

ЦЕНТРАЛЬНЫЙ БАНК РОССИЙСКОЙ ФЕДЕРАЦИИ
(БАНК РОССИИ)

УТВЕРЖДЕН
ВАМБ.00143-06-ЛУ

**«СИГНАТУРА-СЕРТИФИКАТ L» ВЕРСИЯ 6
ПРИКЛАДНОЙ ПРОГРАММНЫЙ ИНТЕРФЕЙС**

**БИБЛИОТЕКА
ДЛЯ ЯЗЫКОВ C/C++**

Руководство программиста

ВАМБ.00143-06 33 01

2023

Аннотация

Данный документ содержит описание библиотеки прикладного программного интерфейса (библиотеки) для языков C/C++ к функциям сервисов Центра Сертификации (ЦС) и Центра Регистрации (ЦР) программного комплекса (ПК) ВАМБ.00128-06 «"Сигнатура-сертификат L" версия 6» (далее по тексту — ПК «Сигнатура-сертификат L») для операционных систем (ОС) семейства Linux, а также рекомендации по встраиванию и использованию данной библиотеки в прикладное программное обеспечение (ПО).

Документ предназначен для разработчиков прикладных программ (внешних по отношению к ПК «Сигнатура-сертификат L») как руководство по программированию с использованием библиотеки работы с сервисами Удостоверяющего Центра.

При встраивании библиотеки предполагается, что программист имеет знания о существующей архитектуре системы управления сертификатами (СУС), используемых рекомендациях и стандартах.

Документ разработан специалистами ООО «Валидата».

Содержание

1	БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА	4
1.1	Назначение библиотеки	4
1.2	Характеристики библиотеки	4
1.2.1	Описание формата XML модифицирующего шаблона	5
1.2.2	Описание формата XML запроса на отзыв	5
1.3	Использование библиотеки	5
1.3.1	Основные понятия и определения	5
1.3.2	Условия использования библиотеки	6
1.3.3	Описание состава библиотеки	6
1.4	Описание базовых типов	7
1.5	Описание структур блоков памяти	7
1.6	Описание структур поиска объектов СУС	8
1.7	Инициализация и деинициализация	13
1.7.1	Функции инициализации библиотеки работы с ЦР	13
1.7.2	Функции деинициализации библиотеки работы с ЦР	14
1.7.3	Функции инициализации библиотеки работы с ЦС	14
1.7.4	Функции деинициализации библиотеки работы с ЦС	15
1.8	Тестирование соединения с сервисами	15
1.8.1	Функции тестирования соединения с сервисом ЦР	15
1.8.2	Функции тестирования соединения с сервисом ЦС	16
1.9	Обработка запросов сервиса ЦР	16
1.9.1	Функции обработки запросов на выпуск сертификатов	16
1.9.2	Функции обработки выпущенных сертификатов	17
1.9.3	Функции обработки запросов на отзыв сертификатов	17
1.9.4	Функции получения списка объектов СУС	18
1.10	Обработка запросов сервиса ЦС	19
1.10.1	Функции обработки запросов на выпуск сертификатов	19
1.10.2	Функции обработки запросов на отзыв сертификатов	21
1.11	Управление памятью	21
1.11.1	Функции освобождения блока памяти	21
1.11.2	Функции освобождения массива блоков памяти	22
2	ОПИСАНИЕ ОШИБОЧНЫХ СИТУАЦИЙ	23
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	30

1 БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА

1.1 Назначение библиотеки

Библиотека прикладного программного интерфейса к функциям сервисов ЦР и ЦС для операционных систем (ОС) семейства Linux (далее по тексту - библиотека) предназначена для совместного использования с программным комплексом ВАНБ.00128-06 "Сигнатура-сертификат L" версия 6, предоставляет программный интерфейс работы с запросами на выпуск и отзыв сертификатов.

Библиотека обеспечивает удаленный доступ к функциям ЦР и ЦС обработки запросов на выпуск и отзыв сертификатов. Удаленный доступ к функциям осуществляется посредством использования протокола DCE-RPC поверх протокола ТСР/ІР.

Библиотека обеспечивает доступ к следующим функциям ЦР:

- проверка соединения с ЦР;
- обработка запроса на выпуск сертификата с выработкой запроса для ЦС;
- обработка запроса на выпуск сертификата с модифицирующим шаблоном с выработкой запроса для ЦС;
- обработка выпущенного ЦС сертификата;
- обработка запроса на отзыв сертификата с выработкой запроса для ЦС;
- получение объектов из указанного справочника ЦР по критерию поиска;
- проверка соединения ЦР с ЦС;
- вызов ЦС для выпуска сертификата по запросу на выпуск;
- вызов ЦС для выпуска сертификата по запросу на выпуск с модифицирующим шаблоном;
- вызов ЦС для обработки запроса на отзыв сертификата;

Библиотека обеспечивает доступ к следующим функциям ЦС:

- проверка соединения с ЦС;
- выпуск сертификата по запросу на выпуск;
- выпуск сертификата по запросу на выпуск с модифицирующим шаблоном;
- обработка запроса на отзыв сертификата;

1.2 Характеристики библиотеки

Библиотека предназначена для встраивания в прикладные системы. Требования к аппаратно-программной среде, в которой функционирует библиотека, приведены в документе ВАНБ.00143-06 30 01 «Сигнатура-сертификат L" версия 6. Прикладной программный интерфейс. Формуляр».

Запрос на выпуск сертификата передаётся в формате PKCS#10. Модифицирующий шаблон передаётся в формате XML. Запрос на отзыв сертификата для

ЦС может быть сформирован по запросу на отзыв в формате XML.

Форматы сертификатов ключей проверки ЭП и/или открытых ключей шифрования, списков аннулированных сертификатов (САС) и PKCS#10 запросов на получение сертификатов, формируемых и поддерживаемых библиотекой, соответствуют Рекомендациям по стандартизации Р 1323565.1.023-2022 «Использование алгоритмов ГОСТ Р 34.10-2012, ГОСТ Р 34.11-2012 в сертификате, списке аннулированных сертификатов (CRL) и запросе на сертификат PKCS#10 инфраструктуры открытых ключей X.509», которые в свою очередь соответствуют одноименной спецификации Технического комитета № 26.

1.2.1 Описание формата XML модифицирующего шаблона

Формат XML модифицирующего шаблона совпадает с форматом XML запроса на сертификат, описание которого приведено в документе ВАНБ.00126-06 33 01 «“Сигнатура-клиент L” версия 6. Руководство программиста».

1.2.2 Описание формата XML запроса на отзыв

XML запрос на аннулирование (отзыв) представляет собой текстовый XML документ в кодировке UTF-8, содержащий информацию, достаточную для аннулирования сертификата.

Ниже приведен пример XML запроса на аннулирование без необходимой декларации XML документа (без XML Declaration):

```
<pkiUser>  
  <Certificate>000102030405060708090a0b0c0d0e0f</Certificate>  
  <ReasonCode>1</ReasonCode>  
  <InvalidDate>1673513400</InvalidDate>  
</pkiUser>
```

Аннулируемый сертификат содержится в элементе pkiUser -> Certificate. В этом элементе должна присутствовать шестнадцатеричная текстовая строка (hex-string) байтов сертификата в DER-кодировке.

