers - флаг добавления Base64-заголовков в выходном подписанном файле SignDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции подписи SignExitFormat - выходной формат подписанного сообщения TSPProfile или TSPProfileID - параметры, позволяющие установить профиль Службы Штamped Времени. Данные параметры необходимы, если IncludeTimeStampToken = True. Resource - значение идентификатора (URI) подписываемого файла.

Для выполнения операции заверения подписи необходимо установить в профиль сертификат подписи. Установка сертификата в профиль производится с помощью метода SetCertificate.

```
object.Cosign(lIndex)
```

#### Параметры

`lIndex` - порядковый номер подписи, которую необходимо заверить.

### IPKCS7MESSAGE::DECRYPT [C++]

### PKCS7MESSAGE::DECRYPT [VBS]

[C++]

Метод Decrypt позволяет расшифровывать загруженные данные. Необходимые данные берутся из профиля.

Для установки в профиль сертификата расшифрования используется метод SetCertificate.

```
HRESULT Decrypt(  
VARIANT* pvtDecryptingCertificate  
);
```

#### Параметры

`pvtDecryptingCertificate [out, retval]` - признак наличия в сообщении сертификата расшифрования.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Decrypt Метод позволяет расшифровывать загруженные данные. Необходимые данные берутся из профиля.

Для установки в профиль сертификата расшифрования используется метод SetCertificate.

```
object.Decrypt()
```

#### Возвращаемое значение

Признак наличия в сообщении сертификата расшифрования.

**IPKCS7MESSAGE::DISPLAY [C++]****PKCS7MESSAGE::DISPLAY [VBS]**

[C++]

Метод Display отображает загруженное сообщение с возможностью его редактирования.

```
HRESULT Display(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**Примечание**

Метод Display выполняется только после вызова метода Load интерфейса IPKCS7Message.

[VBS]

Метод Display отображает загруженное сообщение с возможностью его редактирования.

```
object.Display()
```

**Примечание**

Метод Display выполняется только после вызова метода Load интерфейса IPKCS7Message.

**IPKCS7MESSAGE::ENCRYPT [C++]****PKCS7MESSAGE::ENCRYPT [VBS]**

[C++]

Метод Encrypt позволяет шифровать загруженные данные (если они были загружены как PLAIN\_DATA). Необходимые для операции данные берутся из профиля.

Параметры профиля, необходимые для выполнения операции шифрования: EncAlg - идентификатор алгоритма шифрования EncryptIncludeBase64Headers - флаг добавления Base64-заголовков в выходном зашифрованном файле EncryptDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции шифрования EncryptExitFormat - выходной формат зашифрованного сообщения EncryptToSenderAddress - флаг шифрования в свой адрес (отправителя) Provider - имя криптопровайдера, использующегося при шифровании Recipients - коллекция сертификатов получателей зашифрованного сообщения UseCertificateForEncrypt - флаг использования личного сертификата для определения параметров шифрования. Если параметр UseCertificateForEncrypt = True, то необходимо установить в профиль личный сертификат. Установка личного сертификата в профиль производится с помощью метода SetCertificate.

```
HRESULT Encrypt(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Encrypt позволяет шифровать загруженные данные (если они были загружены как PLAIN\_DATA). Необходимые для операции данные берутся из профиля.



Параметры профиля, необходимые для выполнения операции шифрования: EncAlg - идентификатор алгоритма шифрования EncryptIncludeBase64Headers - флаг добавления Base64-заголовков в выходном зашифрованном файле EncryptDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции шифрования EncryptExitFormat - выходной формат шифрованного сообщения EncryptToSenderAddress - флаг шифрования в свой адрес (отправителя) Provider - имя криптопровайдера, использующегося при шифровании Recipients - коллекция сертификатов получателей зашифрованного сообщения UseCertificateForEncrypt - флаг использования личного сертификата для определения параметров шифрования. Если параметр UseCertificateForEncrypt = True, то необходимо установить в профиль личный сертификат. Установка личного сертификата в профиль производится с помощью метода SetCertificate.

```
object.Encrypt()
```

## IPKCS7MESSAGE::EXPORT [C++]

### PKCS7MESSAGE::EXPORT [VBS]

[C++]

Метод Export позволяет экспортировать данные из компонента в область памяти.

```
HRESULT Export(  
    DATA_TYPE dtType,  
    FORMAT fFormat,  
    BSTR* pVal  
);
```

#### Параметры

dtType [in] - тип экспортируемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
fFormat [in] - тип используемой кодировки при сохранении (DER, Base64). Допустимыми значениями являются значения FORMAT.  
pVal [out, retval] - приемник экспортируемых из компонента данных.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать данные из компонента в область памяти.

```
object.Export(dtType, fFormat)
```

#### Параметры

dtType - тип экспортируемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
fFormat - тип используемой кодировки при сохранении (DER, Base64). Допустимыми значениями являются значения FORMAT.

#### Возвращаемое значение

Строка-приемник экспортируемых из компонента данных.

## IPKCS7MESSAGE::IMPORT [C++]

### PKCS7MESSAGE::IMPORT [VBS]

[C++]

Метод Import позволяет импортировать данные в компонент из области памяти.

```
HRESULT Import(  
    DATA_TYPE dtType,  
    BSTR bstrSource  
    BSTR bstrDetachedSourceURI  
    VARIANT_BOOL* pVal  
);
```

### Параметры

dtType [in] – тип импортируемых из памяти данных (зашифрованные данные, открытые данные или подписанные данные). Допустимыми значениями являются значения DATA\_TYPE.

bstrSource [in] – источник импортируемых данных, расположенный в памяти.

bstrDetachedSourceURI [in, optional, defaultvalue("")] – имя файла с исходными данными в случае импорта отсоединенной подписи.

pVal [out, retval] – возвращает флаг, были ли загружены подписанные данные (в составе подписи или из файла bstrDetachedSourceURI) в процессе загрузки подписи. Для типа данных, отличных от подписи, неопределен.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать данные в компонент из области памяти.

```
object.Import(dtType, bstrSource, bstrDetachedSourceURI)
```

### Параметры

dtType – тип импортируемых из памяти данных (зашифрованные данные, открытые данные или подписанные данные). Допустимыми значениями являются значения DATA\_TYPE.

bstrSource – источник импортируемых данных, расположенный в памяти.

bstrDetachedSourceURI – имя файла с исходными данными в случае импорта отсоединенной подписи.

### Возвращаемое значение

Возвращает флаг, были ли загружены подписанные данные (в составе подписи или из файла bstrDetachedSourceURI) в процессе загрузки подписи. Для типа данных, отличных от подписи, неопределен.

## IPKCS7MESSAGE::JOINSIGNATURE [C++]

### PKCS7MESSAGE::JOINSIGNATURE [VBS]

[C++]

Метод JoinSignature позволяет объединить отделенную подпись с исходными данными.

```
HRESULT JoinSignature(  
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод JoinSignature позволяет объединить отделенную подпись с исходными данными.

```
object.JoinSignature
```

## IPKCS7MESSAGE::LOAD [C++]

**PKCS7MESSAGE::LOAD [VBS]**

[C++]

Метод Load позволяет загружать данные для их последующей обработки. Данными могут служить:

- открытые данные, которые необходимо зашифровать;
- зашифрованные данные;
- подписанные данные (отделенная или совмещенная подпись).

```
HRESULT Load(  
    DATA_TYPE dtType,  
    BSTR bstrSourceURI  
    BSTR bstrDetachedSourceURI  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

dtType [in] – тип загружаемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
bstrSourceURI [in] – URI (имя файла), откуда необходимо загрузить сообщение.  
bstrDetachedSourceURI [in, optional, defaultvalue("")] – URI (имя файла) с отделенными (исходными) данными в случае загрузки отсоединенной подписи.  
pVal [out, retval] – возвращает флаг, были ли загружены подписанные данные (в составе подписи или из файла bstrDetachedSourceURI) в процессе загрузки подписи. Для типа данных, отличных от подписи, неопределен.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загружать данные для их последующей обработки. Данными могут служить:

- открытые данные, которые необходимо зашифровать;
- зашифрованные данные;
- подписанные данные (отделенная или совмещенная подпись).

```
object.Load(dtType, bstrSourceURI, bstrDetachedSourceURI)
```

**Параметры**

dtType – тип загружаемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
bstrSourceURI – URI (имя файла), откуда необходимо загрузить сообщение.  
bstrDetachedSourceURI  
URI (имя файла) с отделенными (исходными) данными в случае загрузки отсоединенной подписи.

**Возвращаемое значение**

Возвращает флаг, были ли загружены подписанные данные (в составе подписи или из файла bstrDetachedSourceURI) в процессе загрузки подписи. Для типа данных, отличных от подписи, неопределен.

**IPKCS7MESSAGE::SAVE [C++]****PKCS7MESSAGE::SAVE [VBS]**

**[C++]**

Метод Save позволяет сохранять данные из интерфейса.

```
HRESULT Save(  
    DATA_TYPE dtType,  
    FORMAT fFormat  
    BSTR bstrDestURI  
);
```

**Параметры**

dtType [in] - тип сохраняемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
fFormat [in] - тип используемой кодировки при сохранении (DER, Base64). Допустимыми значениями являются значения FORMAT.  
bstrDestURI [in] - имя файла, куда необходимо сохранить данные.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Save позволяет сохранять данные из интерфейса.

```
object.Save(dtType, fFormat, bstrDestURI)
```

**Параметры**

dtType - тип сохраняемых данных. Допустимыми значениями являются значения DATA\_TYPE.  
fFormat - тип используемой кодировки при сохранении (DER, Base64). Допустимыми значениями являются значения FORMAT.  
bstrDestURI - имя файла, куда необходимо сохранить данные.

**IPKCS7MESSAGE::SIGN [C++]****PKCS7MESSAGE::SIGN [VBS]****[C++]**

Метод Sign позволяет подписать загруженные данные. Данные, необходимые для создания подписи, берутся из профиля.

Параметры профиля, необходимые для выполнения операции подписи: Comment - комментарии к подписи Detach - флаг формирования отдельной подписи HashAlg - идентификатор алгоритма хэширования, используемого для создании подписи IncludeSignatureTime - флаг включения времени создания подписи IncludeTimeStampToken - флаг включения в подпись штампа времени SignIncludeBase64Headers - флаг добавления Base64-заголовков в выходном подписанном файле SignDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции подписи

SignExitFormat - выходной формат подписанного сообщения TSPProfile или TSPProfileID - параметры, позволяющие установить профиль Службы Штamped Времени. Данные параметры необходимы, если IncludeTimeStampToken = True. Resource - значение идентификатора (URI) подписываемого файла.

Для выполнения операции подписи необходимо установить в профиль сертификат подписи. Установка сертификата в профиль производится с помощью метода SetCertificate.

```
HRESULT Sign(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

#### Примечание

Добавление параллельной подписи невозможно для файла размером более 100 МБ!

[VBS]

Метод Sign позволяет подписать загруженные данные. Данные, необходимые для создания подписи, берутся из профиля.

Параметры профиля, необходимые для выполнения операции подписи: Comment - комментарии к подписи Detach - флаг формирования отдельной подписи HashAlg - идентификатор алгоритма хэширования, используемого для создания подписи IncludeSignatureTime - флаг включения времени создания подписи IncludeTimeStampToken - флаг включения в подпись штампа времени SignIncludeBase64Headers - флаг добавления Base64-заголовков в выходном подписанном файле SignDeleteSourceFile - флаг удаления исходного файла после успешного выполнения операции подписи SignExitFormat - выходной формат подписанного сообщения TSPProfile или TSPProfileID - параметры, позволяющие установить профиль Службы Штamped Времени. Данные параметры необходимы, если IncludeTimeStampToken = True. Resource - значение идентификатора (URI) подписываемого файла.

Для выполнения операции подписи необходимо установить в профиль сертификат подписи. Установка сертификата в профиль производится с помощью метода SetCertificate.

```
object.Sign()
```

#### Примечание

Добавление параллельной подписи невозможно для файла размером более 100 МБ!

### IPKCS7MESSAGE::SPLITSIGNATURE [C++]

### PKCS7MESSAGE::SPLITSIGNATURE [VBS]

[C++]

Метод SplitSignature позволяет отделить подпись от исходных данных (сделать подпись отдельной).

```
HRESULT SplitSignature(  
);
```

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод SplitSignature позволяет отделить подпись от исходных данных (сделать подпись отдельной).

```
object.SplitSignature
```

### IPKCS7MESSAGE::GETCONTENTOID [C++]

### PKCS7MESSAGE::GETCONTENTOID [VBS]

[C++]

Метод GetContentOID позволяет получить OID контента из PKCS-7/9 сообщений. Метод не изменяет состояние COM-объекта, для этой цели следует использовать Load или Import.

```
HRESULT GetContentOID(  
    BSTR bstrSource,  
    VARIANT_BOOL vbSourceIsURI,  
    BSTR* pVal  
);
```

#### Параметры

bstrSource [in] – URI файла или строка исходных данных.  
vbSourceIsURI [in, optional, defaultvalue(-1)] – флаг для указания типа источника данных: TRUE – URI файла, FALSE – строка.  
pVal [out, retval] – возвращает OID контента исходных данных.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GetContentOID позволяет получить OID контента из PKCS-7/9 сообщений. Метод не изменяет состояние COM-объекта, для этой цели следует использовать Load или Import..

```
object.GetContentOID(bstrSource, vbSourceIsURI)
```

#### Параметры

bstrSource – URI файла или строка исходных данных.  
vbSourceIsURI – флаг для указания типа источника данных: TRUE – URI файла, FALSE – строка.

#### Возвращаемые значения

Возвращает OID контента исходных данных.

### IPKCS7MESSAGE::CONTENTTYPE [C++]

### PKCS7MESSAGE::CONTENTTYPE [VBS]

[C++]

Свойство ContentType позволяет получить описание типа загруженных в интерфейс данных.

```
HRESULT get_ContentType(  
    DATA_TYPE* pVal  
);
```

#### Параметры

pVal [out, retval] – указатель на тип данных. Допустимыми значениями являются значения DATA\_TYPE.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ContentType позволяет получить описание типа загруженных в интерфейс данных.

```
object.ContentType
```

#### Возвращаемое значение

Указатель на тип данных. Допустимыми значениями являются значения DATA\_TYPE.

### IPKCS7MESSAGE::ENCALG [C++]

**PKCS7MESSAGE::ENCALG [VBS]**

[C++]

Свойство EncAlg позволяет получить алгоритм шифрования загруженного сообщения. Свойство становится доступно после вызова Decrypt(), т.к. в Load() зашифрованное сообщение не прогружается.

```
HRESULT get_ShowProgress(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает строку с OID-ом алгоритма зашифрованного сообщения.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncAlg позволяет получить алгоритм шифрования загруженного сообщения. Свойство становится доступно после вызова Decrypt(), т.к. в Load() зашифрованное сообщение не прогружается.

```
object.EncAlg
```

**Возвращаемое значение**

Возвращает строку с OID-ом алгоритма зашифрованного сообщения.

**IPKCS7MESSAGE::PROFILE [C++]****PKCS7MESSAGE::PROFILE [VBS]**

[C++]

Свойство Profile позволяет получить/установить профиль с настройками.

```
HRESULT get_Profile(  
    IProfile** pVal  
);  
HRESULT put_Profile(  
    IProfile* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает интерфейс IProfile, описывающий текущий профиль.  
pVal [in] - указатель на интерфейс IProfile, описывающий текущий профиль.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет получить/установить профиль с настройками.

```
object.Profile [= pVal]
```

**Параметры**

pVal - указатель на интерфейс Profile, описывающий текущий профиль.

**IPKCS7MESSAGE::RECIPIENTS [C++]****PKCS7MESSAGE::RECIPIENTS [VBS]**

[C++]

Свойство Recipients позволяет получить/установить коллекцию сертификатов получателей шифрованного сообщения.

**Примечание**

Свойство put\_Recipients в текущей версии не имеет эффекта и оставлено для совместимости с предыдущими версиями. Свойство get\_Recipients используется только после того, как в интерфейс IPKCS7Message было загружено шифрованное сообщение.

```
HRESULT put_Recipients(  
    ICertificates* pVal  
);  
HRESULT get_Recipients(  
    ICertificates** pVal  
);
```

**Параметры**

pVal [in] - указатель на интерфейс ICertificates, позволяющий установить сертификаты получателей.  
pVal [out, retval] - указатель на интерфейс ICertificates, позволяющий получить доступ к сертификатам получателей.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Recipients позволяет получить/установить коллекцию сертификатов получателей шифрованного сообщения.

**Примечание**

Свойство put\_Recipients в текущей версии не имеет эффекта и оставлено для совместимости с предыдущими версиями. Свойство get\_Recipients используется только после того, как в интерфейс IPKCS7Message было загружено шифрованное сообщение.

```
object.Recipients [= pVal]
```

**Параметры**

pVal - указатель на интерфейс Certificates.

**IPKCS7MESSAGE::SIGNATURES [C++]****PKCS7MESSAGE::SIGNATURES [VBS]**

[C++]

Свойство Signatures позволяет получить коллекцию подписей загруженного ранее сообщения.

```
HRESULT get_Signatures(  
    ISignatures** pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на интерфейс ISignatures.



**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Signatures позволяет получить коллекцию подписей загруженного ранее сообщения.

`object.Signatures`

**Возвращаемое значение**

Возвращает указатель на интерфейс Signatures.

**IPKCS7MESSAGE::SIGNERS [C++]****PKCS7MESSAGE::SIGNERS [VBS]**

[C++]

Свойство Signers позволяет получить коллекцию сертификатов подписчиков.

```
HRESULT get_Signers(  
    ICertificates** pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на интерфейс ICertificates, позволяющий получить доступ к сертификатам подписчиков сообщения.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Signers позволяет получить коллекцию сертификатов подписчиков.

`object.Signers`

**Возвращаемое значение**

Возвращает указатель на интерфейс Certificates, позволяющий получить доступ к сертификатам подписчиков сообщения.

**IPKCS7MESSAGE::SHOWPROGRESS [C++]****PKCS7MESSAGE::SHOWPROGRESS [VBS]**

[C++]

Свойство ShowProgress позволяет запустить/остановить окно с отображением текущего статуса («процента») выполнения операции, а также получить текущее состояние (запущен или остановлен).

```
HRESULT get_ShowProgress(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_ShowProgress(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [in] - TRUE - запуск окна с отображением текущего статуса, FALSE - остановка.  
pVal [out, retval] - TRUE - окна с отображением текущего статуса запущено, FALSE - остановлено.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ShowProgress позволяет запустить/остановить окно с отображением текущего статуса («процента») выполнения операции, а также получить текущее состояние (запущен или остановлен).

object.ShowProgress

### Параметры

pVal - TRUE - запуск окна с отображением текущего статуса, FALSE - остановка.

### Возвращаемое значение

Возвращает флаг отображения она состояния выполнения операции.

## PKIPROFILE (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

### Свойства

Свойство	Описание
CAType	Свойство позволяет получить/установить тип Удостоверяющего Центра.
CertificateHoldDuration	Свойство позволяет получить/установить период приостановления сертификата
Comment	Свойство позволяет получить/установить комментарий к запросу
EmailMessage	Свойство позволяет получить/установить текст электронного сообщения
EmailRecipients	Свойство позволяет получить/установить адреса получателей электронного сообщения
EmailSubject	Свойство позволяет получить/установить тему электронного сообщения
IncludeBASE64Headers	Свойство позволяет получить/установить флаг добавления Base64-заголовков в выходном файле при сохранении запроса
OperationCertificate	Свойство позволяет получить/установить сертификат, над которым выполняется операция отзыва, приостановления, возобновления или обновления ключа
ProxyAddress	Свойство позволяет получить/установить адрес прокси-сервера, используемый при подключении к Удостоверяющему Центру
ProxyPassword	Свойство позволяет получить/установить пароль для аутентификации на прокси-сервере
ProxyUser	Свойство позволяет получить/установить имя пользователя для аутентификации на прокси-сервере
RequestFilename	Свойство позволяет получить/установить имя файла для сохранения запроса
RequestFilenameFormat	Свойство позволяет получить/установить формат файла для сохранения запроса
RequestTemplate	Свойство позволяет получить/установить шаблон запроса на сертификат
RequestType	Свойство позволяет получить/установить тип запроса на сертификат
RevocationReason	Свойство позволяет получить/установить причину отзыва сертификата при формировании запроса на отзыв
SaveRequest	Свойство позволяет получить/установить флаг сохранения запроса в файл
SendRequest	Свойство позволяет получить/установить флаг отправки запроса по сети
SendRequestByEmail	Свойство позволяет получить/установить флаг отправки запроса по электронной почте

ServiceProfile	Свойство позволяет получить/установить WSDL-адрес Центра Регистрации для отправки запроса по сети
ServiceURL	Свойство позволяет получить/установить WSDL-адрес Центра Регистрации для отправки запроса по сети
SignatureCertificatePIN	Свойство позволяет получить/установить пин-код доступа к ключевому контейнеру сертификата подписи запроса.
SignatureHashAlg	Свойство позволяет получить/установить строку с объектным идентификатором алгоритма хэширования для подписи запроса
SignCertificate	Свойство позволяет получить/установить сертификат подписи запроса
SilentLevel	Свойство позволяет получить/установить уровень интерактивности при создании запроса на сертификат
SSLCertificate	Свойство позволяет получить/установить сертификат аутентификации клиента для создания SSL соединения
UseProxy	Свойство позволяет получить/установить флаг использования прокси-сервера для установки соединения с Удостоверяющим Центром.

## Методы

Метод	Описание
CollectData	Метод позволяет запустить мастер для сбора данных, необходимых для создания PKI запроса.
LoadXMLTemplate	Метод позволяет загрузить данные в PKIProfile из XML шаблона

## IPKIPROFILE::COLLECTDATA [C++]

## PKIPROFILE::COLLECTDATA [VBS]

[C++]

Метод CollectData позволяет запустить мастер для сбора данных, необходимых для создания PKI запроса.

```
HRESULT CollectData(  
    VARIANT_BOOL* pVal  
);
```

### Параметры

pVal [out, retval] - индикатор завершения мастера. TRUE - завершен, FALSE - прерван.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CollectData позволяет запустить мастер для сбора данных, необходимых для создания PKI запроса.

```
object.CollectData()
```

### Возвращаемые значения

Индикатор завершения мастера. TRUE - завершен, FALSE - прерван.

## IPKIPROFILE::LOADXMLTEMPLATE [C++]

## PKIPROFILE::LOADXMLTEMPLATE [VBS]

[C++]

Метод LoadXMLTemplate позволяет загрузить данные в PKIProfile из XML шаблона. Шаблон задает набор атрибутов отличительного имени владельца сертификата и параметры ключа.

При создании шаблона важно помнить, что порядок следования атрибутов отличительного имени, установленный в шаблоне, определяет и порядок атрибутов в запросе на сертификат.

Также шаблон позволяет настраивать вид формы установки параметров пользователя в мастере создания запроса на сертификат: скрывать поля формы, устанавливать для атрибутов значения по умолчанию и устанавливать поля недоступными для изменений.

```
HRESULT LoadXMLTemplate(  
    BSTR bstrFilename  
);
```

### Параметры

bstrFilename [in] - XML файл с шаблоном PKI запроса.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

Пример шаблона запроса на сертификат

```
<?xml version="1.0" encoding="windows-1251" ?>  
<CertificateRequestTemplate>  
    <Subject>  
        <DirectoryName>  
            <RDNSequence>  
                <RDNEntry readonly="false" mandatory="true" hidden="true">  
                    <!--OID>2.5.4.3</OID-->  
                    <OID>CN</OID>  
                    <Name>Имя владельца</Name> <!-- Название поля -->  
                    <!--Value>Нечаев Андрей Евгеньевич</Value-->  
                <!-- Значение, устанавливаемое по умолчанию -->  
                </RDNEntry>  
                <RDNEntry hidden="true">  
                    <OID>2.5.4.11</OID>  
                    <Name>Подразделение</Name>  
                    <Value>ИТ</Value>  
                </RDNEntry>  
                <RDNEntry readonly="true">  
                    <OID>2.5.4.10</OID>  
                    <Name>Организация</Name>  
                    <Value>Цифровые технологии</Value>  
                </RDNEntry>  
                <RDNEntry>  
                    <OID>2.5.4.8</OID>  
                    <Name>Регион</Name>  
                    <Value></Value>  
                </RDNEntry>  
                <RDNEntry>  
                    <OID>2.5.4.7</OID>  
                    <Name>Город</Name>  
                    <Value>Moscow</Value>  
                </RDNEntry>  
                <RDNEntry>  
                    <OID>2.5.4.6</OID>  
                    <Name>Страна</Name>  
                    <Value>RU</Value>  
                    <Values><Value>EN</Value><Value>RU</Value></Values>  
                </RDNEntry>  
            </RDNEntry>  
        </DirectoryName>  
    </Subject>
```

```

        <OID>1.2.840.113549.1.9.1</OID>
        <Name>E-mail</Name>
        <Value>ane@trusted.ru</Value>
    </RDNEntry>
</RDNSequence>
</DirectoryName>
</Subject>
<Extensions>
    <Extension>
        <OID>2.5.29.15</OID>
        <Name>Использование ключа</Name>
        <Critical>True</Critical>
        <Value>
            <KeyUsage>
                <DigitalSignature />
                <NonRepudiation />
                <kuKeyEncipherment />
                <DataEncipherment />
                <KeyAgreement />
                <KeyCertSign />
                <CRLSign />
            </KeyUsage>
        </Value>
    </Extension>
    <Extension>
        <OID>2.5.29.37</OID>
        <Name>Назначение сертификата</Name>
        <Value>
            <ExtendedKeyUsage>
                <KeyPurposeId>1.3.6.1.5.5.7.3.4</KeyPurposeId>
                <KeyPurposeId>1.3.6.1.5.5.7.3.2</KeyPurposeId>
            </ExtendedKeyUsage>
        </Value>
    </Extension>
</Extensions>
</CertificateRequestTemplate>

```

[VBS]

Метод LoadXMLTemplate позволяет загрузить данные в PKIProfile из XML шаблона. Шаблон задает набор атрибутов отличительного имени владельца сертификата и параметры ключа.

При создании шаблона важно помнить, что порядок следования атрибутов отличительного имени, установленный в шаблоне, определяет и порядок атрибутов в запросе на сертификат.

Также шаблон позволяет настраивать вид формы установки параметров пользователя в мастере создания запроса на сертификат: скрывать поля формы, устанавливая для атрибутов значения по умолчанию и устанавливать поля недоступными для изменений.

```
object.LoadXMLTemplate(bstrFilename)
```

### Параметры

bstrFilename

XML файл с шаблоном PKI запроса.

Пример шаблона запроса на сертификат

```
<?xml version="1.0" encoding="windows-1251" ?>
```

```
<CertificateRequestTemplate>
```

```
    <Subject>
```

```
        <DirectoryName>
```

```
            <RDNSequence>
```

```
                <RDNEntry    readonly="false"    mandatory="true"    hid-
```

```

den="true">
    <!--OID>2.5.4.3</OID-->
    <OID>CN</OID>
    <Name>Имя владельца</Name> <!-- Название поля -->
    <!--Value>Нечаев Андрей Евгеньевич</Value--> <!--
Значение, устанавливаемое по умолчанию -->
    </RDNEntry>
    <RDNEntry hidden="true">
        <OID>2.5.4.11</OID>
        <Name>Подразделение</Name>
        <Value>ИТ</Value>
    </RDNEntry>
    <RDNEntry readonly="true">
        <OID>2.5.4.10</OID>
        <Name>Организация</Name>
        <Value>Цифровые технологии</Value>
    </RDNEntry>
    <RDNEntry>
        <OID>2.5.4.8</OID>
        <Name>Регион</Name>
        <Value></Value>
    </RDNEntry>
    <RDNEntry>
        <OID>2.5.4.7</OID>
        <Name>Город</Name>
        <Value>Moscow</Value>
    </RDNEntry>
    <RDNEntry>
        <OID>2.5.4.6</OID>
        <Name>Страна</Name>
        <Value>RU</Value>
        <Values><Value>EN</Value><Value>RU</Value></Values>
    </RDNEntry>
    <RDNEntry>
        <OID>1.2.840.113549.1.9.1</OID>
        <Name>E-mail</Name>
        <Value>ane@trusted.ru</Value>
    </RDNEntry>
    </RDNSequence>
    </DirectoryName>
</Subject>
<Extensions>
    <Extension>
        <OID>2.5.29.15</OID>
        <Name>Использование ключа</Name>
        <Critical>True</Critical>
        <Value>
            <KeyUsage>
                <DigitalSignature />
                <NonRepudiation />
                <kuKeyEncipherment />
                <DataEncipherment />
                <KeyAgreement />
                <KeyCertSign />
                <CRLSign />
            </KeyUsage>
        </Value>
    </Extension>
    <Extension>
        <OID>2.5.29.37</OID>
        <Name>Назначение сертификата</Name>
        <Value>
            <ExtendedKeyUsage>

```

```
<KeyPurposeId>1.3.6.1.5.5.7.3.4</KeyPurposeId>

<KeyPurposeId>1.3.6.1.5.5.7.3.2</KeyPurposeId>
    </ExtendedKeyUsage>
    </Value>
    </Extension>
</Extensions>

</CertificateRequestTemplate>
```

## IPKIPROFILE::CATYPE [C++]

## PKIPROFILE::CATYPE [VBS]

[C++]

Свойство CAType позволяет получить/установить тип Удостоверяющего Центра (Microsoft CA).

```
HRESULT put_CAType (
    CA_TYPE newVal
);
HRESULT get_CAType (
    CA_TYPE* pVal
);
```

### Параметры

newVal [in] - тип УЦ. Принимает значения из CA\_TYPE.

pVal [out, retval] - указатель на тип УЦ. Принимает значения из CA\_TYPE.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CAType позволяет получить/установить тип Удостоверяющего Центра (Microsoft CA).

object.CAType (=Value)

### Параметры

Value - тип УЦ. Принимает значения из CA\_TYPE.

### Возвращаемые значения

Возвращаемое значение - указатель на тип УЦ. Принимает значения из CA\_TYPE.

## IPKIPROFILE::CERTIFICATEHOLDURATION [C++]

## PKIPROFILE::CERTIFICATEHOLDURATION [VBS]

[C++]

Свойство CertificateHoldDuration позволяет получить/установить период приостановления сертификата.

```
HRESULT put_CertificateHoldDuration (
    BSTR newVal
);
HRESULT get_CertificateHoldDuration (
    BSTR* pVal
);
```

### Параметры

`newVal [in]` – продолжительность приостановления действия сертификата. Задается в виде строки вида: Y-M-W-D-H-m, где:  
Y число лет в интервале  
M число месяцев в интервале  
W число недель в интервале  
D число дней в интервале  
H число часов в интервале  
m число минут в интервале  
`pVal [out, retval]` – строка вида Y-M-W-D-H-m, содержащая продолжительность приостановления действия сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство `CertificateHoldDuration` позволяет получить/установить период приостановления сертификата.

```
object.CertificateHoldDuration(=Value)
```

### Параметры

`Value` – продолжительность приостановления действия сертификата. Задается в виде строки вида: Y-M-W-D-H-m, где:  
Y число лет в интервале  
M число месяцев в интервале  
W число недель в интервале  
D число дней в интервале  
H число часов в интервале  
m число минут в интервале

### Возвращаемые значения

Возвращаемое значение – строка вида Y-M-W-D-H-m, содержащая продолжительность приостановления действия сертификата.

## IPKIPROFILE::COMMENT [C++]

### PKIPROFILE::COMMENT [VBS]

[C++]

Свойство `Comment` позволяет получить/установить комментарий к запросу.

```
HRESULT put_Comment(  
    BSTR newVal  
);  
HRESULT get_Comment(  
    BSTR* pVal  
);
```

### Параметры

`newVal [in]` – строка комментария к запросу.  
`pVal [out, retval]` – указатель на строку комментария к запросу.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство `Comment` позволяет получить/установить комментарий к запросу.



```
object.Comment(=Value)
```

### Параметры

Value - строка комментария к запросу.

### Возвращаемые значения

Возвращаемое значение - указатель на строку комментария к запросу.

## IPKIPROFILE::EMAILMESSAGE [C++]

## PKIPROFILE::EMAILMESSAGE [VBS]

[C++]

Свойство EmailMessage позволяет получить/установить текст электронного сообщения.

```
HRESULT put_EmailMessage(  
    BSTR newVal  
);  
HRESULT get_EmailMessage(  
    BSTR* pVal  
);
```

### Параметры

newVal [in] - строка текста электронного сообщения.

pVal [out, retval] - указатель на строку текста электронного сообщения.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EmailMessage позволяет получить/установить текст электронного сообщения.

```
object.EmailMessage(=Value)
```

### Параметры

Value - строка текста электронного сообщения.

### Возвращаемые значения

Возвращаемое значение - указатель на строку текста электронного сообщения.

## IPKIPROFILE::EMAILRECIPIENT [C++]

## PKIPROFILE::EMAILRECIPIENT [VBS]

[C++]

Свойство EmailRecipient позволяет получить/установить адреса получателей электронного сообщения.

```
HRESULT put_EmailRecipient(  
    BSTR newVal  
);  
HRESULT get_EmailRecipient(  
    BSTR* pVal  
);
```

### Параметры

newVal [in] - строка с адресами получателей электронного сообщения.  
pVal [out, retval] - указатель на строку с адресами получателей электронного сообщения.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EmailRecipient позволяет получить/установить адреса получателей электронного сообщения.

```
object.EmailRecipient(=Value)
```

#### Параметры

Value - строка с адресами получателей электронного сообщения.

#### Возвращаемые значения

Возвращаемое значение - указатель на строку с адресами получателей электронного сообщения.

### IPKIPROFILE::EMAILSUBJECT [C++]

### PKIPROFILE::EMAILSUBJECT [VBS]

[C++]

Свойство EmailSubject позволяет получить/установить тему электронного сообщения.

```
HRESULT put_EmailSubject(  
    BSTR newVal  
);  
HRESULT get_EmailSubject(  
    BSTR* pVal  
);
```

#### Параметры

newVal [in] - строка с темой электронного сообщения.  
pVal [out, retval] - указатель на строку с темой электронного сообщения.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EmailSubject позволяет получить/установить тему электронного сообщения.

```
object.EmailSubject(=Value)
```

#### Параметры

Value - строка с темой электронного сообщения.

#### Возвращаемые значения

Возвращаемое значение - указатель на строку с темой электронного сообщения.

### IPKIPROFILE::INCLUDEBASE64HEADERS [C++]

### PKIPROFILE::INCLUDEBASE64HEADERS [VBS]

**[C++]**

Свойство `IncludeBASE64Headers` позволяет получить/установить флаг добавления Base64-заголовков в выходном файле при сохранении запроса.

```
HRESULT put_IncludeBASE64Headers (
    VARIANT_BOOL newVal
);
HRESULT get_IncludeBASE64Headers (
    VARIANT_BOOL* pVal
);
```

**Параметры**

`newVal [in]` - устанавливаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.  
`pVal [out, retval]` - возвращаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство `IncludeBASE64Headers` позволяет получить/установить флаг добавления Base64-заголовков в выходном файле при сохранении запроса.

```
object.IncludeBASE64Headers (=Value)
```

**Параметры**

`Value` - устанавливаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.

**Возвращаемые значения**

Возвращаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.

**IPKIPROFILE::OPERATIONCERTIFICATE [C++]****PKIPROFILE::OPERATIONCERTIFICATE [VBS]****[C++]**

Свойство `OperationCertificate` позволяет получить/установить сертификат, над которым выполняется операция отзыва, приостановления, возобновления или обновления ключа.

```
HRESULT put_OperationCertificate(
    ICertificate* newVal
);
HRESULT get_OperationCertificate(
    ICertificate** pVal
);
```

**Параметры**

`newVal [in]` - указатель на интерфейс ICertificate.  
`pVal [out, retval]` - указатель на указатель на интерфейс ICertificate.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство `OperationCertificate` позволяет получить/установить сертификат, над которым выполняется операция отзыва, приостановления, возобновления или обновления ключа.

```
object.OperationCertificate(=Value)
```

#### Параметры

`Value` – указатель на объект `Certificate`.

#### Возвращаемые значения

Возвращаемое значение – указатель на указатель на объект `Certificate`.

### IPKIPROFILE::PROXYADDRESS [C++]

### PKIPROFILE::PROXYADDRESS [VBS]

[C++]

Свойство `ProxyAddress` позволяет получить/установить адрес прокси-сервера, используемый при подключении к Удостоверяющему Центру.

```
HRESULT put_ProxyAddress(  
    BSTR newVal  
);  
HRESULT get_ProxyAddress(  
    BSTR* pVal  
);
```

#### Параметры

`newVal [in]` – строка с адресом прокси-сервера в формате "<протокол>://<сервер>[:порт]".  
`pVal [out, retval]` – указатель на строку с адресом прокси-сервера в формате "<протокол>://<сервер>[:порт]".

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `ProxyAddress` позволяет получить/установить адрес прокси-сервера, используемый при подключении к Удостоверяющему Центру.

```
object.ProxyAddress(=Value)
```

#### Параметры

`Value` – строка с адресом прокси-сервера в формате "<протокол>://<сервер>[:порт]".

#### Возвращаемые значения

Указатель на строку с адресом прокси-сервера в формате "<протокол>://<сервер>[:порт]".

### IPKIPROFILE::PROXYPASSWORD [C++]

### PKIPROFILE::PROXYPASSWORD [VBS]

[C++]

Свойство `ProxyPassword` позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
HRESULT put_ProxyPassword(  
    BSTR newVal  
);  
HRESULT get_ProxyPassword(  
    BSTR* pVal  
);
```

#### Параметры

newVal [in] - строка пароля для аутентификации на прокси-сервере.  
pVal  
[out, retval] - указатель на строку пароля для аутентификации на прокси-сервере.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
object.ProxyPassword(=Value)
```

#### Параметры

Value - строка пароля для аутентификации на прокси-сервере.

#### Возвращаемые значения

Указатель на строку пароля для аутентификации на прокси-сервере.

### IPKIPROFILE::PROXYUSER [C++]

### PKIPROFILE::PROXYUSER [VBS]

[C++]

Свойство ProxyUser позволяет получить/установить пользователя для аутентификации на прокси-сервере.

```
HRESULT put_ProxyUser(  
    BSTR newVal  
);  
HRESULT get_ProxyUser(  
    BSTR* pVal  
);
```

#### Параметры

newVal [in] - строка с именем пользователя для аутентификации на прокси-сервере.  
pVal [out, retval] - указатель на строку с именем пользователя для аутентификации на прокси-сервере.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyUser позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
object.ProxyUser(=Value)
```

**Параметры**

Value - строка с именем пользователя для аутентификации на прокси-сервере.

**Возвращаемые значения**

Указатель на строку с именем пользователя для аутентификации на прокси-сервере.

**IPKIPROFILE::REQUESTFILENAME [C++]****PKIPROFILE::REQUESTFILENAME [VBS]**

[C++]

Свойство RequestFilename позволяет получить/установить имя файла для сохранения запроса.

```
HRESULT put_RequestFilename(  
    BSTR newVal  
);  
HRESULT get_RequestFilename(  
    BSTR* pVal  
);
```

**Параметры**

newVal [in] - строка с именем файла для сохранения запроса.

pVal [out, retval] - указатель на строку с именем файла для сохранения запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RequestFilename позволяет получить/установить имя файла для сохранения запроса.

```
object.RequestFilename(=Value)
```

**Параметры**

Value - строка с именем файла для сохранения запроса.

**Возвращаемые значения**

Указатель на строку с именем файла для сохранения запроса.

**IPKIPROFILE::REQUESTFILENAMEFORMAT [C++]****PKIPROFILE::REQUESTFILENAMEFORMAT [VBS]**

[C++]

Свойство RequestFilenameFormat позволяет получить/установить формат файла для сохранения запроса.

```
HRESULT put_RequestFilenameFormat(  
    FORMAT newVal  
);  
HRESULT get_RequestFilenameFormat(  
    FORMAT* pVal  
);
```

**Параметры**

newVal [in] - формат файла для сохранения запроса. Принимает значения из FORMAT.  
pVal [out, retval] - указатель на формат файла для сохранения запроса. Принимает значения из FORMAT.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RequestFilenameFormat позволяет получить/установить формат файла для сохранения запроса.

```
object.RequestFilenameFormat(=Value)
```

**Параметры**

Value - формат файла для сохранения запроса. Принимает значения из FORMAT.

**Возвращаемые значения**

Указатель на формат файла для сохранения запроса. Принимает значения из FORMAT.

**IPKIPROFILE::REQUESTTEMPLATE [C++]****PKIPROFILE::REQUESTTEMPLATE [VBS]**

[C++]

Свойство RequestTemplate позволяет получить/установить шаблон запроса на сертификат.

```
HRESULT put_RequestTemplate(  
    IRequestTemplate* newVal  
);  
HRESULT get_RequestTemplate(  
    IRequestTemplate** pVal  
);
```

**Параметры**

newVal [in] - указатель на интерфейс IRequestTemplate.  
pVal [out, retval] - указатель на указатель на интерфейс IRequestTemplate.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RequestTemplate позволяет получить/установить шаблон запроса на сертификат.

```
object.RequestTemplate(=Value)
```

**Параметры**

Value - указатель на объект RequestTemplate.

**Возвращаемые значения**

Указатель на указатель на объект RequestTemplate.

**IPKIPROFILE::REQUESTTYPE [C++]****PKIPROFILE::REQUESTTYPE [VBS]**

**[C++]**

Свойство RequestType позволяет получить/установить тип запроса на сертификат.

```
HRESULT put_RequestType(  
    REQUEST_TYPE newVal  
);  
HRESULT get_RequestType(  
    REQUEST_TYPE* pVal  
);
```

**Параметры**

newVal [in] - тип запроса. Принимает значения из REQUEST\_TYPE.  
pVal [out, retval] - указатель на тип запроса. Принимает значения из REQUEST\_TYPE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство RequestType позволяет получить/установить тип запроса на сертификат.

```
object.RequestType(=Value)
```

**Параметры**

Value - тип запроса. Принимает значения из REQUEST\_TYPE.

**Возвращаемые значения**

Указатель на тип запроса. Принимает значения из REQUEST\_TYPE.

**IPKIPROFILE::SAVEREQUEST [C++]****PKIPROFILE::SAVEREQUEST [VBS]****[C++]**

Свойство SaveRequest позволяет получить/установить флаг сохранения запроса в файл.

```
HRESULT put_SaveRequest(  
    VARIANT_BOOL newVal  
);  
HRESULT get_SaveRequest(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение TRUE или FALSE.  
pVal [out, retval] - принимает значения TRUE или FALSE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство SaveRequest позволяет получить/установить флаг сохранения запроса в файл.

```
object.SaveRequest(=Value)
```

**Параметры**

Value - принимает значения TRUE или FALSE.



**Возвращаемые значения**

Возвращает значения TRUE или FALSE.

**IPKIPROFILE::SAVEREQUEST [C++]****PKIPROFILE::SAVEREQUEST [VBS]**

[C++]

Свойство SaveRequest позволяет получить/установить флаг сохранения запроса в файл.

```
HRESULT put_SaveRequest(  
    VARIANT_BOOL newVal  
);  
HRESULT get_SaveRequest(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение TRUE или FALSE.  
pVal [out, retval] - принимает значения TRUE или FALSE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SaveRequest позволяет получить/установить флаг сохранения запроса в файл.

```
object.SaveRequest(=Value)
```

**Параметры**

Value - принимает значения TRUE или FALSE.

**Возвращаемые значения**

Возвращает значения TRUE или FALSE.

**IPKIPROFILE::SENDREQUEST [C++]****PKIPROFILE::SENDREQUEST [VBS]**

[C++]

Свойство SendRequest позволяет получить/установить флаг отправки запроса по сети.

```
HRESULT put_SendRequest(  
    VARIANT_BOOL newVal  
);  
HRESULT get_SendRequest(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение TRUE или FALSE.  
pVal [out, retval] - принимает значения TRUE или FALSE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SendRequest позволяет получить/установить флаг отправки запроса по сети.

```
object.SendRequest(=Value)
```

#### Параметры

Value - принимает значения TRUE или FALSE.

#### Возвращаемые значения

Возвращает значения TRUE или FALSE.

### IPKIPROFILE::SENDREQUESTBYEMAIL [C++]

### PKIPROFILE::SENDREQUESTBYEMAIL [VBS]

[C++]

Свойство SendRequestByEmail позволяет получить/установить флаг отправки запроса по электронной почте.

```
HRESULT put_SendRequestByEmail(  
    VARIANT_BOOL newVal  
);  
HRESULT get_SendRequestByEmail(  
    VARIANT_BOOL* pVal  
);
```

#### Параметры

newVal [in] - устанавливаемое значение TRUE или FALSE.  
pVal [out, retval] - принимает значения TRUE или FALSE.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SendRequestByEmail позволяет получить/установить флаг отправки запроса по электронной почте.

```
object.SendRequestByEmail(=Value)
```

#### Параметры

Value - принимает значения TRUE или FALSE.

#### Возвращаемые значения

Возвращает значения TRUE или FALSE.

### IPKIPROFILE::SERVICEPROFILE [C++]

### PKIPROFILE::SERVICEPROFILE [VBS]

[C++]

Свойство ServiceProfile позволяет получить/установить параметры доступа к Веб-сервису.

```
HRESULT put_ServiceProfile(  
    IServiceProfile* newVal  
);  
HRESULT get_ServiceProfile(  
    IServiceProfile** pVal  
);
```

```
IServiceProfile** pVal  
);
```

### Параметры

newVal [in] - указатель на интерфейс IServiceProfile.

pVal [out, retval] - указатель на указатель на интерфейс IServiceProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ServiceProfile позволяет получить/установить параметры доступа к Веб-сервису.

```
object.ServiceProfile(=Value)
```

### Параметры

Объект ServiceProfile.

### Возвращаемые значения

Объект ServiceProfile.

## IPKIPROFILE::SERVICEURL [C++]

## PKIPROFILE::SERVICEURL [VBS]

[C++]

Свойство ServiceURL позволяет получить/установить WSDL-адрес Центра Регистрации ПАК "КриптоПро УЦ" для отправки запроса по сети. Пример адреса "https://localhost/RA/RA.wsdl".

```
HRESULT put_ServiceURL(  
    BSTR newVal  
);  
HRESULT get_ServiceURL(  
    BSTR* pVal  
);
```

### Параметры

newVal [in] - устанавливаемое значение - строка с WSDL-адресом Центра Регистрации ПАК «КриптоПро УЦ».

pVal [out, retval] - возвращаемое значение - указатель на строку с WSDL-адресом Центра Регистрации ПАК «КриптоПро УЦ».

