

NATIVE функции

компоненты CipherLabTCP8

Большинство функций аналогичны функциям из 1С, за исключением специфичных для настройки и обработки событий.

Принцип работы компоненты

Вначале создаем объект функцией **cipherlabtcpCreate** и получаем дескриптор сервера-компоненты **handle**.

Далее указываем функцию Колбека **cipherlabtcpSetEventCallback**, или параметры окна **cipherlabtcpSetEventMessage**, которые будут перехватывать сообщения от терминала. Указывать нужно только один из механизмов. Если указать оба механизма, то будет работать только Колбек. Если не указать ни один из механизмов, то терминал просто не будет получать ответы, а компонента не будет их транслировать программе.

Затем, если необходимо, устанавливаем параметры компоненты функциями **cipherlabtcpGetProperties** и **cipherlabtcpSetProperties**.

Далее запускается сам сервер обработки запросов от терминала сбора данных функцией **cipherlabtcpStart**. Если сервер необходимо остановить, вызываем функцию **cipherlabtcpStop**. После остановки сервера и изменения параметров его можно запустить повторно.

Остальные функции работают внутри обработчика событий: функции Колбека или окне, перехватывающем сообщения. Вначале обработки события необходимо получить имя события функцией **cipherlabtcpGetEvent**. Далее получаем данные от ТСД, функцией **cipherlabtcpGetData**. После обработки данных, подготавливаем ответ терминалу и отсылаем его функцией **cipherlabtcpAnswer**. Эти три функции минимально необходимы для правильной работы процесса обмена данными с терминалом. Функции **cipherlabtcpSendMessage**, **cipherlabtcpSendWarning**, **cipherlabtcpGetSerialNumber** являются вспомогательными и не являются обязательными.

Отправить сообщение или предупреждение на ТСД можно в любое время, если сервер запущен. Но из-за того, что принципа работы ТСД (включение-запрос-ответ-отключение), сообщения не поступят на ТСД сразу, а только тогда, когда терминал проявит активность и пошлет запрос на сервер. В ответ будет добавлена строка с сообщением.

Функция **cipherlabtcpDestroy** вызывается после остановки сервера, после того как надобность в сервере отпала и ресурсы необходимо освободить.

Функции компоненты

1. Название: **int cipherlabtcpCreate(void** ptrthandle)**

Входные параметры:

- **ptrthandle (OUT)** - указатель на дескриптор сервера-компоненты (дескриптор имеет тип void*), который используется в последующих вызовах, может быть равен NULL. Значение дескриптора должно быть инициализировано NULL.

Выходные параметры: код ошибки

Описание: Инициализация компоненты-объекта сервера. Создается объект-сервера. Если **ptrthandle** равен NULL, подразумевается использование одной "глобальной" компоненты-сервера.

2. Название: **int cipherlabtcpDestroy(void* handle)**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты

Выходные параметры: код ошибки

Описание: Освобождение ресурсов. Удаление ранее созданного объекта-сервера. Если равен NULL, подразумевается удаление "глобальной" компоненты-сервера.

3. Название: **int cipherlabtcpSetEventCallback(void* handle, void* callbackfunc, void* param1)**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **callbackfunc (IN)** - указатель на callback-функцию, которая будет вызвана при появлении события от терминала
- **param1 (IN)** - произвольное значение, передаваемое в callback-функцию при ее вызове (в первом параметре)

Выходные параметры: код ошибки

Описание: Устанавливает callback-функцию для события от терминала. Параметр функции **eventhandle** - дескриптор "объекта-события". Прототип callback-функции: **int callbackfunc(void* param1, void* eventhandle);**

4. Название: **int cipherlabtcpSetEventMessage(void* handle, void* hwnd, int messageid, void* lparam)**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **hwnd (IN)** - дескриптор окна, которому посылается сообщение
- **messageid (IN)** - идентификатор отправляемого сообщения
- **lparam (IN)** - заданный (постоянный) параметр

Выходные параметры: код ошибки

Описание: Задание оконного сообщения для события. Задаёт окно, идентификатор сообщения и параметр LPARAM для оконного сообщения, которое отсылается при наступлении события. В параметре WPARAM сообщения передается параметр (void*) **eventhandle**.