Причина аннулирования содержится в элементе (опциональном) pkiUser -> ReasonCode и представляет собой число, описанное в документе ВАНБ.00126-06 33 01 «“Сигнатура-клиент L” версия 6. Руководство программиста». Если данный элемент не задан, причина аннулирования не задается.

Момент времени аннулирования содержится в элементе (опциональном) pkiUser -> InvalidDate и представляет собой время в секундах, прошедшее с 00:00 01.01.1970 UTC. Если данный элемент не задан, используется текущий момент времени.

1.3 Использование библиотеки

1.3.1 Основные понятия и определения

Все операции, выполняемые посредством вызовов функций библиотеки, оперируют объектами **системы управления сертификатами (СУС)** - сертификатами

тами, САС, запросами PKCS#10, запросами на аннулирование. Также для некоторых операций используются модифицирующие шаблоны в формате XML.

1.3.2 Условия использования библиотеки

При использовании библиотеки необходимо соблюдать следующие условия:

- если не указано обратное, все строковые данные, получаемые и возвращаемые библиотекой, должны быть в кодировке Windows Code Page 1251 (Windows-1251, CP1251);
- библиотека предназначена для использования в приложениях либо написанных на языках программирования C/C++, либо совместимых с ними по форматам данных и параметрам вызовов функций. При использовании библиотеки в приложениях, написанных на других языках программирования, должно выполняться требуемое преобразование форматов данных и параметров вызовов функций.

Приложение должно вызывать функции библиотеки последовательно:

- начало работы приложения;
- инициализация сессии библиотеки с локальным или удалённым сервисом ЦР/ЦС одной из функций **RACLI_OpenSessionLocal()**, **CACLI_OpenSessionLocal()**, **RACLI_OpenSessionRemote()**, **CACLI_OpenSessionRemote()**;
- использование сессии - вызов функций библиотеки для проверки соединения, обработки запросов на выпуск или отзыв сертификатов или получения списка объектов;
- освобождение сессии библиотеки функцией **RACLI_CloseSession()** или **CACLI_CloseSession()**;
- окончание работы приложения.

Примечание - В процессе своей работы приложению разрешается инициализировать и деинициализировать несколько различных сессий библиотеки, в том числе в целях их одновременного (параллельного) использования.

1.3.3 Описание состава библиотеки

В состав библиотеки входят следующие файлы:

- **ra_cli_extiop.a** - модуль статической библиотеки функций работы с ЦР, используемый для линковки;
- **ca_cli_extiop.a** - модуль статической библиотеки функций работы с ЦС, используемый для линковки;
- **ra_cli_extiop.h** - файл заголовков с определениями констант, структур и прототипов функций библиотеки работы с ЦР;
- **ca_cli_extiop.h** - файл заголовков с определениями констант, структур и прототипов функций библиотеки работы с ЦС;
- **cara_cli_extiop.h** - вспомогательный файл заголовков с определениями констант, структур и прототипов функций библиотеки. Используется в 2 преды-

дующих файлах заголовков;

- **caracom.h, nbase.h, idlbase.h, ndrtypes.h** – вспомогательные файлы заголовков с определениями констант, структур библиотек DCE RPC;
- **ra_test.cpp, ca_test.cpp, makefile.tests** – файлы с исходными текстами тестовых утилит командной строки.
- **sra_test, sca_test** – исполняемые файлы тестовых утилит командной строки, собранных из вышеуказанных исходных текстов.

1.4 Описание базовых типов

```
#define CRSV_HRESULT unsigned int
```

Код возврата, код ошибки.

```
#define CRSV_LPVOID void*
```

Безтиповая ссылка, используется также как дескриптор сессии и возвращается функцией инициализации.

```
typedef idl_ulong_int flag_t
```

Параметры, модификаторы или флаги выполнения функций. 32-битное беззнаковое целое число.

```
typedef idl_hyper_int time_t
```

Время в секундах, прошедшее с 00:00 01.01.1970 UTC - данный тип аналогичен 64-битному time_t. 64-битное целое число.

1.5 Описание структур блоков памяти

Структура mem_blk_t

Непрерывный блок памяти (блок данных):

```
– idl_ulong_int len
```

Длина блока памяти в байтах - 32-бит целое беззнаковое число. Должна быть в интервале от 0 до $2^{31} - 1$.

```
– idl_byte * buf
```

Указатель на блок памяти, может быть равен NULL только если длина блока равна 0.

Структура блока памяти используется для обмена бинарными данными между приложением и библиотекой. При передаче данных из приложения в библиотеку структура должна содержать корректный указатель на блок памяти и его длину. В случае отсутствия передаваемых данных или при получении данных из библиотеки указатель и длина должны быть обнулены.

Структура mem_blk_array_t

Массив блоков памяти типа mem_blk_t:

```
– idl_ulong_int len
```

Количество блоков памяти в массиве - 32-бит целое беззнаковое число.

– `mem_blk_t * buf`

Указатель на массив блоков памяти, может быть равен NULL только если массив пустой (количество блоков памяти равно 0).

Структура массива блоков памяти используется для обмена списками объектов СУС между приложением и библиотекой. При передаче данных из приложения в библиотеку структура должна содержать корректный указатель на массив блоков памяти и количество блоков. В случае отсутствия передаваемых данных или при получении данных из библиотеки указатель и длина должны быть обнулены.

1.6 Описание структур поиска объектов СУС

Перечисление CARASEARCH_STORE

Определяет хранилище объектов СУС для выполнения поиска:

– **CARASEARCH_STORE_PSE**

ПСП.

– **CARASEARCH_STORE_LOCAL**

Локальное хранилище.

– **CARASEARCH_STORE_CERT**

База сертификации.

– **CARASEARCH_STORE_REG**

База регистрации.

Перечисление CARACOMMON_OBJECT

Определяет тип объекта СУС для поиска:

– **CARACOMMON_OBJECT_X509**

Сертификат.

– **CARACOMMON_OBJECT_CRL**

CAC.

– **CARACOMMON_OBJECT_REQ**

Запрос на выпуск сертификата.

– **CARACOMMON_OBJECT_REVREQ**

Запрос на отзыв сертификата.

– **CARACOMMON_OBJECT_XMLREQ**

Запрос в формате XML.

– **CARACOMMON_OBJECT_UNKNOWN**

Тип неопределён.

Перечисление CARASEARCH_VALUE

Определяет критерий поиска объектов СУС:

– **CARASEARCH_VALUE_ANY_OBJECT**

Все найденные объекты - дополнительный фильтр отсутствует.

– **CARASEARCH_VALUE_LAST_RECORDS**

Указанное количество последних добавленных объектов.

– **CARASEARCH_VALUE_SUBJECT_STR**

Полное имя владельца.

– **CARASEARCH_VALUE_SUBJECT_SUBSTR**

Подстрока полного имени владельца.

– **CARASEARCH_VALUE_ISSUER_STR**

Полное имя издателя.

– **CARASEARCH_VALUE_ISSUER_SUBSTR**

Подстрока полного имени издателя.

– **CARASEARCH_VALUE_ISSUER_AND_SERIAL**

Полное имя издателя и серийный номер сертификата в структуре tuna issuer_and_serial_t.