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ServiceURL позволяет получить/установить WSDL-адрес Центра Регистрации для отправки запроса по сети. Пример адреса «https://localhost/RA/RA.wsdl».

```
object.ServiceURL(=Value)
```

### Параметры

Value - устанавливаемое значение - строка с WSDL-адресом Центра Регистрации.

### Возвращаемые значения

Возвращаемое значение - указатель на строку с WSDL-адресом Центра Регистрации ПАК «КриптоПро УЦ».

**IPKIPROFILE::SIGNATURECERTIFICATEPIN [C++]****PKIPROFILE::SIGNATURECERTIFICATEPIN [VBS]**

[C++]

Свойство SignatureCertificatePIN позволяет получить/установить пин-код доступа к ключевому контейнеру сертификата подписи запроса.

```
HRESULT put_SignatureCertificatePIN(  
    BSTR newVal  
);  
HRESULT get_SignatureCertificatePIN(  
    BSTR* pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение - строка с пин-кодом доступа к ключевому контейнеру сертификата подписи запроса.

pVal [out, retval] - возвращаемое значение - указатель на строку с пин-кодом доступа к ключевому контейнеру сертификата подписи запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureCertificatePIN позволяет получить/установить пин-код доступа к ключевому контейнеру сертификата подписи запроса.

```
object.SignatureCertificatePIN(=Value)
```

**Параметры**

Value - устанавливаемое значение - строка с пин-кодом доступа к ключевому контейнеру сертификата подписи запроса.

**Возвращаемые значения**

Возвращаемое значение - указатель на строку с пин-кодом доступа к ключевому контейнеру сертификата подписи запроса.

**IPKIPROFILE::SIGNATUREHASHALG [C++]****PKIPROFILE::SIGNATUREHASHALG [VBS]**

[C++]

Свойство SignatureHashAlg позволяет получить/установить строку с объектным идентификатором алгоритма хэширования для подписи запроса.

```
HRESULT put_SignatureHashAlg(  
    BSTR newVal  
);  
HRESULT get_SignatureHashAlg(  
    BSTR* pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение - строка с объектным идентификатором алгоритма хэширования для подписи запроса.

pVal [out, retval] - возвращаемое значение - указатель на строку с объектным идентификатором алгоритма хэширования для подписи запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureHashAlg позволяет получить/установить строку с объектным идентификатором алгоритма хэширования для подписи запроса.

```
object.SignatureHashAlg(=Value)
```

**Параметры**

Value - устанавливаемое значение - строка с объектным идентификатором алгоритма хэширования для подписи запроса.

**Возвращаемые значения**

Возвращаемое значение - указатель на строку с объектным идентификатором алгоритма хэширования для подписи запроса

**IPKIPROFILE::SIGNCERTIFICATE [C++]****PKIPROFILE::SIGNCERTIFICATE [VBS]**

[C++]

Свойство SignCertificate позволяет получить/установить сертификат подписи запроса.

```
HRESULT put_SignCertificate(  
    ICertificate* newVal  
);  
HRESULT get_SignCertificate(  
    ICertificate** pVal  
);
```

**Параметры**

newVal [in] - устанавливаемое значение - указатель на интерфейс ICertificate.  
pVal [out, retval] - возвращаемое значение - указатель на указатель на интерфейс ICertificate.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignCertificate позволяет получить/установить сертификат подписи запроса.

```
object.SignCertificate(=Value)
```

**Параметры**

Value - устанавливаемое значение - указатель на объект ICertificate.

**Возвращаемые значения**

Возвращаемое значение - указатель на указатель на объект ICertificate.

**IPKIPROFILE::SILENTLEVEL [C++]****PKIPROFILE::SILENTLEVEL [VBS]**

[C++]

Свойство `SilentLevel` позволяет получить/установить уровень интерактивности при создании запроса на сертификат.

```
HRESULT put_SilentLevel (
    ICertificate* newVal
);
HRESULT get_SilentLevel (
    ICertificate** pVal
);
```

#### Параметры

`newVal [in]` - устанавливаемое значение - уровень интерактивности. Допустимыми значениями являются значения `SILENT_LEVEL`  
`pVal [out, retval]` - возвращаемое значение - указатель на уровень интерактивности. Принимает значения из `SILENT_LEVEL`.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SilentLevel` позволяет получить/установить уровень интерактивности при создании запроса на сертификат.

```
object.SilentLevel (=Value)
```

#### Параметры

`Value` - уровень интерактивности. Допустимыми значениями являются значения `SILENT_LEVEL`

#### Возвращаемые значения

Возвращаемое значение - указатель на уровень интерактивности. Принимает значения из `SILENT_LEVEL`.

### IPKIPROFILE::SSLCERTIFICATE [C++]

### PKIPROFILE::SSLCERTIFICATE [VBS]

[C++]

Свойство `SSLCertificate` позволяет получить/установить сертификат аутентификации клиента для создания SSL соединения.

```
HRESULT put_SSLCertificate(
    ICertificate* newVal
);
HRESULT get_SSLCertificate(
    ICertificate** pVal
);
```

#### Параметры

`newVal [in]` - устанавливаемое значение - указатель на интерфейс `ICertificate`.  
`pVal [out, retval]` - возвращаемое значение - указатель на указатель на интерфейс `ICertificate`.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SSLCertificate` позволяет получить/установить сертификат аутентификации клиента для создания SSL соединения.

```
object.SSLCertificate(=Value)
```

#### Параметры

`Value` - устанавливаемое значение - указатель на объект `ICertificate`.

#### Возвращаемые значения

Возвращаемое значение - указатель на указатель на объект `ICertificate`.

### IPKIPROFILE::UseProxy [C++]

### PKIPROFILE::UseProxy [VBS]

[C++]

Свойство `UseProxy` позволяет получить/установить флаг использования прокси-сервера для установки соединения с Удостоверяющим Центром.

```
HRESULT put_UseProxy(  
    ICertificate* newVal  
);  
HRESULT get_UseProxy(  
    ICertificate** pVal  
);
```

#### Параметры

`newVal [in]` - принимает `TRUE` или `FALSE`. `TRUE` - использовать прокси-сервер, `FALSE` - не использовать.

`pVal [out, retval]` - возвращает `TRUE` или `FALSE`. `TRUE` - использовать прокси-сервер, `FALSE` - не использовать.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `UseProxy` позволяет получить/установить флаг использования прокси-сервера для установки соединения с Удостоверяющим Центром.

```
object.UseProxy(=Value)
```

#### Параметры

`Value`

Принимает `TRUE` или `FALSE`. `TRUE` - использовать прокси-сервер, `FALSE` - не использовать.

#### Возвращаемые значения

Возвращает `TRUE` или `FALSE`. `TRUE` - использовать прокси-сервер, `FALSE` - не использовать.

### PKIREQUEST (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

#### Свойства

Свойство	Описание
<code>CertificateTemplate</code>	Свойство позволяет получить шаблон запроса на сертификат, полученного после генерации запроса на сертификат, обновление сертификата или ключа сертификата

GenTime	Свойство позволяет получить дату и время генерации запроса
ID	Свойство позволяет получить уникальный идентификатор запроса
PKIProfile	Свойство позволяет получить/установить PKIProfile
RequestID	Свойство позволяет заполнить номер запроса в УЦ
Status	Свойство позволяет получить/установить общее имя владельца сертификата

## Методы

Метод	Описание
Display	Метод позволяет открыть форму просмотра PKI запроса
Export	Метод позволяет произвести экспорт PKI запроса в строку
Generate	Метод позволяет создать PKI запрос
Import	Метод позволяет импортировать запрос из строки
IsEqual	Метод позволяет сравнивать пару запросов
Load	Метод позволяет загрузить данные запроса с указанного идентификатора ресурса (URI)
RefreshStatus	Метод позволяет обновить статус PKI запроса
RunSendingPKIRequestWizard	Метод позволяет запустить мастер отправки запроса в УЦ
Save	Метод позволяет сохранить запрос по указанному идентификатору ресурса (URI).
Send	Метод позволяет отправить PKI запрос на обработку в УЦ
SendEmail	Метод позволяет отправить PKI запрос по электронной почте

## IPKIREQUEST::DISPLAY [C++]

## PKIREQUEST::DISPLAY [VBS]

[C++]

Метод Display позволяет открыть форму просмотра свойств PKI запроса.

```
HRESULT Display(  
    );
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет открыть форму просмотра свойств PKI запроса.

```
object.Display()
```

## IPKIREQUEST::EXPORT [C++]

## PKIREQUEST::EXPORT [VBS]

[C++]

Метод Export позволяет произвести экспорт PKI запроса в строку.

```
HRESULT Export(  
    FORMAT fFormat,  
    BSTR* pbstrExportedRequest,  
    );
```



### Параметры

fFormat [in, defaultvalue(BASE64\_TYPE)] - формат загружаемого PKI запроса. Принимает значения из FORMAT.  
pbstrExportedRequest [out, retval] - строка с PKI запросом.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет произвести экспорт PKI запроса в строку.

```
object.Export(fFormat)
```

### Параметры

fFormat [in, optional, defaultvalue(0)] - формат загружаемого PKI запроса. Принимает значения из FORMAT.

### Возвращаемое значение

Строка с PKI запросом.

## IPKIREQUEST::GENERATE [C++]

## PKIREQUEST::GENERATE [VBS]

[C++]

Метод Generate позволяет создать PKI запрос.

```
HRESULT Generate(  
    REQUEST_GENERATION_RESULT* pResult,  
);
```

### Параметры

pResult [out, retval] - результат генерации PKI запроса. Принимает значения из REQUEST\_GENERATION\_RESULT

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Generate позволяет создать PKI запрос.

```
object.Generate()
```

### Возвращаемое значение

Результат генерации PKI запроса. Принимает значения из REQUEST\_GENERATION\_RESULT.

## IPKIREQUEST::IMPORT [C++]

## PKIREQUEST::IMPORT [VBS]

[C++]

Метод Import позволяет импортировать запрос из строки.

```
HRESULT Import(  
    BSTR bstrReq,
```

```
);
```

### Параметры

bstrReq [in] – строка, содержащая PKI запрос.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать запрос из строки.

```
object.Import(bstrReq)
```

### Параметры

bstrReq – строка, содержащая PKI запрос.

## IPKIREQUEST::IsEqual [C++]

## PKIREQUEST::IsEqual [VBS]

[C++]

Метод IsEqual позволяет сравнивать пару запросов.

```
HRESULT IsEqual(  
    IPKIRequest* pIPKIRequest,  
    VARIANT_BOOL* pVal,  
);
```

### Параметры

pIPKIRequest [in] – запрос, который необходимо сравнить с запросом, загруженным ранее в данный объект.  
pVal [out, retval] – результат сравнения.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод IsEqual позволяет сравнивать пару запросов.

```
object.IsEqual(pIPKIRequest)
```

### Параметры

pIPKIRequest – строка, содержащая PKI запрос.

### Возвращаемые значения

Результат сравнения.

## IPKIREQUEST::Load [C++]

## PKIREQUEST::Load [VBS]

[C++]

Метод Load позволяет загрузить данные запроса с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

```
HRESULT Load (
```

```
BSTR bstrFilename  
);
```

### Параметры

bstrFilename [in] – URI (имя файла), откуда необходимо загрузить запрос.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить данные запроса с указанного идентификатора ресурса (URI). Параметром может являться строка вида file://.

```
object.Load (bstrFilename)
```

### Параметры

bstrFilename  
URI (имя файла), откуда необходимо загрузить запрос.

## IPKIREQUEST::REFRESHSTATUS [C++]

## PKIREQUEST::REFRESHSTATUS [VBS]

[C++]

Метод RefreshStatus позволяет обновить статус PKI запроса.

```
HRESULT RefreshStatus(  
    REFRESH_STATUS_TYPE enmRST  
    VARIANT* pvResult  
);
```

### Параметры

enmRST [in] – тип проверки статуса. Принимает значения из REFRESH\_STATUS\_TYPE.  
pvResult [out, retval] – результат обновления статуса запроса. Содержит ICertificate, если по запросу был выпущен сертификат, иначе – пустую ссылку.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RefreshStatus позволяет обновить статус PKI запроса.

```
object.RefreshStatus(enmRST)
```

### Параметры

enmRST – тип проверки статуса. Принимает значения из REFRESH\_STATUS\_TYPE.

### Возвращаемые значения

Результат обновления статуса запроса. Содержит Certificate, если по запросу был выпущен сертификат, иначе – "Nothing".

## IPKIREQUEST::RUNSENDINGPKIREQUESTWIZARD [C++]

## PKIREQUEST::RUNSENDINGPKIREQUESTWIZARD [VBS]

[C++]

Метод RunSendingPKIRequestWizard позволяет запустить мастер отправки запроса в УЦ.

```
HRESULT RunSendingPKIRequestWizard(  
    IPKIRequests* pVal  
    IPKIRequests** ppVal  
);
```

#### Параметры

pVal [in] - коллекция запросов, которая будет отображаться первоначально на странице выбора запросов. Может быть пустая.

ppVal [out, retval] - коллекция запросов, которая выбрана в мастере для отправки.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RunSendingPKIRequestWizard позволяет запустить мастер отправки запроса в УЦ.

```
object.RunSendingPKIRequestWizard(pVal)
```

#### Параметры

pVal - коллекция запросов, которая будет отображаться первоначально на странице выбора запросов. Может быть пустая.

#### Возвращаемые значения

Результат - коллекция запросов, которая выбрана в мастере для отправки.

### IPKIREQUEST::SAVE [C++]

### PKIREQUEST::SAVE [VBS]

[C++]

Метод Save позволяет сохранить запрос по указанному идентификатору ресурса (URI).

```
HRESULT Save(  
    BSTR bstrFilename  
    FORMAT fFormat  
);
```

#### Параметры

bstrFilename [in] - URI (имя файла), куда необходимо сохранить сертификат.  
fFormat [in, defaultvalue(DER\_TYPE)] - указывает формат сохранения данных сертификата (DER или Base64). Допустимыми значениями являются значения FORMAT.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить запрос по указанному идентификатору ресурса (URI).

```
object.Save(bstrFilename, fFormat)
```

#### Параметры

bstrFilename - URI (имя файла), куда необходимо сохранить сертификат.

fFormat - указывает формат сохранения данных сертификата (DER или Base64). Допустимыми значениями являются значения FORMAT. По умолчанию используется значение DER\_TYPE.

## IPKIREQUEST::SEND [C++]

## PKIREQUEST::SEND [VBS]

[C++]

Метод Send позволяет отправить PKI запрос на обработку в УЦ.

```
HRESULT Send(  
    VARIANT* pResult  
);
```

### Параметры

pResult [out, retval] - результат отправки PKI запроса. Содержит ICertificate, если по запросу был выпущен сертификат, иначе - идентификатор запроса.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Send позволяет отправить PKI запрос на обработку в УЦ.

```
object.Send()
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## IPKIREQUEST::SENDEMAIL [C++]

## PKIREQUEST::SENDEMAIL [VBS]

[C++]

Метод SendEmail позволяет отправить PKI запрос по электронной почте.

```
HRESULT SendEmail(  
    REQUEST_SENDING_RESULT* pResult  
);
```

### Параметры

pResult [out, retval] Результат отправки PKI запроса по электронной почте. Принимает значения из REQUEST\_SENDING\_RESULT.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод SendEmail позволяет отправить PKI запрос на обработку в УЦ.

```
object.SendEmail()
```

### Возвращаемые значения

Результат отправки PKI запроса по электронной почте. Принимает значения из REQUEST\_SENDING\_RESULT

## IPKIREQUEST::CERTIFICATETEMPLATE [C++]

## PKIREQUEST::CERTIFICATETEMPLATE [VBS]

[C++]

Свойство CertificateTemplate позволяет получить шаблон запроса на сертификат, полученного после генерации запроса на сертификат, обновление сертификата или ключа сертификата.

```
HRESULT CertificateTemplate(  
    ICertificate** pVal  
);
```

### Параметры

pVal  
[out, retval] Указатель на указатель на интерфейс ICertificate, в который помещен шаблон сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CertificateTemplate позволяет получить шаблон запроса на сертификат, полученного после генерации запроса на сертификат, обновление сертификата или ключа сертификата.

```
object.CertificateTemplate()
```

### Возвращаемое значение

Указатель на указатель на объект Certificate, в который помещен шаблон сертификата.

## IPKIREQUEST::GENTIME [C++]

## PKIREQUEST::GENTIME [VBS]

[C++]

Свойство GenTime позволяет получить/установить дату и время генерации запроса.

```
HRESULT put_GenTime(  
    DATE pVal  
);  
HRESULT get_GenTime(  
    DATE* pVal  
);
```

### Параметры

pVal [in] - дата и время генерации запроса.  
pVal [out, retval] - дата и время генерации запроса.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство GenTime позволяет получить/установить дату и время генерации запроса.

```
object.GenTime[=Value]
```

### Параметры

Дата и время генерации запроса.

### Возвращаемое значение

Дата и время генерации запроса.

## IPKIREQUEST::ID [C++]

## PKIREQUEST::ID [VBS]

[C++]

Свойство ID позволяет получить уникальный идентификатор запроса.

```
HRESULT ID(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - уникальный идентификатор запроса.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ID позволяет получить уникальный идентификатор запроса.

```
object.ID()
```

### Возвращаемое значение

Уникальный идентификатор запроса.

## IPKIREQUEST::PKIPROFILE [C++]

## PKIREQUEST::PKIPROFILE [VBS]

[C++]

Свойство PKIProfile позволяет получить/установить PKIProfile.

```
HRESULT put_PKIProfile(  
    IPKIProfile** newVal  
);  
HRESULT get_PKIProfile(  
    IPKIProfile* pVal  
);
```

### Параметры

newVal [in] - указатель на указатель на интерфейс IPKIProfile.  
pVal [out, retval] - указатель на интерфейс IPKIProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PKIProfile позволяет получить уникальный идентификатор запроса.

```
object.PKIProfile(=Value)
```

#### Параметры

newVal [in] - указатель на указатель на интерфейс IPKIProfile.

#### Возвращаемое значение

Указатель на интерфейс IPKIProfile.

### IPKIREQUEST::REQUESTID [C++]

### PKIREQUEST::REQUESTID [VBS]

[C++]

Свойство RequestID позволяет получить номер данного запроса в УЦ.

```
HRESULT RequestID(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] - строка с номером запроса в УЦ.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RequestID позволяет получить номер данного запроса в УЦ.

```
object.RequestID()
```

#### Возвращаемое значение

Строка с номером запроса в УЦ.

### IPKIREQUEST::STATUS [C++]

### PKIREQUEST::STATUS [VBS]

[C++]

Свойство Status позволяет получить/установить статус запроса.

```
HRESULT put_Status(  
    REQUEST_STATUS newVal  
);  
HRESULT get_Status(  
    REQUEST_STATUS* pVal  
);
```

#### Параметры

newVal [in] - статус запроса. Принимает значения из REQUEST\_STATUS.

pVal [out, retval] - указатель на значение статуса запроса. Принимает значения из REQUEST\_STATUS.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.



[VBS]

Свойство Status позволяет получить/установить статус PKI запроса.

```
object.Status (=Value)
```

#### Параметры

Value – статус запроса. Принимает значения из REQUEST\_STATUS.

#### Возвращаемое значение

Указатель на значение статуса запроса. Принимает значения из REQUEST\_STATUS.

### IPKIREQUESTS (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

#### Методы

Метод	Описание
Add	Метод позволяет добавлять PKI запрос в коллекцию
Remove	Метод позволяет удалять PKI запрос из коллекции
Clear	Метод позволяет очистить коллекцию объектов типа PKIRequest

#### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество объектов типа PKIRequest в коллекции
Item	Свойство позволяет получить указатель на объект типа PKIRequest

### IPKIREQUESTS::ADD [C++]

### PKIREQUESTS::ADD [VBS]

[C++]

Метод Add позволяет добавлять PKI запросы в коллекцию.

```
HRESULT Add(  
    IPKIRequest* addVal  
);
```

#### Параметры

addVal [in] – указатель на интерфейс типа IPKIRequest.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

#### Примечания

Вызов метода возможен после того, как получен интерфейс IPKIRequest, например из коллекции PKI запросов.

[VBS]

Метод Add позволяет добавлять PKI запросы в коллекцию.

```
object.Add(addVal)
```

#### Параметры

addVal – указатель на объект типа PKIRequest.

## Примечания

Вызов метода возможен после того, как получен объект PKIRequest, например из коллекции PKI запросов.

## IPKIREQUESTS::REMOVE [C++]

## PKIREQUESTS::REMOVE [VBS]

[C++]

Метод Remove позволяет удалять PKI запросы из коллекции.

```
HRESULT Remove(  
    long lIndex  
);
```

## Параметры

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1]

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалять PKI запросы из коллекции.

```
object.Remove(lIndex)
```

## Параметры

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1]

## IPKIREQUESTS::CLEAR [C++]

## PKIREQUESTS::CLEAR [VBS]

[C++]

Метод Clear позволяет очистить коллекцию PKI запросов.

```
HRESULT Clear(  
);
```

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clear позволяет очистить коллекцию PKI запросов.

```
object.Clear()
```

## IPKIREQUESTS::COUNT [C++]

## PKIREQUESTS::COUNT [VBS]

[C++]

Свойство Count позволяет получить количество элементов коллекции PKI запросов.

```
HRESULT get_Count(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - количество PKI запросов в коллекции.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество элементов коллекции PKI запросов.

object.Count

#### Возвращаемое значение

Количество PKI запросов в коллекции.

### IPKIREQUESTS::ITEM [C++]

### PKIREQUESTS::ITEM [VBS]

[C++]

Свойство Item позволяет получить указатель на интерфейс типа IPKIRequest.

```
HRESULT get_Item(  
    long lIndex,  
    ICertificate** pVal  
);
```

#### Параметры

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1]  
pVal [out, retval] - объект-приемник.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить указатель на объект типа PKIRequest.

object.Item ( lIndex)

#### Параметры

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1]

#### Возвращаемое значение

Объект-приемник.

### PKIREQUESTSTORE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)

#### Методы

Метод	Описание
Close	Метод позволяет закрыть открытое хранилище PKI запросов
Open	Метод позволяет открыть хранилище PKI запросов
Save	Метод позволяет сохранить в хранилище текущую коллекцию PKI запросов. Старые

	данные, находящиеся в хранилище, удаляются.
--	---------------------------------------------

**Свойства**

Свойство	Описание
Store	Свойство позволяет получить/установить коллекцию PKI запросов, содержащуюся в хранилище.

**IPKIREQUESTSTORE::CLOSE [C++]****PKIREQUESTSTORE::CLOSE [VBS]**

[C++]

Метод Close позволяет закрыть хранилище PKI запросов.

```
HRESULT Close(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Close позволяет закрыть хранилище PKI запросов.

```
object.Close()
```

**IPKIREQUESTSTORE::OPEN [C++]****PKIREQUESTSTORE::OPEN [VBS]**

[C++]

Метод Open позволяет открыть хранилище PKI запросов.

```
HRESULT Open(  
    PROFILESTORETYPE stStoreType,  
    BSTR bstrPuth  
);
```

**Параметры**

stStoreType [in] – тип хранилища PKI запросов. Допустимыми значениями являются значения PROFILESTORETYPE.  
bstrPuth [in, optional, defaultvalue("")] – зарезервировано для указания имени xml-файла с коллекцией PKI запросов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет открыть хранилище PKI запросов.

```
object.Open(stStoreType, bstrURI)
```

**Параметры**

stStoreType – тип хранилища PKI запросов. Допустимыми значениями являются значения PKIRequestStoreTYPE.

bstrURI - зарезервировано для указания имени xml-файла с коллекцией PKI запросов.

## IPKIREQUESTSTORE::SAVE [C++]

## PKIREQUESTSTORE::SAVE [VBS]

[C++]

Метод Save позволяет сохранить в хранилище текущую коллекцию PKI запросов. Старые данные, находящиеся в хранилище, удаляются.

```
HRESULT Save(  
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить в хранилище текущую коллекцию PKI запросов. Старые данные, находящиеся в хранилище, удаляются.

```
object.Save()
```

## IPKIREQUESTSTORE::STORE [C++]

## PKIREQUESTSTORE::STORE [VBS]

[C++]

Свойство Store позволяет получить/установить коллекцию PKI запросов, содержащуюся в хранилище.

```
HRESULT get_Store(  
    IPKIRequests** pVal  
);  
HRESULT put_Store(  
    IPKIRequests* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на коллекцию PKI запросов.

pVal [in] - указатель на коллекцию PKI запросов.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Store позволяет получить/установить коллекцию PKI запросов, содержащуюся в хранилище.

```
object.Store [= pVal]
```

### Параметры

pVal - указатель на коллекцию PKI запросов.

**POLICYPROFILE (GUID: AA92D123-AF6D-4054-AF70-8AADF9B511F8)**

**Свойства**

Свойство	Описание
EncryptRequiredOIDs	Свойство позволяет получить/установить список назначений сертификата (OID), требуемых для шифрования
SignatureRequiredOIDs	Свойство позволяет получить/установить список назначений сертификата (OID), требуемых для подписи

**IPOLICYPROFILE::ENCRYPTREQUIREDOIDs [C++]****POLICYPROFILE::ENCRYPTREQUIREDOIDs [VBS]****[C++]**

Свойство EncryptRequiredOIDs позволяет получить/установить список назначений сертификата (OID), необходимых для выполнения шифрования.

```
HRESULT get_EncryptRequiredOIDs (
    IOIDs** pVal
);
HRESULT put_EncryptRequiredOIDs (
    IOIDs* pVal
);
```

**Параметры**

pVal [out, retval] - полученная коллекция назначений сертификата (OID).  
pVal [in] - устанавливаемая коллекция назначений сертификата (OID).

**Возвращаемое значение**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство EncryptRequiredOIDs позволяет получить/установить список назначений сертификата (OID), необходимых для выполнения шифрования.

```
object.EncryptRequiredOIDs [= pVal]
```

**Параметры**

pVal - коллекция назначений сертификата (OID-ы).

**Возвращаемое значение**

Коллекция назначений сертификата (OID).

**IPOLICYPROFILE::SIGNATUREREQUIREDOIDs [C++]****POLICYPROFILE::SIGNATUREREQUIREDOIDs [VBS]****[C++]**

Свойство SignatureRequiredOIDs позволяет получить/установить коллекцию назначений сертификата (OID), необходимых для выполнения операции подписи.

```
HRESULT get_SignatureRequiredOIDs (
    IOIDs** pVal
);
HRESULT put_SignatureRequiredOIDs (
    IOIDs* pVal
);
```

## Параметры

pVal [out, retval] – возвращаемая коллекция назначений сертификата (OID), необходимых для выполнения операции подписи.

pVal [in] – устанавливаемая коллекция назначений сертификата (OID), необходимых для выполнения операции подписи.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureRequiredOIDs позволяет получить/установить коллекцию назначений сертификата (OID), необходимых для выполнения операции подписи.

```
object.SignatureRequiredOIDs [= pVal]
```

## Параметры

pVal – коллекция назначений сертификата (OID), необходимых для выполнения операции подписи.

## PROFILE (GUID: E68CCDAE-8769-4344-9ED3-F13A0E1A84C5)

## Методы

Метод	Описание
CheckCertificate	Метод проверяет возможность использования сертификата в определенной операции в соответствии с заданной политикой.
CheckData	Метод позволяет проверить полноту заполнения параметров профиля для выполнения указанной операции
Clone	Метод позволяет создать копию текущего профиля
CollectData	Метод позволяет запустить мастер для сбора данных, необходимых для выполнения заданной операции
Display	Метод позволяет запустить просмотреть свойства текущего профиля
GetCertificate	Метод позволяет получить сертификат одного из трёх типов (ключа подписи, шифрования или расшифрования).
GetPIN	Метод позволяет получить PIN-код для доступа к ключу одного из трёх типов (подписи, шифрования или расшифрования).
GetVerifiedCertificates	Метод позволяет получить список сертификатов, которые необходимо проверить с заданным уровнем безопасности
SetCertificate	Метод позволяет установить сертификат и PIN-код ключа одного из трёх типов (подписи, шифрования или расшифрования).
SetVerifiedCertificates	Метод позволяет установить список сертификатов, которые необходимо проверять с заданным уровнем безопасности
UpdateToPolicy	Метод позволяет обновить параметры профиля в соответствии с групповой политикой

## Свойства

Свойство	Описание
AdditionalArchiveFileNames	Свойство содержит массив имен файлов, которые необходимо дополнительно добавить в создаваемый архив
ArchiveFilename	Свойство позволяет получить/установить имя архива, который будет создан при выполнении операции
CheckByCTL	Свойство позволяет получить/установить флаг необходимости проверки сертификатов по CTL.
Comment	Свойство позволяет получить/установить комментарии к подписи

CTLs	Свойство позволяет получить/установить коллекцию CTL-ей, по которым необходимо проверять сертификаты, если свойство ChaescByCTL установлено в TRUE.
DefaultPath	Свойство позволяет получить/установить путь для результирующих файлов при обратных операциях
DisableInputFilesWindow	Свойство позволяет получить/установить флаг запрета отображения окна выбора входных файлов
Description	Свойство позволяет получить/установить строку с кратким описанием профиля
Detach	Свойство позволяет получить/установить флаг формирования отдельной подписи
DetachedFiles	Свойство позволяет получить/установить массив строк, содержащих полный путь к файлам исходных данных для отдельной подписи
EncAlg	Свойство позволяет получить/установить идентификатор алгоритма шифрования
EncryptArchiveFiles	Свойство позволяет получить/установить флаг архивирования исходных файлов перед выполнением операции шифрования
EncryptIncludeBase64Headers	Свойство позволяет получить/установить флаг добавления Base64-заголовков в выходном зашифрованном файле
EncryptInputFilesWindow	Свойство позволяет получить/установить флаг отображения окна выбора входных файлов операции шифрования
EncryptDeleteSourceFile	Свойство позволяет получить/установить флаг отображения окна с выбором формата подписанного файла мастера шифрования
EncryptedExtension	Свойство позволяет получить/установить значение расширения выходных файлов для операции шифрования
EncryptExitFormat	Свойство позволяет получить/установить формат файла, полученного в результате шифрования
EncryptFinalWindow	Свойство позволяет получить/установить флаг отображения финального окна мастера шифрования
EncryptFormatWindow	Свойство позволяет получить/установить флаг отображения окна выбора формата результирующего файла мастера шифрования
EncryptMessageType	Свойство позволяет получить/установить тип криптографического сообщения, которое будет создано при шифровании файла
EncryptPropertiesWindow	Свойство позволяет получить/установить флаг отображения окна параметров шифрования одноименного мастера
EncryptRecipientsWindow	Свойство позволяет получить/установить флаг отображения окна со списком получателей мастера шифрования
EncryptStartWindow	Свойство позволяет получить/установить флаг отображения стартового окна мастера шифрования
EncryptToSenderAddress	Свойство позволяет получить/установить флаг шифрования в свой адрес (отправителя).
HashAlg	Свойство позволяет получить/установить идентификатор алгоритма хэширования, используемого для создания подписи
ID	Свойство позволяет получить идентификатор профиля
IncludeSignatureTime	Свойство позволяет получить/установить флаг включения времени создания подписи
IncludeTimeStampToken	Свойство позволяет получить/установить флаг включения штампа времени в подпись
InputFiles	Свойство позволяет получить/установить коллекцию входных файлов операции
InputFolder	Свойство позволяет получить/установить путь к каталогу входящих файлов
Name	Свойство позволяет получить/установить пользовательское (дружественное) имя профиля
OCSPProfile	Свойство позволяет получить/установить OCSP профиль
OCSPProfileID	Свойство позволяет получить/установить идентификатор OCSP профиля
OutputFiles	Свойство позволяет получить/установить коллекцию выходных файлов операции



OutputFolder	Свойство позволяет получить/установить путь к каталогу исходящих файлов
Policy	Свойство позволяет получить/установить настройки политики использования сертификатов
PrintSignInDoc	Свойство позволяет получить/установить флаг необходимости включения подписи в просматриваемый (сохраняемый) подписанный документ *.doc или *.xls
Provider	Свойство позволяет получить/установить имя криптопровайдера, использующегося при шифровании
ReadOnlyUI	Свойство позволяет получить/установить флаг открытия формы просмотра свойств профиля только на чтение.
Recipients	Свойство позволяет получить/установить коллекцию сертификатов получателей зашифрованного сообщения
Resource	Свойство позволяет получить/установить значение идентификатор (URI) подписываемого файла
ResourceIsFile	Свойство позволяет получить/установить флаг помещения в атрибут подписи "Идентификатор ресурса" имени исходного файла
RevocationTimeout	Свойство позволяет получить/установить время ожидания ответа при проверке статуса сертификата с помощью Revocation Provider
ShowStatusReport	Свойство позволяет получить/установить флаг отображения окна с результатами выполнения операции
SignatureExtension	Свойство позволяет получить/установить значение расширения выходных файлов для операции подписи
SignatureManagerFlags	Свойство позволяет получить/установить тип криптографического сообщения, которое будет создано при шифровании файла
SignatureType	Свойство позволяет получить/установить тип подписи
SignArchiveFiles	Свойство позволяет получить/установить флаг архивирования созданных подписей, после выполнения операции подписывания
SignCertificateWindow	Свойство позволяет получить/установить флаг отображения окна выбора сертификата подписи в одноименном мастере
SignIncludeBase64Headers	Свойство позволяет получить/установить флаг добавления Base64-заголовков в выходном подписанном файле
SignInputFilesWindow	Свойство позволяет получить/установить флаг отображения окна выбора входных файлов операции подписи
SignDeleteSourceFile	Свойство позволяет получить/установить флаг удаления исходного файла после успешного выполнения операции подписи
SignExitFormat	Свойство позволяет получить/установить флаг отображения окна с выбором формата подписанного файла мастера подписи
SignFinalWindow	Свойство позволяет получить/установить флаг отображения финального окна мастера подписи
SignFormatWindow	Свойство позволяет получить/установить выходной формат подписанного сообщения
SignPropertiesWindow	Свойство позволяет получить/установить флаг отображения окна параметров подписи одноименного мастера
SignMessageType	Свойство позволяет получить/установить тип подписанного сообщения (PKCS7 или стандарт ФНС).
SignStartWindow	Свойство позволяет получить/установить флаг отображения стартового окна мастера подписи
SignTSPWindow	Свойство позволяет получить/установить флаг отображения окна установки параметров доступа к службе Штатпов времени
SignUsage	Свойство позволяет получить/установить OID назначения подписи
SilentLevel	Свойство позволяет получить/установить уровень интерактивности
TempFolder	Свойство позволяет получить/установить путь к временному каталогу

TimeStampTypes	Свойство позволяет получить/установить тип штампа времени
UseCertificateForEncrypt	Свойство позволяет получить/установить флаг использования личного сертификата для определения параметров шифрования
VerifyCertByOCSP	Свойство позволяет получить/установить флаг проверки сертификата подписи в OCSP службе
WipeLevel	Свойство позволяет получить/установить уровень гарантированного удаления файлов

## IPROFILE::CHECKCERTIFICATE [C++]

## PROFILE::CHECKCERTIFICATE [VBS]

[C++]

Метод CheckCertificate проверяет возможность использования сертификата в определенной операции в соответствии с заданной политикой.

```
HRESULT CheckCertificate(  
    ICertificate* pCheckingCertificate,  
    CERTIFICATE_PUPROSE enmCertType,  
    CERTIFICATE_WORKABILITY* pCheckingResult  
);
```

### Параметры

pCheckingCertificate [in] – указатель на сертификат  
enmCertType [in] – определяет тип сертификата в зависимости от операции. Допустимыми значениями являются значения CERTIFICATE\_PUPROSE.  
pCheckingResult [out, retval] – возвращает возможность использования сертификата в операции. Допустимыми значениями являются значения CERTIFICATE\_WORKABILITY.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CheckCertificate проверяет возможность использования сертификата в определенной операции в соответствии с заданной политикой.

```
object.CheckCertificate(pCheckingCertificate, enmCertType)
```

### Параметры

pCheckingCertificate – указатель на сертификат.  
enmCertType – определяет тип сертификата в зависимости от операции. Допустимыми значениями являются значения CERTIFICATE\_PUPROSE.

### Возвращаемые значения

Возвращает возможность использования сертификата в операции. Допустимыми значениями являются значения CERTIFICATE\_WORKABILITY.

## IPROFILE::CHECKDATA [C++]

## PROFILE::CHECKDATA [VBS]

[C++]

Метод CheckData позволяет проверить полноту заполнения параметров профиля для выполнения указанной операции.

```
HRESULT CheckData(  
    WIZARD_TYPE enmOperation,  
    CHECKING_RESULT* pStatus  
);
```

#### Параметры

enmOperation [in] – тип операции, для которой проверяется полнота заполнения параметров профиля. Допустимыми значениями являются значения WIZARD\_TYPE.  
pStatus [out, retval] – статус завершения проверки. Допустимыми значениями являются значения CHECKING\_RESULT.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CheckData позволяет проверить полноту заполнения параметров профиля для выполнения указанной операции.

```
object.CheckData(enmOperation)
```

#### Параметры

enmOperation – тип операции, для которой проверяется полнота заполнения параметров профиля. Допустимыми значениями являются значения WIZARD\_TYPE.

#### Возвращаемое значение

Статус завершения проверки. Допустимыми значениями являются значения CHECKING\_RESULT.

### IPROFILE::CLONE [C++]

### PROFILE::CLONE [VBS]

[C++]

Метод Clone позволяет создать копию профиля.

```
HRESULT Clone(  
    IProfile** pVal  
);
```

#### Параметры

pVal [out, retval] – возвращаемое значение – профиль, являющийся копией текущего профиля.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clone позволяет создать копию профиля.

```
object.Clone(1)
```

#### Возвращаемое значение

Возвращаемое значение – профиль, являющийся копией текущего профиля.

### IPROFILE::COLLECTDATA [C++]

**PROFILE::COLLECTDATA [VBS]**

[C++]

Метод CollectData позволяет запустить мастер для сбора данных, необходимых для выполнения заданной операции.

```
HRESULT CollectData(  
    WIZARD_TYPE enmWizardType  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

enmWizardType [in] - тип запускаемого мастера. Допустимыми значениями являются значения WIZARD\_TYPE.

enmWizardType [out, retval] - индикатор завершения мастера. TRUE - завершен, FALSE - прерван.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CollectData позволяет запустить мастер для сбора данных, необходимых для выполнения заданной операции.

```
object.CollectData(enmWizardType)
```

**Параметры**

enmWizardType - тип запускаемого мастера. Допустимыми значениями являются значения WIZARD\_TYPE.

**Возвращаемое значение**

[out, retval] Индикатор завершения мастера. TRUE - завершен, FALSE - прерван.

**IPROFILE::DISPLAY [C++]****PROFILE::DISPLAY [VBS]**

[C++]

Метод Display позволяет просмотреть параметры профиля.

```
HRESULT Display(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает TRUE, если в окне нажали кнопку «ОК» или «Применить» и FALSE, если была нажата «Отмена».

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет просмотреть параметры профиля.

```
object.Display ()
```

**Возвращаемое значение**

Возвращает TRUE, если в окне нажали кнопку «ОК» или «Применить» и FALSE, если была нажата «Отмена».

## IPROFILE::GETCERTIFICATE [C++]

## PROFILE::GETCERTIFICATE [VBS]

[C++]

Метод GetCertificate позволяет получить сертификат одного из трёх типов (ключа подписи, шифрования или расшифрования).

```
HRESULT GetCertificate(  
    CERTTYPE ctType,  
    VARIANT* pCertificate  
);
```

### Параметры

ctType [in] - тип сертификата. Допустимыми значениями являются значения CERTTYPE.  
pCertificate [out, retval] - возвращаемый сертификат.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GetCertificate позволяет получить сертификат одного из трёх типов (ключа подписи, шифрования или расшифрования).

```
object. GetCertificate(ctType)
```

### Параметры

ctType - тип сертификата. Допустимыми значениями являются значения CERTTYPE.

### Возвращаемое значение

Возвращаемый сертификат.

## IPROFILE::GETPIN [C++]

## PROFILE::GETPIN [VBS]

[C++]

Метод GetPIN позволяет получить PIN-код для доступа к ключу одного из трёх типов (подписи, шифрования или расшифрования).

```
HRESULT GetPIN(  
    CERTTYPE ctType,  
    BSTR* pVal  
);
```

### Параметры

ctType [in] - тип сертификата. Допустимыми значениями являются значения CERTTYPE.  
pVal [out, retval] - возвращаемый PIN-код.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GetPIN позволяет получить PIN-код для доступа к ключу одного из трёх типов (подписи, шифрования или расшифрования).

```
object. GetPIN(ctType)
```

#### Параметры

ctType - тип сертификата. Допустимыми значениями являются значения CERTTYPE.

#### Возвращаемое значение

Возвращаемый PIN-код.

### IPROFILE::GETVERIFIEDCERTIFICATES [C++]

### PROFILE::GETVERIFIEDCERTIFICATES [VBS]

[C++]

Метод GetVerifiedCertificates позволяет получить список сертификатов, которые необходимо проверить с заданным уровнем безопасности.

```
HRESULT GetVerifiedCertificates(  
    CERTIFICATE_VERIFY_LEVEL cvLevel,  
    ICertificates** pVal  
);
```

#### Параметры

cvLevel [in] - уровень проверки. Допустимыми значениями являются значения CERTIFICATE\_VERIFY\_LEVEL.

pVal [out, retval] - коллекция сертификатов, у которых заполнено только два поля IssuerName и, опционально, SerialNumber.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GetVerifiedCertificates позволяет получить список сертификатов, которые необходимо проверить с заданным уровнем безопасности.

```
object. GetVerifiedCertificates(cvLevel)
```

#### Параметры

cvLevel - уровень проверки. Допустимыми значениями являются значения CERTIFICATE\_VERIFY\_LEVEL.

#### Возвращаемое значение

Коллекция сертификатов, у которых заполнено только два поля IssuerName и, опционально, SerialNumber.

### IPROFILE::SETCERTIFICATE [C++]

### PROFILE::SETCERTIFICATE [VBS]

[C++]

Метод SetCertificate позволяет установить сертификат и PIN-код ключа одного из трёх типов (подписи, шифрования или расшифрования).

```
HRESULT SetCertificate(  
    CERTTYPE ctType,  
    BSTR bstrPIN  
    VARIANT newVal  
);
```

#### Параметры

ctType [in] - тип сертификата. Допустимыми значениями являются значения CERTTYPE.  
bstrPIN [in] - новое значение PIN-кода.  
newVal [in] - новый сертификат.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод SetCertificate позволяет установить сертификат и PIN-код ключа одного из трёх типов (подписи, шифрования или расшифрования).

```
object.SetCertificate(ctType, bstrPIN, newVal)
```

#### Параметры

ctType - тип сертификата. Допустимыми значениями являются значения CERTTYPE.  
bstrPIN - новое значение PIN-кода.  
newVal - новый сертификат.

### IPROFILE::SETVERIFIEDCERTIFICATES [C++]

### PROFILE::SETVERIFIEDCERTIFICATES [VBS]

[C++]

Метод SetVerifiedCertificates позволяет установить список сертификатов, которые необходимо проверять с заданным уровнем безопасности.

```
HRESULT SetVerifiedCertificates(  
    CERTIFICATE_VERIFY_LEVEL cvLevel,  
    ICertificates* pVal  
);
```

#### Параметры

cvLevel [in] - уровень проверки. Допустимыми значениями являются значения CERTIFICATE\_VERIFY\_LEVEL.  
pVal [in] - коллекция сертификатов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод SetVerifiedCertificates позволяет установить список сертификатов, которые необходимо проверять с заданным уровнем безопасности.

```
object.SetVerifiedCertificates(cvLevel, pVal)
```

#### Параметры

cvLevel - уровень проверки. Допустимыми значениями являются значения CERTIFICATE\_VERIFY\_LEVEL.  
pVal - коллекция сертификатов.

**IPROFILE::UPDATETOPOLICY [C++]****PROFILE::UPDATETOPOLICY [VBS]**

[C++]

Метод UpdateToPolicy позволяет обновить параметры профиля в соответствии с групповой политикой.

```
HRESULT UpdateToPolicy(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод UpdateToPolicy позволяет обновить параметры профиля в соответствии с групповой политикой.

```
object.UpdateToPolicy()
```

**IPROFILE::ADDITIONALARCHIVEFILENAMESE [C++]****PROFILE::ADDITIONALARCHIVEFILENAMES [VBS]**

[C++]

Свойство AdditionalArchiveFileNames содержит массив имен файлов, которые необходимо дополнительно добавить в создаваемый архив.

```
HRESULT get_AdditionalArchiveFileNames(  
    VARIANT* pVal  
);  
HRESULT put_AdditionalArchiveFileNames(  
    VARIANT pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает массив строк, содержащих полный путь к файлам, которые дополнительно должны быть добавлены в архив, при выполнении архивирования.

pVal [in] - устанавливаемое значение - массив строк, содержащих полный путь к файлам, которые дополнительно должны быть добавлены в архив, при выполнении архивирования.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство AdditionalArchiveFileNames содержит массив имен файлов, которые необходимо дополнительно добавить в создаваемый архив.

object.AdditionalArchiveFileNames [= pVal]

**Параметры**

pVal - устанавливает массив с именами дополнительных файлов.

**Возвращаемое значение**

Возвращает массив с именами дополнительных файлов.



**IPROFILE::ARCHIVEFILENAME [C++]****PROFILE::ARCHIVEFILENAME [VBS]****[C++]**

Свойство ArchiveFilename позволяет получить/установить имя архива, который будет создан при выполнении операции.

```
HRESULT get_ArchiveFilename(  
    BSTR* pVal  
);  
HRESULT put_ArchiveFilename(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает строку, содержащую имя файла архива, который будет создан в результате выполнения операции архивирования.

pVal [in] - устанавливаемое значение - строка, содержащая имя файла архива, который будет создан в результате выполнения операции архивирования.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ArchiveFilename позволяет получить/установить имя архива, который будет создан при выполнении операции.

```
object.ArchiveFilename [= pVal]
```

**Параметры**

pVal - устанавливает строку с именем файла архива.

**Возвращаемое значение**

Строка с именем файла архива.

**IPROFILE::CHECKBYCTL [C++]****PROFILE::CHECKBYCTL [VBS]****[C++]**

Свойство CheckByCTL позволяет получить/установить флаг необходимости проверки сертификатов по CTL.

```
HRESULT get_CheckByCTL(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_CheckByCTL(  
    VARIANT_BOOL newVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - сертификаты будут проверяться по CTL, FALSE - сертификаты не будут проверяться по CTL.

newVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - сертификаты будут проверяться по CTL, FALSE - сертификаты не будут проверяться по CTL.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CheckByCTL позволяет получить/установить флаг необходимости проверки сертификатов по CTL.

```
object.Detach [= newVal]
```

#### Параметры

newVal - принимаемое значение TRUE или FALSE. TRUE - сертификаты будут проверяться по CTL, FALSE - сертификаты не будут проверяться по CTL.

### IPROFILE::CTLs [C++]

### PROFILE::CTLs [VBS]

[C++]

Свойство CTLs позволяет получить/установить коллекцию CTL-ей, по которым необходимо проверять сертификаты, если свойство CheckByCTL установлено в TRUE.

```
HRESULT get_CTLs(  
    ICTLs** pVal  
);  
HRESULT put_CTLs(  
    ICTLs* newVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - указатель на указатель на интерфейс ICTLs.  
newVal [in] - устанавливаемое значение - указатель на интерфейс ICTLs.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CTLs позволяет получить/установить коллекцию CTL-ей, по которым необходимо проверять сертификаты, если свойство CheckByCTL установлено в TRUE.

```
object.CTLs [= newVal]
```

#### Параметры

pVal - принимаемое значение - указатель на интерфейс ICTLs.

#### Возвращаемые значения

Возвращаемое значение - указатель на интерфейс ICTLs.

### IPROFILE::COMMENT [C++]

### PROFILE::COMMENT [VBS]

[C++]

Свойство Comment позволяет получить/установить комментарии к подписи.

```
HRESULT get_Comment(  
    BSTR* pVal
```

```
);  
HRESULT put_Comment(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое строковое значение комментария  
pVal [in] - устанавливаемое строковое значение комментария

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Comment позволяет получить/установить комментарии к подписи.

```
object.Comment [= pVal]
```

#### Параметры

pVal - строковое значение комментария.

### IPROFILE::DEFAULTPATH [C++]

### PROFILE::DEFAULTPATH [VBS]

[C++]

Свойство DefaultPath позволяет получить/установить путь для результирующих файлов при обратных операциях.

```
HRESULT get_DefaultPath(  
    BSTR* pVal  
);  
HRESULT put_DefaultPath(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемая строка, содержащая путь для сохранения файла. Если на конце строки не указан слэш, то выражение после последнего слэша считается префиксом к сохраняемому файлу.  
newVal [in] - устанавливаемая строка, содержащая путь для сохранения файла. Если на конце строки не указан слэш, то выражение после последнего слэша считается префиксом к сохраняемому файлу.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DefaultPath позволяет получить/установить путь для результирующих файлов при обратных операциях.

```
object.DefaultPath [= pVal]
```

#### Параметры

Value - строка, содержащая путь для сохранения файла. Если на конце строки не указан слэш, то выражение после последнего слэша считается префиксом к сохраняемому файлу.

**IPROFILE::DISABLEINPUTFILESWINDOW [C++]****PROFILE::DISABLEINPUTFILESWINDOW [VBS]**

[C++]

Свойство DisableInputFilesWindow позволяет получить/установить флаг запрета отображения окна выбора входных файлов, не зависимо от значений флага EncryptInputFilesWindow/SignInputFilesWindow.

```
HRESULT get_DisableInputFilesWindow(
    VARIANT_BOOL* pVal
);
HRESULT put_DisableInputFilesWindow(
    VARIANT_BOOL newVal
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - запретить показывать окно выбора входных файлов, FALSE - разрешить.  
newVal [in] - устанавливаемое значение TRUE или FALSE. запретить показывать окно выбора входных файлов, FALSE - разрешить.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DisableInputFilesWindow позволяет получить/установить флаг запрета отображения окна выбора входных файлов, не зависимо от значений флага EncryptInputFilesWindow/SignInputFilesWindow.

```
object.DisableInputFilesWindow [= pVal]
```

**Параметры**

pVal - принимаемое значение TRUE или FALSE. TRUE - запретить показывать окно выбора входных файлов, FALSE - разрешить.

**IPROFILE::DESCRIPTION [C++]****PROFILE::DESCRIPTION [VBS]**

[C++]

Свойство Description позволяет получить/установить краткое описание профиля.

```
HRESULT get_Description(
    VARIANT* pVal
);
HRESULT put_Description(
    VARIANT pVal
);
```

**Параметры**

pVal [out, retval] - возвращает строку, содержащую краткое описание профиля.  
pVal [in] - устанавливаемое значение - строка с кратким описанием профиля

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Description позволяет получить/установить краткое описание профиля.

```
object.Description [= pVal]
```

#### Параметры

pVal - устанавливаемое значение - строка с кратким описание профиля

#### Возвращаемое значение

Возвращает строку, содержащую краткое описание профиля.

### IPROFILE::DETACH [C++]

### PROFILE::DETACH [VBS]

[C++]

Свойство Detach позволяет получить/установить флаг формирования отдельной подписи.

```
HRESULT get_Detach(  
    VARIANT* pVal  
);  
HRESULT put_Detach(  
    VARIANT pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - отсоединенная подпись, FALSE - совмещенная.

newVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - отсоединенная подпись, FALSE - совмещенная.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Detach позволяет получить/установить флаг формирования отдельной подписи.

```
object.Detach [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - отсоединенная подпись, FALSE - совмещенная.

### IPROFILE::DETACHEDFILES [C++]

### PROFILE::DETACHEDFILES [VBS]

[C++]

Свойство DetachedFiles позволяет получить/установить коллекцию файлов исходных данных для отдельной подписи.

```
HRESULT get_DetachedFiles(  
    VARIANT* pVal  
);  
HRESULT put_DetachedFiles(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращает массив строк, содержащих полный путь к файлам исходных данных.  
pVal [in] - устанавливаемое значение - массив строк, содержащих полный путь к файлам исходных данных.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DetachedFiles позволяет получить/установить коллекцию файлов исходных данных для отдельной подписи.

```
object.DetachedFiles [= pVal]
```

### Параметры

pVal - устанавливаемое значение - массив строк, содержащих полный путь к файлам исходных данных.

### Возвращаемое значение

Возвращает массив строк, содержащих полный путь к файлам исходных данных .

## IPROFILE::ENCALG [C++]

### PROFILE::ENCALG [VBS]

[C++]

Свойство EncAlg позволяет получить/установить идентификатор алгоритма шифрования.

```
HRESULT get_EncAlg(  
    BSTR* pVal  
);  
HRESULT put_EncAlg(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] - строка, содержащая идентификатор алгоритма шифрования.  
pVal [in] - строка, содержащая идентификатор алгоритма шифрования.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncAlg позволяет получить/установить идентификатор алгоритма шифрования.

```
object.EncAlg [= pVal]
```

### Параметры

pVal - строка, содержащая идентификатор алгоритма шифрования.

## IPROFILE::ENCRYPTARCHIVEFILES [C++]

### PROFILE::ENCRYPTARCHIVEFILES [VBS]

[C++]

Свойство `EncryptArchiveFiles` позволяет получить/установить флаг архивирования исходных файлов перед выполнением операции шифрования.

```
HRESULT get_EncryptArchiveFiles(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_EncryptArchiveFiles(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

`pVal [out, retval]` - возвращаемое значение `TRUE` или `FALSE`. `TRUE` - файлы перед шифрованием будут запакованы в архив. `FALSE` - Архивирование не будет проводиться.

`pVal [in]` - устанавливаемое значение `TRUE` или `FALSE`. `TRUE` - файлы перед шифрованием будут запакованы в архив. `FALSE` - Архивирование не будет проводиться.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `EncryptArchiveFiles` позволяет получить/установить флаг архивирования исходных файлов перед выполнением операции шифрования.

```
object.EncryptArchiveFiles [= pVal]
```

#### Параметры

`pVal` - устанавливает флаг архивирования исходных файлов перед выполнением операции шифрования.

#### Возвращаемое значение

Возвращает флаг архивирования исходных файлов перед выполнением операции шифрования.

### IPROFILE::ENCRYPTINCLUDEBASE64HEADERS [C++]

### PROFILE::ENCRYPTINCLUDEBASE64HEADERS [VBS]

[C++]

Свойство `EncryptIncludeBase64Headers` позволяет получить/установить флаг добавления Base64-заголовков в выходном зашифрованном файле.

```
HRESULT get_EncryptIncludeBase64Headers(  
    VARIANT* pVal  
);  
HRESULT put_EncryptIncludeBase64Headers(  
    VARIANT pVal  
);
```

#### Параметры

`pVal [out, retval]` - возвращаемое значение `TRUE` или `FALSE`. `TRUE` - устанавливать заголовки, `FALSE` - не устанавливать.

`newVal [in]` - устанавливаемое значение `TRUE` или `FALSE`. `TRUE` - устанавливать заголовки, `FALSE` - не устанавливать.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `EncryptIncludeBase64Headers` позволяет получить/установить флаг добавления Base64-заголовков в выходном зашифрованном файле.

```
object.EncryptIncludeBase64Headers [= pVal]
```

#### Параметры

`pVal` - принимает значения `TRUE` или `FALSE`. `TRUE` - устанавливать заголовки, `FALSE` - не устанавливать.

### **IPROFILE::ENCRYPTINPUTFILESWINDOW [C++]**

### **PROFILE::ENCRYPTINPUTFILESWINDOW [VBS]**

[C++]

Свойство `EncryptInputFilesWindow` позволяет получить/установить флаг отображения окна выбора входных файлов операции шифрования.

```
HRESULT get_EncryptInputFilesWindow(  
    VARIANT* pVal  
);  
HRESULT put_EncryptInputFilesWindow(  
    VARIANT pVal  
);
```

#### Параметры

`pVal [out, retval]` - возвращаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно с выбора входных файлов операции шифрования, `FALSE` - не показывать.

`pVal [in]` - устанавливаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно выбора входных файлов операции шифрования, `FALSE` - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `EncryptInputFilesWindow` позволяет получить/установить флаг отображения окна выбора входных файлов операции шифрования.

```
object.EncryptInputFilesWindow [= pVal]
```

#### Параметры

`pVal` - принимаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно выбора входных файлов операции шифрования, `FALSE` - не показывать.

### **IPROFILE::ENCRYPTDELETESOURCEFILE [C++]**

### **PROFILE::ENCRYPTDELETESOURCEFILE [VBS]**

[C++]

Свойство `EncryptDeleteSourceFile` позволяет получить/установить флаг удаления исходного файла после успешного выполнения операции шифрования.

```
HRESULT get_EncryptDeleteSourceFile(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_EncryptDeleteSourceFile(  
    VARIANT_BOOL pVal  
);
```



### Параметры

pVal  
[out, retval] Возвращаемое значение TRUE или FALSE. TRUE - установить флаг удаления исходного файла, FALSE - не устанавливать.  
pVal  
[in] Устанавливаемое значение TRUE или FALSE. TRUE - установить флаг удаления исходного файла, FALSE - не устанавливать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptDeleteSourceFile позволяет получить/установить флаг удаления исходного файла после успешного выполнения операции шифрования.

```
object.EncryptDeleteSourceFile [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - установить флаг удаления исходного файла, FALSE - не устанавливать.

## IPROFILE::ENCRYPTEDEXTENSION [C++]

### PROFILE::ENCRYPTEDEXTENSION [VBS]

[C++]

Свойство EncryptedExtension позволяет получить/установить значение расширения выходных файлов для операции шифрования.

```
HRESULT get_EncryptedExtension(  
    PROFILEEXITFORMAT enmFormat  
    BSTR* pVal  
);  
HRESULT put_EncryptedExtension(  
    PROFILEEXITFORMAT enmFormat  
    BSTR newVal  
);
```

### Параметры

enmFormat1 [in] - формат выходного файла. Может принимать значения из набора PROFILEEXITFORMAT.  
newVal [in] - устанавливаемое значение - строка расширения выходного файла шифрования.  
pVal [out, retval] - возвращаемое значение - строка расширения выходного файла шифрования.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptedExtension позволяет получить/установить значение расширения выходных файлов для операции шифрования.

```
object.EncryptedExtension [enmFormat, = pVal]
```

### Параметры

enmFormat - формат выходного файла. Может принимать значения из набора PROFILEEXITFORMAT.

pVal - устанавливаемое значение - строка расширения выходного файла шифрования.

### Возвращаемое значение

Строка расширения выходного файла шифрования.

## IPROFILE::ENCRYPTEXITFORMAT [C++]

## PROFILE::ENCRYPTEXITFORMAT [VBS]

[C++]

Свойство EncryptExitFormat позволяет получить/установить формат файла, полученного в результате шифрования.

```
HRESULT get_EncryptExitFormat(  
    PROFILEEXITFORMAT* pVal  
);  
HRESULT put_EncryptExitFormat(  
    PROFILEEXITFORMAT pVal  
);
```

### Параметры

pVal [out, retval] - DER или Base64. Допустимыми значениями являются значения PROFILEEXITFORMAT.

pVal [in] - DER или Base64. Допустимыми значениями являются значения PROFILEEXITFORMAT.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptExitFormat позволяет получить/установить формат файла, полученного в результате шифрования.

```
object.EncryptExitFormat [=pVal]
```

### Параметры

pVal - DER или Base64. Допустимыми значениями являются значения PROFILEEXITFORMAT.

## IPROFILE::ENCRYPTFINALWINDOW [C++]

## PROFILE::ENCRYPTFINALWINDOW [VBS]

[C++]

Свойство EncryptFinalWindow позволяет получить/установить флаг отображения финального окна мастера шифрования.

```
HRESULT get_EncryptFinalWindow(  
    VARIANT* pVal  
);  
HRESULT put_EncryptFinalWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - принимаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера шифрования. FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера шифрования. FALSE - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptFinalWindow позволяет получить/установить флаг отображения финального окна мастера шифрования.

```
object.EncryptFinalWindow [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера шифрования. FALSE - не показывать.

### IPROFILE::ENCRYPTFORMATWINDOW [C++]

### PROFILE::ENCRYPTFORMATWINDOW [VBS]

[C++]

Свойство EncryptFormatWindow позволяет получить/установить флаг отображения окна выбора формата результирующего файла мастера шифрования.

```
HRESULT get_EncryptFormatWindow(  
    VARIANT* pVal  
);  
HRESULT put_EncryptFormatWindow(  
    VARIANT pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно выбора формата шифрования. FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно выбора формата шифрования. FALSE - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptFormatWindow позволяет получить/установить флаг отображения окна выбора формата результирующего файла мастера шифрования.

```
object.EncryptFormatWindow [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно выбора формата шифрования. FALSE - не показывать.

### IPROFILE::ENCRYPTPROPERTIESWINDOW [C++]

### PROFILE::ENCRYPTPROPERTIESWINDOW [VBS]

[C++]

Свойство `EncryptPropertiesWindow` позволяет получить/установить флаг отображения окна параметров шифрования одноименного мастера.

#### **HRESULT get\_EncryptPropertiesWindow(**

```
VARIANT* pVal
);
HRESULT put_EncryptPropertiesWindow(
    VARIANT pVal
);
```

#### **Параметры**

`pVal [out, retval]` - возвращаемое значение TRUE или FALSE. TRUE - показывать окно установки параметров шифрования мастера шифрования. FALSE - не показывать.

`pVal [in]` - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно установки параметров шифрования мастера шифрования. FALSE - не показывать.

#### **Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство `EncryptPropertiesWindow` позволяет получить/установить флаг отображения окна параметров шифрования одноименного мастера.

```
object.EncryptPropertiesWindow [= pVal]
```

#### **Параметры**

`pVal` - принимаемое значение TRUE или FALSE. TRUE - показывать окно установки параметров шифрования мастера шифрования. FALSE - не показывать.

### **IPROFILE::ENCRYPTRECIPIENTSWINDOW [C++]**

#### **PROFILE::ENCRYPTRECIPIENTSWINDOW [VBS]**

[C++]

Свойство `EncryptRecipientsWindow` позволяет получить/установить флаг отображения окна со списком получателей мастера шифрования.

```
HRESULT get_EncryptRecipientsWindow(
    VARIANT* pVal
);
HRESULT put_EncryptRecipientsWindow(
    VARIANT pVal
);
```

#### **Параметры**

`pVal [out, retval]` - возвращаемое значение TRUE или FALSE. TRUE - показывать окно выбора получателей шифрованного сообщения мастера шифрования. FALSE - не показывать.

`pVal [in]` - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно выбора получателей шифрованного сообщения мастера шифрования. FALSE - не показывать.

#### **Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptRecipientsWindow позволяет получить/установить флаг отображения окна со списком получателей мастера шифрования.

```
object.EncryptRecipientsWindow [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно выбора получателей шифрованного сообщения мастера шифрования. FALSE - не показывать.

### IPROFILE::ENCRYPTSTARTWINDOW [C++]

### PROFILE::ENCRYPTSTARTWINDOW [VBS]

[C++]

Свойство EncryptStartWindow позволяет получить/установить флаг отображения стартового окна мастера шифрования.

```
HRESULT get_EncryptStartWindow(  
    VARIANT* pVal  
);  
HRESULT put_EncryptStartWindow(  
    VARIANT pVal  
);
```

#### Параметры

pVal [out, retval] - принимаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера шифрования. FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера шифрования. FALSE - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptStartWindow позволяет получить/установить флаг отображения стартового окна мастера шифрования.

```
object.EncryptStartWindow [= Value]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера шифрования. FALSE - не показывать.

### IPROFILE::ENCRYPTTOSENDERADDRESS [C++]

### PROFILE::ENCRYPTTOSENDERADDRESS [VBS]

[C++]

Свойство EncryptToSenderAddress позволяет получить/установить флаг шифрования в свой адрес (отправителя).

```
HRESULT get_EncryptToSenderAddress(  
    VARIANT* pVal  
);  
HRESULT put_EncryptToSenderAddress(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - принимаемое значение TRUE или FALSE. TRUE - установить флаг шифрования в свой адрес. FALSE - не устанавливать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - установить флаг шифрования в свой адрес. FALSE - не устанавливать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство EncryptToSenderAddress позволяет получить/установить флаг шифрования в свой адрес (отправителя).

```
object.EncryptToSenderAddress [= Value]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - установить флаг шифрования в свой адрес. FALSE - не устанавливать.

## IPROFILE::HASHALG [C++]

## PROFILE::HASHALG [VBS]

[C++]

Свойство HashAlg позволяет получить/установить идентификатор алгоритма хэширования, используемого для создания подписи.

```
HRESULT get_HashAlg(  
    BSTR* pVal  
);  
HRESULT put_HashAlg(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемая строка, содержащая идентификатор хэш алгоритма.  
pVal [in] - устанавливаемая строка, содержащая идентификатор хэш алгоритма.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HashAlg позволяет получить/установить идентификатор алгоритма хэширования, используемого для создания подписи.

```
object.HashAlg [= pVal]
```

### Параметры

pVal - строка, содержащая идентификатор хэш алгоритма.

## IPROFILE::ID [C++]

## PROFILE::ID [VBS]

[C++]

Свойство ID позволяет получить идентификатор профиля.

```
HRESULT get_ID(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] - идентификатор профиля.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ID позволяет получить идентификатор профиля.

```
object.ID
```

#### Возвращаемое значение

Идентификатор профиля.

### IPROFILE::INCLUDESIGNATURETIME [C++]

### PROFILE::INCLUDESIGNATURETIME [VBS]

[C++]

Свойство IncludeSignatureTime позволяет получить/установить флаг включения времени создания подписи.

```
HRESULT get_IncludeSignatureTime(  
    VARIANT* pVal  
);  
HRESULT put_IncludeSignatureTime(  
    VARIANT pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - включать подпись, FALSE - не включать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - включать подпись, FALSE - не включать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство IncludeSignatureTime позволяет получить/установить флаг включения времени создания подписи.

```
object.IncludeSignatureTime [= pVal]
```

#### Параметры

pVal - принимаемые значения TRUE или FALSE. TRUE - включать подпись, FALSE - не включать

### IPROFILE::INCLUDETIMESTAMP\_TOKEN [C++]

### PROFILE::INCLUDETIMESTAMP\_TOKEN [VBS]

**[C++]**

Свойство IncludeTimeStampToken позволяет получить/установить флаг включения в подпись штампа времени.

```
HRESULT get_IncludeTimeStampToken(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_IncludeTimeStampToken(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - установлен флаг включения в подпись штампа времени, FALSE - не установлен флаг.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - установить флаг включения в подпись штампа времени, FALSE - не устанавливать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство IncludeTimeStampToken позволяет получить/установить флаг включения в подпись штампа времени.

```
object.IncludeTimeStampToken [= pVal]
```

**Параметры**

pVal - устанавливаемое значение TRUE или FALSE. TRUE - установить флаг включения в подпись штампа времени, FALSE - не устанавливать.

**Возвращаемые значения**

Возвращаемое значение TRUE или FALSE. TRUE - установлен флаг включения в подпись штампа времени, FALSE - не установлен флаг.

**IPROFILE::INPUTFILES [C++]****PROFILE::INPUTFILES [VBS]****[C++]**

Свойство InputFiles позволяет получить/установить коллекцию входных файлов операции.

```
HRESULT get_InputFiles(  
    VARIANT* pVal  
);  
HRESULT put_InputFiles(  
    VARIANT pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает массив строк, содержащих полный путь к входным файлам операции.

pVal [in] - устанавливаемое значение - массив строк, содержащих полный путь к входным файлам операции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**



Свойство InputFiles позволяет получить/установить коллекцию входных файлов операции.

```
object.InputFiles [= pVal]
```

#### Параметры

pVal - устанавливаемое значение - массив строк, содержащих полный путь к входным файлам операции.

#### Возвращаемое значение

Возвращает массив строк, содержащих полный путь к входным файлам операции.

### IPROFILE::INPUTFOLDER [C++]

### PROFILE::INPUTFOLDER [VBS]

[C++]

Свойство InputFolder Свойство позволяет получить/установить путь к каталогу входящих файлов

```
HRESULT get_InputFolder(  
    BSTR* pVal  
);  
HRESULT put_InputFolder(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - возвращает указатель на строку, содержащую путь к каталогу входящих файлов.

pVal [in] - устанавливаемое значение строка, содержащая путь к каталогу входящих файлов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство InputFolder позволяет получить/установить путь к каталогу, используемому для хранения входящих файлов.

```
object.InputFolder [= pVal]
```

#### Параметры

pVal - устанавливаемое значение строка, содержащая путь к каталогу входящих файлов.

#### Возвращаемое значение

Возвращает указатель на строку, содержащую путь к каталогу входящих файлов.

### IPROFILE::NAME [C++]

### PROFILE::NAME [VBS]

[C++]

Свойство Name позволяет получить/установить пользовательское (дружественное) имя профиля.

```
HRESULT get_Name(  
    BSTR* pVal  
);
```

```
HRESULT put_Name(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку, содержащую имя профиля.  
pVal [in] - строка, содержащая имя профиля.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Name позволяет получить/установить пользовательское (дружественное) имя профиля.

```
object.Name [= pVal]
```

#### Параметры

pVal - строка, содержащая имя профиля.

### IPROFILE::OCSPPROFILE [C++]

### PROFILE::OCSPPROFILE [VBS]

[C++]

Свойство OCSPProfile позволяет получить/установить параметры доступа к Службе актуальных статусов (OCSP).

```
HRESULT get_OCSPProfile(  
    IOCSPPProfile* pVal  
);  
HRESULT put_OCSPProfile(  
    IOCSPPProfile pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - указатель на интерфейс IOCSPPProfile.  
pVal [in] - устанавливаемое значение - указатель на интерфейс IOCSPPProfile.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPProfile позволяет получить/установить параметры доступа к Службе актуальных статусов (OCSP).

```
object.OCSPProfile [= pVal]
```

#### Параметры

pVal - принимаемое значение - указатель на объект OCSPProfile.

#### Возвращаемые значения

Возвращаемое значение - указатель на объект OCSPProfile.

### IPROFILE::OCSPPROFILEID [C++]

**PROFILE::OCSPProfileID [VBS]**

[C++]

Свойство OCSPProfileID позволяет получить/установить идентификатор профиля Службы Актуальных Статусов (OCSP).

```
HRESULT get_OCSPProfileID(  
    BSTR* pVal  
);  
HRESULT put_OCSPProfileID(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на строку, содержащую идентификатор OCSP профиля.

pVal [in] - устанавливаемое значение строка, содержащая идентификатор OCSP профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPProfileID позволяет получить/установить идентификатор профиля Службы Актуальных Статусов (OCSP).

```
object.OCSPProfileID [= pVal]
```

**Параметры**

pVal - устанавливаемое значение строка, содержащая идентификатор OCSP профиля.

**Возвращаемое значение**

Возвращает указатель на строку, содержащую идентификатор OCSP профиля.

**IPROFILE::OUTPUTFILES [C++]****PROFILE::OUTPUTFILES [VBS]**

[C++]

Свойство OutputFiles позволяет получить/установить коллекцию выходных файлов операции.

```
HRESULT get_OutputFiles(  
    VARIANT* pVal  
);  
HRESULT put_OutputFiles(  
    VARIANT pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает массив строк, содержащих полный путь к выходным файлам операции.

pVal [in] - устанавливаемое значение - массив строк, содержащих полный путь к выходным файлам операции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство OutputFiles позволяет получить/установить коллекцию выходных файлов операции.

```
object.OutputFiles [= pVal]
```

**Параметры**

pVal - устанавливаемое значение - массив строк, содержащих полный путь к выходным файлам операции.

**Возвращаемое значение**

Возвращает массив строк, содержащих полный путь к выходным файлам операции.

**IPROFILE::OUTPUTFOLDER [C++]****PROFILE::OUTPUTFOLDER [VBS]****[C++]**

Свойство OutputFolder Свойство позволяет получить/установить путь к каталогу исходящих файлов

```
HRESULT get_OutputFolder(  
    BSTR* pVal  
);  
HRESULT put_OutputFolder(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на строку, содержащую путь к каталогу исходящих файлов.

pVal [in] - устанавливаемое значение строка, содержащая путь к каталогу исходящих файлов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство OutputFolder позволяет получить/установить путь к каталогу, используемому для хранения исходящих файлов.

```
object.OutputFolder [= pVal]
```

**Параметры**

pVal - устанавливаемое значение строка, содержащая путь к каталогу исходящих файлов.

**Возвращаемое значение**

Возвращает указатель на строку, содержащую путь к каталогу исходящих файлов.

**IPROFILE::POLICY [C++]****PROFILE::POLICY [VBS]****[C++]**

Свойство Policy позволяет получить/установить настройки политики использования сертификатов.

```
HRESULT get_Policy(  

```

```
IPolicyProfile** pVal  
);  
HRESULT put_Policy(  
    IPolicyProfile* pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на интерфейс IPolicyProfile.  
pVal [in] - указатель на интерфейс IPolicyProfile.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Policy позволяет получить/установить настройки политики использования сертификатов.

object.Policy [= pVal]

#### Параметры

pVal - указатель на объект PolicyProfile.

### IPROFILE::PROVIDER [C++]

### PROFILE::PROVIDER [VBS]

[C++]

Свойство Provider позволяет получить/установить имя криптопровайдера, использующегося при шифровании.

```
HRESULT get_Provider(  
    BSTR* pVal  
);  
HRESULT put_Provider(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - имя криптопровайдера.  
pVal [in] - имя криптопровайдера.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Provider позволяет получить/установить имя криптопровайдера, использующегося при шифровании.

object.Provider [= pVal]

#### Параметры

pVal - имя криптопровайдера.

### IPROFILE::READONLYUI [C++]

### PROFILE::READONLYUI [VBS]

**[C++]**

Свойство ReadOnlyUI позволяет получить/установить флаг открытия формы просмотра свойств профиля только на чтение. Используется только для метода Display.

```
HRESULT get_ReadOnlyUI(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_ReadOnlyUI(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - TRUE или FALSE. TRUE - форма просмотра свойств профиля будет открыта только на чтение, FALSE - на чтение и запись.

pVal [in] - устанавливаемое значение - TRUE или FALSE. TRUE - форма просмотра свойств профиля будет открыта только на чтение, FALSE - на чтение и запись.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ReadOnlyUI позволяет получить/установить флаг открытия формы просмотра свойств профиля только на чтение. Используется только для метода Display.

```
object.ReadOnlyUI [= pVal]
```

**Параметры**

pVal - принимаемое значение - TRUE или FALSE. TRUE - форма просмотра свойств профиля будет открыта только на чтение, FALSE - на чтение и запись.

**Возвращаемые значения**

Возвращаемое значение - TRUE или FALSE. TRUE - форма просмотра свойств профиля будет открыта только на чтение, FALSE - на чтение и запись.

**IPROFILE::RECIPIENTS [C++]****PROFILE::RECIPIENTS [VBS]****[C++]**

Свойство Recipients позволяет получить/установить коллекцию сертификатов получателей зашифрованного сообщения.

```
HRESULT get_Recipients(  
    ICertificates** pVal  
);  
HRESULT put_Recipients(  
    ICertificates* pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на интерфейс ICertificates.

pVal [in] - указатель на интерфейс ICertificates.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство Recipients позволяет получить/установить коллекцию сертификатов получателей зашифрованного сообщения.

```
object.Recipients [= pVal]
```

#### Параметры

pVal - указатель на объект Certificates.

### IPROFILE::RESOURCE [C++]

### PROFILE::RESOURCE [VBS]

[C++]

Свойство Resource позволяет получить/установить значение идентификатора (URI) подписываемого файла.

```
HRESULT get_Resource(  
    BSTR* pVal  
);  
HRESULT put_Resource(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на значение идентификатора.

pVal [in] - строка со значением идентификатора.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Resource позволяет получить/установить значение идентификатора (URI) подписываемого файла.

```
object.Resource [= pVal]
```

#### Параметры

pVal - значение идентификатора.

### IPROFILE::RESOURCEISFILE [C++]

### PROFILE::RESOURCEISFILE [VBS]

[C++]

Свойство ResourceIsFile позволяет получить/установить флаг помещения имени исходного файла в атрибут подписи "Идентификатор ресурса".

```
HRESULT get_ResourceIsFile(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_ResourceIsFile(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - помещать в атрибут "Идентификатор ресурса" имя исходного файла, FALSE - не помещать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - помещать в атрибут "Идентификатор ресурса" имя исходного файла, FALSE - не помещать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ResourceIsFile позволяет получить/установить флаг помещения имени исходного файла в атрибут подписи «Идентификатор ресурса».

```
object.ResourceIsFile [= pVal]
```

### Параметры

pVal - устанавливаемое значение TRUE или FALSE. TRUE - помещать в атрибут «Идентификатор ресурса» имя исходного файла, FALSE - не помещать.

### Возвращаемое значение

Возвращает TRUE или FALSE. TRUE - помещать в атрибут «Идентификатор ресурса» имя исходного файла, FALSE - не помещать.

## IPROFILE::REVOCATIONTIMEOUT [C++]

### PROFILE::REVOCATIONTIMEOUT [VBS]

[C++]

Свойство RevocationTimeout позволяет получить/установить время ожидания ответа при проверке статуса сертификата с помощью Revocation Provider.

```
HRESULT get_RevocationTimeout(  
    long* pVal  
);  
HRESULT put_RevocationTimeout(  
    long pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение - время ожидания ответа в миллисекундах.

pVal [in] - устанавливаемое значение - время в миллисекундах.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RevocationTimeout позволяет получить/установить время ожидания ответа при проверке статуса сертификата с помощью Revocation Provider.

```
object.RevocationTimeout [= pVal]
```

### Параметры

pVal - устанавливаемое значение - время в миллисекундах.

### Возвращаемое значение

Возвращаемое значение - время ожидания ответа в миллисекундах.

## IPROFILE::SHOWSTATUSREPORT [C++]



**PROFILE::SHOWSTATUSREPORT [VBS]**

[C++]

Свойство ShowStatusReport позволяет получить/установить флаг отображения окна с результатами выполнения операции.

HRESULT get\_ShowStatusReport(

VARIANT\* pVal

```
);  
HRESULT put_ShowStatusReport(  
    VARIANT pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно статуса завершения операции, FALSE - не показывать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно статуса завершения операции, FALSE - не показывать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ShowStatusReport позволяет получить/установить флаг отображения окна с результатами выполнения операции.

```
object.ShowStatusReport [= pVal]
```

**Параметры**

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно статуса завершения операции, FALSE - не показывать.

**IPROFILE::SIGNATUREEXTENSION [C++]****PROFILE::SIGNATUREEXTENSION [VBS]**

[C++]

Свойство SignatureExtension позволяет получить/установить значение расширения выходных файлов для операции подписи.

```
HRESULT get_SignatureExtension(  
    PROFILEEXITFORMAT enmFormat  
    BSTR* pVal  
);  
HRESULT put_SignatureExtension(  
    PROFILEEXITFORMAT enmFormat  
    BSTR newVal  
);
```

**Параметры**

enmFormat [in] - формат выходного файла. Может принимать значения из набора PROFILEEXITFORMAT.

newVal [in] - устанавливаемое значение - строка расширения выходного файла подписи.

pVal [out, retval] - возвращаемое значение - строка расширения выходного файла подписи.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureExtension позволяет получить/установить значение расширения выходных файлов для операции подписи.

```
object.SignatureExtension [enmFormat, = pVal]
```

**Параметры**

enmFormat - формат выходного файла. Может принимать значения из набора PROFILEEXITFORMAT.

pVal - устанавливаемое значение - строка расширения выходного файла подписи.

**Возвращаемое значение**

Строка расширения выходного файла подписи.

**IPROFILE::SIGNATUREMANAGERFLAGS [C++]****PROFILE::SIGNATUREMANAGERFLAGS [VBS]**

[C++]

Свойство SignatureManagerFlags позволяет получить/установить тип криптографического сообщения, которое будет создано при шифровании файла.

```
HRESULT get_SignatureManagerFlags(  
    LONG* pVal  
);  
HRESULT put_SignatureManagerFlags(  
    LONG pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает флаги диалога управления подписанными данными.

pVal [in] - устанавливает флаги диалога управления подписанными данными.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureManagerFlags позволяет получить/установить тип криптографического сообщения, которое будет создано при шифровании файла.

```
object.SignatureManagerFlags [= pVal]
```

**Параметры**

pVal - устанавливаемое значение - флаги диалога управления подписанными данными.

**Возвращаемое значение**

Указатель на (езде возвращается указатель) число, содержащее флаги диалога управления подписанными данными.

**IPROFILE::SIGNATURETYPE [C++]**

**PROFILE::SIGNATURETYPE [VBS]**

[C++]

Свойство **SignatureType** позволяет получить/установить тип подписи (обычная или улучшенная (CAAdES-Long)).

```
HRESULT get_SignatureType(  
    SIGNATURE_TYPE* pVal  
);  
HRESULT put_SignatureType(  
    SIGNATURE_TYPE pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает тип подписи. Значения берутся из SIGNATURE\_TYPE.  
pVal [in] - устанавливает тип подписи. Значения берутся из SIGNATURE\_TYPE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство **SignatureType** позволяет получить/установить тип подписи (обычная или улучшенная (CAAdES-Long)).

```
object.SignatureType [= pVal]
```

**Параметры**

pVal - устанавливает тип подписи. Значения берутся из SIGNATURE\_TYPE.

**Возвращаемое значение**

Возвращает тип подписи. Значения берутся из SIGNATURE\_TYPE.

**IPROFILE::SIGNARCHIVEFILES [C++]****PROFILE::SIGNARCHIVEFILES [VBS]**

[C++]

Свойство **SignArchiveFiles** позволяет получить/установить флаг архивирования созданных подписей, после выполнения операции подписывания .

```
HRESULT get_SignArchiveFiles(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_SignArchiveFiles(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - файлы после подписывания будут запакованы в архив. FALSE - Архивирование не будет проводиться.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - файлы после подписывания будут запакованы в архив. FALSE - Архивирование не будет проводиться.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignArchiveFiles позволяет получить/установить флаг архивирования созданных подписей, после выполнения операции подписывания .

```
object.SignArchiveFiles [= pVal]
```

#### Параметры

pVal - устанавливает флаг архивирования созданных подписей, после выполнения операции подписывания.

#### Возвращаемое значение

Возвращает флаг архивирования созданных подписей, после выполнения операции подписывания. Может принимать значение TRUE, FALSE.

### IPROFILE::SIGNCERTIFICATEWINDOW [C++]

### PROFILE::SIGNCERTIFICATEWINDOW [VBS]

[C++]

Свойство SignCertificateWindow позволяет получить/установить флаг отображения окна выбора сертификата подписи в одноименном мастере.

```
HRESULT get_SignCertificateWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignCertificateWindow(  
    VARIANT pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно выбора сертификата подписи, FALSE - не показывать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно выбора сертификата подписи, FALSE - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignCertificateWindow позволяет получить/установить флаг отображения окна выбора сертификата подписи в одноименном мастере.

```
object.SignCertificateWindow [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно выбора сертификата подписи, FALSE - не показывать.

### IPROFILE::SIGNDELETESOURCEFILE [C++]

### PROFILE::SIGNDELETESOURCEFILE [VBS]

[C++]

Свойство `SignDeleteSourceFile` позволяет получить/установить флаг удаления исходного файла после успешного выполнения операции подписи.

```
HRESULT get_SignDeleteSourceFile(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_SignDeleteSourceFile(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

`pVal [out, retval]` – возвращаемое значение TRUE или FALSE. TRUE – установить флаг удаления исходного файла, FALSE – не устанавливать.

`pVal [in]` – устанавливаемое значение TRUE или FALSE. TRUE – установить флаг удаления исходного файла, FALSE – не устанавливать.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SignDeleteSourceFile` позволяет получить/установить флаг удаления исходного файла после успешного выполнения операции подписи.

```
object.SignDeleteSourceFile [= pVal]
```

#### Параметры

`pVal` – принимаемое значение TRUE или FALSE. TRUE – установить флаг удаления исходного файла, FALSE – не устанавливать.

### IPROFILE::SIGNEXITFORMAT [C++]

### PROFILE::SIGNEXITFORMAT [VBS]

[C++]

Свойство `SignExitFormat` позволяет получить/установить выходной формат подписанного сообщения.

```
HRESULT get_SignExitFormat(  
    PROFILEEXITFORMAT* pVal  
);  
HRESULT put_SignExitFormat(  
    PROFILEEXITFORMAT pVal  
);
```

#### Параметры

`pVal [out, retval]` – DER или Base64. Допустимыми значениями являются значения `PROFILEEXITFORMAT`.

`pVal [in]` – DER или Base64. Допустимыми значениями являются значения `PROFILEEXITFORMAT`.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SignExitFormat` позволяет получить/установить выходной формат подписанного сообщения.

```
object.SignExitFormat [= pVal]
```

### Параметры

pVal - DER или Base64. Допустимыми значениями являются значения PROFILEEXITFORMAT.

### IPROFILE::SIGNINCLUDEBASE64HEADERS [C++]

### PROFILE::SIGNINCLUDEBASE64HEADERS [VBS]

[C++]

Свойство SignIncludeBase64Headers позволяет получить/установить флаг добавления Base64-заголовков в выходном подписанном файле.

```
HRESULT get_SignIncludeBase64Headers(  
    VARIANT* pVal  
);  
HRESULT put_SignIncludeBase64Headers(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignIncludeBase64Headers позволяет получить/установить флаг добавления Base64-заголовков в выходном подписанном файле.

```
object.SignIncludeBase64Headers [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - вставлять Base64-заголовки, FALSE - не вставлять.

### IPROFILE::SIGNINPUTFILESWINDOW [C++]

### PROFILE::SIGNINPUTFILESWINDOW [VBS]

[C++]

Свойство SignInputFilesWindow позволяет получить/установить флаг отображения окна выбора входных файлов операции подписи.

```
HRESULT get_SignInputFilesWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignInputFilesWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно с выбора входных файлов операции подписи, FALSE - не показывать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно выбора входных файлов операции подписи, FALSE - не показывать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignInputFilesWindow позволяет получить/установить флаг отображения окна выбора входных файлов операции подписи.

```
object.SignInputFilesWindow [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно выбора входных файлов операции подписи, FALSE - не показывать.

## IPROFILE::SIGNFINALWINDOW [C++]

### PROFILE::SIGNFINALWINDOW [VBS]

[C++]

Свойство SignFinalWindow позволяет получить/установить флаг отображения финального окна мастера подписи.

```
HRESULT get_SignFinalWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignFinalWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера подписи, FALSE - не показывать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера подписи, FALSE - не показывать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignFinalWindow позволяет получить/установить флаг отображения финального окна мастера подписи.

```
object.SignFinalWindow [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать финальное окно мастера подписи, FALSE - не показывать.

## IPROFILE::SIGNFORMATWINDOW [C++]

### PROFILE::SIGNFORMATWINDOW [VBS]

[C++]

Свойство `SignFormatWindow` позволяет получить/установить флаг отображения окна с выбором формата подписанного файла мастера подписи.

```
HRESULT get_SignFormatWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignFormatWindow(  
    VARIANT pVal  
);
```

#### Параметры

`pVal [out, retval]` - возвращаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно выбора формата подписи. `FALSE` - не показывать.

`pVal [in]` - устанавливаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно выбора формата подписи. `FALSE` - не показывать.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SignFormatWindow` позволяет получить/установить флаг отображения окна с выбором формата подписанного файла мастера подписи.

```
object.SignFormatWindow [= pVal]
```

#### Параметры

`pVal` - принимаемое значение `TRUE` или `FALSE`. `TRUE` - показывать окно выбора формата подписи. `FALSE` - не показывать.

### IPROFILE::SIGNMESSAGE TYPE [C++]

### PROFILE::SIGNMESSAGE TYPE [VBS]

[C++]

Свойство `SignMessageType` позволяет получить/установить тип подписанного сообщения (PKCS7 или стандарт ФНС).

```
HRESULT get_SignMessageType(  
    MESSAGE_TYPE* pVal  
);  
HRESULT put_SignMessageType(  
    MESSAGE_TYPE newVal  
);
```

#### Параметры

`pVal [out, retval]` - возвращает тип подписи. Значения берутся из `MESSAGE_TYPE`.

`newVal [in]` - устанавливает тип подписи. Значения берутся из `MESSAGE_TYPE`.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SignMessageType` позволяет получить/установить тип подписанного сообщения (PKCS7 или стандарт ФНС).

```
object.SignMessageType [= newVal]
```

#### Параметры



newVal - устанавливает тип подписи. Значения берутся из MESSAGE\_TYPE.

### Возвращаемое значение

Возвращает тип подписи. Значения берутся из MESSAGE\_TYPE.

## IPROFILE::SIGNPROPERTIESWINDOW [C++]

## PROFILE::SIGNPROPERTIESWINDOW [VBS]

[C++]

Свойство SignPropertiesWindow позволяет получить/установить флаг отображения окна параметров подписи одноименного мастера.

```
HRESULT get_SignPropertiesWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignPropertiesWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно параметров подписи, FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно параметров подписи, FALSE - не показывать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignPropertiesWindow позволяет получить/установить флаг отображения окна параметров подписи одноименного мастера.

```
object.SignPropertiesWindow [= pVal]
```

### Параметры

pVal  
Принимаемое значение TRUE или FALSE. TRUE - показывать окно параметров подписи, FALSE - не показывать.

## IPROFILE::SIGNSTARTWINDOW [C++]

## PROFILE::SIGNSTARTWINDOW [VBS]

[C++]

Свойство SignStartWindow позволяет получить/установить флаг отображения стартового окна мастера подписи.

```
HRESULT get_SignStartWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignStartWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера подписи, FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера подписи, FALSE - не показывать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignStartWindow позволяет получить/установить флаг отображения стартового окна мастера подписи.

```
object.SignStartWindow [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать стартовое окно мастера подписи, FALSE - не показывать.

## IPROFILE::SIGNTSPWINDOW [C++]

### PROFILE::SIGNTSPWINDOW [VBS]

[C++]

Свойство SignTSPWindow позволяет получить/установить флаг отображения окна настройки параметров доступа к службе Штампов времени мастера подписи.

```
HRESULT get_SignTSPWindow(  
    VARIANT* pVal  
);  
HRESULT put_SignTSPWindow(  
    VARIANT pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - показывать окно параметров доступа к службе Штампов времени, FALSE - не показывать.  
pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - показывать окно параметров доступа к службе Штампов времени, FALSE - не показывать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignTSPWindow позволяет получить/установить флаг отображения окна настройки параметров доступа к службе Штампов времени мастера подписи.

```
object.SignTSPWindow [= pVal]
```

### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - показывать окно параметров доступа к службе Штампов времени, FALSE - не показывать.

## IPROFILE::SIGNUSAGE [C++]

### PROFILE::SIGNUSAGE [VBS]

[C++]

Свойство SignUsage позволяет получить/установить OID назначения подписи.

```
HRESULT get_SignUsage(  
    BSTR* pVal  
);  
HRESULT put_TempFolder(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - возвращает указатель на строку, содержащую OID назначения подписи.

pVal [in] - устанавливаемое значение строка, содержащая OID назначения подписи.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TempFolder позволяет получить/установить OID назначения подписи.

```
object.TempFolder [= pVal]
```

#### Параметры

pVal - устанавливаемое значение - строка, содержащая OID назначения подписи.

#### Возвращаемое значение

Возвращает указатель на строку, содержащую OID назначения подписи.

### IPROFILE::SILENTLEVEL [C++]

### PROFILE::SILENTLEVEL [VBS]

[C++]

Свойство SilentLevel позволяет получить/установить уровень интерактивности.

```
HRESULT get_SilentLevel(  
    SILENT_LEVEL* pVal  
);  
HRESULT put_SilentLevel(  
    SILENT_LEVEL pVal  
);
```

#### Параметры

pVal [out, retval] - уровень интерактивности. Допустимыми значениями являются значения SILENT\_LEVEL.

pVal [in] - уровень интерактивности. Допустимыми значениями являются значения SILENT\_LEVEL.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SilentLevel позволяет получить/установить уровень интерактивности.

```
object.SilentLevel [= pVal]
```

#### Параметры

pVal - уровень интерактивности. Допустимыми значениями являются значения SILENT\_LEVEL.