– **CARASEARCH_VALUE_KEY_IDENTIFIER**

Строка - идентификатор ключа сертификата.

– **CARASEARCH_VALUE_SERIAL_NUMBER**

Строка - серийный номер сертификата.

– **CARASEARCH_VALUE_SUBJECT_KEYID**

Строка - KEYID владельца.

– **CARASEARCH_VALUE_ISSUER_KEYID**

Строка - KEYID издателя.

– **CARASEARCH_VALUE_ALTNAME_EMAIL**

Строка - эл.почта из альтернативного имени владельца.

– **CARASEARCH_VALUE_ALTNAME_COMPANY**

Строка - название организации из альтернативного имени владельца.

– **CARASEARCH_VALUE_ALTNAME_SURNAME**

Строка - отчество из альтернативного имени владельца.

– **CARASEARCH_VALUE_EXPIRES_WITHIN**

Количество дней до конца действия сертификата - 32-бит целое число.

– **CARASEARCH_VALUE_KEY_EXPIRES_WITHIN**

Количество дней до конца действия закрытого ключа сертификата - 32-бит целое число.

– **CARASEARCH_VALUE_VALID_FROM**

Интервал времени начала действия сертификата в структуре tuna interval_t.

– **CARASEARCH_VALUE_VALID_TO**

Интервал времени конца действия сертификата в структуре tuna interval_t.

– **CARASEARCH_VALUE_KEY_VALID_FROM**

Интервал времени начала действия закрытого ключа сертификата в структуре tuna interval_t.

– **CARASEARCH_VALUE_KEY_VALID_TO**

Интервал времени конца действия закрытого ключа сертификата в структуре tuna interval_t.

– **CARASEARCH_VALUE_SIGNING_TIME**

Время подписи сертификата издателем - число.

Перечисление CARAU_PARSE_NAME

Флаг, указывающий часть полного имени/альтернативного имени для дополнительных критериев поиска:

– **CARAU_PARSE_NAME_RDN_INN**

ИНН полного имени.

– **CARAU_PARSE_NAME_RDN_OGRN**

ОГРН полного имени.

– **CARAU_PARSE_NAME_RDN_OGRNIP**

ОГРН ИП полного имени.

– **CARAU_PARSE_NAME_RDN_SNILS**

СНИЛС полного имени.

– **CARAU_PARSE_NAME_RDN_INNLE**

ИНН ИП полного имени.

– **CARAU_PARSE_NAME_RDN_TITLE**

Должность полного имени.

– **CARAU_PARSE_NAME_RDN_SURNAME**

Отчество полного имени.

– **CARAU_PARSE_NAME_RDN_GIVEN_NAME**

Имя полного имени.

– **CARAU_PARSE_NAME_RDN_COMMON_NAME**

CN полного имени.

– **CARAU_PARSE_NAME_RDN_EMAIL_ADDRESS**

Эл.почта полного имени.

– **CARAU_PARSE_NAME_RDN_UNSTRUCTURED_NAME**

Неструктурированное имя полного имени.

– **CARAU_PARSE_NAME_RDN_UNSTRUCTURED_ADDRESS**

Неструктурированный адрес полного имени.

– **CARAU_PARSE_NAME_RDN_ORGANIZATIONAL_UNIT**

Название подразделения полного имени.

– **CARAU_PARSE_NAME_RDN_ORGANIZATION_NAME**

Название организации полного имени.

– **CARAU_PARSE_NAME_RDN_STREET_ADDRESS**

Название улицы полного имени.

– **CARAU_PARSE_NAME_RDN_LOCALITY_NAME**

Название города полного имени.

– **CARAU_PARSE_NAME_RDN_STATE_PROVINCE_NAME**

Название области/региона полного имени.

– **CARAU_PARSE_NAME_RDN_COUNTRY_NAME**

Название страны полного имени.

– **CARAU_PARSE_NAME_RDN_DOMAIN_COMPONENT**

Имя домена полного имени.

– **CARAU_PARSE_NAME_ALT_EMAIL**

Эл.почта альтернативного имени.

- **CARAU_PARSE_NAME_ALT_DNS**
DNS альтернативного имени.
- **CARAU_PARSE_NAME_ALT_URI**
URI альтернативного имени.
- **CARAU_PARSE_NAME_ALT_IP**
IP адрес альтернативного имени.
- **CARAU_PARSE_NAME_ALT_ORGANIZATION_NAME**
Название организации альтернативного имени.
- **CARAU_PARSE_NAME_ALT_REGISTERED_ADDRESS**
Адрес регистрации альтернативного имени.
- **CARAU_PARSE_NAME_ALT_SURNAME**
Отчество альтернативного имени.
- **CARAU_PARSE_NAME_ALT_BUSINESS_CATEGORY**
Область деятельности альтернативного имени.
- **CARAU_PARSE_NAME_ALT_TELEPHONE_NUMBER**
Номер телефона альтернативного имени.
- **CARAU_PARSE_NAME_ALT_DESCRIPTION**
Описание альтернативного имени.
- **CARAU_PARSE_NAME_ALT_ACCOUNT_NUMBER**
Номер расчетного счета альтернативного имени.
- **CARAU_PARSE_NAME_ALT_BANK_ID**
БИК альтернативного имени.
- **CARAU_PARSE_NAME_ALT_PHYSICAL_DELIVERY**
Почтовый адрес альтернативного имени.
- **CARAU_PARSE_NAME_ALT_EXCHANGE_ADDRESS**
Адрес Microsoft Exchange альтернативного имени.
- **CARAU_PARSE_NAME_ALT_NOTES_ADDRESS**
Адрес Lotus Notes альтернативного имени.
- **CARAU_PARSE_NAME_ALT_PASSPORT_INFORMATION**
Данные паспорта альтернативного имени.
- **CARAU_PARSE_NAME_ALT_UPN**
UPN альтернативного имени.
- **CARAU_PARSE_NAME_ALT_INN**
ИНН альтернативного имени.
- **CARAU_PARSE_NAME_ALT_OGRN**
ОГРН альтернативного имени.
- **CARAU_PARSE_NAME_ALT_OGRNIP**
ОГРН ИП альтернативного имени.
- **CARAU_PARSE_NAME_ALT_SNILS**
СНИЛС альтернативного имени.
- **CARAU_PARSE_NAME_ALT_INNLE**
ИНН ИП альтернативного имени. альтернативного имени.

– **CARAU_PARSE_NAME_ISSUER**

Флаг использования полного имени издателя вместо полного имени владельца.

Перечисление find_object_param_type_t

Определяет данные, передаваемые для поиска объектов:

– **FOP_IAS(0)**

Передаётся значение типа issuer_and_serial_t.

– **FOP_LONG(1)**

Передаётся значение типа 32-бит целое число.

– **FOP_STRING(2)**

Передаётся значение типа idl_char.*

– **FOP_INTERVAL(3)**

Передаётся значение типа interval_t.

Структура issuer_and_serial_t

Пара издателя и серийного номера сертификата:

– idl_char * **issuer**

Строка с X.500-именем издателя сертификата.

– idl_char * **serialNumber**

Строка с серийным номером сертификата.

Структура interval_t

Временной интервал:

– time_t **not_before**

Начало интервала - 64-бит целое число.

– time_t **not_after**

Конец интервала - 64-бит целое число.

Структура find_object_param_t

Параметр поиска объектов СУС:

– find_object_param_type_t **type**

*Определяет используемое поле последующего **union**.*

– issuer_and_serial_t **ias_p**

*Пара издателя и серийного номера сертификата. Используется если **type** равен **FOP_IAS***

– idl_long_int **long_p**

*32-бит целое число. Используется если **type** равен **FOP_LONG***

– idl_char * **string_p**

*Строка c-style. Используется если **type** равен **FOP_STRING***

– interval_t **interval_p**

*Временной интервал. Используется если **type** равен **FOP_INTERVAL***

1.7 Инициализация и деинициализация

Перед использованием любой из функций библиотеки (за исключением функций инициализации) библиотеку необходимо инициализировать - открыть сессию (получить дескриптор сессии) с сервисом ЦР или ЦС.

Для инициализации сессии с удалённым сервисом ЦР или ЦС указывается адрес сервиса в формате **ncasn_ip_tcp:<ip-адрес узла сервиса>[<номер порта сервиса>]**. Для сервиса ЦР стандартный номер порта **13434**, для сервиса ЦС **13333**.

Сессия возвращаемая любой из таких функций, является безопасной для параллельного (многопоточного) использования - т.е. данная сессия может быть использована в вызовах функций библиотеки параллельно из нескольких потоков. При этом сами функции инициализации и деинициализации не являются безопасными для параллельного (многопоточного) использования:

- в рамках данного конкретного процесса операционной системы (ОС) все вызовы функций инициализации и деинициализации должны быть сериализованы, т.е. должны выполняться последовательно.

- в рамках различных процессов ОС, выполняющихся от имени данного конкретного пользователя, функции инициализации и деинициализации можно вызывать параллельно.

Для сессии с удаленным сервисом ЦР/ЦС возможна установка тайм-аута - периода времени ожидания ответа сервиса, по истечении которого при отсутствии ответа возвратится соответствующая ошибка. Тайм-аут задается целым числом от 0 до 10, по умолчанию используется значение 5:

- значение 0 (**crsv_rpc_c_binding_min_timeout**) соответствует тайм-ауту 1 сек;

- значениям 1-5 соответствует тайм-аут 12*<значение> сек (**crsv_rpc_c_binding_default_timeout** равно 5 соответствует 60 сек);

- значениям 6-9 соответствует тайм-аут 15*<значение> сек (**crsv_rpc_c_binding_max_timeout** равно 9 соответствует 135 сек);

- значение 10 (**crsv_rpc_c_binding_infinite_timeout**) соответствует тайм-ауту 86400 сек (одни сутки 24 часа).

1.7.1 Функции инициализации библиотеки работы с ЦР

CRSV_HRESULT RACLI_OpenSessionLocal (CRSV_LPVOID *pSessionHandle)

Функция инициализации сессии с локальным сервисом ЦР

Аргументы:

- **pSessionHandle** (out) указатель на возвращаемый дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT RACLI_OpenSessionRemote (CRSV_LPCSTR lpszBindstring, CRSV_LPVOID *pSessionHandle)

Функция инициализации сессии с удаленным сервисом ЦР

Аргументы:

- ***lpzBindstring*** (in) адрес сервиса;
- ***pSessionHandle*** (out) указатель на возвращаемый дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **RACLI_SetTimeout** (CRSV_LPVOID pSessionHandle, unsigned iTimeout)

Функция установки тайм-аута операций для сессии с удаленным сервисом ЦР

Аргументы:

- ***pSessionHandle*** (in) дескриптор сессии;
- ***iTimeout*** (in) значение тайм-аута (от 0 до 10);

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.7.2 Функции деинициализации библиотеки работы с ЦР

CRSV_HRESULT **RACLI_CloseSession** (CRSV_LPVOID pSessionHandle)

Функция инициализации сессии с локальным сервисом ЦР

Аргументы:

- ***pSessionHandle*** (in) дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.7.3 Функции инициализации библиотеки работы с ЦС

CRSV_HRESULT **CACLI_OpenSessionLocal** (CRSV_LPVOID *pSessionHandle)

Функция инициализации сессии с локальным сервисом ЦС

Аргументы:

- ***pSessionHandle*** (out) указатель на возвращаемый дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **CACLI_OpenSessionRemote** (CRSV_LPCSTR lpzBindstring, CRSV_LPVOID *pSessionHandle)

Функция инициализации сессии с удаленным сервисом ЦС

Аргументы:

- ***lpzBindstring*** (in) адрес сервиса;
- ***pSessionHandle*** (out) указатель на возвращаемый дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **CACLI_SetTimeout** (CRSV_LPVOID pSessionHandle, unsigned iTimeout)

Функция установки тайм-аута операций для сессии с удаленным сервисом ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **iTimeout** (in) значение тайм-аута (от 0 до 10);

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.7.4 Функции деинициализации библиотеки работы с ЦС

CRSV_HRESULT **CACLI_CloseSession** (CRSV_LPVOID pSessionHandle)

Функция инициализации сессии с локальным сервисом ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.8 Тестирование соединения с сервисами

1.8.1 Функции тестирования соединения с сервисом ЦР

CRSV_HRESULT **RACLI_PingService** (CRSV_LPVOID pSessionHandle)

Функция тестирования соединения с сервисом ЦР

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **RACLI_CAPingService** (CRSV_LPVOID pSessionHandle)

Функция тестирования соединения сервиса ЦР с сервисом ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.8.2 Функции тестирования соединения с сервисом ЦС

CRSV_HRESULT **CACLI_PingService** (CRSV_LPVOID pSessionHandle)

Функция тестирования соединения с сервисом ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.9 Обработка запросов сервиса ЦР

1.9.1 Функции обработки запросов на выпуск сертификатов

CRSV_HRESULT **RACLI_REQBlockToREQBlock** (CRSV_LPVOID pSessionHandle, const mem_blk_t *pInREQBlock, mem_blk_t *pOutREQBlock)

Функция обработки запроса на выпуск сертификата

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **pInREQBlock** (in) запрос на выпуск сертификата;
- **pOutREQBlock** (out) подготовленный запрос на выпуск сертификата для ЦС;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Запрос на выпуск сертификата в формате **PKCS#10** должен быть подписан на сертификате Оператора ЦР и передаваться в контейнере **PKCS#7** с присоединенной подписью.

CRSV_HRESULT **RACLI_XmlBlockMergeWithREQBlock** (CRSV_LPVOID pSessionHandle, const mem_blk_t *pXmlBlock, const mem_blk_t *pInREQBlock, mem_blk_t *pOutREQBlock, flag_t fXmlFlags)

Функция обработки запроса на выпуск сертификата с модифицирующим шаблоном

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **pXmlBlock** (in) модифицирующий шаблон запроса в формате XML;
- **pInREQBlock** (in) запрос на выпуск сертификата;
- **pOutREQBlock** (out) подготовленный запрос на выпуск сертификата для ЦС;
- **fXmlFlags** (in) флаги обработки запроса - должен передаваться 0;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Запрос на выпуск сертификата в формате **PKCS#10** должен быть подписан на сертификате Оператора ЦР и передаваться в контейнере **PKCS#7** с присоединенной подписью.