## IPROFILE::TEMPFOLDER [C++]

## PROFILE::TEMPFOLDER [VBS]

[C++]

Свойство TempFolder позволяет получить/установить путь к каталогу, используемому для хранения временных файлов.

```
HRESULT get_TempFolder(
    BSTR* pVal
);
HRESULT put_TempFolder(
    BSTR pVal
);
```

### Параметры

pVal [out, retval] - возвращает указатель на строку, содержащую путь к временному каталогу.

pVal [in] - устанавливаемое значение строка, содержащая путь к временному каталогу.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TempFolder позволяет получить/установить путь к каталогу, используемому для хранения временных файлов.

```
object.TempFolder [= pVal]
```

### Параметры

pVal - устанавливаемое значение строка, содержащая путь к временному каталогу.

### Возвращаемое значение

Возвращает указатель на строку, содержащую путь к временному каталогу.

## IPROFILE::TIMESTAMPTYPES [C++]

## PROFILE::TIMESTAMPTYPES [VBS]

[C++]

Свойство TimestampTypes позволяет получить/установить тип штампа времени, который определяется по типу исходных данных.

```
HRESULT get_TimestampTypes(
    TIME_STAMP_TYPE* pVal
);
HRESULT put_TimestampTypes(
    TIME_STAMP_TYPE pVal
);
```

### Параметры

pVal [out, retval] - возвращает тип штампа времени в подписи. Значения берутся из TIME\_STAMP\_TYPE.  
pVal [in] - устанавливает тип штампа времени в подписи. Значения берутся из TIME\_STAMP\_TYPE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TimeStampTypes позволяет получить/установить тип штампа времени, который определяется по типу исходных данных.

```
object.TimeStampTypes [= pVal]
```

**Параметры**

pVal - устанавливает тип штампа времени в подписи. Значения берутся из TIME\_STAMP\_TYPE.

**Возвращаемое значение**

Возвращает тип штампа времени в подписи. Значения берутся из TIME\_STAMP\_TYPE.

**IPROFILE::TSPPROFILE [C++]****PROFILE::TSPPROFILE [VBS]**

[C++]

Свойство TSPProfile позволяет получить/установить параметры доступа к Службе штампов времени.

```
HRESULT get_TSPProfile(  
    ITSPProfile* pVal  
);  
HRESULT put_TSPProfile(  
    ITSPProfile pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - указатель на интерфейс ITSPProfile.  
pVal [in] - устанавливаемое значение - указатель на интерфейс ITSPProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSPProfile позволяет получить/установить параметры доступа к Службе штампов времени.

```
object.TSPProfile [= pVal]
```

**Параметры**

pVal - принимаемое значение - указатель на объект TSPProfile.

**Возвращаемые значения**

Возвращаемое значение - указатель на объект TSPProfile.

**IPROFILE::TSPProfileID [C++]****PROFILE::TSPProfileID [VBS]****[C++]**

Свойство TSPProfileID позволяет получить/установить идентификатор профиля Службы Штamped Времени.

```
HRESULT get_TSPProfileID(  
    BSTR* pVal  
);  
HRESULT put_TSPProfileID(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение идентификатор профиля Службы Штamped Времени.

pVal [in] - устанавливаемое значение идентификатор профиля Службы Штamped Времени.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство TSPProfileID позволяет получить/установить идентификатор профиля Службы Штamped Времени.

```
object.TSPProfileID [= pVal]
```

**Параметры**

pVal - принимаемое значение идентификатор профиля Службы Штamped Времени.

**Возвращаемые значения**

Возвращаемое значение идентификатор профиля Службы Штamped Времени.

**IPROFILE::UseCertificateForEncrypt [C++]****PROFILE::UseCertificateForEncrypt [VBS]****[C++]**

Свойство UseCertificateForEncrypt позволяет получить/установить флаг использования личного сертификата для определения параметров шифрования.

```
HRESULT get_UseCertificateForEncrypt(  
    VARIANT* pVal  
);  
HRESULT put_UseCertificateForEncrypt(  
    VARIANT pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - использовать личный сертификат, FALSE - не использовать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - использовать личный сертификат, FALSE - не использовать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseCertificateForEncrypt позволяет получить/установить флаг использования личного сертификата для определения параметров шифрования.

```
object.UseCertificateForEncrypt [= pVal]
```

#### Параметры

pVal - принимаемое значение TRUE или FALSE. TRUE - использовать личный сертификат, FALSE - не использовать.

### IPROFILE::VERIFYCERTBYOCSP [C++]

#### PROFILE::VERIFYCERTBYOCSP [VBS]

[C++]

Свойство VerifyCertByOCSP позволяет получить/установить флаг проверки сертификата подписи в Службе актуальных статусов (OCSP). Флаг используется только в случае получения обычной подписи (SIGNATURE\_TYPE\_BASIC = 1). Проверка сертификата производится перед созданием подписи. Если сертификат окажется недействителен, то подпись создана не будет.

```
HRESULT get_VerifyCertByOCSP(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_VerifyCertByOCSP(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - TRUE или FALSE. TRUE - сертификат проверяется с OCSP службе, FALSE - не проверяется.

pVal [in] - устанавливаемое значение - TRUE или FALSE. TRUE - сертификат будет проверяться с OCSP службе, FALSE - не будет проверяться.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство VerifyCertByOCSP позволяет получить/установить флаг проверки сертификата подписи в Службе актуальных статусов (OCSP). Флаг используется только в случае получения обычной подписи (SIGNATURE\_TYPE\_BASIC = 1). Проверка сертификата производится перед созданием подписи. Если сертификат окажется недействителен, то подпись создана не будет.

```
object.VerifyCertByOCSP [= pVal]
```

#### Параметры

pVal - принимаемое значение - TRUE или FALSE. TRUE - сертификат будет проверяться с OCSP службе, FALSE - не будет проверяться.

#### Возвращаемые значения

Возвращаемое значение - TRUE или FALSE. TRUE - сертификат проверяется с OCSP службе, FALSE - не проверяется.

### PROFILE::WIPELEVEL [C++]

**PROFILE::WIPELEVEL [VBS]**

[C++]

Свойство WipeLevel позволяет получить/установить уровень гарантированного удаления файлов - количество перезаписываний файла случайными данными перед удалением.

```
HRESULT get_WipeLevel(  
    LONG* pVal  
);  
HRESULT put_WipeLevel(  
    LONG newVal  
);
```

**Параметры**

pVal [out, retval] - уровень гарантированного удаления файлов.  
newVal [in] - уровень гарантированного удаления файлов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство WipeLevel позволяет получить/установить уровень гарантированного удаления файлов - количество перезаписываний файла случайными данными перед удалением.

```
object.WipeLevel [= Value]
```

**Параметры**

Value - уровень гарантированного удаления файлов.

**PROFILES (GUID: 4BD67068-C7BC-4658-B2FB-5095A993980C)****Методы**

Метод	Описание
Add	Метод позволяет добавить профиль в коллекцию
Clear	Метод позволяет очистить коллекцию профилей
Remove	Метод позволяет удалить из коллекции элемент (профиль) с указанным индексом

**Свойства**

Свойство	Описание
Count	Свойство позволяет получить количество профилей в коллекции
DefaultProfile	Свойство позволяет получить/установить профиль по умолчанию
Item	Свойство позволяет получить элемент коллекции
Profile	Свойство позволяет получить профиль по его идентификатору в коллекции

**IProfiles::ADD [C++]****PROFILES::ADD [VBS]**

[C++]

Метод Add позволяет добавить профиль в коллекцию.

```
HRESULT Add(  
    IProfile* Profile
```



```
);
```

### Параметры

Profile [in] – добавляемый элемент. Указатель на интерфейс IProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить профиль в коллекцию.

```
object.Add(Profile)
```

### Параметры

Profile – добавляемый элемент. Указатель на объект Profile.

### IProfiles::Clear [C++]

### Profiles::Clear [VBS]

[C++]

Метод Clear позволяет очистить коллекцию профилей.

```
HRESULT Clear(  
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clear позволяет очистить коллекцию профилей.

```
object.Clear()
```

### IProfiles::Remove [C++]

### Profiles::Remove [VBS]

[C++]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
HRESULT Remove(  
    long lIndex  
);
```

### Параметры

lIndex [in] – номер элемента в коллекции минус 1 [0 .. Count-1].

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
object.Remove(lIndex)
```

### Параметры

lIndex – номер элемента в коллекции минус 1 [0 .. Count-1].

**IProfiles::COUNT [C++]****PROFILES::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество профилей в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

**Параметры**

pVal [out, retval] - количество профилей в коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество профилей в коллекции.

```
object.Count
```

**Возвращаемое значение**

Количество профилей в коллекции.

**IProfiles::DEFAULTPROFILE [C++]****PROFILES::DEFAULTPROFILE [VBS]**

[C++]

Свойство DefaultProfile позволяет получить/установить профиль по умолчанию.

```
HRESULT get_DefaultProfile(  
    VARIANT* pProfile  
);  
HRESULT put_DefaultProfile(  
    VARIANT pProfile  
);
```

**Параметры**

pProfile [out, retval] - если есть профиль по умолчанию, то содержит ссылку на IProfile, в противном случае - пустую ссылку.

pProfile [in] - указатель на профиль по умолчанию.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DefaultProfile позволяет получить/установить профиль по умолчанию.

```
object.DefaultProfile [= pProfile]
```

**Параметры**

pProfile - указатель на профиль по умолчанию.

**Возвращаемые значения**

Если есть профиль по умолчанию, то содержит ссылку на IProfile, в противном случае - пустую ссылку («Nothing»).

**IProfiles::Item [C++]****Profiles::Item [VBS]**

[C++]

Свойство Item позволяет получить элемент коллекции.

```
HRESULT get_Item(  
    long lIndex,  
    IProfile** pVal  
);
```

**Параметры**

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].  
pVal [out, retval] - указатель на IProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент коллекции.

```
object.Item (lIndex)
```

**Параметры**

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1].

**Возвращаемое значение**

Указатель на Profile.

**IProfiles::Profile [C++]****Profiles::Profile [VBS]**

[C++]

Свойство Profile позволяет получить профиль по его идентификатору в коллекции.

```
HRESULT get_Profile(  
    BSTR bstrID,  
    VARIANT* pProfile  
);
```

**Параметры**

bstrID [in] - идентификатор искомого профиля.  
pProfile [out, retval] - в случае успешного поиска - указатель на IProfile, в случае отсутствия - пустую ссылку

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет получить профиль по его идентификатору в коллекции.

```
object.Profile (bstrID)
```

**Параметры**

bstrID - идентификатор искомого профиля.

**Возвращаемое значение**

В случае успешного поиска - указатель на Profile, в случае отсутствия - Nothing.

**PROFILESTORE (GUID: A1D00952-CCD7-49F4-B2B8-A8F661686FAB)****Методы**

Метод	Описание
Close	Метод позволяет закрыть хранилище профилей
Display	Метод позволяет открыть графический редактор профилей
Download	Метод позволяет выполнить загрузку в хранилище профилей коллекции профилей с Веб-сервиса
Open	Метод позволяет открыть хранилище профилей
Save	Метод позволяет сохранить в хранилище текущую коллекцию профилей. Старые данные, находящиеся в хранилище, удаляются
Upload	Метод позволяет выполнить загрузку профилей из локального хранилища пользователя в хранилище профилей данного пользователя в системе Веб-сервис

**Свойства**

Свойство	Описание
OIDsDictionary	Свойство позволяет получить/установить справочник назначений сертификатов в хранилище настроек
Store	Свойство позволяет получить/установить коллекцию профилей, содержащуюся в хранилище
TSPProfileStore	Свойство позволяет получить/установить в хранилище настроек коллекцию профилей Штатпов времени
OCSPProfileStore	Свойство позволяет получить/установить в хранилище настроек коллекцию профилей Службы актуальных статусов

**IPROFILESTORE::CLOSE [C++]****PROFILESTORE::CLOSE [VBS]**

[C++]

Метод Close позволяет закрыть хранилище профилей.

```
HRESULT Close(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Close позволяет закрыть хранилище профилей.

```
object.Close()
```

**IPROFILESTORE::DISPLAY [C++]****PROFILESTORE::DISPLAY [VBS]**

[C++]

Метод Display позволяет открыть графический редактор профилей.

```
HRESULT Display(  
);
```

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет открыть графический редактор профилей.

```
object.Display()
```

### IPROFILESTORE::DOWNLOAD [C++]

### PROFILESTORE::DOWNLOAD [VBS]

[C++]

Метод Download позволяет выполнить загрузку в хранилище профилей коллекции профилей с Веб-сервиса.

```
HRESULT Download(  
    IServiceProfile* pServiceProfile  
);
```

#### Параметры

pServiceProfile [in] – указатель на интерфейс IServiceProfile, в котором установлены параметры доступа к хранилищу профилей Веб-сервиса.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Download позволяет выполнить загрузку в хранилище профилей коллекции профилей с Веб-сервиса.

```
object.Download(pServiceProfile)
```

#### Параметры

pServiceProfile – указатель на интерфейс IServiceProfile, в котором установлены параметры доступа к хранилищу профилей Веб-сервиса.

### IPROFILESTORE::OPEN [C++]

### PROFILESTORE::OPEN [VBS]

[C++]

Метод Open позволяет открыть хранилище профилей.

```
HRESULT Open(  
    PROFILESTORETYPE stStoreType,  
    BSTR bstrURI  
);
```

#### Параметры

stStoreType [in] – тип хранилища профилей. Допустимыми значениями являются значения PROFILESTORETYPE.

bstrURI [in, optional, defaultvalue("")] - строка для указания имени xml-файла с профилями.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет открыть хранилище профилей.

```
object.Open(stStoreType, bstrURI)
```

### Параметры

stStoreType - тип хранилища профилей. Допустимыми значениями являются значения PROFILESTORETYPE.

bstrURI - строка для указания имени xml-файла с профилями.

### IPROFILESTORE::SAVE [C++]

#### PROFILESTORE::SAVE [VBS]

[C++]

Метод Save позволяет сохранить в хранилище текущую коллекцию профилей. Старые данные, находящиеся в хранилище, удаляются.

```
HRESULT Save(  
);
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить в хранилище текущую коллекцию профилей. Старые данные, находящиеся в хранилище, удаляются.

```
object.Save()
```

### IPROFILESTORE::UPLOAD [C++]

#### PROFILESTORE::UPLOAD [VBS]

[C++]

Метод Upload позволяет выполнить загрузку профилей из локального хранилища пользователя в хранилище профилей данного пользователя в системе Веб-сервис.

```
HRESULT Upload(  
    IServiceProfile* pServiceProfile  
);
```

### Параметры

pServiceProfile [in] - указатель на интерфейс IServiceProfile, в котором установлены параметры доступа к Веб-сервису.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Upload позволяет выполнить загрузку профилей из локального хранилища пользователя в хранилище профилей данного пользователя в системе Веб-сервис.

```
object.Upload(pServiceProfile)
```

#### Параметры

pServiceProfile - указатель на объект ServiceProfile, в котором установлены параметры доступа к Веб-сервису.

### IPROFILESTORE::OIDSDICTIONARY [C++]

### PROFILESTORE::OIDSDICTIONARY [VBS]

[C++]

Свойство OIDSDictionary позволяет получить/установить справочник назначений сертификатов в хранилище настроек компоненты DigtCrytpo. В результате при сохранении хранилища настроек с помощью метода Save, будут сохраняться и изменения в справочнике назначений.

```
HRESULT get_OIDSDictionary(  
    IOIDs** pVal  
);  
HRESULT put_OIDSDictionary(  
    IOIDs* pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на коллекцию назначений сертификатов IOIDs.  
pVal [in] - указатель на коллекцию назначений сертификатов IOIDs.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OIDSDictionary позволяет получить/установить справочник назначений сертификатов в хранилище настроек компоненты DigtCrytpo. В результате при сохранении хранилища настроек с помощью метода Save, будут сохраняться и изменения в справочнике назначений.

```
object.OIDSDictionary [= pVal]
```

#### Параметры

pVal - указатель на коллекцию назначений сертификатов IOIDs.

### IPROFILESTORE::STORE [C++]

### PROFILESTORE::STORE [VBS]

[C++]

Свойство Store позволяет получить/установить коллекцию профилей, содержащуюся в хранилище.

```
HRESULT get_Store(  
    IProfiles** pVal  
);  
HRESULT put_Store(  
    IProfiles* pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на коллекцию профилей IProfiles.  
pVal [in] - указатель на коллекцию профилей IProfiles.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Store позволяет получить/установить коллекцию профилей, содержащуюся в хранилище.

```
object.Store [= pVal]
```

### Параметры

pVal - указатель на коллекцию профилей Profiles.

### IPROFILETSPPROFILESTORE::TSPPROFILESTORE[C++]

### PROFILETSPPROFILESTORE::TSPPROFILESTORE[VBS]

[C++]

Свойство TSPProfileStore позволяет получить/установить коллекцию профилей Штампов времени, содержащуюся в хранилище.

```
HRESULT get_TSPProfileStore(  
    IUnknown** pVal  
);  
HRESULT put_TSPProfileStore(  
    IProfiles* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на коллекцию профилей ITSPProfile.  
pVal [in] - указатель на коллекцию профилей ITSPProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSPProfileStore позволяет получить/установить коллекцию профилей Штампов времени, содержащуюся в хранилище.

```
object.TSPProfileStore [= pVal]
```

### Параметры

pVal - указатель на коллекцию профилей TSPPProfile.

### IPROFILEOCSPPROFILESTORE::OCSPPROFILESTORE[C++]

### PROFILEOCSPPROFILESTORE::OCSPPROFILESTORE[VBS]

[C++]

Свойство OCSPProfileStore позволяет получить/установить коллекцию OCSPP профилей, содержащуюся в хранилище.

```
HRESULT get_OCSPProfileStore(  
    IUnknown** pVal  
);  
HRESULT put_OCSPProfileStore(  
    IProfiles* pVal  
);
```



```
IProfiles* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на коллекцию профилей IOCSPPProfile.  
pVal [in] - указатель на коллекцию профилей IOCSPPProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPPProfileStore позволяет получить/установить коллекцию OCSPP профилей, содержащуюся в хранилище.

```
object.OCSPPProfileStore [= pVal]
```

### Параметры

pVal - указатель на коллекцию профилей OCSPPProfile.

## REQUESTCERTIFICATE (GUID: 5D065C32-3BD5-44A4-92CD-EC8222EDEB27)

### Методы

Метод	Описание
CreateSelfSignedCertificate	Метод позволяет сгенерировать самоподписанный сертификат
CollectData	Метод позволяет вывести на экран мастер сбора данных, необходимых для генерации PKI запроса
Display	Метод позволяет запустить мастер сбора данных, необходимых для генерации запроса на сертификат, и сгенерировать запрос
Export	Метод позволяет экспортировать запрос из компонента в область памяти
Generate	Метод позволяет сформировать и закодировать запрос на сертификат
GenerateSelfSigned	Метод позволяет сгенерировать самоподписанный сертификат
Import	Метод позволяет импортировать запрос из области памяти
Load	Метод позволяет загрузить запрос на сертификат из файла
RetrievePending	Метод позволяет проверить состояние обработки запроса в УЦ
Retry	Метод позволяет проверить состояние обработки запроса в УЦ
Save	Метод позволяет сохранить запрос на сертификат в файл
Send	Метод позволяет осуществить транспортировку запроса в указанный ЦС
SendInstall	Метод позволяет осуществить транспортировку запроса на сертификат в указанный ЦС и установить полученный по запросу сертификат

### Свойства

Свойство	Описание
CADisposition	Свойство позволяет получить код статуса обработки запроса УЦ, полученный в ответ на отсылку запроса на сертификат методами Send и Retry
PublicKey	Свойство позволяет получить значение открытого ключа сертификата
PublicKeyAlg	Свойство позволяет получить алгоритм открытого ключа сертификата
Template	Свойство позволяет получить/установить шаблон заполнения запроса на сертифи-

	кат
--	-----

**IREQUESTCERTIFICATE::CREATESELFSIGNEDCERTIFICATE [C++]****REQUESTCERTIFICATE::CREATESELFSIGNEDCERTIFICATE [VBS]**

[C++]

Метод CreateSelfSignedCertificate позволяет запустить мастер для сбора данных, необходимых для генерации самоподписанного сертификата, и производит генерацию самоподписанного сертификата.

```
HRESULT CreateSelfSignedCertificate(  
    VARIANT* pvGeneratedCertificate  
);
```

**Параметры**

pvGeneratedCertificate [out, retval] - в случае успешной генерации содержит созданный сертификат в формате ICertificate, в случае отказа пользователя - пустую ссылку.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CreateSelfSignedCertificate позволяет запустить мастер для сбора данных, необходимых для генерации самоподписанного сертификата, и производит генерацию самоподписанного сертификата.

```
object.CreateSelfSignedCertificate()
```

**Возвращаемые значения**

В случае успешной генерации содержит созданный сертификат в формате Certificate, в случае отказа пользователя - «Nothing».

**IREQUESTCERTIFICATE::COLLECTDATA [C++]****REQUESTCERTIFICATE::COLLECTDATA [VBS]**

[C++]

Метод CollectData позволяет вывести на экран мастер сбора данных, необходимых для генерации PKI запроса.

```
HRESULT CollectData(  
    REQUEST_TYPE enmRT  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

enmRT [in] - тип запроса. Принимает значения из REQUEST\_TYPE  
pVal [out, retval] - возвращает TRUE, если запрос был сформирован успешно, и FALSE если возникли ошибки при формировании запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод CollectData позволяет вывести на экран мастер сбора данных, необходимых для генерации PKI запроса.

```
object.CollectData(enmRT)
```

**Параметры**

enmRT - тип запроса. Принимает значения из REQUEST\_TYPE

**Возвращаемое значение**

Возвращает TRUE, если запрос был сформирован успешно, и FALSE если возникли ошибки при формировании запроса.

**IREQUESTCERTIFICATE::DISPLAY [C++]**

**REQUESTCERTIFICATE::DISPLAY [VBS]**

[C++]

Метод Display позволяет запустить мастер сбора данных, необходимых для генерации запроса на сертификат, и сгенерировать запрос.

```
HRESULT Display(  
    VARIANT_BOOL* pVal  
);
```

**Параметры**

pVal [out, retval] - в случае успешной генерации содержит True, иначе False.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет запустить мастер сбора данных, необходимых для генерации запроса на сертификат, и сгенерировать запрос.

```
object.Display()
```

**Возвращаемые значения**

В случае успешной генерации содержит True, иначе False.

**IREQUESTCERTIFICATE::EXPORT [C++]**

**REQUESTCERTIFICATE::EXPORT [VBS]**

[C++]

Метод Export позволяет экспортировать запрос из компонента в область памяти.

```
HRESULT Export(  
    FORMAT fFormat,  
    BSTR* pVal  
);
```

**Параметры**

fFormat [in] - выходной формат. Допустимыми значениями являются значения FORMAT.

pVal [out, retval] – приемник экспортируемого запроса.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать запрос из компонента в область памяти.

```
object.Export(fFormat)
```

#### Параметры

fFormat – выходной формат. Допустимыми значениями являются значения FORMAT.

#### Возвращаемое значение

Приемник экспортируемого запроса.

### IREQUESTCERTIFICATE::GENERATE [C++]

#### REQUESTCERTIFICATE::GENERATE [VBS]

[C++]

Метод Generate позволяет сформировать и закодировать запрос на сертификат.

```
HRESULT Generate(  
);
```

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Generate позволяет сформировать и закодировать запрос на сертификат.

```
object.Generate()
```

### IREQUESTCERTIFICATE::GENERATESELF\_SIGNED [C++]

#### REQUESTCERTIFICATE::GENERATESELF\_SIGNED [VBS]

[C++]

Метод GenerateSelfSigned позволяет сгенерировать самоподписанный сертификат.

```
HRESULT GenerateSelfSigned(  
ICertificate** pVal  
);
```

#### Параметры

pVal [out, retval] – в случае успешной генерации содержит созданный сертификат в формате ICertificate, в случае отказа пользователя – пустую ссылку.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GenerateSelfSigned позволяет сгенерировать самоподписанный сертификат.

```
object.GenerateSelfSigned()
```

**Возвращаемые значения**

В случае успешной генерации содержит созданный сертификат в формате Certificate, в случае отказа пользователя – "Nothing".

**IREQUESTCERTIFICATE::IMPORT [C++]****REQUESTCERTIFICATE::IMPORT [VBS]**

[C++]

Метод Import позволяет импортировать запрос из области памяти.

```
HRESULT Import(  
    BSTR bstrSource  
);
```

**Параметры**

bstrSource  
[in] Источник импортируемого запроса, расположенного в памяти.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать запрос из области памяти.

```
object.Import(bstrSource)
```

**Параметры**

bstrSource – источник импортируемого запроса, расположенного в памяти.

**IREQUESTCERTIFICATE::LOAD [C++]****REQUESTCERTIFICATE::LOAD [VBS]**

[C++]

Метод Load позволяет загрузить запрос на сертификат из файла.

```
HRESULT Load(  
    BSTR bstrURI  
);
```

**Параметры**

bstrURI [in] – URI идентификатор, откуда необходимо загрузить запрос. В качестве URI можно указывать идентификатор вида "file://" .

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить запрос на сертификат из файла.

```
object.Load(bstrURI)
```

**Параметры**

bstrURI – URI идентификатор, откуда необходимо загрузить запрос. В качестве URI можно указывать идентификатор вида "file://" .

**IREQUESTCERTIFICATE::RETRIEVEPENDING [C++]****REQUESTCERTIFICATE::RETRIEVEPENDING [VBS]**

[C++]

Метод RetrievePending позволяет проверить состояние обработки запроса в УЦ, работающем в режиме политики отложенного выпуска сертификатов. Метод аналогичен Retry, только в качестве параметра надо передать запрос (шаблон сертификата) из хранилища запросов, созданный ранее.

```
HRESULT RetrievePending(  
    ICertificate* pRequest,  
    long Flags,  
    VARIANT* pVal  
);
```

**Параметры**

pRequest [in] - шаблон сертификата.

Flags [in] - флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 ( 0x1 )	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER ( 0 )	Формат Unicode BASE64 с заголовками
CR_IN_BINARY ( 0x2 )	Формат DER
CR_IN_ENCODEANY ( 0xff )	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

pVal [out, retval] - в случае, если сертификат выпущен, содержит строку с сертификатом, иначе - пустую ссылку.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RetrievePending позволяет проверить состояние обработки запроса в УЦ, работающем в режиме политики отложенного выпуска сертификатов. Метод аналогичен Retry, только в качестве параметра надо передать запрос (шаблон сертификата) из хранилища запросов, созданный ранее.

```
object.RetrievePending(pRequest, Flags)
```

**Параметры**

pRequest - шаблон сертификата.

Flags - флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 ( 0x1 )	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER ( 0 )	Формат Unicode BASE64 с заголовками
CR_IN_BINARY ( 0x2 )	Формат DER
CR_IN_ENCODEANY ( 0xff )	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

## Возвращаемое значение

В случае, если сертификат выпущен, содержит строку с сертификатом, иначе – «Nothing».

## IREQUESTCERTIFICATE::RETRY [C++]

## REQUESTCERTIFICATE::RETRY [VBS]

[C++]

Метод Retry позволяет проверить состояние обработки запроса в УЦ. Он должен вызываться с теми же параметрами, что и метод Send в случае, если УЦ настроен на отложенный выпуск сертификатов (CADisposition == 5 (CR\_DISP\_UNDER\_SUBMISSION)).

```
HRESULT Retry(  
    long lRequestID,  
    BSTR bstrCAAddress,  
    ICertificate* pIssuer,  
    long Flags,  
    VARIANT* pVal  
);
```

## Параметры

lRequestID [in] – номер запроса в очереди, полученный при вызове Send.  
bstrCAAddress [in] – адрес ЦС в формате DNS или IP.  
pIssuer [in] – сертификат издателя (УЦ).  
Flags [in] – флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 ( 0x1 )	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER ( 0 )	Формат Unicode BASE64 с заголовками
CR_IN_BINARY ( 0x2 )	Формат DER
CR_IN_ENCODEANY ( 0xff )	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

pVal [out, retval] – в случае, если сертификат выпущен, содержит строку с сертификатом, иначе – пустую ссылку.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Retry позволяет проверить состояние обработки запроса в УЦ. Он должен вызываться с теми же параметрами, что и метод Send в случае, если УЦ настроен на отложенный выпуск сертификатов (CADisposition == 5 (CR\_DISP\_UNDER\_SUBMISSION)).

```
object.Retry(lRequestID, bstrCAAddress, pIssuer, Flags)
```

## Параметры

lRequestID – номер запроса в очереди, полученный при вызове Send.  
bstrCAAddress – адрес ЦС.  
pIssuer – сертификат издателя (УЦ).  
Flags – флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 ( 0x1 )	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER ( 0 )	Формат Unicode BASE64 с заголовками
CR_IN_BINARY ( 0x2 )	Формат DER
CR_IN_ENCODEANY ( 0xff )	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

### Возвращаемое значение

В случае, если сертификат выпущен, содержит строку с сертификатом, иначе – «Nothing».

### IREQUESTCERTIFICATE::SAVE [C++]

### REQUESTCERTIFICATE::SAVE [VBS]

[C++]

Метод Save позволяет сохранить запрос на сертификат в файл.

```
HRESULT Save(  
    BSTR bstrURI,  
    FORMAT fFormat  
);
```

### Параметры

bstrURI [in] – URI идентификатор, куда необходимо сохранить запрос. В качестве URI можно указывать идентификатор вида "file:///".  
fFormat [in] – тип используемой кодировки при сохранении: DER или Base64. Допустимыми значениями являются значения FORMAT.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить запрос на сертификат в файл.

```
object.Save(bstrURI, fFormat)
```

### Параметры

bstrURI – URI идентификатор, куда необходимо сохранить запрос. В качестве URI можно указывать идентификатор вида "file:///".  
fFormat – тип используемой кодировки при сохранении: DER или Base64. Допустимыми значениями являются значения FORMAT.

### IREQUESTCERTIFICATE::SEND [C++]

### REQUESTCERTIFICATE::SEND [VBS]

[C++]

Метод Send позволяет осуществить транспортировку запроса в указанный ЦС.

```
HRESULT Send(  
    BSTR bstrCAName,  
    ICertificate* pIssuer,  
    long Flags,
```



```
VARIANT* pVal  
);
```

### Параметры

bstrCAName [in] - адрес ЦС в формате DNS или IP.

pIssuer [in] - сертификат УЦ, которым должен быть подписан запрашиваемый Вами сертификат.

Flags [in] - флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 = 1	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER = 0	Формат Unicode BASE64 с заголовками
CR_IN_BINARY = 2	Формат DER
CR_IN_ENCODEANY = 0xff	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

pVal [out, retval] - в случае незамедлительного выпуска сертификата УЦ содержит строку с выпущенным сертификатом, в случае отложенного выпуска - номер запроса в очереди (тип - LONG), по которому позже можно будет получить выпущенный сертификат при помощи метода Retry.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Send позволяет осуществить транспортировку запроса в указанный ЦС.

```
object.Send(bstrCAName, pIssuer, Flags)
```

### Параметры

bstrCAName - адрес ЦС в формате DNS или IP.

pIssuer - сертификат УЦ, которым должен быть подписан запрашиваемый Вами сертификат.

Flags [in] - флаг формата запроса на сертификат. Может принимать следующие значения:

Значение	Описание
CR_IN_BASE64 ( 0x1 )	Формат Unicode BASE64 без заголовков
CR_IN_BASE64HEADER ( 0 )	Формат Unicode BASE64 с заголовками
CR_IN_BINARY ( 0x2 )	Формат DER
CR_IN_ENCODEANY ( 0xff )	Формат не определен. Проверяет все три формата CR_IN_BASE64, CR_IN_BASE64HEADER, CR_IN_BINARY

### Возвращаемое значение

В случае незамедлительного выпуска сертификата УЦ содержит строку с выпущенным сертификатом, в случае отложенного выпуска - номер запроса в очереди (тип - LONG), по которому позже можно будет получить выпущенный сертификат при помощи метода Retry.

## IREQUESTCERTIFICATE::SENDINSTALL [C++]

**REQUESTCERTIFICATE::SENDINSTALL [VBS]****[C++]**

Метод SendInstall позволяет вывести на экран редактор запроса.

```
HRESULT SendInstall(  
    VARIANT* pVal  
);
```

**Параметры**

pVal  
[out, retval] Возвращает TRUE, если запрос был сформирован успешно, и FALSE если возникли ошибки при формировании запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод SendInstall позволяет вывести на экран редактор запроса.

```
object.SendInstall(enmRT)
```

**Возвращаемое значение**

Возвращает TRUE, если апрос был сформирован успешно, и FALSE если возникли ошибки при формировании запроса.

**IREQUESTCERTIFICATE::CADISPOSITION [C++]****REQUESTCERTIFICATE::CADISPOSITION [VBS]****[C++]**

Свойство CADisposition позволяет получить код статуса обработки запроса УЦ, полученный в ответ на отсылку запроса на сертификат методами Send и Retry.

```
HRESULT get_CADisposition(  
    long* pVal  
);
```

**Параметры**

pVal

Значение	Описание
CR_DISP_INCOMPLETE (0)	Запрос содержит недостаточно данных
CR_DISP_ERROR (1)	Ошибка обработки запроса
CR_DISP_DENIED (2)	Запрос отклонен
CR_DISP_ISSUED (3)	Сертификат выпущен
CR_DISP_ISSUED_OUT_OF_BAND (4)	Сертификат выпущен отдельно
CR_DISP_UNDER_SUBMISSION (5)	Запрос будет обработан позже

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство CAdisposition позволяет получить код статуса обработки запроса УЦ, полученный в ответ на отсылку запроса на сертификат методами Send и Retry.

```
object.CAdisposition
```

#### Возвращаемое значение

Значение	Описание
CR_DISP_INCOMPLETE (0)	Запрос содержит недостаточно данных
CR_DISP_ERROR (1)	Ошибка обработки запроса
CR_DISP_DENIED (2)	Запрос отклонен
CR_DISP_ISSUED (3)	Сертификат выпущен
CR_DISP_ISSUED_OUT_OF_BAND (4)	Сертификат выпущен отдельно
CR_DISP_UNDER_SUBMISSION (5)	Запрос будет обработан позже

#### IREQUESTCERTIFICATE::PUBLICKEY [C++]

#### REQUESTCERTIFICATE::PUBLICKEY [VBS]

[C++]

Свойство PublicKey позволяет получить значение открытого ключа сертификата.

```
HRESULT get_PublicKey(  
    FORMAT fmtFormat  
    BSTR* pVal  
);
```

#### Параметры

fmtFormat [in] – формат значения ключа. Принимает значения BASE64\_TYPE = 0 и DER\_TYPE = 1 из набора значений FORMAT.  
pVal [out, retval] – строка, содержащая значение открытого ключа сертификата.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PublicKey позволяет получить значение открытого ключа сертификата.

```
object.PublicKey [fmtFormat]
```

#### Параметры

Формат значения ключа. Принимает значения BASE64\_TYPE = 0 и DER\_TYPE = 1 из набора значений FORMAT.

#### Возвращаемое значение

Строка, содержащая значение открытого ключа сертификата.

#### IREQUESTCERTIFICATE::PUBLICKEYALG [C++]

#### REQUESTCERTIFICATE::PUBLICKEYALG [VBS]

[C++]

Свойство PublicKeyAlg позволяет получить алгоритм открытого ключа сертификата.

```
HRESULT get_PublicKeyAlg(  

```

```
BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] – строка, содержащая идентификатор алгоритма.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PublicKeyAlg позволяет получить алгоритм открытого ключа сертификата.

```
object.PublicKeyAlg
```

#### Возвращаемое значение

Строка, содержащая идентификатор алгоритма.

### IREQUESTCERTIFICATE::TEMPLATE [C++]

### REQUESTCERTIFICATE::TEMPLATE [VBS]

[C++]

Свойство Template позволяет получить/установить шаблон параметров запроса на сертификат.

```
HRESULT get_Template(  
    IRequestTemplate** pVal  
);  
HRESULT put_Template(  
    IRequestTemplate* pVal  
);
```

#### Параметры

pVal [out, retval] – указатель на интерфейс IRequestTemplate, используемый в качестве шаблона.

pVal [in] – указатель на интерфейс IRequestTemplate, используемый в качестве шаблона.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Template позволяет получить/установить шаблон параметров запроса на сертификат.

```
object.Template [= pVal]
```

#### Параметры

pVal – указатель на объект RequestTemplate, используемый в качестве шаблона запроса на сертификат.

### REQUESTTEMPLATE (GUID: AF777772-19ED-4D88-9E2D-C3DBF94B80CA)

#### Свойства

Свойство	Описание
С	Свойство позволяет получить/установить страну владельца сертификата

CAAddress	Свойство позволяет получить/установить строку с адресом УЦ, куда будет отправлен запрос
CACertificate	Свойство позволяет получить/установить корневой сертификат УЦ, куда будет отправлен запрос.
Certificate	Свойство позволяет заполнить шаблон запроса на сертификат данными из сертификата
CN	Свойство позволяет получить/установить общее имя владельца сертификата
CreateNewKeyset	Свойство позволяет получить/установить флаг, нужно ли создавать новый ключевой контейнер с новым ключевым набором
CryptoProvider	Свойство позволяет получить/установить имя криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то имя криптопровайдера, который управляет этим ключевым контейнером
CryptoProviderType	Свойство позволяет получить/установить тип криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то тип криптопровайдера, который управляет этим ключевым контейнером
DN	Свойство позволяет получить/установить отличительное имя сертификата
E	Свойство позволяет получить/установить e-mail владельца сертификата
ExtendedKeyUsage	Свойство позволяет получить/установить список идентификаторов расширенного использования открытого ключа сертификата (в виде XML строки)
Extentions	Свойство позволяет получить/установить коллекцию OID-ов расширений сертификата
KeyLength	Свойство позволяет получить/установить длину ключа
Keyset	Свойство позволяет получить/установить имя ключевого контейнера
KeySpec	Свойство позволяет получить/установить значение типа ключа.
KeyUsage	Свойство позволяет получить/установить использование ключа
L	Свойство позволяет получить/установить название города владельца сертификата
MarkKeysExportable	Свойство позволяет получить/установить флаг экспортируемости закрытого ключа
O	Свойство позволяет получить/установить название организации владельца сертификата
OU	Свойство позволяет получить/установить название отдела организации, к которому принадлежит владелец сертификата
RDNEntries	Свойство позволяет получить/установить относительное отличительное имя (RDN) сертификата
RDNEntryMaxLengths	Свойство позволяет получить/установить максимальный размер значения атрибута относительного отличительного имени (RDN) сертификата
S	Свойство позволяет получить/установить название региона владельца сертификата
ShowSendRequestWindow	Свойство позволяет получить/установить флаг отображения окна отправки запроса в УЦ
ValidFrom	Свойство позволяет получить/установить дату начала действия самоподписанного сертификата
ValidTo	Свойство позволяет получить/установить дату окончания действия самоподписанного сертификата

## Методы

Метод	Описание
LoadXMLTemplate	Метод позволяет загрузить данные для формирования запроса на сертификат из XML шаблона

**IREQUESTTEMPLATE::LOADXMLTEMPLATE [C++]****REQUESTTEMPLATE::LOADXMLTEMPLATE [VBS]****[C++]**

Метод LoadXMLTemplate позволяет загрузить данные, необходимые для формирования запроса на сертификат, из XML шаблона.

```
HRESULT LoadXMLTemplate(  
    BSTR bstrFilename  
);
```

**Параметры**

bstrFilename  
[in] XML строка с шаблоном запроса на сертификат.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод LoadXMLTemplate позволяет загрузить данные, необходимые для формирования запроса на сертификат, из XML шаблона.

```
object.LoadXMLTemplate [ Value]
```

**Параметры**

Value - XML строка с шаблоном запроса на сертификат.

**IREQUESTTEMPLATE::C [C++]****REQUESTTEMPLATE::C [VBS]****[C++]**

Свойство C позволяет получить/установить страну владельца сертификата.

```
HRESULT get_C(  
    BSTR* pVal  
);  
HRESULT put_C(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку.  
newVal [in] - строка с именем страны владельца сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство C позволяет получить/установить страну владельца сертификата.

```
object.C [= Value]
```

**Параметры**

Value - строка с именем страны владельца сертификата.

**IREQUESTTEMPLATE::CAADDRESS [C++]****REQUESTTEMPLATE::CAADDRESS [VBS]****[C++]**

Свойство CAAddress позволяет получить/установить строку с адресом УЦ, куда будет отправлен запрос.

```
HRESULT get_CAAddress(  
    BSTR* pVal  
);  
HRESULT put_CAAddress(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на строку с адресом УЦ.  
newVal [in] - устанавливает строку с адресом УЦ

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство CAAddress позволяет получить/установить строку с адресом УЦ, куда будет отправлен запрос.

```
object.CAAddress [= pVal]
```

**Параметры**

pVal - строка с адресом УЦ.

**IREQUESTTEMPLATE::CACERTIFICATE [C++]****REQUESTTEMPLATE::CACERTIFICATE [VBS]****[C++]**

Свойство CACertificate позволяет получить/установить корневой сертификат УЦ, куда будет отправлен запрос.

```
HRESULT get_CACertificate(  
    ICertificate** pVal  
);  
HRESULT put_CACertificate(  
    ICertificate* newVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на указатель на интерфейс ICertificate  
newVal [in] - устанавливает указатель на интерфейс ICertificate

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство CACertificate позволяет получить/установить корневой сертификат УЦ, куда будет отправлен запрос.

```
object.CACertificate [= pVal]
```

### Параметры

pVal - указатель на объект Certificate.

## IREQUESTTEMPLATE::CERTIFICATE [C++]

## REQUESTTEMPLATE::CERTIFICATE [VBS]

[C++]

Свойство Certificate позволяет заполнить шаблон запроса на сертификат данными из сертификата.

```
HRESULT put_Certificate(  
    ICertificate* pICertificate  
);
```

### Параметры

pICertificate [in] - указатель на интерфейс ICertificate.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Certificate позволяет заполнить шаблон запроса на сертификат данными из сертификата.

```
object.Certificate [pICertificate]
```

### Параметры

pICertificate - указатель на объект Certificate.

## IREQUESTTEMPLATE::CN [C++]

## REQUESTTEMPLATE::CN [VBS]

[C++]

Свойство CN позволяет получить/установить общее имя владельца сертификата.

```
HRESULT get_CN(  
    BSTR* pVal  
);  
HRESULT put_CN(  
    BSTR newVal  
);
```

### Параметры

pVal [out, retval] - указатель на строку.

newVal [in] - строка с общим именем владельца сертификата.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CN позволяет получить/установить общее имя владельца сертификата.

```
object.CN [= Value]
```



## Параметры

Value - строка с общим именем владельца сертификата.

## IREQUESTTEMPLATE::CREATENewKEYSET [C++]

## REQUESTTEMPLATE::CREATENewKEYSET [VBS]

[C++]

Свойство CreateNewKeyset позволяет получить/установить флаг, нужно ли создавать новый ключевой контейнер с новым ключевым набором.

```
HRESULT get_CreateNewKeyset(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_CreateNewKeyset(  
    VARIANT_BOOL newVal  
);
```

## Параметры

pVal [out, retval] - указатель на строку.

newVal [in] - True - формировать новый ключевой набор; False - использовать существующий ключевой набор.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CreateNewKeyset позволяет получить/установить флаг, нужно ли создавать новый ключевой контейнер с новым ключевым набором.

```
object.CreateNewKeyset [= Value]
```

## Параметры

Value - True - формировать новый ключевой набор; False - использовать существующий ключевой набор.

## IREQUESTTEMPLATE::CRYPTOProvider [C++]

## REQUESTTEMPLATE::CRYPTOProvider [VBS]

[C++]

Свойство CryptoProvider позволяет получить/установить имя криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то имя криптопровайдера, который управляет этим ключевым контейнером.

```
HRESULT get_CryptoProvider(  
    BSTR* pVal  
);  
HRESULT put_CryptoProvider(  
    BSTR newVal  
);
```

## Параметры

pVal [out, retval] - указатель на строку.

newVal [in] - строка с именем криптопровайдера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CryptoProvider позволяет получить/установить имя криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то имя криптопровайдера, который управляет этим ключевым контейнером.

```
object.CryptoProvider [= Value]
```

**Параметры**

Value - строка с именем криптопровайдера.

**IREQUESTTEMPLATE::CRYPTOPROVIDERTYPE [C++]****REQUESTTEMPLATE::CRYPTOPROVIDERTYPE [VBS]**

[C++]

Свойство CryptoProviderType позволяет получить/установить тип криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то тип криптопровайдера, который управляет этим ключевым контейнером.

```
HRESULT get_CryptoProviderType(  
    LONG* pVal  
);  
HRESULT put_CryptoProviderType(  
    LONG newVal  
);
```

**Параметры**

pVal [out, retval] - тип криптопровайдера.  
newVal [in] - тип криптопровайдера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CryptoProviderType позволяет получить/установить имя криптопровайдера, который будет использоваться для генерации новой ключевой пары, или, если указан существующий контейнер, то имя криптопровайдера, который управляет этим ключевым контейнером.

```
object.CryptoProviderType [= Value]
```

**Параметры**

Value - тип криптопровайдера.

**IREQUESTTEMPLATE::DN [C++]****REQUESTTEMPLATE::DN [VBS]**

[C++]

Свойство DN позволяет получить/установить отличительное имя сертификата.

```
HRESULT get_DN(  
    BSTR* pVal
```

```
);  
HRESULT put_DN(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - строка, содержащая отличительное имя сертификата

newVal [in] - устанавливаемое значение - строка, содержащая отличительное имя сертификата.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство DN позволяет получить/установить строку, содержащую отличительное имя сертификата.

```
object.DN [= Value]
```

#### Параметры

Value - принимаемые значения - строка, содержащая отличительное имя сертификата.

### IREQUESTTEMPLATE::E [C++]

### REQUESTTEMPLATE::E [VBS]

[C++]

Свойство E позволяет получить/установить e-mail владельца сертификата.

```
HRESULT get_E(  
    BSTR* pVal  
);  
HRESULT put_E(  
    BSTR newVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку.

newVal [in] - строка с e-mail адресом владельца сертификата.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство E позволяет получить/установить e-mail владельца сертификата.

```
object.E [= Value]
```

#### Параметры

Value - строка с e-mail адресом владельца сертификата.

### IREQUESTTEMPLATE::EXTENDEDKEYUSAGE [C++]

### REQUESTTEMPLATE::EXTENDEDKEYUSAGE [VBS]

**[C++]**

Свойство ExtendedKeyUsage позволяет получить/установить список идентификаторов расширенного использования открытого ключа сертификата (в виде XML строки).

```
HRESULT get_ExtendedKeyUsage(  
    BSTR* pVal  
);  
HRESULT put_ExtendedKeyUsage(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - строка, содержащая список объектных идентификаторов расширенного использования открытого ключа в XML виде.  
newVal [in] - строка с именем.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ExtendedKeyUsage позволяет получить/установить список идентификаторов расширенного использования открытого ключа сертификата (в виде XML строки).