1.9.2 Функции обработки выпущенных сертификатов

CRSV_HRESULT **RACLI_X509BlockToX509Block** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pInX509Block, mem_blk_t *pOutX509Block)

Функция обработки полученного от ЦС блока данных

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **pInX509Block** (in) полученный от ЦС блок данных, содержащий выпущенный сертификат;
- **pOutX509Block** (out) сертификат для передачи владельцу;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.9.3 Функции обработки запросов на отзыв сертификатов

CRSV_HRESULT **RACLI_REVREQBlockToREVREQBlock** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pInREVREQBlock, mem_blk_t *pOutREVREQBlock)

Функция обработки запроса на отзыв сертификата

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **pInREVREQBlock** (in) запрос на отзыв сертификата;
- **pOutREVREQBlock** (out) подготовленный запрос на отзыв сертификата для ЦС;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Запрос на отзыв сертификата должен быть подписан на сертификате Оператора ЦР и передаваться в контейнере **PKCS#7** с присоединенной подписью.

CRSV_HRESULT **RACLI_XmlBlockToREVREQBlock** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pXmlBlock, mem_blk_t *pREVREQBlock, flag_t fXmlFlags)

Функция обработки запроса на отзыв сертификата в формате XML

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **pXmlBlock** (in) запрос на отзыв сертификата в формате XML;
- **pREVREQBlock** (out) подготовленный запрос на отзыв сертификата для ЦС;
- **fXmlFlags** (in) флаги обработки запроса - должен передаваться 0;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Запрос на отзыв сертификата в формате XML должен быть подписан на сертификате Оператора ЦР и передаваться в контейнере **PKCS#7** с присоединенной подписью.

1.9.4 Функции получения списка объектов СУС

CRSV_HRESULT **RACLI_FindObjects** (CRSV_LPVOID pSessionHandle, CARASEARCH_STORE storetype, CARACOMMON_OBJECT objecttype, CARASEARCH_VALUE valuetype, CRSV_LPVOID findparam, idl_long_int iFltCount, CARAUI_PARSE_NAME *pFltParse, idl_char **ppszFltValue, idl_long_int *piFltOffset, mem_blk_t **ppobjects, idl_long_int *piobjcount)

Функция получения объектов СУС по указанным критериям

Аргументы:

- **pSessionHandle** (in) дескриптор сессии;
- **storetype** (in) хранилище, откуда будут получены объекты СУС. Одно из значений перечисления **CARASEARCH_STORE**;
- **objecttype** (in) тип объектов СУС. Одно из значений перечисления **CARACOMMON_OBJECT**;
- **valuetype** (in) дополнительный критерий поиска объектов, определяет значение параметра **findparam**. Одно из значений перечисления **CARASEARCH_VALUE**;
- **findparam** (in) данные дополнительного критерия поиска;
- **iFltCount** (in) количество элементов в массивах **pFltParse**, **ppszFltValue** и **piFltOffset**. Должно быть в диапазоне от 0 до 255;
- **pFltParse** (in) указатель на массив значений перечисления **CARAUI_PARSE_NAME**. Должен быть **NULL** если **iFltCount** равен 0;
- **ppszFltValue** (in) указатель на массив строк. Должен быть **NULL** если **iFltCount** равен 0;
- **piFltOffset** (in) указатель на массив смещений (чисел в диапазоне от 0 до 255). Должен быть **NULL** если **iFltCount** равен 0;
- **ppobjects** (out) указатель массив блоков памяти - список объектов СУС;
- **piobjcount** (out) количество элементов в массиве **ppobjects**;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Функция выполняет поиск объектов указанного типа в указанном хранилище по указанному критерию. Значение параметра **valuetype CARASEARCH_VALUE_ANY_OBJECT** означает получение всех объектов данного типа из указанного хранилища, при этом параметры **findparam**, **pFltParse**, **ppszFltValue** и **piFltOffset** должны быть **NULL**, параметр **iFltCount** должен быть 0. Для остальных значений параметра **valuetype** дополнительные критерии поиска заполня-

ются соответствующими значениями.

1.10 Обработка запросов сервиса ЦС

Функции сервиса ЦС вызываются двумя способами:

- используя сессию с сервисом ЦС. Для этого требуется сетевой доступ к узлу, на котором сервис работает;
- используя сессию с сервисом ЦР и соединение сервиса ЦР с сервисом ЦС. В этом случае требуется лишь сетевой доступ к узлу сервиса ЦР для вызова функций обоих сервисов;

1.10.1 Функции обработки запросов на выпуск сертификатов

CRSV_HRESULT **CACLI_REQBlockToX509Block** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pReqblock, mem_blk_t *pX509block, idl_
long_int certmonths, idl_long_int keymonths)

Функция обработки подготовленного для ЦС запроса на выпуск сертификата
Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦС;
- **pReqblock** (in) подготовленный запрос на выпуск сертификата для ЦС;
- **pX509block** (out) блок данных для обработки ЦР, содержащий выпущенный сертификат;
- **certmonths** (in) срок действия сертификата, в месяцах;
- **keymonths** (in) срок действия закрытого ключа, в месяцах;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Полученный в случае успеха блок данных, содержащий выпущенный сертификат, должен быть обработан с помощью функции ЦР **RACLI_X509BlockToX509Block**.

CRSV_HRESULT **RACLI_CAREQBlockToX509Block** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pReqblock, mem_blk_t *pX509block, idl_
long_int certmonths, idl_long_int keymonths)

Функция обработки подготовленного для ЦС запроса на выпуск сертификата
Это функция ЦР для вызова функции CACLI_REQBlockToX509Block ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦР;
- **pReqblock** (in) подготовленный запрос на выпуск сертификата для ЦС;
- **pX509block** (out) блок данных для обработки ЦР, содержащий выпущенный сертификат;
- **certmonths** (in) срок действия сертификата, в месяцах;
- **keymonths** (in) срок действия закрытого ключа, в месяцах;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Полученный в случае успеха блок данных, содержащий выпущенный сертификат, должен быть обработан с помощью функции ЦР **RACLI_X509BlockToX509Block**.

CRSV_HRESULT **CACLI_XmlBlockMergeWithREQBlockToX509Block**
(CRSV_LPVOID pSessionHandle, const mem_blk_t *pXmlBlock, const mem_blk_t *pReqblock, mem_blk_t *pX509block, flag_t fXmlFlags, idl_long_int certmonths, idl_long_int keymonths)

Функция обработки подготовленного для ЦС запроса на выпуск сертификата с модифицирующим шаблоном

Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦС;
- **pXmlBlock** (in) модифицирующий шаблон в формате XML;
- **pReqblock** (in) подготовленный запрос на выпуск сертификата для ЦС;
- **pX509block** (out) выпущенный сертификат;
- **fXmlFlags** (in) флаги обработки модифицирующего шаблона, должен быть равен 0;
- **certmonths** (in) срок действия сертификата, в месяцах;
- **keymonths** (in) срок действия закрытого ключа, в месяцах;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Полученный в случае успеха блок данных, содержащий выпущенный сертификат, должен быть обработан с помощью функции ЦР **RACLI_X509BlockToX509Block**.

CRSV_HRESULT **RACLI_CAXmlBlockMergeWithREQBlockToX509Block**
(CRSV_LPVOID pSessionHandle, const mem_blk_t *pXmlBlock, const mem_blk_t *pReqblock, mem_blk_t *pX509block, flag_t fXmlFlags, idl_long_int certmonths, idl_long_int keymonths)

Функция обработки подготовленного для ЦС запроса на выпуск сертификата с модифицирующим шаблоном

Это функция ЦР для вызова функции

CACLI_XmlBlockMergeWithREQBlockToX509Block ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦР;
- **pXmlBlock** (in) модифицирующий шаблон в формате XML;
- **pReqblock** (in) подготовленный запрос на выпуск сертификата для ЦС;
- **pX509block** (out) выпущенный сертификат;
- **fXmlFlags** (in) флаги обработки модифицирующего шаблона, должен быть равен 0;
- **certmonths** (in) срок действия сертификата, в месяцах;

- **keymonths** (in) срок действия закрытого ключа, в месяцах;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

Полученный в случае успеха блок данных, содержащий выпущенный сертификат, должен быть обработан с помощью функции ЦР **RACLI_X509BlockToX509Block**.

1.10.2 Функции обработки запросов на отзыв сертификатов

CRSV_HRESULT **CACLI_ProcessREVREQBlock** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pREVREQBlock)

Функция обработки подготовленного для ЦС запроса на отзыв сертификата

Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦС;
- **pREVREQBlock** (in) подготовленный запрос на отзыв сертификата для ЦС;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **RACLI_CAProcessREVREQBlock** (CRSV_LPVOID
pSessionHandle, const mem_blk_t *pREVREQBlock)

Функция обработки подготовленного для ЦС запроса на отзыв сертификата

Это функция ЦР для вызова функции CACLI_ProcessREVREQBlock ЦС

Аргументы:

- **pSessionHandle** (in) дескриптор сессии ЦР;
- **pREVREQBlock** (in) подготовленный запрос на отзыв сертификата для ЦС;

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.11 Управление памятью

Память, выделенная для возвращаемых данных в выходных параметрах (out) в структуре или массиве структур **mem_blk_t**, должна быть освобождена с помощью описанных ниже функций.

1.11.1 Функции освобождения блока памяти

CRSV_HRESULT **RACLI_FreeMemBlock** (mem_blk_t *pobject)

Функция освобождения блока памяти выходного параметра функций сервиса ЦР

Аргументы:

- **pobject** (in) указатель на блок памяти. Освобождается только память, на которую ссылается поле **buf** структуры.

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **CACLI_FreeMemBlock** (mem_blk_t *pobject)

Функция освобождения блока памяти выходного параметра функций сервиса ЦС

Аргументы:

- **pobject** (in) указатель на блок памяти. Освобождается только память, на которую ссылается поле **buf** структуры.

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

1.11.2 Функции освобождения массива блоков памяти

CRSV_HRESULT **RACLI_FreeMemBlocks** (mem_blk_t *pobject, idl_long_int iObjCount)

Функция освобождения массива блоков памяти

выходного параметра функций сервиса ЦР

Аргументы:

- **pobject** (in) указатель на массив блоков памяти. Освобождается как память, на которую ссылаются поля **buf** каждой структура массива, так и память, указываемая **pobject**;

- **iObjCount** (in) количество элементов в массиве блоков памяти.

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

CRSV_HRESULT **CACLI_FreeMemBlocks** (mem_blk_t *pobject, idl_long_int iObjCount)

Функция освобождения массива блоков памяти

выходного параметра функций сервиса ЦС

Аргументы:

- **pobject** (in) указатель на массив блоков памяти. Освобождается как память, на которую ссылаются поля **buf** каждой структура массива, так и память, указываемая **pobject**;

- **iObjCount** (in) количество элементов в массиве блоков памяти.

Возвращаемые значения:

- **0** в случае успеха или ненулевой код ошибки.

2 ОПИСАНИЕ ОШИБОЧНЫХ СИТУАЦИЙ

Ниже (Таблица 1) приведено описание возможных ошибочных ситуаций. В левой колонке указано символьное имя ошибки и шестнадцатеричное значение ее кода, в правой колонке приведено детальное описание и причина возникновения ошибки.

Таблица 1 – Описание ошибочных ситуаций

Имя и код ошибки	Описание и причина возникновения ошибки
CARASC_E_NO_MEMORY (0xE0D20001)	Ошибка выделения памяти.
CARASC_E_INVALID_PARAM (0xE0D20002)	Недопустимое значение параметра.
CARASC_E_NOT_IMPLEMENTED (0xE0D20003)	Функция отсутствует.
CARASC_E_INTERNAL_ERROR (0xE0D20004)	Внутренняя ошибка.
CARASC_E_DELETE_FOLDER (0xE0D20005)	Ошибка удаления папки.
CARASC_E_CREATE_FOLDER (0xE0D20006)	Ошибка создания папки.
CARASC_E_LOG_INIT (0xE0D20007)	Ошибка инициализации журналирования.
CARASC_E_INVALID_FOLDER (0xE0D20008)	Недопустимое имя папки.
CARASC_E_CFG_READ (0xE0D20009)	Ошибка чтения настроек.
CARASC_E_REG_ACCESS_DENIED (0xE0D2000A)	Недостаточно прав для операции с ключом/значением реестра.
CARASC_E_CFG_WRITE (0xE0D2000B)	Ошибка записи настроек.
CARASC_E_OPERATION_ - PROHIBITED (0xE0D2000C)	Операция запрещена.
CARASC_E_INVALID_SESSION (0xE0D2000D)	Недопустимый дескриптор сессии работы с сервисом.
CARASC_E_RPC_ERROR (0xE0D2000E)	Ошибка RPC.
CARASC_E_SESSION_ - INITIALIZED (0xE0D20010)	Сессия криптопровайдера уже инициализирована.
CARASC_E_INVALID_CRYPT_ - CTX (0xE0D20011)	Недопустимый дескриптор сессии криптопровайдера.
CARASC_EMSG_AUTO_BAD_OBJ_ - TYPE (0xE0D20012)	Неподдерживаемый тип объекта для авто-обработки.
CARASC_E_AUTO_PLUGIN_ - FAILED (0xE0D20013)	Ошибка вызова подключаемого модуля экспорта.