```
object.ExtendedKeyUsage [= Value]
```

**Параметры**

Value - строка, содержащая список объектных идентификаторов расширенного использования открытого ключа в XML виде.

**IREQUESTTEMPLATE::EXTENTIONS [C++]****REQUESTTEMPLATE::EXTENTIONS [VBS]****[C++]**

Свойство Extentions позволяет получить/установить расширения сертификата.

```
HRESULT get_Extentions(  
    IOIDParams** pVal  
);  
HRESULT put_Extentions(  
    IOIDParams* newVal  
);
```

**Параметры**

pVal [out, retval] - коллекция OID-ов расширений сертификатов.  
newVal [in] - коллекция OID-ов расширений сертификатов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство Extentions позволяет получить/установить расширения сертификата.

```
object.Extentions [= Value]
```

**Параметры**

Value - коллекция OID-ов расширений сертификатов.

**IREQUESTTEMPLATE::KEYLENGTH [C++]****REQUESTTEMPLATE::KEYLENGTH [VBS]**

[C++]

Свойство KeyLength позволяет получить/установить длину ключа.

```
HRESULT get_KeyLength(  
    long* pVal  
);  
HRESULT put_KeyLength(  
    long pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - длина ключа.  
pVal [in] - устанавливаемое значение - длина ключа.

**Примечание**

Длина ключа измеряется в битах. Например, если установить свойство KeyLength равным 512, то будет создан 512-ти-битный ключ, если выбранный криптопровайдер поддерживает такую длину ключа. Если при генерации сертификата будет указана длина ключа, не поддерживаемая выбранным криптопровайдером, то возникнет ошибка генерации сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство KeyLength позволяет получить/установить длину ключа.

```
object.KeyLength [= Value]
```

**Параметры**

Value - принимаемое значение - длина ключа.

**Примечание**

Длина ключа измеряется в битах. Например, если установить свойство KeyLength равным 512, то будет создан 512-ти-битный ключ, если выбранный криптопровайдер поддерживает такую длину ключа. Если при генерации сертификата будет указана длина ключа, не поддерживаемая выбранным криптопровайдером, то возникнет ошибка генерации сертификата.

**IREQUESTTEMPLATE::KEYSET [C++]****REQUESTTEMPLATE::KEYSET [VBS]**

[C++]

Свойство Keyset позволяет получить/установить имя ключевого контейнера.

```
HRESULT get_Keyset(  
    BSTR* pVal  
);  
HRESULT put_Keyset(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку.  
newVal [in] - строка с именем ключевого контейнера.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Keyset позволяет получить/установить имя ключевого контейнера.

object.Keyset [= Value]

### Параметры

Value - строка с именем ключевого контейнера.

## IREQUESTTEMPLATE::KEYSPEC [C++]

### REQUESTTEMPLATE::KEYSPEC [VBS]

[C++]

Свойство KeySpec позволяет получить/установить тип ключа.

```
HRESULT get_KeySpec(  
    long* pVal  
);  
HRESULT put_KeySpec(  
    long newVal  
);
```

### Параметры

pVal [out, retval] - возвращает тип ключа: 1- Шифрование, Шифрование и подпись; 2- Подпись.  
newVal [in] - устанавливает тип ключа: 1- Шифрование; 2- Подпись; 3- Шифрование и подпись.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство KeySpec позволяет получить/установить тип ключа.

object.KeySpec [= pVal]

### Параметры

pVal - тип ключа: 1- Шифрование; 2- Подпись; 3- Шифрование и подпись.

### Возвращаемые значения

Возвращаемое значение - тип ключа: 1- Шифрование, Шифрование и подпись; 2- Подпись;

## IREQUESTTEMPLATE::KEYUSAGE [C++]

### REQUESTTEMPLATE::KEYUSAGE [VBS]

[C++]

Свойство KeyUsage позволяет получить/установить использование ключа.

```
HRESULT get_KeyUsage (
    long* pVal
);
HRESULT put_KeyUsage (
    long pVal
);
```

### Параметры

pVal [out, retval] - возвращает сумму битов использования ключа:

Значение	Описание
CERT_DIGITAL_SIGNATURE_KEY_USAGE = 0x80	Цифровая подпись
CERT_NON_REPUDIATION_KEY_USAGE = 0x40	Неотрекаемость
CERT_KEY_ENCIPHERMENT_KEY_USAGE = 0x20	Шифрование ключа
CERT_DATA_ENCIPHERMENT_KEY_USAGE = 0x10	Шифрование данных
CERT_KEY_AGREEMENT_KEY_USAGE = 0x08	Согласование ключей
CERT_KEY_CERT_SIGN_KEY_USAGE = 0x04	Подпись ключа сертификата
CERT_CRL_SIGN_KEY_USAGE = 0x02	Автономное подписание списков отзыва
CERT_ENCIPHER_ONLY_KEY_USAGE = 0x01	Только шифрование

pVal [in] - устанавливает использование ключа:

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство KeyUsage позволяет получить/установить использование ключа.

```
object.KeyUsage [= pVal]
```

### Параметры

pVal - использование ключа:

Значение	Описание
CERT_DIGITAL_SIGNATURE_KEY_USAGE = &H0x80	Цифровая подпись
CERT_NON_REPUDIATION_KEY_USAGE = &H0x40	Неотрекаемость
CERT_KEY_ENCIPHERMENT_KEY_USAGE = &H0x20	Шифрование ключа
CERT_DATA_ENCIPHERMENT_KEY_USAGE = &H0x10	Шифрование данных
CERT_KEY_AGREEMENT_KEY_USAGE = &H0x08	Согласование ключей
CERT_KEY_CERT_SIGN_KEY_USAGE = &H0x04	Подпись ключа сертификата
CERT_CRL_SIGN_KEY_USAGE = &H0x02	Автономное подписание списков отзыва
CERT_ENCIPHER_ONLY_KEY_USAGE = &H0x01	Только шифрование

IREQUESTTEMPLATE::L [C++]

REQUESTTEMPLATE::L [VBS]

[C++]

Свойство L позволяет получить/установить название города владельца сертификата.

```
HRESULT get_L(  
    BSTR* pVal  
);  
HRESULT put_L(  
    BSTR newVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку.

newVal [in] - строка с названием города владельца сертификата.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство L позволяет получить/установить название города владельца сертификата.

```
object.L [= Value]
```

#### Параметры

Value - строка с названием города владельца сертификата.

### IREQUESTTEMPLATE::MARKKEYSEXPORTABLE [C++]

### REQUESTTEMPLATE::MARKKEYSEXPORTABLE [VBS]

[C++]

Свойство MarkKeysExportable позволяет получить/установить флаг экспортируемости закрытого ключа.

```
HRESULT get_MarkKeysExportable(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_MarkKeysExportable(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - пометить закрытый ключ как экспортируемый, FALSE - не помечать.

pVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - пометить закрытый ключ как экспортируемый, FALSE - не помечать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство MarkKeysExportable позволяет получить/установить флаг экспортируемости закрытого ключа.

```
object.MarkKeysExportable [= Value]
```

#### Параметры

Value - принимаемые значения TRUE или FALSE. TRUE - пометить закрытый ключ как экспортируемый, FALSE - не помечать.



**IREQUESTTEMPLATE::O [C++]****REQUESTTEMPLATE::O [VBS]****[C++]**

Свойство O позволяет получить/установить название организации владельца сертификата.

```
HRESULT get_O(  
    BSTR* pVal  
);  
HRESULT put_O(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку.

newVal [in] - строка с названием организации владельца сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство O позволяет получить/установить название организации владельца сертификата.

```
object.O [= Value]
```

**Параметры**

Value - строка с названием организации владельца сертификата.

**IREQUESTTEMPLATE::OU [C++]****REQUESTTEMPLATE::OU [VBS]****[C++]**

Свойство OU позволяет получить/установить название отдела организации, к которому принадлежит владелец сертификата.

```
HRESULT get_OU(  
    BSTR* pVal  
);  
HRESULT put_OU(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку.

newVal [in] - строка с названием отдела организации, к которому принадлежит владелец сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство OU позволяет получить/установить название отдела организации, к которому принадлежит владелец сертификата.

```
object.OU [= Value]
```

## Параметры

Value - строка с названием отдела организации, к которому принадлежит владелец сертификата.

## IREQUESTTEMPLATE::RDNENTRIES [C++]

## REQUESTTEMPLATE::RDNENTRIES [VBS]

[C++]

Свойство RDNEntries позволяет получить/установить относительное отличительное имя (RDN) сертификата.

```
HRESULT get_RDNEntries(  
    IOIDParams** pVal  
);  
HRESULT put_RDNEntries(  
    IOIDParams* newVal  
);
```

## Параметры

pVal [out, retval] - возвращает указатель на указатель на интерфейс IOIDParams, в котором содержится коллекция атрибутов RDN.  
newVal [in] - устанавливает указатель на интерфейс IOIDParams, в котором содержится коллекция атрибутов RDN.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RDNEntries позволяет получить/установить относительное отличительное имя (RDN) сертификата.

```
object.RDNEntries [= pVal]
```

## Параметры

pVal - устанавливаемое значение - указатель на объект IOIDParams, в котором содержится коллекция атрибутов RDN.

## Возвращаемые значения

Возвращаемое значение - указатель на указатель на объект IOIDParams, в котором содержится коллекция атрибутов RDN.

## IREQUESTTEMPLATE::RDNENTRYMAXLENGTHS [C++]

## REQUESTTEMPLATE::RDNENTRYMAXLENGTHS [VBS]

[C++]

Свойство RDNEntryMaxLengths позволяет получить/установить максимальный размер значения атрибута отличительного имени сертификата.

```
HRESULT get_RDNEntryMaxLengths(  
    IOIDParams** pVal  
);  
HRESULT put_RDNEntryMaxLengths(  
    IOIDParams* newVal  
);
```

**Параметры**

pVal [out, retval] - коллекция IOIDParams.  
newVal [in] - коллекция IOIDParams.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RDNEntryMaxLengths позволяет получить/установить максимальный размер значения атрибута отличительного имени сертификата.

```
object.RDNEntryMaxLengths [= Value]
```

**Параметры**

Value - коллекция OIDParams.

**IREQUESTTEMPLATE::S [C++]****REQUESTTEMPLATE::S [VBS]**

[C++]

Свойство S позволяет получить/установить название региона владельца сертификата.

```
HRESULT get_S(  
    BSTR* pVal  
);  
HRESULT put_S(  
    BSTR newVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку.  
newVal [in] - строка с названием региона владельца сертификата.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство S позволяет получить/установить название региона владельца сертификата.

```
object.S [= Value]
```

**Параметры**

Value - строка с названием региона владельца сертификата.

**IREQUESTTEMPLATE::SHOWSENDREQUESTWINDOW [C++]****REQUESTTEMPLATE::SHOWSENDREQUESTWINDOW [VBS]**

[C++]

Свойство ShowSendRequestWindow позволяет получить/установить флаг отображения окна отправки запроса в УЦ.

```
HRESULT get_ShowSendRequestWindow(  
    VARIANT* pVal  
);
```

```
HRESULT put_ShowSendRequestWindow(  
    VARIANT newVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - отображать окно отправки запроса в УЦ, FALSE - не отображать.  
newVal [in] - устанавливаемое значение TRUE или FALSE. TRUE - отображать окно отправки запроса в УЦ, FALSE - не отображать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ShowSendRequestWindow позволяет получить/установить флаг отображения окна отправки запроса в УЦ.

```
object.ShowSendRequestWindow [= Value]
```

### Параметры

Value - принимаемые значения TRUE или FALSE. TRUE - отображать окно отправки запроса в УЦ, FALSE - не отображать.

## IREQUESTTEMPLATE::VALIDFROM [C++]

## REQUESTTEMPLATE::VALIDFROM [VBS]

[C++]

Свойство ValidFrom позволяет получить/установить дату начала действия самоподписанного сертификата.

```
HRESULT get_ValidFrom(  
    DATE* pVal  
);  
HRESULT put_ValidFrom(  
    DATE newVal  
);
```

### Параметры

pVal [out, retval] - возвращает указатель на дату начала действия сертификата  
newVal [in] - устанавливает дату начала действия сертификата

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ValidFrom позволяет получить/установить дату начала действия самоподписанного сертификата.

```
object.ValidFrom [= pVal]
```

### Параметры

pVal - дата начала действия сертификата.

## IREQUESTTEMPLATE::VALIDTO [C++]

**REQUESTTEMPLATE::VALIDTO [VBS]****[C++]**

Свойство ValidTo позволяет получить/установить дату окончания действия самоподписанного сертификата.

```
HRESULT get_ValidTo(  
    DATE* pVal  
);  
HRESULT put_ValidTo(  
    DATE newVal  
);
```

**Параметры**

pVal [out, retval] - возвращает указатель на дату окончания действия сертификата  
newVal [in] - устанавливает дату окончания действия сертификата

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ValidTo позволяет получить/установить дату окончания действия самоподписанного сертификата.

```
object.ValidTo [= pVal]
```

**Параметры**

pVal - дата окончания действия сертификата.

**SERVICEPROFILE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Свойство	Описание
Display	Метод позволяет просмотреть свойства профиля для настройки доступа к Веб-сервису

**Свойства**

Свойство	Описание
ClientCertificate	Свойство позволяет получить/установить сертификат для клиентской аутентификации
ID	Свойство позволяет получить/установить идентификатор профиля для доступа к Веб-сервису
Password	Свойство позволяет получить/установить пароль зарегистрированного пользователя Веб-сервиса
ProxyAddress	Свойство позволяет получить/установить адрес прокси-сервера
ProxyAuthType	Свойство позволяет получить/установить тип аутентификации на прокси-сервере
ProxyPassword	Свойство позволяет получить/установить пароль для аутентификации на прокси-сервере
ProxyUserName	Свойство позволяет получить/установить имя пользователя для аутентификации на прокси-сервере
ServiceAddress	Свойство позволяет получить/установить адрес Веб-сервиса

ServiceAuthType	Свойство позволяет получить/установить тип аутентификации доступа к Веб-сервису
UseClientCertificate	Свойство позволяет получить/установить флаг использования клиентского сертификата для аутентификации
UseProxy	Свойство позволяет получить/установить флаг использования прокси-сервера для обращения к Веб-сервису
UserName	Свойство позволяет получить/установить логин зарегистрированного пользователя Веб-сервиса

## ISERVICEPROFILE::DISPLAY [C++]

## SERVICEPROFILE::DISPLAY [VBS]

[C++]

Метод Display позволяет просмотреть параметры профиля для настройки доступа к Веб-сервису.

```
HRESULT Display(  
    VARIANT_BOOL* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает TRUE, если в окне нажали кнопку "ОК" или "Применить" и FALSE, если была нажата "Отмена".

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display позволяет просмотреть параметры профиля для настройки доступа к Веб-сервису.

```
object.Display ()
```

### Возвращаемое значение

Возвращает TRUE, если в окне нажали кнопку "ОК" или "Применить" и FALSE, если была нажата "Отмена".

## SERVICEPROFILE::CLIENTCERTIFICATE [C++]

## SERVICEPROFILE::CLIENTCERTIFICATE [VBS]

[C++]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище.

```
HRESULT get_ClientCertificate(  
    BSTR* pVal  
);  
HRESULT put_ClientCertificate(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] - указатель на строку с сертификатом клиентской аутентификации.

pVal [in] - строка с сертификатом клиентской аутентификации.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище.

```
object.ClientCertificate [= Value]
```

**Параметры**

Value - строка с сертификатом клиентской аутентификации.

**SERVICEPROFILE::ID [C++]****SERVICEPROFILE::ID [VBS]**

[C++]

Свойство ID позволяет получить/установить идентификатор профиля.

```
HRESULT get_ID(  
    BSTR* pVal  
);  
HRESULT put_ID(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку идентификатора профиля.  
pVal [in] - строка идентификатора профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ID позволяет получить/установить идентификатор профиля.

```
object.ID [= Value]
```

**Параметры**

Value - строка идентификатора профиля.

**SERVICEPROFILE::PASSWORD [C++]****SERVICEPROFILE::PASSWORD [VBS]**

[C++]

Свойство Password позволяет получить/установить пароль зарегистрированного пользователя Веб-сервиса.

```
HRESULT get_Password(  
    BSTR* pVal  
);  
HRESULT put_Password(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку пароля зарегистрированного пользователя Веб-сервиса.

pVal [in] - строка пароля зарегистрированного пользователя Веб-сервиса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Password позволяет получить/установить пароль зарегистрированного пользователя Веб-сервиса.

object.Password [= Value]

**Параметры**

Value - строка пароля зарегистрированного пользователя Веб-сервиса.

**SERVICEPROFILE::PROXYADDRESS [C++]****SERVICEPROFILE::PROXYADDRESS [VBS]**

[C++]

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

```
HRESULT get_ProxyAddress(  
    BSTR* pVal  
);  
HRESULT put_ProxyAddress(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку с адресом прокси-сервера.

pVal [in] - строка с адресом прокси-сервера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

object.ProxyAddress [= Value]

**Параметры**

Value - строка с адресом прокси-сервера.

**SERVICEPROFILE::PROXYAUTHTYPE [C++]****SERVICEPROFILE::PROXYAUTHTYPE [VBS]**

[C++]

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

```
HRESULT get_ProxyAuthType(  
    SERVICE_AUTH_TYPE* pVal  
);
```



```
HRESULT put_ProxyAuthType(  
    SERVICE_AUTH_TYPE pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - тип аутентификации. Допустимыми значениями являются значения SERVICE\_AUTH\_TYPE.  
pVal [in] - тип аутентификации. Допустимыми значениями являются значения SERVICE\_AUTH\_TYPE.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

```
object.ProxyAuthType [= Value]
```

#### Параметры

Value - тип аутентификации. Допустимыми значениями являются значения SERVICE\_AUTH\_TYPE.

### SERVICEPROFILE::PROXYPASSWORD [C++]

### SERVICEPROFILE::PROXYPASSWORD [VBS]

[C++]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
HRESULT get_ProxyPassword(  
    BSTR* pVal  
);  
HRESULT put_ProxyPassword(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку пароля для аутентификации на прокси-сервере.  
pVal - [in] строка пароля для аутентификации на прокси-сервере.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
object.ProxyPassword [= Value]
```

#### Параметры

Value - строка пароля для аутентификации на прокси-сервере.

### SERVICEPROFILE::PROXYUSERNAME [C++]

### SERVICEPROFILE::PROXYUSERNAME [VBS]

**[C++]**

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере .

```
HRESULT get_ProxyUserName(  
    BSTR* pVal  
);  
HRESULT put_ProxyUserName(  
    BSTR pVal  
);
```

**Параметры**

pVal  
[out, retval] Указатель на строку с именем пользователя для аутентификации на прокси-сервере.  
pVal  
[in] Строка с именем пользователя для аутентификации на прокси-сервере.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
object.ProxyUserName [= Value]
```

**Параметры**

Value - строка с именем пользователя для аутентификации на прокси-сервере.

**SERVICEPROFILE::SERVICEADDRESS [C++]****SERVICEPROFILE::SERVICEADDRESS [VBS]****[C++]**

Свойство **ServiceAddress** позволяет получить/установить адрес Веб-службы.

```
HRESULT get_ServiceAddress(  
    BSTR* pVal  
);  
HRESULT put_ServiceAddress(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку адреса Веб-службы.  
pVal [in] - строка адреса Веб-службы.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ServiceAddress позволяет получить/установить адрес Веб-службы.

```
object.ServiceAddress [= Value]
```

**Параметры**

Value - строка адреса Веб-службы.

## SERVICEPROFILE::SERVICEAUTHTYPE [C++]

## SERVICEPROFILE::SERVICEAUTHTYPE [VBS]

[C++]

Свойство ServiceAuthType позволяет получить/установить тип аутентификации доступа к Службе Штатпов Времени.

```
HRESULT get_ServiceAuthType(  
    SERVICE_AUTH_TYPE* pVal  
);  
HRESULT put_ServiceAuthType(  
    SERVICE_AUTH_TYPE pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение - тип аутентификации доступа к Службе Штатпов Времени. Допустимыми значениями являются значения SERVICE\_AUTH\_TYPE.

pVal [in] - тип аутентификации доступа к Службе Штатпов Времени. Допустимыми значениями являются значения SERVICE\_AUTH\_TYPE.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ServiceAuthType позволяет получить/установить тип аутентификации доступа к Службе Штатпов Времени.

```
object.ServiceAuthType [= Value]
```

### Параметры

Value - тип аутентификации доступа к Службе. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.

## SERVICEPROFILE::USECLIENTCERTIFICATE [C++]

## SERVICEPROFILE::USECLIENTCERTIFICATE [VBS]

[C++]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
HRESULT get_UseClientCertificate(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseClientCertificate(  
    VARIANT_BOOL pVal  
);
```

### Параметры

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

pVal [in] - принимает TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
object.UseClientCertificate [= Value]
```

**Параметры**

Value - TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

**SERVICEPROFILE::USEPROXY [C++]****SERVICEPROFILE::USEPROXY [VBS]**

[C++]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Веб-сервису.

```
HRESULT get_UseProxy(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseProxy(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.

pVal [in] - принимает TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Веб-сервису.

```
object.UseProxy [= Value]
```

**Параметры**

Value - TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.

**SERVICEPROFILE::USERNAME [C++]****SERVICEPROFILE::USERNAME [VBS]**

[C++]

Свойство UserName позволяет получить/установить логин зарегистрированного пользователя Веб-сервиса.

```
HRESULT get_UserName(  

```

```
VARIANT_BOOL* pVal  
);  
HRESULT put_UserName(  
    VARIANT_BOOL pVal  
);
```

### Параметры

pVal [out, retval] - строка логина зарегистрированного пользователя Веб-сервиса.

pVal [in] - указатель на строку логина зарегистрированного пользователя Веб-сервиса.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UserName позволяет получить/установить логин зарегистрированного пользователя Веб-сервиса.

```
object.UserName [= Value]
```

### Параметры

Value - строка логина зарегистрированного пользователя Веб-сервиса.

## SIGNATURE (GUID: 1C5E0721-814B-456A-9208-35014DFD342D)

### Методы

Метод	Описание
Display	Метод отображает окно с детальной информацией данной подписи
Export	Метод позволяет произвести экспорт подписи в строку
Verify	Метод позволяет проверить корректность подписи

### Свойства

Свойство	Описание
Certificate	Свойство позволяет получить сертификат подписчика
CertificateValues	Свойство позволяет получить сертификат подписи с цепочкой сертификации в случае улучшенной подписи (CAvES-X Long).
CMSVersion	Свойство позволяет получить версию протокола CMS
Comments	Свойство позволяет получить комментарии к подписи
ContentType	Свойство позволяет получить тип сообщения
Cosignature	Свойство позволяет получить коллекцию встречных подписей для данной ЭЦП
Detached	Свойство позволяет получить тип подписи – отделенная или совмещенная с данными
HashAlg	Свойство позволяет получить алгоритм хэширования подписанных данных
HashEncAlg	Свойство позволяет получить алгоритм шифрования хэша подписанных данных
Resource	Свойство позволяет получить URI ресурса, где расположены подписанные данные
RevocationCRLValue	Свойство позволяет получить из улучшенной подписи (CAvES-X Long) коллекцию СОС, содержащихся в доказательствах подлинности.
RevocationOCSPValue	Свойство позволяет получить из улучшенной подписи (CAvES-X Long) коллекцию OCSP ответов, содержащихся в доказательствах подлинности.
RevocationValue	Свойство позволяет получить из улучшенной подписи (CAvES-X Long) информацию об отзыве сертификата в виде OCSP ответа или СОС.

SignUsage	Свойство позволяет получить OID использования подписи в виде строки.
SignatureType	Свойство позволяет получить тип подписи (обычная или улучшенная CAdES-X Long)
SignerIndex	Свойство позволяет получить индекс подписчика в CMS сообщении
SigningTime	Свойство позволяет получить дату и время подписи
TimeStamp	Свойство позволяет получить штамп времени подписи.
TSPResponse	Свойство позволяет получить штамп времени подписи, предоставленный на подписываемые данные.

## ISignature::Display [C++]

## Signature::Display [VBS]

[C++]

Метод Display отображает окно с детальной информацией данной подписи.

```
HRESULT Display(  
    BSTR bstrDetachedSourceURI  
);
```

### Параметры

bstrDetachedSourceURI [in, optional, defaultvalue("")] - имя файла исходных данных в случае отделенной подписи.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Display отображает окно с детальной информацией данной подписи.

```
object.Display(bstrDetachedSourceURI)
```

### Параметры

bstrDetachedSourceURI - имя файла исходных данных в случае отделенной подписи.

## ISignature::Export [C++]

## Signature::Export [VBS]

[C++]

Метод Export позволяет произвести экспорт подписи в строку.

```
HRESULT Export(  
    FORMAT fFormat,  
    BSTR bstrSourceURI,  
    BSTR* pbstrSignature  
);
```

### Параметры

fFormat [in, defaultvalue(XML\_TYPE)] - формат, в котором будет экспортирована подпись. Принимает значения из FORMAT.  
bstrSourceURI [in, defaultvalue("")] - имя файла исходных данных, если экспортируемая подпись является отделенной.

pbstrSignatures [out, retval] - указатель на буфер в памяти, в который экспортируются подпись.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет произвести экспорт подписи в строку.

```
object.Export(fFormat[,bstrSourceURI])
```

### Параметры

fFormat - формат, в котором будет экспортирована подпись. Принимает значения из FORMAT.

bstrSourceURI - имя файла исходных данных, если экспортируемая подпись является отдельной.

### Возвращаемое значение

Указатель на буфер в памяти, в который экспортируются подпись.

## ISIGNATURE::VERIFY [C++]

### SIGNATURE::VERIFY [VBS]

[C++]

Метод Verify позволяет проверить корректность подписи.

```
HRESULT Verify(  
    VERIFYFLAG vfFlag,  
    BSTR bstrDetachedSourceURI,  
    VERIFYSTATUS* vsStatus  
);
```

### Параметры

vfFlag [in] - параметры проверки: только подписи или подписи и сертификата подписчика. Допустимыми значениями являются значения VERIFYFLAG.

bstrDetachedSourceURI [in, optional, defaultvalue("")] - имя подписанного файла в случае отдельной подписи.

vsStatus [out, retval] - статус завершения проверки. Допустимыми значениями являются значения VERIFYSTATUS.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Verify позволяет проверить корректность подписи.

```
object.Verify(vfFlag, bstrDetachedSourceURI)
```

### Параметры

vfFlag - параметры проверки: только подписи или подписи и сертификата подписчика. Допустимыми значениями являются значения VERIFYFLAG.

bstrDetachedSourceURI - имя подписанного файла в случае отдельной подписи.

### Возвращаемое значение

Статус завершения проверки. Допустимыми значениями являются значения VERIFYSTATUS.

**ISignature::Certificate [C++]****Signature::Certificate [VBS]**

[C++]

Свойство Certificate позволяет получить сертификат подписчика.

```
HRESULT get_Certificate(  
    ICertificate** pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на интерфейс ICertificate.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Certificate позволяет получить сертификат подписчика.

```
object.Certificate
```

**Возвращаемое значение**

Указатель на объект Certificate.

**ISignature::CertificateValues [C++]****Signature::CertificateValues [VBS]**

[C++]

Свойство CertificateValues позволяет получить сертификат подписи с цепочкой сертификации в случае улучшенной подписи (CADES-X Long).

```
HRESULT get_CertificateValues(  
    ICertificates** pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - интерфейс ICertificates.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CertificateValues позволяет получить сертификат подписи с цепочкой сертификации в случае улучшенной подписи (CADES-X Long).

```
object.CertificateValues ()
```

**Возвращаемое значение**

Возвращаемое значение - объект Certificates.

**ISignature::CMSVersion [C++]****Signature::CMSVersion [VBS]**



[C++]

Свойство CMSVersion позволяет получить версию протокола CMS.

```
HRESULT get_CMSVersion(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - номер версии.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство CMSVersion позволяет получить версию протокола CMS.

```
object.CMSVersion
```

#### Возвращаемое значение

Номер версии.

### ISIGNATURE::COMMENTS [C++]

### SIGNATURE::COMMENTS [VBS]

[C++]

Свойство Comments позволяет получить комментарии к подписи.

```
HRESULT get_Comments(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] - комментарий в виде строки.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Comments позволяет получить комментарии к подписи.

```
object.Comments
```

#### Возвращаемое значение

Комментарий в виде строки.

### ISIGNATURE::CONTENTTYPE [C++]

### SIGNATURE::CONTENTTYPE [VBS]

[C++]

Свойство ContentType позволяет получить тип сообщения.

```
HRESULT get_ContentType(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] – идентификатор типа сообщения. Возможные значения " 1.2.840.113549.1.9.3 " – шифрованные данные, " 1.2.840.113549.1.9.2 " – подпись.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ContentType позволяет получить тип сообщения.

object.ContentType

### Возвращаемое значение

Идентификатор типа сообщения. Возможные значения " 1.2.840.113549.1.9.3 " – шифрованные данные, " 1.2.840.113549.1.9.2 " – подпись.

## ISignature::Cosignature [C++]

### SIGNATURE::COSIGNATURE [VBS]

[C++]

Свойство Cosignature позволяет получить коллекцию встречных подписей для данной ЭЦП.

```
HRESULT get_Cosignature(  
    ISignatures** pVal  
);
```

### Параметры

pVal [out, retval] – возвращает указатель на интерфейс ISignatures, который является коллекцией интерфейсов ISignature.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Cosignature позволяет получить коллекцию встречных подписей для данной ЭЦП.

object.Cosignature

### Возвращаемое значение

Возвращает указатель на объектов Signatures, который является коллекцией объектов Signature.

## ISignature::Detached [C++]

### SIGNATURE::DETACHED [VBS]

[C++]

Свойство Detached позволяет получить тип подписи – отделенная или совмещенная с данными.

```
HRESULT get_Detached(  
    VARIANT_BOOL* pVal  
);
```

### Параметры

pVal [out, retval] - возвращаемое значение TRUE или FALSE. TRUE - отдельная подпись, FALSE - подпись, совмещенная с данными.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Detached позволяет получить тип подписи – отдельная или совмещенная с данными.

object.Detached

### Возвращаемое значение

Возвращаемое значение TRUE или FALSE. TRUE - отдельная подпись, FALSE - подпись, совмещенная с данными.

## ISignature::HashAlg [C++]

### SIGNATURE::HASHALG [VBS]

[C++]

Свойство HashAlg позволяет получить алгоритм хэширования подписанных данных.

```
HRESULT get_HashAlg(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает указатель на строку, содержащую идентификатор алгоритма.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HashAlg позволяет получить алгоритм хэширования подписанных данных.

object.HashAlg

### Возвращаемое значение

Возвращает указатель на строку, содержащую идентификатор алгоритма.

## ISignature::HashEncAlg [C++]

### SIGNATURE::HASHENCALG [VBS]

[C++]

Свойство HashEncAlg позволяет получить алгоритм шифрования хэша подписанных данных.

```
HRESULT get_HashEncAlg(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает указатель на строку, содержащую идентификатор алгоритма.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HashEncAlg позволяет получить алгоритм шифрования хэша подписанных данных.

```
object.HashEncAlg
```

**Возвращаемое значение**

Возвращает указатель на строку, содержащую идентификатор алгоритма.

**ISIGNATURE::RESOURCE [C++]****SIGNATURE::RESOURCE [VBS]**

[C++]

Свойство Resource позволяет получить URI ресурса, где расположены подписанные данные.

```
HRESULT get_Resource(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - идентификатор ресурса в виде строки.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Resource позволяет получить URI ресурса, где расположены подписанные данные .

```
object.Resource
```

**Возвращаемое значение**

Идентификатор ресурса в виде строки.

**ISIGNATURE::REVOCATIONCRLVALUES [C++]****SIGNATURE::REVOCATIONCRLVALUES [VBS]**

[C++]

Свойство RevocationCRLValues позволяет получить из улучшенной подписи (CADES-X Long) коллекцию всех Списков отзыва сертификатов, содержащихся в доказательствах подлинности.

```
HRESULT get_RevocationCRLValues  
    ICRLs* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - указатель на указатель на интерфейс ICRLs, содержащий коллекцию СОС.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RevocationCRLValues позволяет получить из улучшенной подписи (CAAdES-X Long) коллекцию всех Списков отзыва сертификатов, содержащихся в доказательствах подлинности.

```
object.RevocationCRLValues ()
```

#### Возвращаемое значение

Возвращаемое значение – указатель на указатель на интерфейс ICRLs, содержащий коллекцию СОС.

### ISIGNATURE::REVOCATIONOCSPVALUES [C++]

### SIGNATURE::REVOCATIONOCSPVALUES [VBS]

[C++]

Свойство RevocationOCSPValues позволяет получить из улучшенной подписи (CAAdES-X Long) коллекцию всех OCSP ответов, содержащихся в доказательствах подлинности.

```
HRESULT get_RevocationOCSPValues  
    IUnknown** pVal  
);
```

Параметры

pVal [out, retval] – возвращаемое значение – указатель на указатель на интерфейс IOCSPPResponses, содержащий коллекцию OCSP ответов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RevocationOCSPValues позволяет получить из улучшенной подписи (CAAdES-X Long) коллекцию всех OCSP ответов, содержащихся в доказательствах подлинности.

```
object.RevocationOCSPValues ()
```

#### Возвращаемое значение

Возвращаемое значение – указатель на указатель на интерфейс IOCSPPResponses, содержащий коллекцию OCSP ответов.

### ISIGNATURE::REVOCATIONVALUE [C++]

### SIGNATURE::REVOCATIONVALUE [VBS]

[C++]

Свойство RevocationValue позволяет получить из улучшенной подписи (CAAdES-X Long) информацию об отзыве сертификата в виде OCSP ответа или СОС.

```
HRESULT get_RevocationValue  
    ICertificate* certificate  
    Variant* pVal  
);
```

#### Параметры

certificate [in] – указатель на интерфейс ICertificate, в котором содержится проверяемый сертификат.

pVal [out, retval] – возвращаемое значение – информация об отзыве сертификата (OCSP ответ или СОС).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RevocationValue позволяет получить из улучшенной подписи (CADES-X Long) информацию об отзыве сертификата в виде OCSP ответа или COC.

```
object.RevocationValue (certificate)
```

**Параметры**

certificate - указатель на объект Certificate, в котором содержится проверяемый сертификат.

**Возвращаемое значение**

Возвращаемое значение - информация об отзыве сертификата (OCSP ответ или COC).

**ISignature::SignatureUsage [C++]****Signature::SignatureUsage [VBS]**

[C++]

Свойство SignatureUsage позволяет получить OID использования подписи в виде строки.

```
HRESULT get_SignatureUsage(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - строка, содержащая использование подписи.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureUsage позволяет получить OID использования подписи в виде строки.

```
object.SignatureUsage
```

**Возвращаемое значение**

Строка, содержащая использование подписи.

**ISignature::SignatureType [C++]****Signature::SignatureType [VBS]**

[C++]

Свойство SignatureType позволяет получить информацию о типе подписи.

```
HRESULT get_SignatureType(  
    SIGNATURE_TYPE* pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - тип подписи. Параметр принимает значения из SIGNATURE\_TYPE.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignatureType позволяет получить информацию о типе подписи.

```
object.SignatureType ()
```

#### Возвращаемое значение

Возвращаемое значение - тип подписи. Параметр принимает значения из SIGNATURE\_TYPE.

### ISignature::SignerIndex [C++]

#### SIGNATURE::SIGNERINDEX [VBS]

[C++]

Свойство SignerIndex позволяет получить индекс подписчика в CMS сообщении.

```
HRESULT get_SignerIndex(  
    unsigned long* pVal  
);
```

#### Параметры

pVal [out, retval] - индекс подписчика.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SignerIndex позволяет получить индекс подписчика в CMS сообщении.

```
object.SignerIndex
```

#### Возвращаемое значение

Индекс подписчика.

### ISignature::SigningTime [C++]

#### SIGNATURE::SIGNINGTIME [VBS]

[C++]

Свойство SigningTime позволяет получить дату и время подписи.

```
HRESULT get_SigningTime(  
    VARIANT* pVal  
);
```

#### Параметры

pVal [out, retval] - дата и время создания ЭЦП.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SigningTime позволяет получить дату и время подписи.

```
object.SigningTime
```

#### Возвращаемое значение

Дата и время создания ЭЦП.

**ISIGNATURE::TIMESTAMP [C++]**

**SIGNATURE::TIMESTAMP [VBS]**

[C++]

Свойство TimeStamp позволяет получить штамп времени подписи.

```
HRESULT get_TimeStamp(  
    TIME_STAMP_TYPE type  
    ITSPResponse** pVal  
);
```

#### Параметры

type [in] - устанавливает тип получаемого штампа времени, который определяется по типу данных, на которые устанавливается штамп. Параметр принимает значения из TIME\_STAMP\_TYPE.

type [out, retval] - в случае успешного поиска содержит указатель на интерфейс ITSPResponse, в противном случае - пустую ссылку.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TimeStamp позволяет получить штамп времени подписи.

```
object.TimeStamp (type)
```

#### Параметры

Type [in] - устанавливает тип получаемого штампа времени, который определяется по типу данных, на которые устанавливается штамп. Параметр принимает значения из TIME\_STAMP\_TYPE.

#### Возвращаемое значение

В случае успешного поиска содержит указатель на объект TSPResponse, в противном случае - «Nothing».

**ISIGNATURE::TSPRESPONSE [C++]**

**SIGNATURE::TSPRESPONSE [VBS]**

[C++]

Свойство TSPResponse позволяет получить штамп времени подписи, проставленный на подписываемые данные.

```
HRESULT get_TSPResponse(  
    ITSPResponse** pVal  
);
```



## Параметры

pVal [out, retval] - в случае успешного поиска содержит указатель на интерфейс ITSPResponse, в противном случае - пустую ссылку.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSPResponse позволяет получить штамп времени подписи, проставленный на подписываемые данные.

object.TSPResponse

## Возвращаемое значение

В случае успешного поиска содержит указатель на объект TSPResponse, в противном случае - «Nothing».

## SIGNATURES (GUID: 43152407-E1D9-4245-BC2A-67D43C2785AE)

### Методы

Метод	Описание
Export	Метод позволяет произвести экспорт подписей в строку

### Свойства

Свойство	Описание
Count	Свойство позволяет получить количество элементов в коллекции подписей
Item	Свойство позволяет получить элемент из коллекции

## ISIGNATURES::EXPORT [C++]

## SIGNATURES::EXPORT [VBS]

[C++]

Метод Export позволяет произвести экспорт подписей в строку.

```
HRESULT Export(  
    FORMAT fFormat,  
    BSTR bstrSourceURI,  
    BSTR* pbstrSignatures,  
);
```

## Параметры

fFormat [in, defaultvalue(XML\_TYPE)] - формат , в котором будут экспортированы подписи. Принимает значения из FORMAT.  
bstrSourceURI [in, defaultvalue("")] - имя файла исходных данных, если экспортируемые подписи являются отделенными.  
pbstrSignatures [out, retval] - указатель на буфер в памяти, в который экспортируются подписи.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет произвести экспорт подписей в строку.

```
object.Export(fFormat[,bstrSourceURI])
```

### Параметры

fFormat - формат , в котором будут экспортированы подписи. Принимает значения из FORMAT.  
bstrSourceURI - имя файла исходных данных, если экспортируемые подписи являются отдельными.

### Возвращаемое значение

Указатель на буфер в памяти, в который экспортируются подписи.

## ISIGNATURES::COUNT [C++]

## SIGNATURES::COUNT [VBS]

[C++]

Свойство Count позволяет получить количество элементов в коллекции подписей.

```
HRESULT get_Count(  
    long* pVal  
);
```

### Параметры

pVal [out, retval] - возвращает количество элементов в коллекции.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество элементов в коллекции подписей.

```
object.Count
```

### Возвращаемое значение

Возвращает количество элементов в коллекции.

## ISIGNATURES::ITEM [C++]

## SIGNATURES::ITEM [VBS]

[C++]

Свойство Item позволяет получить элемент из коллекции.

```
HRESULT get_Item(  
    long lIndex,  
    ISignature** pVal  
);
```

### Параметры

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].  
pVal [out, retval] - указатель на интерфейс ISignature.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент из коллекции.

```
object.Item (lIndex)
```

#### Параметры

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1].

#### Возвращаемое значение

Указатель на объект Signature

### UTIL (GUID: 243CF830-9C2C-4E47-9EF9-2F9FADFCD3B1)

#### Методы

Метод	Описание
ConvertString	Метод ConvertString позволяет преобразовать строку из одного формата в другой
GetLoggerStatements	Метод GetLoggerStatements позволяет получить список событий, сохраненных в журнале событий
RecodeString	Метод RecodeString позволяет преобразовать строку из одной кодировки в другую
StartupLogger	Метод StartupLogger позволяет запустить процесс логирования событий
ShutdownLogger	Метод ShutdownLogger позволяет остановить процесс логирования событий
UpdateCrlByCert	Метод UpdateCrlByCert позволяет обновить СОС для указанного сертификата
UpdateCrlByUrl	Метод UpdateCrlByUrl позволяет обновить СОС по указанному идентификатору ресурса (URL)

#### Свойства

Свойство	Описание
License	Свойство License позволяет получить/установить лицензию ПО "КриптоАРМ"
SupportedCharsets	Свойство SupportedCharsets позволяет получить список кодировок, поддерживаемых методом RecodeString
Version	Свойство Version позволяет получить версию ПО "КриптоАРМ"

### IUTIL::CONVERTSTRING [C++]

### UTIL::CONVERTSTRING [VBS]

[C++]

Метод ConvertString позволяет преобразовать строку из одного формата в другой.

```
HRESULT ConvertString(  
    FORMAT fSrcFormat,  
    BSTR bstrSource,  
    FORMAT fDestFormat,  
    VARIANT vtAdditionalInfo  
    BSTR* pVal  
);
```

#### Параметры

fSrcFormat [in] - формат исходной строки. Допустимыми значениями являются следующие элементы перечисления FORMAT: BASE64\_TYPE, DER\_TYPE, HEX\_TYPE и UNICODE\_STRING\_TYPE.

bstrSource [in] - строка, содержащая исходные для преобразования данные.  
fDestFormat [in] - формат результирующей строки. Допустимыми значениями являются следующие элементы перечисления FORMAT: BASE64\_TYPE, DER\_TYPE, HEX\_TYPE и UNICODE\_STRING\_TYPE.  
vtAdditionalInfo [in, optional] - в случае, если fDestFormat установлен в BASE64\_TYPE и данный параметр имеет числовое значение, то оно воспринимается как длина результирующей строки.  
Если fDestFormat установлен в HEX\_TYPE, то в данном параметре может передаваться длина результирующей строки (если значение числовое) или шаблон для ее формирования (если значение строковое, например, «%02x »).  
pVal [out, retval] - указатель на буфер, в который будет помещен результат преобразования.

### Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод ConvertString позволяет преобразовать строку из одного формата в другой.

```
object.ConvertString(fSrcFormat, bstrSource, fDestFormat[, vtAdditionalInfo])
```

### Параметры

fSrcFormat - формат исходной строки. Допустимыми значениями являются следующие элементы перечисления FORMAT: BASE64\_TYPE, DER\_TYPE, HEX\_TYPE и UNICODE\_STRING\_TYPE.  
bstrSource - строка, содержащая исходные для преобразования данные.  
fDestFormat - формат результирующей строки. Допустимыми значениями являются следующие элементы перечисления FORMAT: BASE64\_TYPE, DER\_TYPE, HEX\_TYPE и UNICODE\_STRING\_TYPE.  
vtAdditionalInfo - необязательный параметр. В случае, если fDestFormat установлен в BASE64\_TYPE и данный параметр имеет числовое значение, то оно воспринимается как длина результирующей строки. Если fDestFormat установлен в HEX\_TYPE, то в данном параметре может передаваться длина результирующей строки (если значение числовое) или шаблон для ее формирования (если значение строковое, например, «%02x »).

### Возвращаемое значение

Строка, содержащая результат преобразования.

## IUTIL::GETLOGGERSTATEMENTS [C++]

## UTIL::GETLOGGERSTATEMENTS [VBS]

[C++]

Метод GetLoggerStatements позволяет получить список событий, сохраненных в журнале событий. Если при вызове метода не было указано имя файла для сохранения журнала событий (параметр bstrDestFilename), то метод вернет пустую строку.

```
HRESULT GetLoggerStatements(  
    VARIANT vtLoggerID,  
    BSTR bstrDestFilename,  
    BSTR* bstrValue  
);
```

### Параметры

vtLoggerID [in] - идентификатор журнала событий. Значение идентификатора возвращается методом StartLogger.  
bstrDestFilename [in, defaultvalue("")] - строка для сохранения файла журнала событий.

bstrValue [out, retval] - возвращает строку, содержащую список произошедших событий.

#### Примечание

Метод вызывается после вызова метода StartLogger.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод GetLoggerStatements позволяет получить список событий, сохраненных в журнале событий. Если при вызове метода не было указано имя файла для сохранения журнала событий (параметр bstrDestFilename), то метод вернет пустую строку.

```
object.GetLoggerStatements(vtLoggerID, bstrDestFilename)
```

#### Параметры

vtLoggerID - идентификатор журнала событий. Значение идентификатора возвращается методом StartLogger.

bstrDestFilename - строка для сохранения файла журнала событий.

#### Возвращаемое значение

Возвращает строку, содержащую список произошедших событий.

#### Примечание

Метод вызывается после вызова метода StartLogger

### IUTIL::RECODESTRING [C++]

### UTIL::RECODESTRING [VBS]

[C++]

Метод RecodeString позволяет преобразовать строку из одной кодировки в другую.

```
HRESULT RecodeString(  
    BSTR bstrInputCharset,  
    BSTR bstrSource,  
    BSTR bstrOutputCharset,  
    VARIANT vtAdditionalInfo  
    BSTR* pVal  
);
```

#### Параметры

bstrInputCharset [in] - название кодировки исходной строки. Список поддерживаемых кодировок можно получить из свойства SupportedCharsets.

bstrSource [in] - строка, содержащая исходные для преобразования данные.

bstrOutputCharset [in] - название кодировки результирующей строки. Список поддерживаемых кодировок можно получить из свойства SupportedCharsets.

vtAdditionalInfo [in, optional] - в текущей версии не используется.

pVal [out, retval] - указатель на буфер, в который будет помещен результат преобразования.

#### Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод RecodeString позволяет преобразовать строку из одной кодировки в другую.