Имя и код ошибки	Описание и причина возникновения ошибки
CARASC_E_SYSTEM_ERROR (0xE0D20015)	Системная ошибка.
CARASC_E_CA_FWD_FAILED (0xE0D20016)	Ошибка вызова функции сервиса ЦС.
CARASC_E_CA_FWD_PING_- FAILED (0xE0D20017)	Сервис ЦС недоступен.
CARASC_E_LIC_DB_ERROR (0xE0D20018)	Ошибка получения количества сертификатов.
CARASC_E_LIC_BAD (0xE0D20019)	Отсутствует валидная лицензия.
CARASC_E_LIC_CERT_LIMIT (0xE0D2001A)	Превышен лимит сертификатов.
CARASC_E_REQ_LOG_CREATE (0xE0D2001B)	Ошибка создания таблиц журналов запросов.
CARASC_E_REQ_LOG_FIND_CERT (0xE0D2001C)	Ошибка чтения данных журнала запросов на выпуск сертификата.
CARASC_E_REQ_LOG_FIND_REV (0xE0D2001D)	Ошибка чтения данных журнала запросов на отзыв сертификата.
CARALIB_E_NO_MEMORY (0xE0C40001)	Недостаточно оперативной памяти.
CARALIB_E_INVALID_PARAM (0xE0C40002)	Передан неверный параметр.
CARALIB_E_INVALID_FLAGS (0xE0C40003)	Переданы неверные флаги.
CARALIB_E_INVALID_CONTEXT (0xE0C40004)	Неверный контекст библиотеки.
CARALIB_E_NOT_IMPLEMENTED (0xE0C40005)	Вызов данной функции не разрешен.
CARALIB_E_INTERNAL_ERROR (0xE0C40006)	Произошла внутренняя ошибка.
CARALIB_E_BUFFER_TOO_SMALL (0xE0C40007)	Размер буфера недостаточен.
CARALIB_E_DATA_MISSING (0xE0C40008)	Отсутствуют требуемые данные.
CARALIB_E_CANCELLED_BY_- USER (0xE0C40009)	Операция отменена пользователем.
CARALIB_E_BOOLEAN_XML_BLOCK (0xE0C4000A)	Неверный формат блока данных XML.
CARALIB_E_SYSTEM_DATE_TIME (0xE0C4000B)	Ошибка получения системного времени.
CARALIB_E_DATE_TIME_FORMAT (0xE0C4000C)	Неверный формат даты или времени.
CARALIB_E_STRING_ENCODING (0xE0C4000D)	Неверная кодировка строковых данных.
CARALIB_E_INVALID_CONFIG (0xE0C4000E)	Неверная конфигурация библиотеки.
CARALIB_E_CSP_INITIALIZE (0xE0C4000F)	Ошибка инициализации СКЗИ.
CARALIB_E_STRING_TO_HEX (0xE0C40010)	Ошибка конвертации строки в 16-ричный формат.

Имя и код ошибки	Описание и причина возникновения ошибки
CARALIB_E_GENERATE_KEYID (0xE0C40011)	Ошибка формирования идентификатора ключа ЭП.
CARALIB_E_CREATE_PRIVATE_KEY (0xE0C40012)	Ошибка формирования ключа ЭП.
CARALIB_E_LOAD_PRIVATE_KEY (0xE0C40013)	Ошибка загрузки ключа ЭП.
CARALIB_E_GET_PUBLIC_KEY (0xE0C40014)	Ошибка получения ключа проверки ЭП.
CARALIB_E_DUPLICATE_PUBLIC_KEY (0xE0C40015)	Сертификат с таким ключом проверки ЭП уже существует.
CARALIB_E_CREATE_NEW_STORE (0xE0C40016)	Ошибка создания нового хранилища.
CARALIB_E_OPEN_EXTANT_STORE (0xE0C40017)	Ошибка открытия существующего хранилища.
CARALIB_E_ADD_STORE_SIGNER (0xE0C40018)	Ошибка вычисления ЭП хранилища.
CARALIB_E_STORE_ADD_CERT (0xE0C40019)	Ошибка добавления сертификата в хранилище.
CARALIB_E_STORE_ADD_CRL (0xE0C4001A)	Ошибка добавления САС в хранилище.
CARALIB_E_STORE_ADD_REQ (0xE0C4001B)	Ошибка добавления запроса PKCS#10 в хранилище.
CARALIB_E_STORE_ADD_REVREQ (0xE0C4001C)	Ошибка добавления запроса на аннулирование в хранилище.
XCARALIB_E_STORE_NOT_SIGNED (0xE0C4001D)	Хранилище не подписано.
CARALIB_E_INVALID_STORE_URI (0xE0C4001E)	Неверный идентификатор хранилища.
CARALIB_E_INVALID_STORE_USAGE (0xE0C4001F)	Неверное использование хранилища.
CARALIB_E_STORE_ADD_FAILED (0xE0C40020)	Ошибка функции добавления объекта в хранилище.
CARALIB_E_STORE_MODIFY_FAILED (0xE0C40021)	Ошибка функции модификации объекта в хранилище.
CARALIB_E_STORE_FIND_FAILED (0xE0C40022)	Ошибка функции поиска объекта в хранилище.
CARALIB_E_STORE_CTRL_FAILED (0xE0C40023)	Ошибка функции управления хранилищем.
CARALIB_E_STORE_DELETE_FAILED (0xE0C40024)	Ошибка функции удаления объекта из хранилища.
CARALIB_E_STORE_EMPTY (0xE0C40025)	В хранилище отсутствуют объекты.
CARALIB_E_CERT_NOT_FOUND (0xE0C40026)	Сертификат не найден.

Имя и код ошибки	Описание и причина возникновения ошибки
CARALIB_E_CRL_NOT_FOUND (0xE0C40027)	CAC не найден.
CARALIB_E_REQ_NOT_FOUND (0xE0C40028)	Запрос PKCS#10 не найден.
CARALIB_E_REVREQ_NOT_FOUND (0xE0C40029)	Запрос на аннулирование не найден.
CARALIB_E_CALCULATE_KEY (0xE0C4002A)	Ошибка вычисления ключа объекта.
CARALIB_E_KEY_NOT_FOUND (0xE0C4002B)	Контекст библиотеки был создан для проверки ЭП.
CARALIB_E_WRITE_FILE. (0xE0C4002C)	Произошла ошибка при записи выходного файла.
CARALIB_E_PRINT_OBJECT (0xE0C4002D)	Произошла ошибка при печати объекта.
CARAASN_E_ASN1_CERT_ENCODE (0xE0C50001)	Ошибка кодирования сертификата X.509.
CARAASN_E_ASN1_CERT_DECODE (0xE0C50002)	Ошибка декодирования сертификата X.509.
CARAASN_E_ASN1_CRL_ENCODE (0xE0C50003)	Ошибка кодирования CAC X.509.
CARAASN_E_ASN1_CRL_DECODE (0xE0C50004)	Ошибка декодирования CAC X.509.
CARAASN_E_ASN1_REQ_ENCODE (0xE0C50005)	Ошибка кодирования запроса PKCS#10.
CARAASN_E_ASN1_REQ_DECODE (0xE0C50006)	Ошибка декодирования запроса PKCS#10.
CARAASN_E_ASN1_REVREQ_- ENCODE (0xE0C50007)	Ошибка кодирования запроса на аннулирование.
CARAASN_E_ASN1_REVREQ_- DECODE (0xE0C50008)	Ошибка декодирования запроса на аннулирование.
CARAASN_E_ASN1_PKCS7_- ENCODE (0xE0C50009)	Ошибка кодирования сообщения CMS/PKCS#7.
CARAASN_E_ASN1_PKCS7_- DECODE (0xE0C5000A)	Ошибка декодирования сообщения CMS/PKCS#7.
CARAASN_E_INVALID_PKCS7_- TYPE (0xE0C5000B)	Неверный тип сообщения CMS/PKCS#7.
CARAASN_E_ASN1_X509V3_- ENCODE (0xE0C5000C)	Ошибка кодирования расширения X.509v3.
CARAASN_E_ASN1_X509V3_- DECODE (0xE0C5000D)	Ошибка декодирования расширения X.509v3.
CARAPKI_E_X500_NAME_CREATE (0xE0C60001)	Ошибка создания X.500 имени владельца.
CARAPKI_E_X509V3_PREV_- SUBJECT (0xE0C60002)	Ошибка расширения "Предыдущее имя владельца".