```
object.RecodeString(bstrInputCharset, bstrSource, bstrOutputCharset[,  
vtAdditionalInfo])
```

### Параметры

**bstrInputCharset** - название кодировки исходной строки. Список поддерживаемых кодировок можно получить из свойства `SupportedCharsets`.  
**bstrSource** - строка, содержащая исходные для преобразования данные.  
**bstrOutputCharset** - название кодировки результирующей строки. Список поддерживаемых кодировок можно получить из свойства `SupportedCharsets`.  
**vtAdditionalInfo** - необязательный параметр. В текущей версии не используется.

### Возвращаемое значение

Строка, содержащая результат преобразования.

## IUTIL::STARTUPLOGGER [C++]

## UTIL::STARTUPLOGGER [VBS]

[C++]

Метод `StartupLogger` позволяет запустить процесс логирования событий.

```
HRESULT StartupLogger(  
    BSTR bstrPattern,  
    LOG_LEVEL lLogLevel,  
    LONG lMaxSize  
    BSTR bstrLoggerName,  
    VARIANT vtExistingLoggerID,  
    VARIANT* vtLoggerID  
);
```

### Параметры

**bstrPattern** [in, defaultvalue("")] - строка с шаблоном журнала событий. Строка может быть следующей:

"log4cplus.appender.toDSL.layout.ConversionPattern=%d [%5t] [%p] [%c] [%x] - %m%n", где параметры означают следующее:

%d - используется для вывода даты журналируемого события,

%5t - используется для вывода имени потока, сгенерировавшего событие,

%p - используется для вывода уровня журналирования события

%c - имя логгера

%x - используется для вывода вложенного контекста, ассоциированного с потоком, который генерирует журналируемое событие.

**lLogLevel** [in, defaultvalue(LOG\_LEVEL\_WARN)] - уровень журналирования. Принимает значения из `LOG_LEVEL`

**lMaxSize** [in, defaultvalue(10 \* 1024 \* 1024)] - максимальный размер файла журнала событий. При превышении максимальной длины журнал очищается.

**bstrLoggerName** [in, defaultvalue("")] - предустановленное имя логгера.

**vtExistingLoggerID** [in, defaultvalue(0)] - устанавливает идентификатор уже существующего журнала событий.

**vtLoggerID** [out, retval] - возвращает идентификатор журнала событий.

### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Метод `StartupLogger` позволяет запустить процесс логирования событий.

```
object.StartupLogger(bstrPattern, lLogLevel, lMaxSize, bstrLoggerName,  
vtExistingLoggerID)
```

## Параметры

bstrPattern - строка с шаблоном журнала событий. Строка может быть следующей: "log4cplus.appender.toDSL.layout.ConversionPattern=%d [%5t] [%p] [%c] [%x] - %m%n", где параметры означают следующее:

%d - используется для вывода даты журналируемого события,  
%5t - используется для вывода имени потока, сгенерировавшего событие,  
%p - используется для вывода уровня журналирования события,  
%c - имя логгера,  
%x - используется для вывода вложенного контекста, ассоциированного с потоком, который генерирует журналируемое событие.

lLogLevel - уровень журналирования. Принимает значения из LOG\_LEVEL

lMaxSize - максимальный размер файла журнала событий. При превышении максимальной длины журнал очищается.

bstrLoggerName - предустановленное имя логгера.

vtExistingLoggerID - устанавливает идентификатор уже существующего журнала событий.

vtLoggerID - возвращает идентификатор журнала событий.

## Возвращаемое значение

Возвращает строку, содержащую список произошедших событий.

## IUTIL::UPDATECRLBYCERT [C++]

## UTIL::UPDATECRLBYCERT [VBS]

[C++]

Метод UpdateCrlByCert позволяет обновить СОС для указанного сертификата.

```
HRESULT UpdateCrlByCert(  
    VARIANT vtCert,  
    VARIANT vtProfile,  
    LONG lFlags,  
    VARIANT* pValue  
);
```

## Параметры

vtCert [in] - указатель на сертификат, для которого будет закачан СОС.

vtProfile [in, defaultvalue(VT\_EMPTY)] - указатель на IServiceProfile с настройками подключения к серверу точки распространения СОС

lFlags [in, defaultvalue(0)] - битовые флаги обновления СОС. Допустимыми значениями являются значения CRL\_UPDATE\_FLAGS.

pValue [out, retval] - указатель на буфер, в который будет помещен скачанный СОС.

## Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод UpdateCrlByCert позволяет сохранять СДС по указанному идентификатору ресурса (URI).

```
object.UpdateCrlByCert(vtCert, vtProfile, lFlags)
```

## Параметры

vtCert - указатель на сертификат, для которого будет закачан СОС.

vtProfile - указатель на IServiceProfile с настройками подключения к серверу точки распространения СОС.

lFlags - битовые флаги обновления СОС. Допустимыми значениями являются значения CRL\_UPDATE\_FLAGS.

**Возвращаемое значение**

Указатель на буфер, в который будет помещен скачанный СОС.

**IUTIL::UPDATECRLBYURL [C++]****UTIL::UPDATECRLBYURL [VBS]**

[C++]

Метод UpdateCrlByUrl позволяет обновить СОС для указанного сертификата.

```
HRESULT UpdateCrlByUrl(  
    BSTR bstrUrl,  
    VARIANT vtProfile,  
    LONG lFlags,  
    VARIANT* pValue  
);
```

**Параметры**

bstrUrl [in] - строка содержащая идентификатор ресурса (URL) загружаемого СОС.  
vtProfile [in, defaultvalue(VT\_EMPTY)] - указатель на IServiceProfile с настройками подключения к серверу точки распространения СОС  
lFlags [in, defaultvalue(0)] - битовые флаги обновления СОС. Допустимыми значениями являются значения CRL\_UPDATE\_FLAGS.  
pValue [out, retval] - указатель на буфер, в который будет помещен скачанный СОС.

**Возвращаемое значение**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод UpdateCrlByUrl позволяет сохранять СДС по указанному идентификатору ресурса (URI).

```
object.UpdateCrlByUrl(bstrUrl, vtProfile, lFlags)
```

**Параметры**

bstrUrl - строка содержащая идентификатор ресурса (URL) загружаемого СОС.  
vtProfile - указатель на IServiceProfile с настройками подключения к серверу точки распространения СОС.  
lFlags - битовые флаги обновления СОС. Допустимыми значениями являются значения CRL\_UPDATE\_FLAGS.

**Возвращаемое значение**

Указатель на буфер, в который будет помещен скачанный СОС.

**IUTIL::SHUTDOWNLOGGER [C++]****UTIL::SHUTDOWNLOGGER [VBS]**

[C++]

Метод ShutdownLogger позволяет завершить процесс логирования событий.

```
HRESULT ShutdownLogger(  
    VARIANT vtLoggerID  
);
```

**Параметры**



vtLoggerID  
[inl] Идентификатор журнала событий.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод ShutdownLogger позволяет завершить процесс логирования событий.

```
object.ShutdownLogger(vtLoggerID)
```

#### Параметры

vtLoggerID  
Идентификатор журнала событий.

#### Возвращаемое значение

Возвращает строку, содержащую список произошедших событий.

### IUTIL::LICENSE [C++]

#### UTIL::LICENSE [VBS]

[C++]

Свойство License позволяет получить/установить лицензию на ПО "КристоАРМ".

```
HRESULT get_License(  
    BSTR* pVal,  
);  
HRESULT put_License(  
    BSTR pVal,  
);
```

#### Параметры

pVal [out, retval] - возвращает строку, содержащую лицензионный ключ для ПО «КристоАРМ»

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство License позволяет получить/установить лицензию на ПО "КристоАРМ".

```
object.License(=Value)
```

#### Параметры

Строка, содержащая лицензионный ключ для ПО «КристоАРМ»

#### Возвращаемое значение

Указатель на строку, содержащую лицензию на ПО «КристоАРМ»

### IUTIL::SUPPORTEDCHARSETS [C++]

#### UTIL::SUPPORTEDCHARSETS [VBS]

[C++]

Свойство SupportedCharsets позволяет получить список кодировок, поддерживаемых методом RecodeString.

```
HRESULT get_SupportedCharsets(  
    VARIANT* pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на буфер, в который будет помещен массив строк с названиями кодировок.

#### Возвращаемое значение

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SupportedCharsets позволяет получить список кодировок, поддерживаемых методом RecodeString.

```
object.SupportedCharsets
```

#### Возвращаемое значение

Массив строк с названиями кодировок.

IUTIL::VERSION [C++]

UTIL::VERSION [VBS]

[C++]

Метод Version позволяет получить строку, содержащую версию ПО "КриптоАРМ".

```
HRESULT get_Version(  
    BSTR* pVal,  
);
```

#### Параметры

pVal [out, retval] - возвращает строку, содержащую версию ПО "КриптоАРМ"

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Version позволяет получить строку, содержащую версию ПО "КриптоАРМ".

```
object.Version
```

#### Возвращаемое значение

Возвращает строку, содержащую версию ПО «КриптоАРМ».

WEBSERVICEPROFILE (GUID: 50AFC3A5-8F1D-4FD3-8D38-8021187128E1)

#### Свойства

Свойство	Описание
SynchronizeWithService	Свойство позволяет установить/получить флаг синхронизации с веб сервисом.

**IWebServiceProfile::SynchronizeWithService [C++]****WebServiceProfile::SynchronizeWithService [VBS]**

[C++]

Свойство SynchronizeWithService позволяет установить/получить флаг синхронизации с веб сервисом.

```
HRESULT get_SynchronizeWithService(  
    VARIANT_BOOL* pVal  
);  
SynchronizeWithService(  
    VARIANT_BOOL newVal  
);
```

**Параметры**

pVal [in] - возвращает TRUE или FALSE. TRUE - использовать синхронизацию с веб сервисом, FALSE - не использовать.  
pVal [out, retval] - представляет СДС, возвращаемый из коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SynchronizeWithService позволяет установить/получить флаг синхронизации с веб сервисом.

```
object.SynchronizeWithService=[newVal]
```

**Параметры**

newVal - позволяет установить флаг синхронизации с веб сервисом. Принимает значения TRUE или FALSE. TRUE - использовать синхронизацию с веб сервисом, FALSE - не использовать.

**Возвращаемое значение**

Флаг синхронизации с веб сервисом. Принимает значения TRUE или FALSE. TRUE - использовать синхронизацию с веб сервисом, FALSE - не использовать.

**Tsl (GUID: -)****Методы и свойства**

Методы	Описание
<a href="#">FinalRealise</a>	Метод позволяет очистить кэшированные данные из списка аккредитованных УЦ (TSL)
<a href="#">Load</a>	Метод позволяет загрузить список аккредитованных УЦ (TSL) из файла
<a href="#">UpgradeFromServer</a>	Метод обновления списка аккредитованных УЦ (выполняется загрузка с официального сайта)

Свойства	Описание
<a href="#">IsValid</a>	Свойство возвращает статус подписи списка аккредитованных УЦ.
<a href="#">IsTslList</a>	Свойство возвращает признак наличия загруженного списка аккредитованных УЦ.
<a href="#">get_ThisUpdate</a>	Свойство возвращает дату формирования списка аккредитованных УЦ
<a href="#">get_Version</a>	Свойство возвращает версию списка аккредитованных УЦ
<a href="#">Count</a>	Свойство возвращает количество сертификатов, содержащихся в списке аккредитованных УЦ

[get\\_VersionFromServer](#)

Свойство возвращает версию списка аккредитованных УЦ (с официального сайта) без загрузки списка целиком

## ITSL::FINALEREALISE [C++]

## TSL::FINALEREALISE [VBS]

[C++][VBS]

Метод FinaleRealise позволяет очистить кэшированные данные из списка аккредитованных УЦ (TSL).

```
HRESULT FinaleRealise ();
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ITSL::LOAD [C++]

## TSL::LOAD [VBS]

[C++][VBS]

Метод Load позволяет загрузить список аккредитованных УЦ (TSL) из файла.

```
HRESULT Load(  
    BSTR bstrURI  
);
```

### Параметры

bstrURI [in] – URL (имя файла ) расположения списка аккредитованных УЦ (TSL).

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ITSL::UPGRADEFROMSERVER [C++]

## TSL::UPGRADEFROMSERVER [VBS]

[C++][VBS]

Метод UpgradeFromServer обновления списка аккредитованных УЦ (выполняется загрузка с официального сайта).

```
HRESULT UpgradeFromServer ();
```

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ITSL::ISVALID [C++]

## TSL::ISVALID [VBS]

[C++][VBS]

Свойство IsValid возвращает статус подписи списка аккредитованных УЦ.

```
HRESULT IsValid(  
    VERIFYSTATUS* pvsStatus  
);
```

#### Параметры

pvsStatus [out] – статус подписи списка аккредитованных УЦ. Допустимы значения VERIFYSTATUS.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### ITSL::IsTslList [C++]

### TSL::IsTslList [VBS]

[C++][VBS]

Свойство IsTslList возвращает признак наличия загруженного списка аккредитованных УЦ.

```
HRESULT IsTslList(  
    BOOL *bIsTslList  
);
```

#### Параметры

bIsTslList [out] – признак наличия загруженного списка аккредитованных УЦ. Значение TRUE означает, что список загружен, FALSE – не загружен.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

#### Примечание.

Если возвращаемое значение FALSE, то не была выполнена инициализация с помощью метода ITSL::Load(BSTR bstrURI), либо список аккредитованных УЦ отсутствует.

### ITSL::GET\_THISUPDATE [C++]

### TSL::GET\_THISUPDATE [VBS]

[C++][VBS]

Свойство get\_ThisUpdate позволяет получить дату формирования списка аккредитованных УЦ.

```
HRESULT get_ThisUpdate(  
    DATE* pVal  
);
```

#### Параметры

pVal [out, retval] – указатель на дату формирования списка.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### ITSL::GET\_VERSION [C++]

### TSL::GET\_VERSION [VBS]

**[C++][VBS]**

Свойство `get_Version` позволяет получить версию списка аккредитованных УЦ.

```
HRESULT get_Version(  
    OUT LONG* pVal  
);
```

**Параметры**

`pVal [out]` – указатель на версию списка аккредитованных УЦ.

**Возвращаемые значения**

Возвращаемое значение имеет тип `HRESULT`.

**ITSL::COUNT [C++]****TSL::COUNT [VBS]****[C++][VBS]**

Свойство `Count` возвращает количество сертификатов, содержащихся в списке аккредитованных УЦ.

```
HRESULT Count(  
    LONG* pVal  
);
```

**Параметры**

`pVal [out]` – указатель на количество сертификатов списка аккредитованных УЦ.

**Возвращаемые значения**

Возвращаемое значение имеет тип `HRESULT`.

**Примечание.**

Свойство можно использовать только после того, как была произведена загрузка списка аккредитованных УЦ (TSL) с помощью метода `HRESULT Load(BSTR bstrURI)`.

**ITSL::GET\_VERSIONFROMSERVER[C++]****TSL::GET\_VERSIONFROMSERVER [VBS]****[C++][VBS]**

Свойство `get_VersionFromServer` возвращает версию списка аккредитованных УЦ (с официального сайта) без загрузки списка целиком.

```
HRESULT get_VersionFromServer(  
    LONG* pVal  
);
```

**Параметры**

`pVal [out]` – указатель на версию списка аккредитованных УЦ.

**Возвращаемые значения**

Возвращаемое значение имеет тип `HRESULT`.

## 1.3.3. СОБЫТИЯ

**\_PKCS7MESSAGE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы и свойства**

Метод	Описание
<a href="#">OnCreateSign</a>	Событие, возникающее при вызове метода Sign объекта PKCS7Message
<a href="#">OnDecrypt</a>	Событие, возникающее при вызове метода Decrypt объекта PKCS7Message
<a href="#">OnEncrypt</a>	Событие, возникающее при вызове метода Sign объекта OnEncrypt
<a href="#">OnGetSignatures</a>	Событие, возникающее при вызове свойства Signatures объекта PKCS7Message
<a href="#">OnGetSigners</a>	Событие, возникающее при вызове свойства Signers объекта PKCS7Message
<a href="#">OnSave</a>	Событие, возникающее при вызове метода Save объекта PKCS7Message
<a href="#">OnSign</a>	Событие, возникающее при вызове метода Sign объекта PKCS7Message

**ONCREATESIGN**

Событие OnCreateSign, которое генерируется объектом PKCS7Message при вызове метода Sign.

```
HRESULT OnCreateSign(  
    VARIANT vtParam  
);
```

**Параметры**

vtParam [in].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**ONDECRYPT**

Событие OnDecrypt, которое генерируется объектом PKCS7Message при вызове метода Decrypt.

```
HRESULT OnDecrypt(  
    VARIANT vtParam  
);
```

**Параметры**

vtParam [in].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**ONENCRYPT**

Событие OnEncrypt, которое генерируется объектом PKCS7Message при вызове метода Encrypt.

```
HRESULT OnEncrypt(  
    VARIANT vtParam  
);
```

**Параметры**

vtParam [in].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

## ONGETSIGNATURES

Событие OnGetSignatures, которое генерируется объектом PKCS7Message при вызове свойства get\_Signatures.

```
HRESULT OnGetSignatures(  
    VARIANT vtParam  
);
```

### Параметры

vtParam [in].

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ONGETSIGNERS

Событие OnGetSigners, которое генерируется объектом PKCS7Message при вызове метода Signers.

```
HRESULT OnGetSigners(  
    VARIANT vtParam  
);
```

### Параметры

vtParam [in].

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ONSAVE

Событие OnSave, которое генерируется объектом PKCS7Message при вызове метода Save.

```
HRESULT OnSave(  
    VARIANT vtParam  
);
```

### Параметры

vtParam [in].

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

## ONSIGN

Событие OnSign, которое генерируется объектом PKCS7Message при вызове метода CoSign.

```
HRESULT OnSign(  
    VARIANT vtParam  
);
```

### Параметры

vtParam [in].



**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**\_SIGNATURE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)**

**Методы и свойства**

Метод	Описание
<a href="#">OnVerify</a>	Событие, возникающее при вызове метода Verify объекта Signature

**ONVERIFY**

Событие OnVerify, которое генерируется объектом Signature при вызове метода Verify.

```
HRESULT OnVerify(  
    VARIANT vtParam  
);
```

**Параметры**

vtParam [in].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

## РАЗДЕЛ 2. ОПИСАНИЕ COM DigtTSPClient

### 2.1. СВОЙСТВА И МЕТОДЫ

Компонент DigtTSPClient предоставляет интерфейсы для отправки запросов на получение штампа времени в Службу Штamped Времени.

Интерфейс	Описание
<a href="#">TSPClient</a>	Описывает интерфейс отправки запроса в Службу Штamped Времени
<a href="#">TSPPProfile</a>	Описывает настройки параметров запроса и доступа к Службе Штamped Времени
<a href="#">TSPPProfiles</a>	Описывает коллекцию настроек TSPPProfile
<a href="#">TSPPProfileStore</a>	Описывает хранилище настроек
<a href="#">TSPResponse</a>	Описывает параметры штампа времени, полученного по запросу
<a href="#">TSPUtil</a>	Предоставляет информацию о версии модуля и описывает методы установки лицен- зии

### 2.2. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

#### 2.2.1. СОЗДАНИЕ И СОХРАНЕНИЕ В ХРАНИЛИЩЕ НОВОГО TSP ПРОФИЛЯ

Данный пример позволяет создать новый TSP профиль, заполнить его параметры и сохранить новый профиль в хранилище TSP профилей. А так же осуществить поиск дан-ного профиля в хранилище профилей и получить свойства найденного профиля.

```
Option explicit

'enum TSP_AUTH_TYPE
const TSP_AUTH_TYPE_ANONYMOUS = 0
const TSP_AUTH_TYPE_BASIC = 1
const TSP_AUTH_TYPE_DIGEST = 2
const TSP_AUTH_TYPE_NTLM = 3
const TSP_AUTH_TYPE_NEGOTIATE = 4
'enum HASH_ALGORITHM
const HASH_ALGORITHM_SHA1 = 0
const HASH_ALGORITHM_MD2 = 1
const HASH_ALGORITHM_MD4 = 2
const HASH_ALGORITHM_MD5 = 3
const HASH_ALGORITHM_GOST3411 = 4

Dim oTSPPProfileStore, oTSPPProfile, oTSPPProfileFinded, oTSPPProfileFinded1

'Создадим новый профиль с настройками доступа к TSP службе
Set oTSPPProfile = CreateObject("DigtTSPClient.TSPPProfile")

'Здесь установим параметры профиля
oTSPPProfile.RequestTSACertificate = true
oTSPPProfile.PolicyID = "1.1.1.1.1"
oTSPPProfile.UseNonce = true
oTSPPProfile.TSAAAddress = "http://www.cryptopro.ru/tsp/tsp.srf"
oTSPPProfile.TSAAuthType = TSP_AUTH_TYPE_ANONYMOUS
oTSPPProfile.TSAUserName = ""
oTSPPProfile.TSAPassword = ""
oTSPPProfile.ProxyAddress = ""
oTSPPProfile.ProxyAuthType = TSP_AUTH_TYPE_DIGEST
oTSPPProfile.ProxyUserName = "UserName"
oTSPPProfile.ProxyPassword = "UserPassword"
oTSPPProfile.UseProxy = true
oTSPPProfile.UseClientCertificate = false
oTSPPProfile.HashAlg = HASH_ALGORITHM_SHA1
```

```
'Добавим новый профиль в хранилище TSP профилей
Set oTSPPProfileStore = CreateObject ("DigtTSPClient.TSPPProfileStore")
oTSPPProfileStore.Open
oTSPPProfileStore.Add oTSPPProfile
oTSPPProfileStore.Save

'Поиск профиля
'Получим идентификатор созданного профиля
Dim sTSPPProfileID
sTSPPProfileID = oTSPPProfile.TSPPProfileID
'Найдем профиль по идентификатору
Set oTSPPProfileFinded = oTSPPProfileStore.Profile (sTSPPProfileID)

'Получим некоторые параметры найденного профиля
Dim GetProfileProps
GetProfileProps = ""
GetProfileProps = GetProfileProps + "RequestTSACertificate = " +
CStr(oTSPPProfile.RequestTSACertificate) + Chr(13)
GetProfileProps = GetProfileProps + "PolicyID = " + oTSPPProfile.PolicyID +
Chr(13)
GetProfileProps = GetProfileProps + "UseNonce = " +
CStr(oTSPPProfile.UseNonce) + Chr(13)
GetProfileProps = GetProfileProps + "TSAAddress = " + oTSPPProfile.TSAAddress
+ Chr(13)
GetProfileProps = GetProfileProps + "TSAAuthType = " +
CStr(oTSPPProfile.TSAAuthType) + Chr(13)
GetProfileProps = GetProfileProps + "ProxyAuthType = " +
CStr(oTSPPProfile.ProxyAuthType) + Chr(13)
GetProfileProps = GetProfileProps + "ProxyUserName = " +
oTSPPProfile.ProxyUserName + Chr(13)
GetProfileProps = GetProfileProps + "ProxyPassword = " +
oTSPPProfile.ProxyPassword + Chr(13)
GetProfileProps = GetProfileProps + "UseProxy = " +
CStr(oTSPPProfile.UseProxy) + Chr(13)
GetProfileProps = GetProfileProps + "UseClientCertificate = " +
CStr(oTSPPProfile.UseClientCertificate) + Chr(13)
GetProfileProps = GetProfileProps + "HashAlg = " + CStr(oTSPPProfile.HashAlg)
+ Chr(13)
GetProfileProps = GetProfileProps + "TSPPProfileID = " +
oTSPPProfile.TSPPProfileID + Chr(13)

MsgBox GetProfileProps
```

## 2.2.2. ПОЛУЧЕНИЕ ШТАМПА ВРЕМЕНИ И ЕГО СВОЙСТВ

Данный пример позволяет запросить штамп времени на произвольные данные, посмотреть свойства полученного штампа времени. Запрос отправляется в тестовую Службу Штампов Времени.

```
Option explicit
'enum TSP_AUTH_TYPE
const TSP_AUTH_TYPE_ANONYMOUS = 0
const TSP_AUTH_TYPE_BASIC = 1
const TSP_AUTH_TYPE_DIGEST = 2
const TSP_AUTH_TYPE_NTLM = 3
const TSP_AUTH_TYPE_NEGOTIATE = 4
'enum HASH_ALGORITHM
const HASH_ALGORITHM_SHA1 = 0
const HASH_ALGORITHM_MD2 = 1
const HASH_ALGORITHM_MD4 = 2
const HASH_ALGORITHM_MD5 = 3
const HASH_ALGORITHM_GOST3411 = 4
```

```
Dim oTSPPProfileStore, oTSPPProfile, oTSPPProfileFinded, oTSPPProfileFinded1

'Создадим новый профиль с настройками доступа к TSP службе
Set oTSPPProfile = CreateObject("DigtTSPClient.TSPPProfile")

'Здесь установим параметры доступа к службе штампов времени в профиль
oTSPPProfile.TSAAAddress = "http://www.cryptopro.ru/tsp/tsp.srf"
oTSPPProfile.TSAAAuthType = TSP_AUTH_TYPE_ANONYMOUS
oTSPPProfile.ProxyAuthType = TSP_AUTH_TYPE_DIGEST
oTSPPProfile.ProxyUserName = """"UserName""
oTSPPProfile.ProxyPassword = """"UserPassword""
oTSPPProfile.UseProxy = true

'Здесь установим параметры запроса службе штампов времени в профиль
oTSPPProfile.RequestTSACertificate = true
oTSPPProfile.PolicyID = "1.1.1.1.1"
oTSPPProfile.UseNonce = true
oTSPPProfile.HashAlg = HASH_ALGORITHM_SHA1

' Отправим запрос и получим Штамп Времени
Dim oTSPPRequest : set oTSPPRequest = CreateObject( "DigtTSPClient.TSPClient" )
oTSPPRequest.TSPPProfile = oTSPPProfile
oTSPPRequest.PlainData = "Некоторые данные, на которые получаем штамп времени"
Dim sPlainData1 : sPlainData1 = oTSPPRequest.PlainData
'Посылаем запрос
oTSPPRequest.Send()

If oTSPPRequest.HTTPStatus = 200 then 'Если запрос был успешным
    ' Получим ответ службы штампов времени
    Dim oTSPPResponse
    set oTSPPResponse = oTSPPRequest.TSPResponse
    Dim GetResponseProps
    GetResponseProps = GetResponseProps + "Accuracy" +
CStr(oTSPPResponse.Accuracy) + Chr(13)
    GetResponseProps = GetResponseProps + "PolicyID = " +
CStr(oTSPPResponse.PolicyID) + Chr(13)
    GetResponseProps = GetResponseProps + "SerialNumber = " +
CStr(oTSPPResponse.SerialNumber) + Chr(13)
    GetResponseProps = GetResponseProps + "Time = " +
CStr(oTSPPResponse.Time) + Chr(13)
    GetResponseProps = GetResponseProps + "TSAName = " +
oTSPPResponse.TSAName + Chr(13)
    GetResponseProps = GetResponseProps + "FailInfo = " +
CStr(oTSPPResponse.FailInfo) + Chr(13)
    GetResponseProps = GetResponseProps + "Status" +
CStr(oTSPPResponse.Status) + Chr(13)
    GetResponseProps = GetResponseProps + "StatusString = " +
oTSPPResponse.StatusString + Chr(13)
    MsgBox GetResponseProps
    'Получим из ответа Службы сертификат службы штампов и посмотрим его
    Dim sTSACert : sTSACert = oTSPPResponse.TSACertificate ' Получили стро-
ку с блобом сертификата
    Dim oTSACert : Set oTSACert = CreateObject ("DigtCrypto.Certificate")
    oTSACert.Import sTSACert
    oTSACert.Display
end if
```

### 2.2.3. ПОЛУЧЕНИЕ ПОДПИСИ СО ВКЛЮЧЕННЫМ В НЕЕ ШТАМПОМ ВРЕМЕНИ

Данный пример позволяет получить электронно-цифровую подпись со включенным штампом времени. Получить подпись со включенным штампом времени можно с помощью метода Sign объекта PKCS7Message, если в настройках подписи установлен флаг включения штампа времени в подпись (свойство IncludeTimeStampToken объекта Profile).

```
Option explicit
'enum TSP_AUTH_TYPE
const TSP_AUTH_TYPE_ANONYMOUS = 0
const TSP_AUTH_TYPE_BASIC = 1
const TSP_AUTH_TYPE_DIGEST = 2
const TSP_AUTH_TYPE_NTLM = 3
const TSP_AUTH_TYPE_NEGOTIATE = 4
'enum HASH_ALGORITHM
const HASH_ALGORITHM_SHA1 = 0
const HASH_ALGORITHM_MD2 = 1
const HASH_ALGORITHM_MD4 = 2
const HASH_ALGORITHM_MD5 = 3
const HASH_ALGORITHM_GOST3411 = 4
'enum DATATYPE (тип данных)
const DT_PLAIN_DATA = 0
const DT_SIGNED_DATA = 2

const ALL_OK = 0
const BASE64 = 0
const DER = 1
const CURRENT_USER_STORE = 1
const CERT_FOR_SIGN = 2
const SIGN_WIZARD_TYPE = 1

const PLAIN_DATA = "test.txt" 'Файл исходных данных
const SIGN_DATA = "test.txt.p7s" 'Файл с подписью

Dim oTSPPProfileStore, oTSPPProfile, oTSPPProfileFinded, oTSPPProfileFinded1, _
oProfile, oCertificate, oCertificateStore, oCertificates, oPKCS7Message

'Создадим новый профиль и заполним параметры получения подписи
Set oProfile = CreateObject("DigtCrypto.Profile")
oProfile.HashAlg = "1.3.14.3.2.26"
oProfile.SignIncludeBase64Headers = false
oProfile.SignExitFormat = DER
oProfile.Detach = false
oProfile.IncludeSignatureTime = true
oProfile.IncludeTimeStampToken = true ' Устанавливаем флаг включения штампа
времени в подпись
oProfile.Comment = "Подпись со штампом времени"
oProfile.SilentLevel = 1
'Выберем из хранилища сертификат подписи и установим его в профиль
Set oCertificateStore = CreateObject ("DigtCrypto.CertificateStore")
oCertificateStore.Open CURRENT_USER_STORE, "my"
' Вызов формы просмотра хранилища
Set oCertificates = oCertificateStore.Display (1)
Set oCertificate = oCertificates.Item (0)
'Установка сертификата подписи в хранилище
oProfile.SetCertificate CERT_FOR_SIGN, "1", oCertificate

'Создадим новый профиль с настройками доступа к TSP службе
Set oTSPPProfile = CreateObject("DigtTSPClient.TSPPProfile")
'Здесь установим параметры доступа к службе штампов времени в профиль
oTSPPProfile.TSAAAddress = "http://www.cryptopro.ru/tsp/tsp.srf"
oTSPPProfile.TSAAuthType = TSP_AUTH_TYPE_ANONYMOUS
```

```

oTSPProfile.ProxyAuthType = TSP_AUTH_TYPE_DIGEST
oTSPProfile.ProxyUserName = "" "UserName"
oTSPProfile.ProxyPassword = "" "UserPassword"
oTSPProfile.UseProxy = true
'Здесь установим параметры запроса службе штампов времени в профиль
oTSPProfile.RequestTSACertificate = true
oTSPProfile.PolicyID = "1.1.1.1.1"
oTSPProfile.UseNonce = true
oTSPProfile.HashAlg = HASH_ALGORITHM_SHA1

'Установим в настройки подписи TSP профиль
oProfile.TSPProfile = oTSPProfile

'Получим подпись
oProfile.CollectData SIGN_WIZARD_TYPE 'Запустим мастер подписи для сбора дан-
ных
Dim CheckResult
CheckResult = oProfile.CheckData (SIGN_WIZARD_TYPE) 'Проверим, все ли данные
собраны
If CheckResult = ALL_OK then
    Set oPKCS7Message = CreateObject ("DigtCrypto.PKCS7Message")
    oPKCS7Message.Profile = oProfile 'Установим профиль с настройками
    oPKCS7Message.Load DT_PLAIN_DATA, PLAIN_DATA, "" 'Загрузим простые
исходные данные
    oPKCS7Message.Sign 'Подпишем данные, используя параметры подписи из
профиля.
    oPKCS7Message.Save DT_SIGNED_DATA, oProfile.SignExitFormat, SIGN_DATA
'Сохраним данные
end if

```

## 2.3. ИНТЕРФЕЙСЫ DIGTSPCLIENT

Константа	Описание
<a href="#">COMPONENT_TYPE</a>	Позволяет задать тип модуля
<a href="#">HASH_ALGORITHM</a>	Алгоритм хеширования данных запроса
<a href="#">RESPONSE_FORMAT</a>	Формат ответа из Службы Штампов Времени
<a href="#">TSP_AUTH_TYPE</a>	Тип аутентфикации Службы Штампов Времени
<a href="#">TSP_PROFILE_STORE_TYPE</a>	Тип хранилища профиля TSP

### 2.3.1. КОНСТАНТЫ И ПЕРЕЧИСЛЕНИЯ

#### COMPONENT\_TYPE

Позволяет задать тип модуля

Значение	Описание
DIGT_TSP_CLIENT = 0	КриптоАРМ Модуль TSP
CP_TSP_CLIENT = 1	Модуль КриптоПро TSP Клиент

#### HASH\_ALGORITHM

Значение	Описание
HASH_ALGORITHM_SHA1 = 0	Алгоритм хэширования SHA1
HASH_ALGORITHM_MD2 = 1	Алгоритм хэширования MD2
HASH_ALGORITHM_MD4 = 2	Алгоритм хэширования MD4
HASH_ALGORITHM_MD5 = 3	Алгоритм хэширования MD5
HASH_ALGORITHM_GOST3411 = 4	Алгоритм хэширования GOST3411

**RESPONSE\_FORMAT**

Значение	Описание
FORMAT_DER = 0	Формат DER
FORMAT_BASE64 = 1	Формат BASE64

**TSP\_AUTH\_TYPE**

Значение	Описание
TSP_AUTH_TYPE_ANONYMOUS = 0	Анонимный доступ
TSP_AUTH_TYPE_BASIC = 1	Обычная аутентификация
TSP_AUTH_TYPE_DIGEST = 2	Краткая проверка подлинности
TSP_AUTH_TYPE_NTLM = 3	Встроенная проверка подлинности Windows
TSP_AUTH_TYPE_NEGOTIATE = 4	Встроенная проверка подлинности Windows или Kerberos

**TSP\_PROFILE\_STORE\_TYPE**

Значение	Описание
TSP_REGISTRY_STORE = 0	Тип хранилища - Реестр Windows
TSP_XML_STORE = 1	Тип хранилища - XML файл

**2.3.2. ОБЪЕКТЫ И ИНТЕРФЕЙСЫ****TSPCLIENT (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
Send	Метод позволяет отправить запрос в Службу Штмпов Времени и получить ответ

**Свойства**

Свойство	Описание
HTTPStatus	Свойство позволяет получить/установить http-статус отправки запроса в Службу Штмпов Времени
PlainData	Свойство позволяет получить/установить данные, на которые необходимо получить штамп времени
PlainDataFile	Свойство позволяет получить/установить путь к документу, на который необходимо получить штамп времени
TSPProfile	Свойство позволяет получить/установить профиль с настройками
TSPResponse	Свойство позволяет получить/установить результат обработки запроса Службой Штмпов Времени

**TSPCLIENT::SEND [C++]****TSPCLIENT::SEND [VBS]****[C++]**

Метод Send позволяет отправить запрос в Службу Штмпов Времени и получить ответ.

```
HRESULT Send(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Send позволяет отправить запрос в Службу Штмпов Времени и получить ответ.

```
object.Send
```

**TSPCLIENT::HTTPSTATUS [C++]****TSPCLIENT::HTTPSTATUS [VBS]**

[C++]

Свойство HTTPStatus позволяет получить/установить http-статус отправки запроса в Службу Штмпов Времени.

```
HRESULT get_HTTPStatus(  
    long* pVal  
);  
HRESULT put_HTTPStatus(  
    long pVal  
);
```

**Параметры**

pVal [out, retval] - значение http-статуса отправки запроса.  
pVal [in] - значение http-статуса отправки запроса.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HTTPStatus позволяет получить/установить http-статус отправки запроса в Службу Штмпов Времени.

```
object.HTTPStatus [= Value]
```

**Параметры**

Value - значение http-статуса отправки запроса.

**TSPCLIENT::PLAINDATA [C++]****TSPCLIENT::PLAINDATA [VBS]**

[C++]

Свойство PlainData позволяет получить/установить данные, на которые необходимо получить штамп времени.

```
HRESULT get_PlainData(  
    BSTR* pVal  
);  
HRESULT put_PlainData(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку с данными.



newVal [in] - строка с данными.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PlainData позволяет получить/установить данные, на которые необходимо получить штамп времени.

object.PlainData [= Value]

### Параметры

Value - строка с данными.

## TSPCLIENT::PLAINDATAFILE [C++]

## TSPCLIENT::PLAINDATAFILE [VBS]

[C++]

Свойство PlainDataFile позволяет получить/установить путь к документу, на который необходимо получить штамп времени.

```
HRESULT get_PlainDataFile(  
    BSTR* pVal  
);  
HRESULT put_PlainDataFile(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] - указатель на строку с путем к документу.  
newVal [in] - строка с путем к документу.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PlainDataFile позволяет получить/установить путь к документу, на который необходимо получить штамп времени.

object.PlainDataFile [= Value]

### Параметры

Value - строка с путем к документу.

## TSPCLIENT::TSPPROFILE [C++]

## TSPCLIENT::TSPPROFILE [VBS]

[C++]

Свойство TSPPProfile позволяет получить/установить профиль.

```
HRESULT get_TSPPProfile(  
    ITSPProfile** pVal  
);  
HRESULT put_TSPPProfile(  
    ITSPProfile* pVal  
);
```

```
ITSPProfile* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на указатель на интерфейс ITSPProfile.  
pVal [in] - указатель на интерфейс ITSPProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSPProfile позволяет получить/установить профиль.

```
object.TSPProfile [= Value]
```

### Параметры

Value - указатель на объект TSPProfile.

## TSPCLIENT::TSPRESPONSE [C++]

## TSPCLIENT::TSPRESPONSE [VBS]

[C++]

Свойство TSPResponse позволяет получить/установить результат обработки запроса Службой Штamped Времени.

```
HRESULT get_TSPResponse(  
    ITSPResponse** pVal  
);  
HRESULT put_TSPResponse(  
    ITSPResponse* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на указатель на интерфейс ITSPResponse.  
pVal [in] - указатель на интерфейс ITSPResponse.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSPResponse позволяет получить/установить результат обработки запроса Службой Штamped Времени.

```
object.TSPResponse [= Value]
```

### Параметры

Value - указатель на интерфейс ITSPResponse.

**TSPPROFILE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
Clone	Метод позволяет создать копию профиля

**Свойства**

Свойство	Описание
ClientCertificate	Свойство позволяет получить/установить сертификат для клиентской аутентификации
HashAlg	Свойство позволяет получить/установить алгоритм хэширования данных, выполняемого перед отправкой запроса к Службе Штатпов Времени
PolicyID	Свойство позволяет получить/установить идентификатор политики (в формате OID), по которой должен быть выдан ответ Службой Штатпов Времени
ProxyAddress	Свойство позволяет получить/установить адрес прокси-сервера
ProxyAuthType	Свойство позволяет получить/установить тип аутентификации на прокси-сервере
ProxyPassword	Свойство позволяет получить/установить пароль для аутентификации на прокси-сервере
ProxyUserName	Свойство позволяет получить/установить имя пользователя для аутентификации на прокси-сервере
TSAAddress	Свойство позволяет получить/установить адрес сервера Службы Штатпов Времени
TSAAuthType	Свойство позволяет получить/установить тип аутентификации доступа к Службе Штатпов Времени
TSAPassword	Свойство позволяет получить/установить пароль для аутентификации доступа к Службе Штатпов Времени
TSAUserName	Свойство позволяет получить/установить имя пользователя для аутентификации доступа к Службе Штатпов Времени
TSPProfileID	Свойство позволяет получить/установить идентификатор TSP-профиля
UseClientCertificate	Свойство позволяет получить/установить флаг использования клиентского сертификата для аутентификации
UseNonce	Свойство позволяет получить/установить флаг использования уникального идентификатора для связывания TSP-запроса и ответа
UseProxy	Свойство позволяет получить/установить флаг использования прокси-сервера для обращения к Службе Штатпов Времени
RequestTSACertificate	Свойство позволяет получить/установить флаг запроса включения сертификата подписи Службы Штатпов Времени в штамп времени

**ITSPPROFILE::CLONE [C++]****TSPPROFILE::CLONE [VBS]****[C++]**

Метод Clone позволяет создать копию профиля.

```
HRESULT Clone(  
    ITSPProfile** pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - профиль, являющийся копией текущего профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clone позволяет создать копию профиля.

```
object.Clone(1)
```

**Возвращаемое значение**

Возвращаемое значение - профиль, являющийся копией текущего профиля.

**TSPPROFILE::CLIENTCERTIFICATE [C++]****TSPPROFILE::CLIENTCERTIFICATE [VBS]**

[C++]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище.

```
HRESULT get_ClientCertificate(  
    BSTR* pVal  
);  
HRESULT put_ClientCertificate(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку с сертификатом клиентской аутентификации.

pVal [in] - строка с сертификатом клиентской аутентификации.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище .

```
object.ClientCertificate [= Value]
```

**Параметры**

Value - строка с сертификатом клиентской аутентификации.

**TSPPROFILE::HASHALG [C++]****TSPPROFILE::HASHALG [VBS]**

[C++]

Свойство HashAlg позволяет получить/установить алгоритм хэширования данных, выполняемого перед отправкой запроса к Службам Штатных Времени.

```
HRESULT get_HashAlg(  
    HASH_ALGORITHM* pVal  
);  
HRESULT put_HashAlg(  
    HASH_ALGORITHM pVal
```

```
);
```

### Параметры

pVal [out, retval] - возвращаемое значение - алгоритм хэширования данных. Допустимыми значениями являются значения HASH\_ALGORITHM.  
pVal [in] - алгоритм хэширования данных. Допустимыми значениями являются значения HASH\_ALGORITHM.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HashAlg позволяет получить/установить алгоритм хэширования данных, выполняемого перед отправкой запроса к Службам Штатных Времени.

```
object.HashAlg [= Value]
```

### Параметры

Value - алгоритм хэширования данных. Допустимыми значениями являются значения HASH\_ALGORITHM.

## TSPPROFILE::POLICYID [C++]

## TSPPROFILE::POLICYID [VBS]

[C++]

Свойство PolicyID позволяет получить/установить идентификатор политики (в формате OID), по которой должен быть выдан ответ Службой Штатных Времени.

```
HRESULT get_PolicyID(  
    BSTR* pVal  
);  
HRESULT put_PolicyID(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] - указатель на строку идентификатора политики (в формате OID).  
pVal [in] - строка идентификатора политики (в формате OID).

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PolicyID позволяет получить/установить идентификатор политики (в формате OID), по которой должен быть выдан ответ Службой Штатных Времени.

```
object.PolicyID [= Value]
```

### Параметры

Value - строка идентификатора политики (в формате OID).

## TSPPROFILE::PROXYADDRESS [C++]

## TSPPROFILE::PROXYADDRESS [VBS]

**[C++]**

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

```
HRESULT get_ProxyAddress(  
    BSTR* pVal  
);  
HRESULT put_ProxyAddress(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку с адресом прокси-сервера.  
pVal [in] - строка с адресом прокси-сервера.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

```
object.ProxyAddress [= Value]
```

**Параметры**

Value - строка с адресом прокси-сервера.

**TSPPROFILE::PROXYAUTHTYPE [C++]****TSPPROFILE::PROXYAUTHTYPE [VBS]****[C++]**

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

```
HRESULT get_ProxyAuthType(  
    TSP_AUTH_TYPE* pVal  
);  
HRESULT put_ProxyAuthType(  
    TSP_AUTH_TYPE pVal  
);
```

**Параметры**

pVal [out, retval] - возвращаемое значение - тип аутентификации. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.  
pVal [in] - тип аутентификации. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

```
object.ProxyAuthType [= Value]
```

**Параметры**

Value - тип аутентификации. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.

**TSPPROFILE::PROXYPASSWORD [C++]****TSPPROFILE::PROXYPASSWORD [VBS]**

[C++]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
HRESULT get_ProxyPassword(  
    BSTR* pVal  
);  
HRESULT put_ProxyPassword(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку пароля для аутентификации на прокси-сервере.

pVal [in] - строка пароля для аутентификации на прокси-сервере.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
object.ProxyPassword [= Value]
```

**Параметры**

Value - строка пароля для аутентификации на прокси-сервере.

**TSPPROFILE::PROXYUSERNAME [C++]****TSPPROFILE::PROXYUSERNAME [VBS]**

[C++]

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
HRESULT get_ProxyUserName(  
    BSTR* pVal  
);  
HRESULT put_ProxyUserName(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] - указатель на строку с именем пользователя для аутентификации на прокси-сервере.

pVal [in] - строка с именем пользователя для аутентификации на прокси-сервере.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
object.ProxyUserName [= Value]
```

#### Параметры

Value - строка с именем пользователя для аутентификации на прокси-сервере.

### TSPPROFILE::TSAADDRESS [C++]

### TSPPROFILE::TSAADDRESS [VBS]

[C++]

Свойство TSAAddress позволяет получить/установить адрес сервера Службы Штампов Времени.

```
HRESULT get_TSAAddress(  
    BSTR* pVal  
);  
HRESULT put_TSAAddress(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] - указатель на строку адреса сервера Службы Штампов Времени.

pVal [in] - строка адреса сервера Службы Штампов Времени.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSAAddress позволяет получить/установить адрес сервера Службы Штампов Времени.

```
object.TSAAddress [= Value]
```

#### Параметры

Value - строка адреса сервера Службы Штампов Времени.

### TSPPROFILE::TSAAUTHTYPE [C++]

### TSPPROFILE::TSAAUTHTYPE [VBS]

[C++]

Свойство TSAAuthType позволяет получить/установить тип аутентификации доступа к Службе Штампов Времени.

```
HRESULT get_TSAAuthType(  
    TSP_AUTH_TYPE* pVal  
);  
HRESULT put_TSAAuthType(  
    TSP_AUTH_TYPE pVal  
);
```

#### Параметры

pVal [out, retval] - возвращаемое значение - тип аутентификации доступа к Службе Штампов Времени. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.



pVal [in] – тип аутентификации доступа к Службе Штamped Времени. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSAAuthType позволяет получить/установить тип аутентификации доступа к Службе Штamped Времени.

```
object.TSAAuthType [= Value]
```

### Параметры

Value – тип аутентификации доступа к Службе. Допустимыми значениями являются значения TSP\_AUTH\_TYPE.

## TSPPROFILE::TSAPASSWORD [C++]

## TSPPROFILE::TSAPASSWORD [VBS]

[C++]

Свойство TSAPassword позволяет получить/установить пароль для аутентификации доступа к Службе Штamped Времени.

```
HRESULT get_TSAPassword(  
    BSTR* pVal  
);  
HRESULT put_TSAPassword(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] – указатель строку пароля для аутентификации доступа к Службе Штamped Времени.

pVal [in] – строка пароля для аутентификации доступа к Службе Штamped Времени.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSAPassword позволяет получить/установить пароль для аутентификации доступа к Службе Штamped Времени.

```
object.TSAPassword [= Value]
```

### Параметры

Value – строка пароля для аутентификации доступа к Службе Штamped Времени.

## TSPPROFILE::TSAUSERNAME [C++]

## TSPPROFILE::TSAUSERNAME [VBS]

[C++]

Свойство `TSAUserName` позволяет получить/установить имя пользователя для аутентификации доступа к Службе Штamped Времени.

```
HRESULT get_TSAUserName(  
    BSTR* pVal  
);  
HRESULT put_TSAUserName(  
    BSTR pVal  
);
```

#### Параметры

`pVal [out, retval]` - указатель на строку с именем пользователя для аутентификации доступа к Службе Штamped Времени.

`pVal [in]` - строка с именем пользователя для аутентификации доступа к Службе Штamped Времени.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `TSAUserName` позволяет получить/установить имя пользователя для аутентификации доступа к Службе Штamped Времени.

```
object.TSAUserName [= Value]
```

#### Параметры

`Value` - строка с именем пользователя для аутентификации доступа к Службе Штamped Времени.

### TSPPROFILE::TSPPROFILEID [C++]

### TSPPROFILE::TSPPROFILEID [VBS]

[C++]

Свойство `TSPProfileID` позволяет получить/установить идентификатор TSP-профиля.

```
HRESULT get_TSPProfileID(  
    BSTR* pVal  
);  
HRESULT put_TSPProfileID(  
    BSTR pVal  
);
```

#### Параметры

`pVal [out, retval]` - указатель на строку идентификатора TSP-профиля.

`pVal [in]` - строка идентификатора TSP-профиля.

#### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `TSPProfileID` позволяет получить/установить идентификатор TSP-профиля.

```
object.TSPProfileID [= Value]
```

#### Параметры

`Value` - строка идентификатора TSP-профиля.

**TSPPROFILE::USECLIENTCERTIFICATE [C++]****TSPPROFILE::USECLIENTCERTIFICATE [VBS]**

[C++]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
HRESULT get_UseClientCertificate(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseClientCertificate(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

pVal [in] - принимает TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
object.UseClientCertificate [= Value]
```

**Параметры**

Value - TRUE или FALSE. TRUE - использовать клиентский сертификат для аутентификации, FALSE - не использовать.

**TSPPROFILE::USENONCE [C++]****TSPPROFILE::USENONCE [VBS]**

[C++]

Свойство UseNonce позволяет получить/установить флаг использования уникального идентификатора для связывания TSP-запроса и ответа.

```
HRESULT get_UseNonce(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseNonce(  
    VARIANT_BOOL pVal  
);
```

**Параметры**

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - использовать уникальный идентификатора, FALSE - не использовать.

pVal [in] - принимает TRUE или FALSE. TRUE - использовать уникальный идентификатора, FALSE - не использовать.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseNonce позволяет получить/установить флаг использования уникального идентификатора для связывания TSP-запроса и ответа.

```
object.UseNonce [= Value]
```

#### Параметры

Value - TRUE или FALSE. TRUE - использовать уникальный идентификатора, FALSE - не использовать.

### TSPPROFILE::USEPROXY [C++]

### TSPPROFILE::USEPROXY [VBS]

[C++]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Службе Штатпов Времени.

```
HRESULT get_UseProxy(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseProxy(  
    VARIANT_BOOL pVal  
);
```

#### Параметры

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.  
pVal [in] - принимает TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Службе Штатпов Времени.

```
object.UseProxy [= Value]
```

#### Параметры

Value - TRUE или FALSE. TRUE - использовать прокси-сервер, FALSE - не использовать.

### TSPPROFILE::REQUESTTSACERTIFICATE [C++]

### TSPPROFILE::REQUESTTSACERTIFICATE [VBS]

[C++]

Свойство RequestTSACertificate позволяет получить/установить флаг запроса включения сертификата подписи Службы Штатпов Времени в штамп времени.

```
HRESULT get_RequestTSACertificate(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_RequestTSACertificate(  
    VARIANT_BOOL pVal  
);
```

```
VARIANT_BOOL pVal  
);
```

### Параметры

pVal [out, retval] - возвращает TRUE или FALSE. TRUE - включать сертификата подписи Службы Штмпов Времени в штамп времени, FALSE - не использовать.  
pVal [in] - принимает TRUE или FALSE. TRUE - включать сертификата подписи Службы Штмпов Времени в штамп времени, FALSE - не использовать.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство RequestTSACertificate позволяет получить/установить флаг запроса включения сертификата подписи Службы Штмпов Времени в штамп времени.

```
object.RequestTSACertificate [= Value]
```

### Параметры

Value - TRUE или FALSE. TRUE - включать сертификата подписи Службы Штмпов Времени в штамп времени, FALSE - не использовать.

**TSPPROFILES (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)**

**Методы**

Метод	Описание
Add	Метод позволяет добавить
Clear	Метод позволяет очистить коллекцию
Remove	Метод позволяет удалить профиль из коллекции

**Свойства**

Свойство	Описание
Count	Свойство позволяет получить число профилей в коллекции
Item	Свойство позволяет получить указатель на элемент коллекции

**ITSPPROFILES::ADD [C++]**

**TSPPROFILES::ADD [VBS]**

[C++]

Метод Add позволяет добавить профиль в коллекцию.

```
HRESULT Add(  
    ITSPProfile* addVal  
);
```

**Параметры**

addVal [in] - добавляемый элемент. Указатель на интерфейс ITSPProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить профиль в коллекцию.

```
object.Add(addVal)
```

**Параметры**

addVal - добавляемый элемент. Указатель на объект Profile.

**ITSPPROFILES::ADD [C++]**

**TSPPROFILES::ADD [VBS]**

[C++]

Метод Add позволяет добавить профиль в коллекцию.

```
HRESULT Add(  
    ITSPProfile* addVal  
);
```

**Параметры**

addVal  
[in] Добавляемый элемент. Указатель на интерфейс ITSPProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить профиль в коллекцию.

```
object.Add(addVal)
```

**Параметры**

addVal - добавляемый элемент. Указатель на объект Profile.

**ITSPPROFILES::CLEAR [C++]****TSPPROFILES::CLEAR [VBS]**

[C++]

Метод Clear позволяет очистить коллекцию профилей.

```
HRESULT Clear(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clear позволяет очистить коллекцию профилей.

```
object.Clear()
```

**ITSPPROFILES::REMOVE [C++]****TSPPROFILES::REMOVE [VBS]**

[C++]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
HRESULT Remove(  
    VARIANT varIndex  
);
```

**Параметры**

varIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
object.Remove(varIndex)
```

**Параметры**

varIndex - номер элемента в коллекции минус 1 [0 .. Count-1].

**ITSPPROFILES::COUNT [C++]****TSPPROFILES::COUNT [VBS]**

[C++]

Свойство Count позволяет получить количество профилей в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

**Параметры**

pVal [out, retval] - количество профилей в коллекции.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество профилей в коллекции.

```
object.Count
```

**Возвращаемое значение**

Количество профилей в коллекции.

**ITSPPROFILES::ITEM [C++]****TSPPROFILES::ITEM [VBS]**

[C++]

Свойство Item позволяет получить элемент коллекции.

```
HRESULT get_Item(  
    long lIndex,  
    ITSPProfile** pVal  
);
```

**Параметры**

lIndex [in] - номер элемента в коллекции минус 1 [0 .. Count-1].  
pVal [out, retval] - указатель на ITSPProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент коллекции.

```
object.Item (lIndex)
```

**Параметры**

lIndex - номер элемента в коллекции минус 1 [0 .. Count-1].

**Возвращаемое значение**

Указатель на TSPPProfile.



**TSPPROFILESTORE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
Add	Метод позволяет добавить новую настройку в хранилище
Close	Метод позволяет закрыть хранилище настроек
Open	Метод позволяет открыть хранилище настроек
Remove	Метод позволяет удалить настройку из хранилища
Save	Метод позволяет сохранить хранилище настроек в реестр

**Свойства**

Свойство	Описание
Profile	Свойство позволяет найти профиль в хранилище по идентификатору
Store	Свойство позволяет получить/установить коллекцию TSP профилей, содержащуюся в хранилище

**ITSPPROFILESTORE::ADD [C++]****TSPPROFILESTORE::ADD [VBS]****[C++]**

Метод Add позволяет добавить профиль в хранилище.

```
HRESULT Add(  
    ITSPProfile* pNewTSPProfile  
);
```

**Параметры**

pNewTSPProfile [in] - указатель на интерфейс ITSPProfile.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Add позволяет добавить профиль в хранилище.

```
object.Add (Value)
```

**Параметры**

Value - указатель на интерфейс ITSPProfile.

**ITSPPROFILESTORE::CLOSE [C++]****TSPPROFILESTORE::CLOSE [VBS]****[C++]**

Метод Close позволяет закрыть хранилище.

```
HRESULT Close(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Close позволяет закрыть хранилище.

```
object.Close ()
```

## ITSPPROFILESTORE::OPEN [C++]

## TSPPROFILESTORE::OPEN [VBS]

[C++]

Метод Open позволяет открыть хранилище.

```
HRESULT Open(  
    TSP_PROFILE_STORE_TYPE stStoreType  
    BSTR bstrPath  
);
```

### Параметры

stStoreType [in, defaultvalue(TSP\_REGISTRY\_STORE)] - тип хранилища. Принимает значения из TSP\_PROFILE\_STORE\_TYPE.  
bstrPath [in, defaultvalue("")] - строка, содержащая путь к XML файлу.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет открыть хранилище.

```
object.Open (stStoreType, bstrPath)
```

### Параметры

stStoreType - тип хранилища. Принимает значения из TSP\_PROFILE\_STORE\_TYPE. По умолчанию принимает значение TSP\_REGISTRY\_STORE  
bstrPath - строка, содержащая путь к XML файлу. По умолчанию принимает значение "".

## ITSPPROFILESTORE::REMOVE [C++]

## TSPPROFILESTORE::REMOVE [VBS]

[C++]

Метод Remove позволяет удалить профиль из хранилища по идентификатору.

```
HRESULT Remove(  
    BSTR bstrID  
);
```

### Параметры

bstrID [in] - строка с идентификатором профиля.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### Примечание

Не реализован.

[VBS]

Метод Remove позволяет удалить профиль из хранилища по идентификатору.

```
object.Remove (Value)
```

#### Параметры

Value – строка с идентификатором профиля.

#### Примечание

Не реализован.

### ITSPPROFILESTORE::SAVE [C++]

### TSPPROFILESTORE::SAVE [VBS]

[C++]

Метод Save позволяет сохранить изменения в хранилище профилей .

```
HRESULT Save (  
);
```

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить изменения в хранилище профилей.

```
object.Save ()
```

### ITSPPROFILESTORE::PROFILE [C++]

### TSPPROFILESTORE::PROFILE [VBS]

[C++]

Свойство Profile позволяет найти профиль в хранилище по идентификатору.

```
HRESULT get_Profile(  
    BSTR bstrID  
    ITSPProfile** pVal  
);
```

#### Параметры

bstrID [in] – идентификатор профиля.

pVal [out, retval] – указатель на указатель на интерфейс ITSPProfile. В случае отсутствия профиля с указанным идентификатором возвращается пустая ссылка.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет найти профиль в хранилище по идентификатору.

```
object.Profile (bstrID)
```

### Параметры

bstrID  
Идентификатор профиля.

### Возвращаемые значения

Возвращаемое значение указатель на указатель на интерфейс ITSPProfile. В случае отсутствия профиля с указанным идентификатором возвращается "Nothing".

## ITSPPROFILESTORE::STORE [C++]

## TSPPROFILESTORE::STORE [VBS]

[C++]

Свойство Store позволяет получить/установить коллекцию TSP профилей, содержащуюся в хранилище.

```
HRESULT get_Store(  
    ITSPProfiles** pVal  
);  
HRESULT put_Store(  
    ITSPProfiles* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на коллекцию TSP профилей.  
pVal [in] - указатель на коллекцию TSP профилей.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Store позволяет получить/установить коллекцию TSP профилей, содержащуюся в хранилище.

```
object.Store [= pVal]
```

### Параметры

pVal - указатель на коллекцию TSP профилей.

TSPRESPONSE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)

Методы

Метод	Описание
Export	Метод позволяет экспортировать штамп времени в область памяти
VerifyStamp	Метод позволяет проверить корректность ЭЦП штампа и сертификата Службы штампов времени

Свойства

Свойство	Описание
Accuracy	Свойство позволяет получить точность штампа времени
FailInfo	Свойство позволяет получить/установить биты ошибки обработки запроса Службой Штампов Времени
HashAlgorithm	Свойство позволяет получить информацию об алгоритме хеширования данных
HashValue	Свойство позволяет получить значение хэша данных, на которые был выдан штамп
PlainData	Свойство позволяет установить строку данных, для которых был получен штамп времени
PlainDataFile	Свойство позволяет установить строку с путем к документу, для которого был получен штамп времени
PolicyID	Свойство позволяет получить политику выдачи штампа времени (в формате OID)
SerialNumber	Свойство позволяет получить серийный номер штампа времени
Status	Свойство позволяет получить/установить статус обработки запроса Службой Штампов Времени
StatusString	Свойство позволяет получить/установить пользовательское описание статуса обработки запроса Службой Штампов Времени
Time	Свойство позволяет получить время выдачи штампа времени
TimeStampToken	Свойство позволяет установить блок штампа времени
TSACertificate	Свойство позволяет получить сертификат подписи Службы Штампов Времени из штампа времени
TSAName	Свойство позволяет получить имя Службы Штампа Времени из штампа времени

ITSPRESPONSE::EXPORT [C++]

TSPRESPONSE::EXPORT [VBS]

[C++]

Метод Export позволяет экспортировать штамп времени в область памяти.

```
HRESULT Export(
    RESPONSE_FORMAT   frmResponseFormat
    BSTR*   pVal
);
```

Параметры

frmResponseFormat [in] - формат штампа времени. Допустимыми значениями являются значения из RESPONSE\_FORMAT  
pVal [out, retval] - указатель на строку, содержащую штамп времени в указанном формате.

Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать штамп времени в область памяти.

```
object.Export (frmResponseFormat)
```

#### Параметры

frmResponseFormat - формат штампа времени. Допустимыми значениями являются значения из RESPONSE\_FORMAT

#### Возвращаемые значения

Возвращаемое значение указатель на строку, содержащую штамп времени в указанном формате.

### ITSPRESPONSE::VERIFYSTAMP [C++]

### TSPRESPONSE::VERIFYSTAMP [VBS]

[C++]

Метод VerifyStamp позволяет проверить корректность ЭЦП штампа и сертификата Службы штампов времени

```
HRESULT VerifyStamp(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - результат проверки: 0 - проверка прошла успешно, иначе - код ошибки. (см. поле dwErrorStatus структуры CERT\_TRUST\_STATUS (CryptoAPI)).

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод VerifyStamp позволяет проверить корректность ЭЦП штампа и сертификата Службы штампов времени

```
object.VerifyStamp ()
```

#### Возвращаемые значения

Возвращаемое значение - результат проверки: 0 - проверка прошла успешно, иначе - код ошибки. (см. поле dwErrorStatus структуры CERT\_TRUST\_STATUS (CryptoAPI)).

### ITSPRESPONSE::ACCURACY [C++]

### TSPRESPONSE::ACCURACY [VBS]

[C++]

Свойство Accuracy позволяет получить точность штампа времени.

```
HRESULT get_Accuracy(  
    long* pVal  
);
```

#### Параметры

pVal [out, retval] - точность штампа времени.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Accuracy позволяет получить точность штампа времени.

object.Accuracy (bstrID)

**Возвращаемые значения**

Возвращаемое значение - точность штампа времени.

**ITSPRESPONSE::FAILINFO [C++]****TSPRESPONSE::FAILINFO [VBS]**

[C++]

Метод FailInfo позволяет получить/установить биты ошибки обработки запроса Службой Штмпов Времени.

```
HRESULT get_FailInfo(  
    long* pVal  
);  
HRESULT put_FailInfo(  
    long pVal  
);
```

**Параметры**

pVal [in] - биты ошибки обработки запроса Службой Штмпов Времени.

pVal [out, retval] - ошибки обработки запроса Службой Штмпов Времени.

Константа	Описание
badAlg = 0	Неизвестный или неподдерживаемый алгоритм
badRequest = 2	Запрос запрещен или неподдерживается
badDataFormat = 5	Неверный формат даты
timeNotAvailable = 14	Служба Штмпов Времени недоступна
unacceptedPolicy = 15	Неподдерживаемая политика
unacceptedExtension = 16	Неподдерживаемое расширение
addInfoNotAvailable = 17	Дополнительная информация непонятна или недоступна
systemFailure = 25	Системная ошибка

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство FailInfo позволяет получить/установить биты ошибки обработки запроса Службой Штмпов Времени.

object.FailInfo [=Value]

**Параметры**

Value - биты ошибки обработки запроса Службой Штмпов Времени.

Константа	Описание
badAlg = 0	Неизвестный или неподдерживаемый алгоритм
badRequest = 2	Запрос запрещен или неподдерживается
badDataFormat = 5	Неверный формат даты
timeNotAvailable = 14	Служба Штампов Времени недоступна
unacceptedPolicy = 15	Неподдерживаемая политика
unacceptedExtension = 16	Неподдерживаемое расширение
addInfoNotAvailable = 17	Дополнительная информация непонятна или недоступна
systemFailure = 25	Системная ошибка

## TSPRESPONSE::HASHALGORITHM [C++]

## TSPRESPONSE::HASHALGORITHM [VBS]

[C++]

Метод HashAlgorithm позволяет получить информацию об алгоритме хеширования данных.

```
HRESULT get_HashAlgorithm(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - строка, содержащая значение OID алгоритма, описанного в библиотеке CAPICOM

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HashAlgorithm позволяет получить информацию об алгоритме хеширования данных.

```
object.HashAlgorithm ()
```

### Возвращаемое значение

Возвращаемое значение - строка, содержащая значение OID алгоритма, описанного в библиотеке CAPICOM

## TSPRESPONSE::HASHVALUE [C++]

## TSPRESPONSE::HASHVALUE [VBS]

[C++]

Метод HashValue позволяет получить значение хэша данных, на которые был выдан штамп.

```
HRESULT get_HashValue(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - строка, содержащая значение хэша данных.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.



[VBS]

Свойство HashValue позволяет получить значение хэша данных, на которые был выдан штамп.

```
object.HashValue ()
```

#### Возвращаемое значение

Возвращаемое значение - строка, содержащая значение хэша данных.

### TSPRESPONSE::PLAINDATA [C++]

### TSPRESPONSE::PLAINDATA [VBS]

[C++]

Метод PlainData позволяет установить строку данных, для которых был получен штамп времени.

```
HRESULT put_PlainData(  
    BSTR rhs  
);
```

#### Параметры

rhs [out, retval] - строка, содержащая данные.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PlainData позволяет установить строку данных, для которых был получен штамп времени.

```
object.PlainData ()
```

#### Возвращаемое значение

Возвращаемое значение - строка, содержащая данные.

### TSPRESPONSE::PLAINDATAFILE [C++]

### TSPRESPONSE::PLAINDATAFILE [VBS]

[C++]

Метод PlainDataFile позволяет установить строку пути к документу, для которого был получен штамп времени.

```
HRESULT put_PlainDataFile(  
    BSTR rhs  
);
```

#### Параметры

rhs [out, retval] - строка пути к документу.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PlainDataFile позволяет установить строку данных, для которых был получен штамп времени.

```
object.PlainDataFile ()
```

**Возвращаемое значение**

Возвращаемое значение – строка пути к документу.

**ITSPRESPONSE::POLICYID [C++]****TSPRESPONSE::POLICYID [VBS]**

[C++]

Свойство PolicyID позволяет получить политику выдачи штампа времени (в формате OID).

```
HRESULT get_PolicyID(  
    BSTR* pVal  
);
```

**Параметры**

pVal [in] – строка с политикой выдачи штампа времени (в формате OID).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство PolicyID позволяет получить политику выдачи штампа времени (в формате OID).

```
object.PolicyID ()
```

**Возвращаемые значения**

Возвращаемое значение строка с политикой выдачи штампа времени (в формате OID).

**ITSPRESPONSE::SERIALNUMBER [C++]****TSPRESPONSE::SERIALNUMBER [VBS]**

[C++]

Свойство SerialNumber позволяет получить серийный номер штампа времени .

```
HRESULT get_SerialNumber(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] – строка с серийным номером штампа времени

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство SerialNumber позволяет получить серийный номер штампа времени.

```
object.SerialNumber ()
```

**Возвращаемые значения**

Возвращаемое значение строка с серийным номером штампа времени.

**ITSPRESPONSE::STATUS [C++]****TSPRESPONSE::STATUS [VBS]**

[C++]

Свойство Status получить/установить статус обработки запроса Службой Штатпов Времени.

```
HRESULT get_Status(  
    long* pVal  
);  
HRESULT put_Status(  
    long pVal  
);
```

**Параметры**

pVal [in] - устанавливаемое битовое значение статуса обработки запроса.

pVal [out, retval] - возвращаемое битовое значение статуса обработки запроса.

Константа	Описание
granted = 0	Запрос успешно обработан , штамп возвращен клиенту
grantedWithMods = 1	Запрос обработан, штамп с изменениями возвращен клиенту
rejection = 2	Отказ обработки запроса
waiting = 3	Ожидание обработки запроса

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Status получить/установить статус обработки запроса Службой Штатпов Времени.

```
object.Status [=pVal]
```

**Параметры**

pVal - битовое значение статуса обработки запроса.

Константа	Описание
granted = 0	Запрос успешно обработан, штамп возвращен клиенту
grantedWithMods = 1	Запрос обработан, штамп с изменениями возвращен клиенту
rejection = 2	Отказ обработки запроса
waiting = 3	Ожидание обработки запроса

**ITSPRESPONSE::STATUSSTRING [C++]****TSPRESPONSE::STATUSSTRING [VBS]**

[C++]

Свойство StatusString позволяет получить/установить пользовательское описание статуса обработки запроса Службой Штатпов Времени.

```
HRESULT get_StatusString(  
    BSTR* pVal  
);  
HRESULT put_StatusString(  
    BSTR pVal  
);
```

**Параметры**

pVal [in] - строка с пользовательским описанием статуса обработки запроса Службой Штатпов Времени.

pVal [out, retval] - указатель на строку с пользовательским описанием статуса обработки запроса Службой Штатпов Времени.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство StatusString позволяет получить/установить пользовательское описание статуса обработки запроса Службой Штатпов Времени.

```
object.StatusString [=pVal]
```

**Параметры**

pVal - строка с пользовательским описанием статуса обработки запроса Службой Штатпов Времени.

**ITSPRESPONSE::TIME [C++]****TSPRESPONSE::TIME [VBS]**

[C++]

Свойство Time позволяет получить время выдачи штампа времени в локальном представлении.

```
HRESULT get_Time(  
    DATE* pVal  
);
```

**Параметры**

pNewTSPProfile [out, retval] - время выдачи штампа.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство **Time** позволяет получить время выдачи штампа времени в локальном представлении.

```
object.Time ()
```

**Возвращаемые значения**

Возвращаемое значение время выдачи штампа.

## ITSPRESPONSE::TIMESTAMP\_TOKEN [C++]

## TSPRESPONSE::TIMESTAMP\_TOKEN [VBS]

[C++]

Свойство TimeStampToken позволяет установить блок штампа времени.

```
HRESULT put_TimeStampToken(  
    BSTR rhs  
);
```

### Параметры

rhs [in] - строка блока штампа времени.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TimeStampToken позволяет установить блок штампа времени.

```
object.TimeStampToken (rhs)
```

### Параметры

rhs - строка блока штампа времени.

## ITSPRESPONSE::TSACERTIFICATE [C++]

## TSPRESPONSE::TSACERTIFICATE [VBS]

[C++]

Свойство TSACertificate позволяет получить сертификат подписи Службы Штamped Времени из штампа времени.

```
HRESULT get_TSACertificate(  
    BSTR* pVal  
);
```

### Параметры

pVal [out, retval] - указатель на строку с блоком сертификата подписи Службы Штamped Времени из штампа времени.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство TSACertificate позволяет получить сертификат подписи Службы Штamped Времени из штампа времени .

```
object.TSACertificate ()
```

### Возвращаемые значения

Возвращаемое значение указатель на строку с блоком сертификата подписи Службы Штamped Времени из штампа времени.

**ITSPRESPONSE::TSANAME [C++]****TSPRESPONSE::TSANAME [VBS]****[C++]**

Свойство TSAName позволяет получить имя Службы Штампа Времени из штампа времени.

```
HRESULT get_TSAName(  
    BSTR* pVal  
);
```

**Параметры**

pVal [out, retval] - указатель строку с именем Службы Штампа Времени.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство TSAName позволяет получить имя Службы Штампа Времени из штампа времени.

```
object.TSAName (bstrID)
```

**Параметры**

bstrID - идентификатор профиля.

**Возвращаемые значения**

Возвращаемое значение указатель строку с именем Службы Штампа Времени.

**TSPUTIL (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Свойства**

Свойство	Описание
License	Свойство позволяет получить\установить лицензию на модуль TSP
Version	Свойство позволяет получить версию на модуль TSP

**ITSPUTIL::LICENSE [C++]****TSPUTIL::LICENSE [VBS]****[C++]**

Свойство License позволяет получить/ установить лицензию на КриптоАРМ модуль TSP и на модуль КриптоПро TSP Клиент.

```
HRESULT get_License(  
    COMPONENT_TYPE  enmComponentType  
    BSTR* pVal,  
);  
HRESULT put_License(  
    COMPONENT_TYPE  enmComponentType  
    BSTR pVal,  
);
```

**Параметры**

enmComponentType [in] - тип модуля. Принимает значения из набора COMPONENT\_TYPE.  
pVal [out, retval] - строка, содержащую лицензионный ключ для модуля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство License позволяет получить/ установить лицензию на КриптоАРМ модуль TSP и на модуль КриптоПро TSP Клиент.

```
object.License(enmComponentType, =Value)
```

**Параметры**

enmComponentType - тип модуля. Принимает значения из набора COMPONENT\_TYPE.  
pVal - строку, содержащую лицензионный ключ для модуля.

**Возвращаемое значение**

Указатель на строку, содержащую лицензию на ПО «КриптоАРМ».

**ITSPUTIL::VERSION [C++]****TSPUTIL::VERSION [VBS]****[C++]**

Свойство Version позволяет получить строку, содержащую версию КриптоАРМ модуль TSP и версию модуля КриптоПро TSP Клиент.

```
HRESULT get_Version(  

```

**COMPONENT\_TYPE enmComponentType**

```
BSTR* pVal,  
);
```

**Параметры**

enmComponentType [in, optional, defaultvalue(0)] - тип модуля. Принимает значения из набора COMPONENT\_TYPE.

pVal [out, retval] - возвращает строку, содержащую версию ПО "КриптоАРМ"

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство Version позволяет получить строку, содержащую версию КриптоАРМ модуль TSP и версию модуля КриптоПро TSP Клиент

```
object.Version
```

**Параметры**

enmComponentType - тип модуля. Принимает значения из набора COMPONENT\_TYPE.

**Возвращаемое значение**

Возвращает строку, содержащую версию ПО «КриптоАРМ».



## РАЗДЕЛ 3. ОПИСАНИЕ COM DIGTOCSPCLIENT

### 3.1. СВОЙСТВА И МЕТОДЫ

Компонент DigtOCSPClient предоставляет интерфейсы для отправки запросов на проверку статуса сертификата в Службе Актуальных статусов.

Интерфейс	Описание
<a href="#">OCSPClient</a>	Описывает интерфейс отправки запроса в Службу актуальных статусов
<a href="#">OCSPProfile</a>	Описывает настройки параметров запроса и доступа к Службе актуальных статусов
<a href="#">OCSPProfiles</a>	Описывает коллекцию настроек параметров запроса и доступа к Службе актуальных статусов
<a href="#">OCSPProfileStore</a>	Описывает хранилище настроек
<a href="#">OCSPResponse</a>	Описывает работу и получение параметров OCSP ответа
<a href="#">OCSPResponses</a>	Описывает коллекцию OCSP ответов
<a href="#">OCSPSingleResponse</a>	Описывает параметры OCSP ответа, содержащего информацию об одном сертификате.

### 3.2. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

#### 3.2.1. СОЗДАНИЕ И СОХРАНЕНИЕ OCSP ПРОФИЛЯ

Данный пример позволяет создать новый OCSP профиль, заполнить его параметры и сохранить новый профиль в хранилище OCSP профилей. А так же осуществить поиск данного профиля в хранилище профилей и получить свойства найденного профиля.

```
Option explicit
'enum OCSP_AUTH_TYPE
const OCSP_AUTH_TYPE_ANONYMOUS = 0
const OCSP_AUTH_TYPE_BASIC = 1
const OCSP_AUTH_TYPE_DIGEST = 2
const OCSP_AUTH_TYPE_NTLM = 3
const OCSP_AUTH_TYPE_NEGOTIATE = 4
Dim oOCSPProfileStore, oOCSPProfile, oOCSPProfileFinded, oOCSPProfileFinded1
'Создадим новый профиль с настройками доступа к OCSP службе
Set oOCSPProfile = CreateObject("DigtOCSPClient.OCSPProfile")
'Здесь установим параметры профиля
oOCSPProfile.OCSPAddress = "http://www.cryptopro.ru/ocspnc/ocsp.srf"
oOCSPProfile.OCSPAuthType = OCSP_AUTH_TYPE_ANONYMOUS
oOCSPProfile.OCSPUserName = ""
oOCSPProfile.OCSPPassword = ""
oOCSPProfile.ProxyAddress = ""
oOCSPProfile.ProxyAuthType = OCSP_AUTH_TYPE_DIGEST
oOCSPProfile.ProxyUserName = "UserName"
oOCSPProfile.ProxyPassword = "UserPassword"
oOCSPProfile.UseProxy = true
oOCSPProfile.UseClientCertificate = false
'Добавим новый профиль в хранилище OCSP профилей
Set oOCSPProfileStore = CreateObject ("DigtOCSPClient.OCSPProfileStore")
oOCSPProfileStore.Open
oOCSPProfileStore.Add oOCSPProfile
oOCSPProfileStore.Save
'Поиск профиля
'Получим идентификатор созданного профиля
Dim sOCSPProfileID
sOCSPProfileID = oOCSPProfile.OCSPProfileID
'Найдем профиль по идентификатору
Set oOCSPProfileFinded = oOCSPProfileStore.Profile (sOCSPProfileID)
```

```
'Получим некоторые параметры найденного профиля
Dim GetProfileProps
GetProfileProps = ""
GetProfileProps = GetProfileProps + "OCSPAddress = " +
oOCSPProfile.OCSPAddress + Chr(13)
GetProfileProps = GetProfileProps + "OCSPAuthType = " +
CStr(oOCSPProfile.OCSPAuthType) + Chr(13)
GetProfileProps = GetProfileProps + "ProxyAuthType = " +
CStr(oOCSPProfile.ProxyAuthType) + Chr(13)
GetProfileProps = GetProfileProps + "ProxyUserName = " +
oOCSPProfile.ProxyUserName + Chr(13)
GetProfileProps = GetProfileProps + "ProxyPassword = " +
oOCSPProfile.ProxyPassword + Chr(13)
GetProfileProps = GetProfileProps + "UseProxy = " +
CStr(oOCSPProfile.UseProxy) + Chr(13)
GetProfileProps = GetProfileProps + "UseClientCertificate = " +
CStr(oOCSPProfile.UseClientCertificate) + Chr(13)
GetProfileProps = GetProfileProps + "OCSPProfileID = " +
oOCSPProfile.OCSPProfileID + Chr(13)
MsgBox GetProfileProps
```

### 3.2.2. ПОЛУЧЕНИЕ OCSP ОТВЕТА И ЕГО СВОЙСТВА

Данный пример позволяет отправить запрос в службу OCSP на сертификат, посмотреть свойства полученного ответа и статус сертификата. Запрос отправляется в тестовую службу актуальных статусов.

```
Option explicit
'enum OCSP_AUTH_TYPE
const OCSP_AUTH_TYPE_ANONYMOUS = 0
const OCSP_AUTH_TYPE_BASIC = 1
const OCSP_AUTH_TYPE_DIGEST = 2
const OCSP_AUTH_TYPE_NTLM = 3
const OCSP_AUTH_TYPE_NEGOTIATE = 4
'enum FORMAT
const BASE64_TYPE = 0
const DER_TYPE = 1
Dim oOCSPProfileStore, oOCSPProfile, oOCSPProfileFinded, oOCSPProfileFinded1
'Создадим новый профиль с настройками доступа к OCSP службе
Set oOCSPProfile = CreateObject("DigtOCSPClient.OCSPProfile")
'Здесь установим параметры профиля
oOCSPProfile.OCSPAddress = "http://www.cryptopro.ru/ocspnc/ocsp.srf"
oOCSPProfile.OCSPAuthType = OCSP_AUTH_TYPE_ANONYMOUS
oOCSPProfile.OCSPUserName = ""
oOCSPProfile.OCSPPassword = ""
oOCSPProfile.ProxyAddress = ""
oOCSPProfile.ProxyAuthType = OCSP_AUTH_TYPE_DIGEST
oOCSPProfile.ProxyUserName = "UserName"
oOCSPProfile.ProxyPassword = "UserPassword"
oOCSPProfile.UseProxy = true
oOCSPProfile.UseClientCertificate = false
'сертификат, который будем проверять
Dim oCertificate : Set oCertificate = CreateObject("DigtCrypto.Certificate")
oCertificate.Load("C:\test1.cer")
Dim oOCSPClient : Set oOCSPClient = CreateObject("DigtOCSPClient.OCSPClient")
oOCSPClient.OCSPProfile = oOCSPProfile
oOCSPClient.Certificate = oCertificate.Export(BASE64_TYPE)
'отсылаем запрос
oOCSPClient.Send
if oOCSPClient.HTTPStatus = 200 then
Dim oOCSPResponse
set oOCSPResponse = oOCSPClient.OCSPResponse
```

```
Dim strResponseProps
strResponseProps = ""

'дата выпуска
strResponseProps = "Produce date: " &
FormatDateTime(oOCSPResponse.ProducedAt, vbLongDate)
strResponseProps = strResponseProps & vbCrLf & "Produce time: " &
FormatDateTime(oOCSPResponse.ProducedAt, vbLongTime)
Dim oResponderCertificate : Set oResponderCertificate =
CreateObject("DigtCrypto.Certificate")
oResponderCertificate.Import( oOCSPResponse.ResponderCertificate )

'информация о сертификате OCSP службы
strResponseProps = strResponseProps & vbCrLf
strResponseProps = strResponseProps & vbCrLf & "Responder certificate owner:
" & oResponderCertificate.SubjectName
strResponseProps = strResponseProps & vbCrLf & "Responder certificate issuer:
" & oResponderCertificate.IssuerName
strResponseProps = strResponseProps & vbCrLf & "Responder certificate serial
number: " & oResponderCertificate.SerialNumber

Dim oSingleResponse
set oSingleResponse = oOCSPResponse.SingleResponse(0)

'статус проверяемого сертификата, возвращенные OCSP службой
strResponseProps = strResponseProps & vbCrLf
strResponseProps = strResponseProps & vbCrLf & "Certificate status: " &
oSingleResponse.CertificateStatus

MsgBox strResponseProps, vbOKOnly, "Response properties"
else
MsgBox "Ststus: " & oOCSPClient.HTTPStatus, vbOKOnly or vbExclamation, "Error"
end if
```

### 3.3. ИНТЕРФЕЙСЫ DIGTOCSPCLIENT

#### 3.3.1. КОНСТАНТЫ И ПЕРЕЧИСЛЕНИЯ

Константа	Описание
<a href="#">OCSP_AUTH_TYPE</a>	Тип аутентификации Службы актуальных статусов
<a href="#">OCSP_CERTIFICATE_STATUS_TYPE</a>	Статус сертификата, полученный от Службы актуальных статусов
<a href="#">OCSP_COMPONENT_TYPE</a>	Формат ответа из Службы Штампов Времени
<a href="#">OCSP_PROFILE_STORE_TYPE</a>	Тип хранилища профиля хранения настроек доступа к Службе актуальных статусов
<a href="#">OCSP_RESPONSE_FORMAT</a>	Тип аутентификации Службы Штампов Времени

#### OCSP\_AUTH\_TYPE

Значение	Описание
OCSP_AUTH_TYPE_ANONYMOUS = 0	Анонимный доступ
OCSP_AUTH_TYPE_BASIC = 1	Обычная аутентификация
OCSP_AUTH_TYPE_DIGEST = 2	Краткая проверка подлинности
OCSP_AUTH_TYPE_NTLM = 3	Встроенная проверка подлинности Windows
OCSP_AUTH_TYPE_NEGOTIATE = 4	Встроенная проверка подлинности Windows или Kerberos

**OCSP\_CERTIFICATE\_STATUS\_TYPE**

Значение	Описание
OCSP_CERTIFICATE_STATUS_GOOD = 0	Сертификат действителен
OCSP_CERTIFICATE_STATUS_REVOKED = 1	Сертификат отозван
OCSP_CERTIFICATE_STATUS_UNKNOWN = 2	Статус сертификата неизвестен

**OCSP\_COMPONENT\_TYPE**

Значение	Описание
DIGT_OCSP_CLIENT = 0	КриптоАРМ Модуль OCSP
CP_OCSP_CLIENT = 1	Модуль КриптоПро OCSP Клиент

**OCSP\_PROFILE\_STORE\_TYPE**

Значение	Описание
OCSP_REGISTRY_STORE = 0	Тип хранилища - Реестр Windows
OCSP_XML_STORE = 1	Тип хранилища - XML файл

**OCSP\_RESPONSE\_FORMAT**

Значение	Описание
OCSP_FORMAT_DER = 0	Формат DER
OCSP_FORMAT_BASE64 = 1	Формат BASE64

**3.3.2. ОБЪЕКТЫ И ИНТЕРФЕЙСЫ**

Компонент DigtOCSPClient предоставляет интерфейсы для отправки запросов на проверку статуса сертификата в Службе Актуальных статусов.

Интерфейс	Описание
<a href="#">OCSPClient</a>	Описывает интерфейс отправки запроса в Службу актуальных статусов
<a href="#">OCSPProfile</a>	Описывает настройки параметров запроса и доступа к Службе актуальных статусов
<a href="#">OCSPProfiles</a>	Описывает коллекцию настроек параметров запроса и доступа к Службе актуальных статусов
<a href="#">OCSPProfileStore</a>	Описывает хранилище настроек
<a href="#">OCSPResponse</a>	Описывает работу и получение параметров OCSP ответа
<a href="#">OCSPResponses</a>	Описывает коллекцию OCSP ответов
<a href="#">OCSPSingleResponse</a>	Описывает параметры OCSP ответа, содержащего информацию об одном сертификате.

**OCSPPCLient (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
<a href="#">Send</a>	Метод позволяет отправить запрос в Службу актуальных статусов и получить ответ

**Свойства**

Свойство	Описание
<a href="#">Certificate</a>	Свойство позволяет получить/установить сертификат, статус которого необходимо проверить в Службе актуальных статусов
<a href="#">HTTPStatus</a>	Свойство позволяет получить/установить http-статус отправки запроса в Службу актуальных статусов
<a href="#">OCSPProfile</a>	Свойство позволяет получить/установить профиль с настройками
<a href="#">OCSPResponse</a>	Свойство позволяет получить/установить результат обработки запроса Службой актуальных статусов

**OCSPPCLient::SEND [C++][VBS]****[C++]**

Метод Send позволяет отправить запрос в Службу актуальных статусов и получить ответ.

```
HRESULT Send(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Send позволяет отправить запрос в Службу актуальных статусов и получить ответ.

```
object.Send
```

**OCSPPCLient::CERTIFICATE [C++][VBS]****[C++]**

Свойство Certificate позволяет получить/установить сертификат, статус которого необходимо проверить в Службе актуальных статусов.

```
HRESULT get_Certificate(  
    BSTR* pVal  
);  
HRESULT put_Certificate(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] Указатель на строку с сертификатом.  
newVal [in] Строка с сертификатом.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство Certificate позволяет получить/установить сертификат, статус которого необходимо проверить с Службе актуальных статусов.

```
object.Certificate [= Value]
```

#### Параметры

Value Строка с сертификатом.

### OCSPCLIENT::HTTPSTATUS [C++][VBS]

[C++]

Свойство HTTPStatus позволяет получить/установить http-статус отправки запроса в Службу актуальных статусов.

```
HRESULT get_HTTPStatus(  
    long* pVal  
);  
HRESULT put_HTTPStatus(  
    long pVal  
);
```

#### Параметры

pVal [out, retval] Значение http-статуса отправки запроса.

pVal [in] Значение http-статуса отправки запроса.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство HTTPStatus позволяет получить/установить http-статус отправки запроса в Службу актуальных статусов.

```
object.HTTPStatus [= Value]
```

#### Параметры

Value Значение http-статуса отправки запроса.

### OCSPCLIENT::OCSPPROFILE [C++][VBS]

[C++]

Свойство OCSPProfile позволяет получить/установить профиль.

```
HRESULT get_OCSPProfile(  
    IOCSPPProfile** pVal  
);  
HRESULT put_OCSPProfile(  
    IOCSPPProfile* pVal  
);
```

#### Параметры

pVal [out, retval] Указатель на указатель на интерфейс IOCSPPProfile.

pVal [in] Указатель на интерфейс IOCSPPProfile.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPProfile позволяет получить/установить профиль.

```
object.OCSPProfile [= Value]
```

#### Параметры

Value Указатель на объект OCSPProfile.

### OCSPCLIENT::OCSPRESPONSE [C++][VBS]

[C++]

Свойство OCSPResponse позволяет получить/установить результат обработки запроса Службой актуальных статусов.

```
HRESULT get_OCSPResponse(  
    IOCSPPResponse** pVal  
);  
HRESULT put_OCSPResponse(  
    IOCSPPResponse* pVal  
);
```

#### Параметры

pVal  
[out, retval] Указатель на указатель на интерфейс IOCSPPResponse.  
pVal  
[in] Указатель на интерфейс IOCSPPResponse.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPResponse позволяет получить/установить результат обработки запроса Службой актуальных статусов.

```
object.OCSPResponse [= Value]
```

#### Параметры

Value Указатель на интерфейс IOCSPPResponse.

**OCSPPROFILE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
<a href="#">Clone</a>	Метод позволяет создать копию профиля

**Свойства**

Свойство	Описание
<a href="#">ClientCertificate</a>	Свойство позволяет получить/установить сертификат для клиентской аутентификации
<a href="#">ProxyAddress</a>	Свойство позволяет получить/установить адрес прокси-сервера
<a href="#">ProxyAuthType</a>	Свойство позволяет получить/установить тип аутентификации на прокси-сервере
<a href="#">ProxyPassword</a>	Свойство позволяет получить/установить пароль для аутентификации на прокси-сервере
<a href="#">ProxyUserName</a>	Свойство позволяет получить/установить имя пользователя для аутентификации на прокси-сервере
<a href="#">OCSPAddress</a>	Свойство позволяет получить/установить адрес сервера Службы актуальных статусов
<a href="#">OCSPAuthType</a>	Свойство позволяет получить/установить тип аутентификации доступа к Службе актуальных статусов
<a href="#">OCSPPassword</a>	Свойство позволяет получить/установить пароль для аутентификации доступа к Службе актуальных статусов
<a href="#">OCSPUserName</a>	Свойство позволяет получить/установить имя пользователя для аутентификации доступа к Службе актуальных статусов
<a href="#">OCSPProfileID</a>	Свойство позволяет получить/установить идентификатор OSCP-профиля
<a href="#">UseClientCertificate</a>	Свойство позволяет получить/установить флаг использования клиентского сертификата для аутентификации
<a href="#">UseProxy</a>	Свойство позволяет получить/установить флаг использования прокси-сервера для обращения к Службе актуальных статусов

**IOCSPPROFILE::CLONE [C++][VBS]****[C++]**

Метод Clone позволяет создать копию профиля.

```
HRESULT Clone(  
    IOCSPPProfile** pVal  
);
```

**Параметры**

pVal [out, retval] Возвращаемое значение – профиль, являющийся копией текущего профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод Clone позволяет создать копию профиля.

```
object.Clone()
```

**Возвращаемое значение**



Возвращаемое значение – профиль, являющийся копией текущего профиля.

## OCSPPROFILE::CLIENTCERTIFICATE [C++][VBS]

### [C++]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище.

```
HRESULT get_ClientCertificate(  
    BSTR* pVal  
);  
HRESULT put_ClientCertificate(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] Указатель на строку с сертификатом клиентской аутентификации.

pVal [in] Строка с сертификатом клиентской аутентификации.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### [VBS]

Свойство ClientCertificate позволяет получить/установить сертификат для клиентской аутентификации. Сертификат должен находиться в личном хранилище .

```
object.ClientCertificate [= Value]
```

### Параметры

Value Строка с сертификатом клиентской аутентификации.

## OCSPPROFILE::PROXYADDRESS [C++][VBS]

### [C++]

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

```
HRESULT get_ProxyAddress(  
    BSTR* pVal  
);  
HRESULT put_ProxyAddress(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] Указатель на строку с адресом прокси-сервера.

pVal [in] Строка с адресом прокси-сервера.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

### [VBS]

Свойство ProxyAddress позволяет получить/установить адрес прокси-сервера.

```
object.ProxyAddress [= Value]
```

### Параметры

Value Строка с адресом прокси-сервера.