Имя и код ошибки	Описание и причина возникновения ошибки
CARAPKI_E_X509V3_ALT_NAME (0xE0C60003)	Ошибка расширения "Альтернативное имя владельца".
CARAPKI_E_X509V3_KEY_USAGE (0xE0C60004)	Ошибка расширения "Использование ключа".
CARAPKI_E_X509V3_BASIC_- CONSTR (0xE0C60005)	Ошибка расширения "Базовые ограничения".
CARAPKI_E_X509V3_KEY_- USAGE_PERIOD (0xE0C60006)	Ошибка расширения "Период использования ключа".
CARAPKI_E_X509V3_EXT_KEY_- USAGE (0xE0C60007)	Ошибка расширения "Расширенное использование ключа".
CARAPKI_E_X509V3_CERT_- POLICY (0xE0C60008)	Ошибка расширения "Регламенты сертификата".
CARAPKI_E_X509V3_CUSTOM_- EXTENSION (0xE0C60009)	Ошибка расширения "Собственное расширение".
CARAPKI_E_X509V3_RA_- CERTIFICATE (0xE0C6000A)	Ошибка расширения "Сертификат Центра Регистрации".
CARAPKI_E_X509V3_PRIVATE_- KEYID (0xE0C6000B)	Ошибка расширения "Идентификатор ключа ЭП".
CARAPKI_E_X509V3_SUBJECT_- KEYID (0xE0C6000C)	Ошибка расширения "Идентификатор ключа владельца".
CARAPKI_E_X509V3_- AUTHORITY_KEYID (0xE0C6000D)	Ошибка расширения "Идентификатор ключа издателя".
CARAPKI_E_KEY_NOT_YET_- VALID (0xE0C6000E)	Ключ ЭП еще не действителен.
CARAPKI_E_KEY_HAS_EXPIRED (0xE0C6000F)	Ключ ЭП уже истек.
CARAPKI_E_KEY_USAGE_PERIOD (0xE0C60010)	Ошибка периода использования ключа.
CARAPKI_E_CERT_NOT_YET_- VALID (0xE0C60011)	Сертификат еще не действителен.
CARAPKI_E_CERT_HAS_EXPIRED (0xE0C60012)	Сертификат уже истек.
CARAPKI_E_CERT_USAGE_- PERIOD (0xE0C60013)	Ошибка периода использования сертификата.
CARAPKI_E_CERT_IS DAMAGED (0xE0C60014)	Сертификат поврежден или искажен.
CARAPKI_E_CERT_IS MISSING (0xE0C60015)	Отсутствует сертификат издателя.
CARAPKI_W_CERT_IS_ON_HOLD (0xA0C60016)	Действие сертификата приостановлено.
CARAPKI_E_CERT_IS_- UNTRUSTED (0xE0C60017)	Сертификат не является доверенным.

Имя и код ошибки	Описание и причина возникновения ошибки
CARAPKI_E_CRL_NOT_YET_VALID (0xE0C60018)	CAC еще не действителен.
CARAPKI_E_CRL_HAS_EXPIRED (0xE0C60019)	CAC уже истек.
CARAPKI_E_CRL_IS_DAMAGED (0xE0C6001A)	CAC поврежден или искажен.
CARAPKI_E_CRL_IS_MISSING (0xE0C6001B)	Отсутствует CAC издателя.
CARAPKI_E_INVALID_USAGE (0xE0C6001C)	Неверное использование ключа или сертификата.
CARAPKI_E_INVALID_SIGNATURE (0xE0C6001D)	Произошла ошибка проверки ЭП.
CARAPKI_E_INVALID_SIGNERS_COUNT (0xE0C6001E)	Неверное количество подписантов.
CARAPKI_E_CHAIN_TOO_LONG (0xE0C6001F)	Цепочка сертификации слишком длинная.
CARAPKI_E_BROKEN_CONSTRAINT (0xE0C60020)	Нарушены базовые ограничения.
CARAPKI_E_BROKEN_HIERARCHY (0xE0C60021)	Нарушены ограничения иерархии.
CARAPKI_E_INVALID_CA_CERT (0xE0C60022)	Неверный сертификат Центра Сертификации.
CARAPKI_E_INVALID_RA_CERT (0xE0C60023)	Неверный сертификат Центра Регистрации.
CARAPKI_E_INVALID_CERT_TYPE (0xE0C60024)	Неверный тип сертификата.
CARAPKI_E_CERT_SIGNING_FAILED (0xE0C60025)	Произошла ошибка вычисления ЭП сертификата.
CARAPKI_E_CRL_SIGNING_FAILED (0xE0C60026)	Произошла ошибка вычисления ЭП CAC.
CARAPKI_E_REQ_SIGNING_FAILED (0xE0C60027)	Произошла ошибка вычисления ЭП запроса PKCS#10.
CARAPKI_E_REVREQ_SIGNING_FAILED (0xE0C60028)	Произошла ошибка вычисления ЭП запроса на аннулирование.
CARAPKI_E_DIGEST_FAILED (0xE0C60029)	Произошла ошибка вычисления хэш-значения.
CARAPKI_E_SIGNING_FAILED (0xE0C6002A)	Произошла общая ошибка вычисления ЭП.
CARAPKI_E_INVALID_ISSUER (0xE0C6002B)	Неверное имя издателя.
CARAPKI_E_SUBJECT_MISSING (0xE0C6002C)	Имя владельца отсутствует.
CARAPKI_E_INVALID_SUBJECT (0xE0C6002D)	Неверное имя владельца.

Имя и код ошибки	Описание и причина возникновения ошибки
CARAPKI_E_PUBLIC_KEY_- MISSING (0xE0C6002E)	Отсутствует ключ проверки ЭП.
CARAPKI_E_INVALID_PUBLIC_- KEY (0xE0C6002F)	Неверный ключ проверки ЭП.
CARAPKI_E_EXTENSION_- MISSING (0xE0C60030)	Отсутствует расширение X.509v3.
CARAPKI_E_INVALID_- EXTENSION (0xE0C60031)	Неверное расширение X.509v3.
CARAPKI_E_ATTRIBUTE_- MISSING. (0xE0C60032)	Отсутствует требуемый атрибут.
CARAPKI_E_REVOCATION_- MISSING (0xE0C60033)	Отсутствуют данные аннулирования.
CARAPKI_E_INVALID_- REVOCATION (0xE0C60034)	Неверные данные аннулирования.
CARAPKI_E_HANDLING_- PROHIBITED (0xE0C60035)	Обработка запрещена.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС	Операционная система (Operating System)
ПК	Программный комплекс
ПО	Программное обеспечение
САС	Список аннулированных сертификатов (Certificate Revocation List)
СКЗИ	Система криптографической защиты информации
СУС	Система управления сертификатами (Public Key Infrastructure)
ЦР	Центр регистрации (Registration Authority)
ЦС	Центр сертификации (Certification Authority)
ЭП	Электронная подпись

[illegible][illegible]