## OCSPPROFILE::PROXYAUTHTYPE [C++][VBS]

[C++]

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

```
HRESULT get_ProxyAuthType(  
    OCSP_AUTH_TYPE* pVal  
);  
HRESULT put_ProxyAuthType(  
    OCSP_AUTH_TYPE pVal  
);
```

### Параметры

pVal [out, retval] Возвращаемое значение – тип аутентификации. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

pVal [in] Тип аутентификации. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyAuthType позволяет получить/установить тип аутентификации на прокси-сервере.

object.ProxyAuthType [= Value]

### Параметры

Value Тип аутентификации. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

## OCSPPROFILE::PROXYPASSWORD [C++][VBS]

[C++]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
HRESULT get_ProxyPassword(  
    BSTR* pVal  
);  
HRESULT put_ProxyPassword(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] Указатель на строку пароля для аутентификации на прокси-сервере.

pVal [in] Строка пароля для аутентификации на прокси-сервере.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyPassword позволяет получить/установить пароль для аутентификации на прокси-сервере.

```
object.ProxyPassword [= Value]
```

### Параметры

Value Строка пароля для аутентификации на прокси-сервере.

## OCSPPROFILE::PROXYUSERNAME [C++][VBS]

[C++]

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
HRESULT get_ProxyUserName(  
    BSTR* pVal  
);  
HRESULT put_ProxyUserName(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] Указатель на строку с именем пользователя для аутентификации на прокси-сервере.

pVal [in] Строка с именем пользователя для аутентификации на прокси-сервере.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProxyUserName позволяет получить/установить имя пользователя для аутентификации на прокси-сервере.

```
object.ProxyUserName [= Value]
```

### Параметры

Value Строка с именем пользователя для аутентификации на прокси-сервере.

## OCSPPROFILE::OCSPADDRESS [C++][VBS]

[C++]

Свойство OCSPAddress позволяет получить/установить адрес сервера Службы актуальных статусов.

```
HRESULT get_OCSPAddress(  
    BSTR* pVal  
);  
HRESULT put_OCSPAddress(  
    BSTR pVal  
);
```

### Параметры

pVal [out, retval] Указатель на строку адреса сервера Службы актуальных статусов.

pVal [in] Строка адреса сервера Службы актуальных статусов.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPAddress позволяет получить/установить адрес сервера Службы актуальных статусов.

```
object.OCSPAddress [= Value]
```

#### Параметры

Value

Строка адреса сервера Службы актуальных статусов.

### OCSPPROFILE::OCSPAUTHTYPE [C++][VBS]

[C++]

Свойство OCSPAuthType позволяет получить/установить тип аутентификации доступа к Службе актуальных статусов.

```
HRESULT get_OCSPAuthType(  
    OCSP_AUTH_TYPE* pVal  
);  
HRESULT put_OCSPAuthType(  
    OCSP_AUTH_TYPE pVal  
);
```

#### Параметры

pVal [out, retval] Возвращаемое значение – тип аутентификации доступа к Службе актуальных статусов. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

pVal [in] Тип аутентификации доступа к Службе актуальных статусов. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPAuthType позволяет получить/установить тип аутентификации доступа к Службе актуальных статусов.

```
object.OCSPAuthType [= Value]
```

#### Параметры

Value

Тип аутентификации доступа к Службе актуальных статусов. Допустимыми значениями являются значения OCSP\_AUTH\_TYPE.

### OCSPPROFILE::OCSPPASSWORD [C++][VBS]

[C++]

Свойство OCSPPassword позволяет получить/установить пароль для аутентификации доступа к Службе актуальных статусов.

```
HRESULT get_OCSPPassword(  
    BSTR* pVal  
);  
HRESULT put_OCSPPassword(  
    BSTR pVal  
);
```

#### Параметры

pVal [out, retval] Указатель строку пароля для аутентификации доступа к Службе актуальных статусов.  
pVal [in] Строка пароля для аутентификации доступа к Службе актуальных статусов.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство OCSPPassword позволяет получить/установить пароль для аутентификации доступа к Службе актуальных статусов.

object.OCSPPassword [= Value]

### Параметры

Value Строка пароля для аутентификации доступа к Службе актуальных статусов.

**OCSPPROFILE::OCSPUSERNAME [C++][VBS]****[C++]**

Свойство OCSPUserName позволяет получить/установить имя пользователя для аутентификации доступа к Службе актуальных статусов.

```
HRESULT get_OCSPUserName(  
    BSTR* pVal  
);  
HRESULT put_OCSPUserName(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] Указатель на строку с именем пользователя для аутентификации доступа к Службе актуальных статусов.

pVal [in] Стока с именем пользователя для аутентификации доступа к Службе актуальных статусов.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство OCSPUserName позволяет получить/установить имя пользователя для аутентификации доступа к Службе актуальных статусов.

```
object.OCSPUserName [= Value]
```

**Параметры**

Value Стока с именем пользователя для аутентификации доступа к Службе актуальных статусов.

**OCSPPROFILE::OCSPPROFILEID [C++][VBS]****[C++]**

Свойство OCSPProfileID позволяет получить/установить идентификатор OSCP-профиля.

```
HRESULT get_OCSPProfileID(  
    BSTR* pVal  
);  
HRESULT put_OCSPProfileID(  
    BSTR pVal  
);
```

**Параметры**

pVal [out, retval] Указатель на строку идентификатора OSCP-профиля.

pVal [in] Строка идентификатора OSCP-профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Свойство OCSPProfileID позволяет получить/установить идентификатор OSCP-профиля.

```
object.OCSPProfileID [= Value]
```

## Параметры

Value Строка идентификатора OSCP-профиля.

## OCSPPROFILE::USECLIENTCERTIFICATE [C++][VBS]

[C++]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
HRESULT get_UseClientCertificate(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseClientCertificate(  
    VARIANT_BOOL pVal  
);
```

## Параметры

pVal [out, retval] Возвращает TRUE или FALSE. TRUE – использовать клиентский сертификат для аутентификации, FALSE – не использовать.

pVal [in] Принимает TRUE или FALSE. TRUE – использовать клиентский сертификат для аутентификации, FALSE – не использовать.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseClientCertificate позволяет получить/установить флаг использования клиентского сертификата для аутентификации.

```
object.UseClientCertificate [= Value]
```

## Параметры

Value TRUE или FALSE. TRUE – использовать клиентский сертификат для аутентификации, FALSE – не использовать.

## OCSPPROFILE::USEPROXY [C++][VBS]

[C++]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Службе актуальных статусов.

```
HRESULT get_UseProxy(  
    VARIANT_BOOL* pVal  
);  
HRESULT put_UseProxy(  
    VARIANT_BOOL pVal  
);
```

## Параметры

pVal [out, retval] Возвращает TRUE или FALSE. TRUE – использовать прокси-сервер, FALSE – не использовать.

pVal [in] Принимает TRUE или FALSE. TRUE – использовать прокси-сервер, FALSE – не использовать.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство UseProxy позволяет получить/установить флаг использования прокси-сервера для обращения к Службе актуальных статусов.

```
object.UseProxy [= Value]
```

### Параметры

Value TRUE или FALSE. TRUE – использовать прокси-сервер, FALSE – не использовать.



OCSPPROFILES (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)

Методы

Метод	Описание
<a href="#">Add</a>	Метод позволяет добавить профиль в коллекцию
<a href="#">Clear</a>	Метод позволяет очистить коллекцию
<a href="#">Remove</a>	Метод позволяет удалить профиль из коллекции

Свойства

Свойство	Описание
<a href="#">Count</a>	Свойство позволяет получить число профилей в коллекции
<a href="#">Item</a>	Свойство позволяет получить указатель на элемент коллекции

IOCSPPROFILES::ADD [C++]

OCSPPROFILES::ADD [VBS]

[C++]

Метод Add позволяет добавить профиль в коллекцию.

```
HRESULT Add(  
    IOCSPPProfile* addVal  
);
```

Параметры

addVal [in] Добавляемый элемент. Указатель на интерфейс IOCSPPProfile.

Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить профиль в коллекцию.

```
object.Add(addVal)
```

Параметры

addVal Добавляемый элемент. Указатель на объект Profile.

IOCSPPROFILES::CLEAR [C++]

OCSPPROFILES::CLEAR [VBS]

[C++]

Метод Clear позволяет очистить коллекцию профилей.

```
HRESULT Clear(  
);
```

Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Clear позволяет очистить коллекцию профилей.

```
object.Clear()
```

## IOCSPPROFILES::REMOVE [C++]

## OCSPPROFILES::REMOVE [VBS]

[C++]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
HRESULT Remove(  
    VARIANT varIndex  
);
```

### Параметры

varIndex [in] Номер элемента в коллекции минус 1 [0 .. Count-1].

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Remove позволяет удалить из коллекции элемент (профиль) с указанным индексом.

```
object.Remove(varIndex)
```

### Параметры

varIndex Номер элемента в коллекции минус 1 [0 .. Count-1].

## IOCSPPROFILES::COUNT [C++]

## OCSPPROFILES::COUNT [VBS]

[C++]

Свойство Count позволяет получить количество профилей в коллекции.

```
HRESULT get_Count(  
    long* pVal  
);
```

### Параметры

pVal [out, retval] Количество профилей в коллекции.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Count позволяет получить количество профилей в коллекции.

```
object.Count
```

### Возвращаемое значение

Количество профилей в коллекции.

## IOCSPPROFILES::ITEM [C++]

## OCSPPROFILES::Item [VBS]

[C++]

Свойство Item позволяет получить элемент коллекции.

```
HRESULT get_Item(  
    long lIndex,  
    IOCSPPProfile** pVal  
);
```

### Параметры

lIndex [in] Номер элемента в коллекции минус 1 [0 .. Count-1].  
pVal [out, retval] Указатель на IOCSPPProfile.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Item позволяет получить элемент коллекции.

```
object.Item (lIndex)
```

### Параметры

lIndex Номер элемента в коллекции минус 1 [0 .. Count-1].

### Возвращаемое значение

Указатель на OCSPPProfile.

OCSPPROFILESTORE (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)

Методы

Метод	Описание
<a href="#">Add</a>	Метод позволяет добавить новый OSCP профиль в хранилище
<a href="#">Close</a>	Метод позволяет закрыть хранилище OSCP профилей
<a href="#">Open</a>	Метод позволяет открыть хранилище OSCP профилей
<a href="#">Remove</a>	Метод позволяет удалить OSCP профиль из хранилища
<a href="#">Save</a>	Метод позволяет сохранить хранилище OSCP профилей в реестр

Свойства

Свойство	Описание
<a href="#">Profile</a>	Свойство позволяет найти профиль в хранилище по идентификатору
<a href="#">Store</a>	Свойство позволяет получить/установить коллекцию OSCP профилей, содержащуюся в хранилище

IOCSPPROFILESTORE::ADD [C++]

OCSPPROFILESTORE::ADD [VBS]

[C++]

Метод Add позволяет добавить профиль в хранилище.

```
HRESULT Add(  
    IOCSPPProfile*  pNewOCSPProfile  
);
```

Параметры

pNewOCSPProfile  
[in] Указатель на интерфейс IOCSPPProfile.

Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Add позволяет добавить профиль в хранилище.

```
object.Add (Value)
```

Параметры

Value Указатель на интерфейс IOCSPPProfile.

IOCSPPROFILESTORE::CLOSE [C++]

OCSPPROFILESTORE::CLOSE [VBS]

[C++]

Метод Close позволяет закрыть хранилище.

```
HRESULT Close(  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Close позволяет закрыть хранилище.

```
object.Close ()
```

**IOCSPPROFILESTORE::OPEN [C++]****OCSPPROFILESTORE::OPEN [VBS]**

[C++]

Метод Open позволяет открыть хранилище.

```
HRESULT Open(  
    OCSPPROFILESTORE_TYPE stStoreType  
    BSTR bstrPath  
);
```

**Параметры**

stStoreType [[in, defaultvalue(0)] Тип хранилища. Принимает значения из OCSPPROFILESTORE\_TYPE. По умолчанию принимает значение OCSPPROFILESTORE\_REGISTRY\_STORE.  
bstrPath [in, defaultvalue("")] Строка, содержащая путь к XML файлу.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Open позволяет открыть хранилище.

```
object.Open (stStoreType, bstrPath)
```

**Параметры**

stStoreType Тип хранилища. Принимает значения из OCSPPROFILESTORE\_TYPE. По умолчанию принимает значение OCSPPROFILESTORE\_REGISTRY\_STORE  
bstrPath Строка, содержащая путь к XML файлу. По умолчанию принимает значение "".

**IOCSPPROFILESTORE::REMOVE [C++]****OCSPPROFILESTORE::REMOVE [VBS]**

[C++]

Метод Remove позволяет удалить профиль из хранилища по идентификатору.

```
HRESULT Remove(  
    BSTR bstrID  
);
```

**Параметры**

bstrID [in] Строка с идентификатором профиля.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**Примечание**

Не реализован.

[VBS]

Метод Remove позволяет удалить профиль из хранилища по идентификатору.

```
object.Remove (Value)
```

**Параметры**

Value Строка с идентификатором профиля.

**Примечание**

Не реализован.

**IOCSPPROFILESTORE::SAVE [C++]****OCSPPROFILESTORE::SAVE [VBS]**

[C++]

Метод Save позволяет сохранить изменения в хранилище профилей.

```
HRESULT Save (  
);
```

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить изменения в хранилище профилей.

```
object.Save ()
```

**IOCSPPROFILESTORE::PROFILE [C++]****OCSPPROFILESTORE::PROFILE [VBS]**

[C++]

Свойство Profile позволяет найти профиль в хранилище по идентификатору.

```
HRESULT get_Profile(  
    BSTR bstrID  
    IOCSPPProfile** pVal  
);
```

**Параметры**

bstrID [in] Идентификатор профиля.

pVal [out, retval] Указатель на указатель на интерфейс IOCSPPProfile. В случае отсутствия профиля с указанным идентификатором возвращается пустая ссылка.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Profile позволяет найти профиль в хранилище по идентификатору.

```
object.Profile (bstrID)
```

## Параметры

bstrID Идентификатор профиля.

## Возвращаемые значения

Возвращаемое значение указатель на указатель на интерфейс IOCSPPProfile. В случае отсутствия профиля с указанным идентификатором возвращается "Nothing".

## IOCSPPROFILESTORE::STORE [C++]

## OCSPPROFILESTORE::STORE [VBS]

[C++]

Свойство Store позволяет получить/установить коллекцию OSCP профилей, содержащуюся в хранилище.

```
HRESULT get_Store(  
    IOCSPPProfiles** pVal  
);  
HRESULT put_Store(  
    IOCSPPProfiles* pVal  
);
```

## Параметры

pVal [out, retval] Указатель на коллекцию OSCP профилей.  
pVal [in] Указатель на коллекцию OSCP профилей.

## Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство Store позволяет получить/установить коллекцию OSCP профилей, содержащуюся в хранилище.

```
object.Store [= pVal]
```

## Параметры

pVal Указатель на коллекцию OSCP профилей.

**OCSPResponse (GUID: 4B115A3E-2212-483A-B7AB-8FDE57BE2586)****Методы**

Метод	Описание
<a href="#">AddCertificateForSearch</a>	Метод позволяет добавить сертификат к списку сертификатов, среди которых осуществляется поиск в свойстве ResponderCertificate
<a href="#">Export</a>	Метод позволяет экспортировать ответ Службы актуальных статусов в область памяти
<a href="#">Import</a>	Метод позволяет импортировать ответ Службы актуальных статусов в область памяти
<a href="#">Load</a>	Метод позволяет загрузить ответ Службы актуальных статусов из файла
<a href="#">Save</a>	Метод позволяет сохранить ответ Службы актуальных статусов в файл
<a href="#">Verify</a>	Метод позволяет проверить корректность ЭЦП ответ Службы актуальных статусов и сертификата Службы актуальных статусов

**Свойства**

Свойство	Описание
<a href="#">ProducedAt</a>	Свойство позволяет получить дату и время выдачи ответа Службой актуальных статусов
<a href="#">ResponderCertificate</a>	Свойство позволяет получить сертификат оператора Службы актуальных статусов
<a href="#">SingleResponse</a>	Свойство позволяет получить из общего OCSP ответа OCSP ответ по одному сертификату.

**IOCSPPRESPONSE::ADDCERTIFICATEFORSEARCH [C++]****OCSPRESPONSE::ADDCERTIFICATEFORSEARCH [VBS]****[C++]**

Метод AddCertificateForSearch позволяет добавить сертификат к списку сертификатов, среди которых осуществляется поиск сертификата оператора Службы актуальных статусов, возвращаемый свойством ResponderCertificate.

```
HRESULT AddCertificateForSearch(  
    BSTR bstrCertificate  
);
```

**Параметры**

bstrCertificate [in] Строка, содержащая сертификат (в кодировке DER или Base64).

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

**[VBS]**

Метод AddCertificateForSearch позволяет добавить сертификат к списку сертификатов, среди которых осуществляется поиск сертификата оператора Службы актуальных статусов, возвращаемый свойством ResponderCertificate.

```
object.AddCertificateForSearch(bstrCertificate)
```

**Параметры**

bstrCertificate Строка, содержащая сертификат (в кодировке DER или Base64).



**IOCSPPRESPONSE::EXPORT [C++]****OCSPRESPONSE::EXPORT [VBS]**

[C++]

Метод Export позволяет экспортировать ответ Службы актуальных статусов в область памяти.

```
HRESULT Export(  
    RESPONSE_FORMAT frmResponseFormat  
    BSTR* pbstrResponse  
);
```

**Параметры**

frmResponseFormat [in] Формат OCSP ответа. Допустимыми значениями являются значения из OCSP\_RESPONSE\_FORMAT  
pbstrResponse [out, retval] Указатель на строку, содержащую OCSP ответ в указанном формате.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Export позволяет экспортировать ответ Службы актуальных статусов в область памяти.

```
object.Export (frmResponseFormat)
```

**Параметры**

frmResponseFormat Формат OCSP ответа. Допустимыми значениями являются значения из OCSP\_RESPONSE\_FORMAT

**Возвращаемые значения**

Возвращаемое значение указатель на строку, содержащую OCSP ответ в указанном формате.

**IOCSPPRESPONSE::IMPORT [C++]****OCSPRESPONSE::IMPORT [VBS]**

[C++]

Метод Import позволяет импортировать ответ Службы актуальных статусов из области памяти.

```
HRESULT Import(  
    BSTR bstrResponse  
);
```

**Параметры**

bstrResponse [in] Строка, содержащая OCSP ответ.

**Возвращаемые значения**

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Import позволяет импортировать ответ Службы актуальных статусов из области памяти.

```
object.Import (bstrResponse)
```

**Параметры**

bstrResponse – строка, содержащая OCSP ответ.

### Возвращаемые значения

Возвращаемое значение указатель на строку, содержащую OCSP ответ в указанном формате.

## IOCSPPRESPONSE::LOAD [C++]

### OCSPRESPONSE::LOAD [VBS]

[C++]

Метод Load позволяет загрузить ответ Службы актуальных статусов из файла.

```
HRESULT Load(  
    BSTR bstrFileName  
);
```

### Параметры

bstrResponse [in] – строка, содержащая путь к файлу OCSP ответа.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Load позволяет загрузить ответ Службы актуальных статусов из файла.

```
object.Load (bstrResponse)
```

### Параметры

bstrResponse Строка, содержащая путь к файлу OCSP ответа.

### Возвращаемые значения

Возвращаемое значение указатель на строку, содержащую OCSP ответ в указанном формате.

## IOCSPPRESPONSE::SAVE [C++]

### OCSPRESPONSE::SAVE [VBS]

[C++]

Метод Save позволяет сохранить ответ Службы актуальных статусов в файл.

```
HRESULT Save(  
    RESPONSE_FORMAT frmResponseFormat  
    BSTR bstrFileName  
);
```

### Параметры

frmResponseFormat [in] Формат OCSP ответа. Допустимыми значениями являются значения из OCSP\_RESPONSE\_FORMAT.

bstrFileName [in] Строка, содержащая путь к файлу OCSP ответа.

### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Save позволяет сохранить ответ Службы актуальных статусов в файл.

```
object.Save (frmResponseFormat)
```

#### Параметры

**frmResponseFormat** Формат OSCP ответа. Допустимыми значениями являются значения из OSCP\_RESPONSE\_FORMAT  
**bstrFileName** Строка, содержащая путь к файлу OSCP ответа.

#### Возвращаемые значения

Возвращаемое значение указатель на строку, содержащую OSCP ответ в указанном формате.

### IOCSPPRESPONSE::VERIFY [C++]

### OCSPPRESPONSE::VERIFY [VBS]

[C++]

Метод Verify позволяет проверить корректность ЭЦП OSCP ответа и сертификата Службы актуальных статусов.

```
HRESULT Verify(  
    long* pVal  
);
```

#### Параметры

**pVal [out, retval]** результат проверки: 0 – проверка прошла успешно, иначе – код ошибки. (см. поле dwErrorStatus структуры CERT\_TRUST\_STATUS (CryptoAPI)).

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Метод Verify позволяет проверить корректность ЭЦП OSCP ответа и сертификата Службы актуальных статусов.

```
object.Verify ()
```

#### Возвращаемые значения

Возвращаемое значение – результат проверки: 0 – проверка прошла успешно, иначе – код ошибки. (см. поле dwErrorStatus структуры CERT\_TRUST\_STATUS (CryptoAPI)).

### IOCSPPRESPONSE::PRODUCEDAT [C++]

### OCSPPRESPONSE::PRODUCEDAT [VBS]

[C++]

Свойство ProducedAt позволяет получить время выдачи ответа Службой актуальных статусов.

```
HRESULT get_ProducedAt (  
    DATE* pVal  
);
```

#### Параметры

pNewOCSPProfile [out, retval] Время выдачи ответа Службой актуальных статусов.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ProducedAt позволяет получить время выдачи ответа Службой актуальных статусов.

```
object.ProducedAt ()
```

#### Возвращаемые значения

Возвращаемое значение время выдачи ответа Службой актуальных статусов.

### IOCSPPRESPONSE::RESPONDERCERTIFICATE [C++]

#### OCSPRESPONSE::RESPONDERCERTIFICATE [VBS]

[C++]

Свойство ResponderCertificate позволяет получить сертификат подписи Службы актуальных статусов из OCSP ответа.

```
HRESULT get_ResponderCertificate(  
    BSTR* pVal  
);
```

#### Параметры

pVal [out, retval] Указатель на строку с блобом сертификата подписи Службы актуальных статусов из OCSP ответа.

#### Возвращаемые значения

Возвращаемое значение имеет тип HRESULT.

[VBS]

Свойство ResponderCertificate позволяет получить сертификат подписи Службы актуальных статусов из OCSP ответа.

```
object.ResponderCertificate ()
```

#### Возвращаемые значения

Возвращаемое значение указатель на строку с блобом сертификата подписи Службы актуальных статусов из OCSP ответа.

### IOCSPPRESPONSE::SINGLERESPONSE [C++]

#### OCSPRESPONSE::SINGLERESPONSE [VBS]

[C++]

Свойство SingleResponse позволяет получить отдельный OCSP ответ по каждому сертификату.

```
HRESULT get_SingleResponse(  
    VARIANT vtCriterion  
    IOCSPSingleResponse** ppVal  
);
```

#### Параметры

`vtCriterion [in]` Индекс элемента (OCSP ответа в коллекции) или строка с сертификатом, для которого был получен OCSP ответ.  
`ppVal [out, retval]` Указатель на интерфейс `IOCSPSingleResponse`.

### Возвращаемые значения

Возвращаемое значение имеет тип `HRESULT`.

[VBS]

Свойство `SingleResponse` позволяет получить отдельный OCSP ответ по каждому сертификату.

`object.SingleResponse (vtCriterion)`

### Параметры

`vtCriterion` Индекс элемента (OCSP ответа в коллекции) или строка с сертификатом, для которого был получен OCSP ответ.

### Возвращаемые значения

Возвращаемое значение указатель на объект `OCSPSingleResponse`

## РАЗДЕЛ 4. ОПИСАНИЕ COM- и NPAPI-БИБЛИОТЕКИ DIGTCAPICOM

В компоненте `npCAPICOMExtension.dll` содержится COM- и NPAPI-библиотека `DigtCAPICOM`, которая предназначена для работы с сертификатами, хранилищами сертификатов, формирования и проверки электронных подписей, зашифрования и расшифрования данных.

Программный интерфейс `DigtCAPICOM` в основном совпадает с интерфейсом `CAPICOM`, описание которого приводится на сайте Microsoft (см. <http://msdn.microsoft.com/en-us/library/windows/desktop/aa380256%28v=vs.85%29.aspx>, англ.), но все же имеется несколько отличий.

### 4.1. ИНИЦИАЛИЗАЦИЯ DIGTCAPICOM

COM-библиотека не требует специальной инициализации. Создавать COM-объекты в Internet Explorer следует как обычно:

```
var obj = new ActiveXObject(<идентификатор_объекта>);
```

Но для универсального создания объектов в различных браузерах следует разработать специальную функцию. Пример такой функции приведен ниже.

Для создания NPAPI-объектов необходимо выполнить следующие действия:

- разместить на странице объект с типом «application/x-digtcapicom»;
- у DOM-объекта проинициализировать свойство `createDate` ссылкой на функцию, которая принимает 7 аргументов (год, месяц, день, часы, минуты, секунды и миллисекунды) и возвращает объект даты;
- у DOM-объекта вызывать метод `CreateObject(<идентификатор_объекта>)`.

Пример инициализации и работы с объектами плагина:

```
<html>
<head>
  <title>test page for DigtCAPICOM</title>
</head>
<script type="text/javascript">
  function plugin0()
  {
    return document.getElementById('plugin0');
  }
  plugin = plugin0;

  function pluginLoaded()
  {
    alert("Plugin loaded!");
    plugin().createDate = createDate;
  }

  function isComAvailable()
  {
    return (navigator.appName == "Microsoft Internet Explorer");
  }

  function createDate(year, month, day, hrs, min, sec, ms)
  {
    return new Date(Date.UTC(year, month, day, hrs, min, sec, ms));
  }

  function getVarDate(date)
  {
    if ( isComAvailable() )
    {
      return date.getVarDate();
    }
  }
</script>
```

```
    }
    else
    {
        return date;
    }
}

function CreateObject(strTypeName)
{
    switch (navigator.appName)
    {
        case "Microsoft Internet Explorer":
            return new ActiveXObject(strTypeName);

        default:
            return plugin().CreateObject(strTypeName);
    }
}
</script>

<script type="text/javascript">
    // CAPICOM_ATTRIBUTE
    var CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME          = 0;
    var CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_NAME         = 1;
    var CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_DESCRIPTION = 2;

    function createAttr( Name, Value )
    {
        var oAttr = CreateObject("CAPICOM.Attribute");
        oAttr.Name = Name;
        oAttr.Value = Value;
        return oAttr;
    }

    function dumpAttr(oAttr)
    {
        alert("Name: " + oAttr.Name + ", Value: " + oAttr.Value);
    }

    function testAttribute()
    {
        var oAttr = createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME, getVarDate(new
Date()));
        dumpAttr(oAttr);

        oAttr = createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_NAME, "Document1");
        dumpAttr(oAttr);

        oAttr = createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_DESCRIPTION, "Описание доку-
мента");
        dumpAttr(oAttr);
    }
</script>

<body>
    <object id="plugin0" type="application/x-digtcapicom" width="200" height="200">
        <param name="onload" value="pluginLoaded" />
    </object>

    <br /><a href="#" onclick="javascript:testAttribute()">testAttribute</a>
</body>
</html>
```

## 4.2. ОБЩИЕ ПРАВИЛА ВЫЗОВА API ПЛАГИНА

Названия свойств и методов NPAPI-объектов чувствительны к регистру.

## 4.3. СПЕЦИФИКА РАБОТЫ С ДАТАМИ

Отличие между COM- и NPAPI-интерфейсами состоит в работе с датами. COM-методы принимают и возвращают дату в виде объекта VARIANT с типом VT\_DATE. NPAPI-методы оперируют датой в виде объекта Date.

Для преобразования объекта даты в вид, соответствующий браузеру, рекомендуется разработать специальную функцию. Пример такой функции приведен в п.4.1.

#### 4.4. ПЕРЕЧИСЛЕНИЯ И ОБЪЕКТЫ

Назначение и числовые коды перечислений, используемых в DigtCAPICOM, совпадают с перечислениями MS CAPICOM (см. описание на странице <http://msdn.microsoft.com/en-us/library/windows/desktop/aa380250%28v=vs.85%29.aspx>, англ.).

В DigtCAPICOM реализована поддержка следующих COM- и NPAPI-объектов:

Интерфейс	Описание
- для работы с сертификатами:	
<a href="#">BasicConstraints</a>	Описывает расширение сертификата «Basic Constraints».
<a href="#">Certificate</a>	Описывает сертификата.
<a href="#">Certificates</a>	Описывает коллекцию объектов с типом Certificate.
<a href="#">CertificatePolicies</a>	Описывает расширение сертификата «Certificate Policies».
<a href="#">CertificateStatus</a>	Предоставляет информацию по проверке сертификата.
<a href="#">Chain</a>	Описывает цепочку сертификатов.
<a href="#">EKU</a>	Описывает расширенное использование ключа сертификата.
<a href="#">EKUs</a>	Описывает коллекцию объектов с типом EKU.
<a href="#">ExtendedKeyUsage</a>	Описывает расширение сертификата «Extended Key Usage».
<a href="#">ExtendedProperty</a>	Описывает свойства сертификата, специфичные для ОС Windows.
<a href="#">ExtendedProperties</a>	Описывает коллекцию объектов с типом ExtendedProperty.
<a href="#">Extension</a>	Описывает расширение сертификата.
<a href="#">Extensions</a>	Описывает коллекцию объектов с типом Extension.
<a href="#">KeyUsage</a>	Описывает расширение сертификата «Key Usage».
<a href="#">PolicyInformation</a>	Описывает элемент объекта с типом CertificatePolicies.
<a href="#">PrivateKey</a>	Описывает закрытый ключ.
<a href="#">PublicKey</a>	Описывает открытый ключ.
<a href="#">Store</a>	Описывает хранилище сертификатов.
<a href="#">Template</a>	Описывает расширение «Template» (версий 1 и 2).
- для работы с подписанными данными:	
<a href="#">Attribute</a>	Описывает атрибут подписи.
<a href="#">Attributes</a>	Описывает коллекцию объектов с типом Attribute.
<a href="#">SignedData</a>	Описывает CMS-сообщение с одной или несколькими электронными подписями.
<a href="#">Signer</a>	Предоставляет расширенную информацию об электронной подписи.
<a href="#">Signers</a>	Описывает коллекцию объектов с типом Signer.
- для работы с шифрованными данными:	
<a href="#">Algorithm</a>	Описывает криптографический алгоритм.
<a href="#">EnvelopedData</a>	Описывает CMS-сообщение, содержащие шифрованные данные.
<a href="#">Recipients</a>	Описывает коллекцию объектов с типом Certificate.
- вспомогательные объекты:	
<a href="#">EncodedData</a>	Описывает блок закодированных данных.
<a href="#">OID</a>	Описывает объектный идентификатор.
<a href="#">OIDs</a>	Описывает коллекцию объектов с типом OID.
<a href="#">Settings</a>	Позволяет изменять настройки функционирования плагина.
<a href="#">Utilities</a>	Предоставляет методы по преобразованию данных из одного формата в другой.

#### Algorithm (GUID: 6A0EF8CC-164D-4D86-9093-E32E8BCFC507)

Описывает криптографический алгоритм. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa375546\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375546(v=vs.85).aspx) (англ.).



**Attribute (GUID: F789CF2E-A127-4A2F-863F-704A10FF7E4C)**

Описывает атрибут подписи. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa375567\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375567(v=vs.85).aspx) (англ.). Особенности работы с датами описаны в п.4.3.

**Attributes (GUID: 791350EC-3252-450C-A038-2A4C17A4035A)**

Описывает коллекцию объектов с типом Attribute. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa374834\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374834(v=vs.85).aspx) (англ.).

**BasicConstraints (GUID: EAB4BF2B-B821-4B7B-8857-578E35D5A59C)**

Описывает расширение сертификата «Basic Constraints». Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa375602\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375602(v=vs.85).aspx) (англ.).

**Certificate (GUID: 4B23E890-3266-4FC2-97C4-37869B94E451)**

Описывает сертификата. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa376092\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376092(v=vs.85).aspx) (англ.).

**Certificates (GUID: CB72E4D1-F24D-4B6C-BCD7-C94C046C9AF8)**

Описывает коллекцию объектов с типом Certificate. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa376489\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376489(v=vs.85).aspx) (англ.).

**CertificatePolicies (GUID: 7DB0843E-5CCB-4678-B5D0-8116029ECB25)**

Описывает расширение сертификата «Certificate Policies». Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa376093\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376093(v=vs.85).aspx) (англ.).

**CertificateStatus (GUID: 60AAB705-5219-4BFB-B70E-BCF930FCE90B)**

Предоставляет информацию по проверке сертификата. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa376490\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376490(v=vs.85).aspx) (англ.).

Особенности работы с датами описаны в п.4.3.

**Chain (GUID: EEAE79FE-F9B2-4591-AA76-0E96A7CD18B0)**

Описывает цепочку сертификатов. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa377611\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa377611(v=vs.85).aspx) (англ.).

Особенности обращения к свойству Status: в NPAPI-объекте оно реализовано в виде метода, т.к. имеет входной параметр, поэтому при его вызове следует использовать круглые скобки.

**EKU (GUID: 1A868CF3-D8F8-4332-B40A-5FF0D9280C81)**

Описывает расширенное использование ключа сертификата. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa381990\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa381990(v=vs.85).aspx) (англ.).

**EKUs (GUID: 985CCCD-E596-4E60-A5FB-20EB136A5F8F)**

Описывает коллекцию объектов с типом ECU. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa381991\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa381991(v=vs.85).aspx) (англ.).

**EncodedData (GUID: 40A7CE64-D1A8-4D97-B15C-32D05C8F57B2)**

Описывает блок закодированных данных. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa381997\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa381997(v=vs.85).aspx) (англ.).

Особенности обращения к свойству Value: в NPAPI-объекте оно реализовано в виде метода, т.к. имеет входной параметр, поэтому при его вызове следует использовать круглые скобки.

**EnvelopedData (GUID: 000DD7DC-16A7-46F9-B81F-8CCA33BCC378)**

Описывает CMS-сообщение, содержащие шифрованные данные. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa382024\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa382024(v=vs.85).aspx) (англ.).

**ExtendedKeyUsage (GUID: 54176970-66C3-4DAA-892D-825422D735E4)**

Описывает расширение сертификата «Extended Key Usage». Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa382388\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa382388(v=vs.85).aspx) (англ.).

**ExtendedProperty (GUID: F5FF48A4-7385-4CCD-9FDE-897FF28B0982)**

Описывает Microsoft-расширение сертификата. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa382393\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa382393(v=vs.85).aspx) (англ.).

Особенности обращения к свойству Value:

- чтение свойства NPAPI-объекта реализовано в виде метода Value(), т.к. имеет входной параметр, поэтому при его вызове следует использовать круглые скобки;
- запись данного свойства в NPAPI-объект следует осуществлять через вызов метода SetValue().

**ExtendedProperties (GUID: E15FB369-4EBE-458F-8866-790590A0D9FB)**

Описывает коллекцию объектов с типом ExtendedProperty. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa382393\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa382393(v=vs.85).aspx) (англ.).

**Extension (GUID: C82B3221-3747-4C6C-96A3-B808787DB7E5)**

Описывает расширение сертификата. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa382408\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa382408(v=vs.85).aspx) (англ.).

**Extensions (GUID: 528D03B8-E484-4988-907A-9DFCEF7CD1E6)**

Описывает коллекцию объектов с типом Extension. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa374978\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374978(v=vs.85).aspx) (англ.).

**KeyUsage (GUID: 575174B8-296E-4948-9E2B-F86C7675FAEE)**

Описывает расширение сертификата «Key Usage». Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa386950\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa386950(v=vs.85).aspx) (англ.).

**OID (GUID: 065A8A05-D356-4570-B4ED-1923D6DAF0C4)**

Описывает объектный идентификатор. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387022\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387022(v=vs.85).aspx) (англ.).

**OIDs (GUID: F507582D-55C1-4656-95D7-BFCB4236A325)**

Описывает коллекцию объектов с типом OID. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387023\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387023(v=vs.85).aspx) (англ.).

**PolicyInformation (GUID: 674CA77A-20BB-40A7-83A6-4242B030C184)**

Описывает элемент объекта с типом CertificatePolicies. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387332\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387332(v=vs.85).aspx) (англ.).

**Внимание!** Свойство Qualifiers не реализовано.

**PrivateKey (GUID: 25820977-F093-4ABE-9743-ABF5D067F17F)**

Описывает закрытый ключ. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387352\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387352(v=vs.85).aspx) (англ.).

**PublicKey (GUID: 2D905C43-A98A-4D02-AB97-18DD2C019B5E)**

Описывает открытый ключ. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387452\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387452(v=vs.85).aspx) (англ.).

**Recipients (GUID: AB300C73-7085-4494-8792-4B68A3FD9E25)**

Описывает коллекцию объектов с типом Certificate. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387474\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387474(v=vs.85).aspx) (англ.).

**Settings (GUID: 5A7C39C8-A8CD-4476-B394-E51B522B04FA))**

Позволяет изменять настройки функционирования плагина. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387701\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387701(v=vs.85).aspx) (англ.).

**SignedData (GUID: F80E8EE1-4DE6-4B62-95CF-B8C0D691663A)**

Описывает CMS-сообщение с одной или несколькими электронными подписями. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387722\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387722(v=vs.85).aspx) (англ.).

**Signer (GUID: 65AA895B-AFAF-4251-9CBF-528F63EFE2C4)**

Предоставляет расширенную информацию об электронной подписи. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387730\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387730(v=vs.85).aspx) (англ.).

**Signers (GUID: 9EC9E933-6271-4208-98DF-892A9FE94291)**

Описывает коллекцию объектов с типом Signer. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa387732\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa387732(v=vs.85).aspx) (англ.).

#### **Store (GUID: 833343C8-A7D1-484D-A838-FE29BBC4C6A9)**

Описывает хранилище сертификатов. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa388120\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa388120(v=vs.85).aspx) (англ.).

#### **Template (GUID: 6C217385-EAD1-4E3F-ACC4-4B817384B0C3)**

Описывает расширение «Template» (версий 1 и 2). Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa388137\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa388137(v=vs.85).aspx) (англ.).

#### **Utilities (GUID: AC39CD62-254B-4CDC-B3BD-FE7F644A5F7C)**

Предоставляет методы по преобразованию данных из одного формата в другой. Функционирует в соответствии с описанием на MSDN: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa388176\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa388176(v=vs.85).aspx) (англ.).

Особенности работы с датами описаны в п.4.3.

### **4.5. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ**

#### **4.5.1. СОЗДАНИЕ ПОДПИСИ, СОВМЕЩЕННОЙ С ДОКУМЕНТОМ**

В следующем примере используется инициализация плагина и функция createAttr() из примера в п.4.1.

```
var oSigner = CreateObject("CAPICOM.Signer");

oSigner.AuthenticatedAttributes.Add(createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME,
getVarDate(new Date())));

oSigner.AuthenticatedAttributes.Add(createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_NAME,
"signature1"));

oSigner.AuthenticatedAttributes.Add(createAttr(CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_DESCRIPTION,
"description1"));

var oSignedData = CreateObject("CAPICOM.SignedData");
oSignedData.Content = "данные для подписи";

var sSignedData = oSignedData.Sign(oSigner, false);
alert(sSignedData);
```

#### **4.5.2. ПРОВЕРКА ПОДПИСИ, СОВМЕЩЕННОЙ С ДОКУМЕНТОМ**

В следующем примере используется инициализация плагина и функция dumpAttr() из примера в п.4.1, а также переменная sSignedData из примера в п.4.5.1.

```
var oSignedData = CreateObject("CAPICOM.SignedData");
oSignedData.Verify(sSignedData);

dumpAttr(oSignedData.Signers.Item(1).AuthenticatedAttributes.Item(1));
dumpAttr(oSignedData.Signers.Item(1).AuthenticatedAttributes.Item(2));
dumpAttr(oSignedData.Signers.Item(1).AuthenticatedAttributes.Item(3));
```

**РАЗДЕЛ 5. СТРУКТУРА ХРАНЕНИЯ ПРОФИЛЕЙ В РЕЕСТРЕ WINDOWS**

[HKEY\_CURRENT\_USER\Software\DIGT SmartServices\TrustedDesktop\ProfileStore] – ключ, в котором хранятся параметры профилей.

[HKEY\_CURRENT\_USER\Software\DIGT SmartServices\TrustedDesktop\ProfileStore\{GUID}] – ключ, в котором хранятся параметры отдельного профиля. Профиль идентифицируется по GUID.

<i>DecryptCertificate</i> "=hex:	сертификат для расшифрования в сериализованном хранилище (SST)
<i>DecryptFinalWindow</i> "=dword:00000000	показывать финальное окно мастера расшифрования
<i>DecryptPIN</i> "="111111"	PIN-код к закрытому ключу сертификата для расшифрования
<i>DecryptStartWindow</i> "=dword:00000000	показывать первое окно мастера расшифрования
<i>DefaultPath</i> " = "D:\doc"	путь по умолчанию для сохранения файлов, полученных в случае выполнения операций расшифрования и/или снятия ЭЦП
<i>Detach</i> "=dword:00000000	тип подписи (отсоединённая или присоединённая)/span>
<i>EncryptAlgorithm</i> "="1.2.643.2.2.6"	алгоритм шифрования по умолчанию
<i>EncryptCAPICOMSupport</i> "=dword:00000000	отключать служебные заголовки при шифровании в формате PEM(Base64)
<i>EncryptCertificate</i> "=hex:	сертификат для шифрования в сериализованном хранилище (SST) (сертификат отправителя)
<i>EncryptCryptoProviderName</i> "="Microsoft Base Cryptographic Provider"	символьное наименование крипто провайдера, используемого при шифровании (без сертификата отправителя)
<i>EncryptDeleteSourceFile</i> "=dword:00000002	удалять исходный файл после зашифрования
<i>EncryptFinalWindow</i> "=dword:00000000	показывать финальное окно мастера шифрования
<i>EncryptFormatOptions</i> "=dword:00000000	показывать окно выбора выходного формата сообщения при шифровании
<i>EncryptP7M</i> "=dword:00000000	выходной формат зашифрованного сообщения – Base64 (PEM)
<i>EncryptPEM</i> "=dword:00000000	при шифровании данные сообщения сохранять в формате PEM(Base64)
<i>EncryptPropsWindow</i> "=dword:00000000	показывать окно свойств шифрования в мастере шифрования
<i>EncryptRecipientsWindow</i> "=dword:00000000	показывать окно формирования списка получателей в мастере шифрования
<i>EncryptStartWindow</i> "=dword:00000000	показывать первое окно мастера шифрования
<i>EncryptSwitch</i> "=dword:00000000	не использовать сертификат при шифровании
<i>EncryptToAddress</i> "=dword:00000001	шифровать в адрес отправителя
<i>IncludeSignatureTime</i> " =dword:00000000	параметр указывает на включение времени создания ЭЦП в неинтерактивном режиме
<i>ProfileName</i> "="Мой второй профиль"	символьное имя профиля
<i>RecipientsCertificates</i> "=hex:	сертификаты получателей по умолчанию в сериализованном хранилище (SST)
<i>ShowStatusReport</i> " =dword:00000000	параметр указывает на отображение окна о статусе выполнения операции в полуинтерактивных режимах
<i>SignAndEncryptFinalWindow</i> "=dword:00000000	показывать финальное окно мастера ЭЦП и шифрования
<i>SignAndEncryptFormatOptions</i> "=dword:00000000	показывать окно выбора выходного формата сообщения при подписи и шифровании
<i>SignAndEncryptP7M</i> "=dword:00000000	выходной формат подписанного и зашифрованного сообщения – DER (P7M)
<i>SignAndEncryptPEM</i> "=dword:00000000	выходной формат подписанного и зашифрованного сообщения – Base64 (PEM)
<i>SignAndEncryptStartWindow</i> "=dword:00000000	показывать первое окно мастера подписи и шифрования
<i>SignCAPICOMSupport</i> "=dword:00000000	отключать служебные заголовки при создании ЭЦП в формате PEM(Base64)
<i>SignCertificate</i> "=hex:	сертификат для создания ЭЦП в сериализованном хранилище (SST)

<i>SignCertificateStep</i> "=hex:	показывать окно выбора сертификата подписи
<i>SignComment</i> "="комментарий"	комментарии по умолчанию для ЭЦП
<i>SignDeleteSourceFile</i> "="комментарий"	удалять исходный файл после получения присоединенной подписи
<i>SignFinalWindow</i> "=dword:00000000	показывать финальное окно мастера создания ЭЦП
<i>SignFormatOptions</i> "=dword:00000000	показывать окно выбора выходного формата сообщения
<i>SignHashAlgorithm</i> "="1.2.643.2.2.6"	алгоритм хэширования по умолчанию для ЭЦП
<i>SignP7S</i> "=dword:00000000	выходной формат подписанного сообщения – DER (P7S)
<i>SignPEM</i> "=dword:00000000	при создании ЭЦП данные сообщения сохранять в формате PEM(Base64)
<i>SignPIN</i> "=""	PIN-код доступа к закрытому ключу сертификата для создания ЭЦП
<i>SignPropertiesStep</i> "=dword:00000000 "	показывать окно установки свойств подписи в мастере подписи
<i>SignResource</i> "="ресурс"	PIN-код доступа к закрытому ключу сертификата для создания ЭЦП
<i>SignStartWindow</i> "=dword:00000000	показывать стартовое окно мастера ЭЦП
<i>VerifiedCertificates</i> "=hex:	список сертификатов для проверки по СОС, полученному из УЦ

HKEY\_CURRENT\_USER\Software\DIGT SmartServices\TrustedDesktop\ProfileStore\{294D0760-479B-4483-817E-27AEDA09144D}\Policy\Certificate Policies\{34F11A3A-737C-4567-AFB2-9718E48FF56E\EKU} - ключ, в котором хранится политика назначений сертификатов подписи.

HKEY\_CURRENT\_USER\Software\DIGT SmartServices\TrustedDesktop\ProfileStore\{33FAA997-175F-4A20-A4E5-0914BA1DEFC9}\Policy\Certificate Policies\{9DD0C13A-4A68-43d9-BF5C-CEE99B44CAA3}\EKU - ключ, в котором хранится политика назначений сертификатов шифрования.