5. Название: **int cipherlabtcpGetLastError(void* handle, int* lasterror)**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **lasterror (OUT)** - указатель на переменную, куда будет помещен код системной ошибки

Выходные параметры: код ошибки

Описание: Получение кода последней системной ошибки. Если какая-либо функция возвращает код ошибки -1, то данной функцией можно узнать код системной ошибки

6. Название: `int cipherlabtcpGetProperties(void* handle, int* Port, int* SaveMethod, int* Debug)`

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **Port (OUT)** - указатель на свойство Порт/Port (допустимо NULL)
- **SaveMethod (OUT)** - указатель на свойство МетодЗаписи/SaveMethod (допустимо NULL)
- **Debug (OUT)** - указатель на свойство Отладка/Debug (допустимо NULL)

Выходные параметры: код ошибки

Описание: Получение параметров. Если указатель задан (не NULL), функция вернет значение

7. Название: `int cipherlabtcpSetProperties(void* handle, int Port, int SaveMethod, int Debug)`

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **Port (IN)** - свойство Порт/Port
- **SaveMethod (IN)** - свойство МетодЗаписи/SaveMethod
- **Debug (IN)** - свойство Отладка/Debug

Выходные параметры: код ошибки

Описание: Задание параметров. Устанавливает значение описанных параметров

8. Название: `int cipherlabtcpStart(void* handle)`

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты

Выходные параметры: код ошибки

Описание: Запуск сервера

9. Название: `int cipherlabtcpStop(void* handle)`

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты

Выходные параметры: код ошибки

Описание: Остановка сервера

**10. Название: `int cipherlabtcpSendMessageA(void* handle, char* NumberTSD, char* TextOfMessage)`
`int cipherlabtcpSendMessage(void* handle, wchar_t* NumberTSD, wchar_t* TextOfMessage)`**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **NumberTSD (IN)** - серийный номер ТСД
- **TextOfMessage (IN)** - текстовое сообщение

Выходные параметры: код ошибки

Описание: Отправка текстового сообщения на терминал. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

11. Название: `int cipherlabtcpSendWarningA(void* handle, char* NumberTSD, char* TextOfMessage)`

int cipherlabtcpSendWarning(void* handle, wchar_t* NumberTSD, wchar_t* TextOfMessage)

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **NumberTSD (IN)** - серийный номер ТСД
- **TextOfMessage (IN)** - текстовое сообщение

Выходные параметры: код ошибки

Описание: Отправка текстового предупреждения на терминал. Отличие от функции отправки сообщения в том, что данная функция заставляет терминал вибрировать и мигать красным светодиодом. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

12. Название: int cipherlabtcpGetSerialNumberA(void* handle, void* eventhandle, char* NumberTSD)
int cipherlabtcpGetSerialNumber(void* handle, void* eventhandle, wchar_t* NumberTSD)

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **eventhandle (IN)** - дескриптор события, возникшего в результате запроса от терминала
- **NumberTSD (OUT)** - адрес буфера должен указывать на выделенную область памяти, достаточную для получения как минимум 256 символов; если NULL - значение не формируется

Выходные параметры: код ошибки

Описание: Получение серийного номера терминала в обработчике события. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

13. Название: int cipherlabtcpGetEventA(void* handle, void* eventhandle, char* messagebuf)
int cipherlabtcpGetEvent(void* handle, void* eventhandle, wchar_t* messagebuf)

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **eventhandle (IN)** - дескриптор события, возникшего в результате запроса от терминала
- **messagebuf (OUT)** - буфер должен указывать на выделенную область памяти, достаточную для получения как минимум 256 символов; если NULL - значение не формируется

Выходные параметры: код ошибки

Описание: Получение строки-события в обработчике события. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

14. Название: int cipherlabtcpGetDataA(void* handle, void* eventhandle, int* NumberDB, int* NumberForm, char* Field1, char* Field2, char* Field3, char* Field4, char* Field5, char* Field6, char* Field7, char* Field8)
int cipherlabtcpGetData(void* handle, void* eventhandle, int* NumberDB, int* NumberForm, wchar_t* Field1, wchar_t* Field2, wchar_t* Field3, wchar_t* Field4, wchar_t* Field5, wchar_t* Field6, wchar_t* Field7, wchar_t* Field8)

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **eventhandle (IN)** - дескриптор события, возникшего в результате запроса от терминала
- **NumberDB (OUT)** - номер базы данных в терминале (всего до трех БД), допустимо NULL

- **NumberForm (OUT)** - номер формы в терминале (всего до 5 форм), допустимо NULL
- **Field1 : Field8 (OUT)** - поле базы данных терминала, указатель должен указывать на выделенную область памяти, достаточную для получения как минимум 256 символов

Выходные параметры: код ошибки

Описание: Получение данных, поступивших с терминала, в обработчике события. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

15. *Название:* **int cipherlabtcpAnswerA(void* handle, void* eventhandle, char* OutField1, char* OutField2, char* OutField3, char* OutField4, char* OutField5, char* OutField6, char* OutField7, char* OutField8, char* NewRow, char* NewForm)**

int cipherlabtcpAnswer(void* handle, void* eventhandle, wchar_t* OutField1, wchar_t* OutField2, wchar_t* OutField3, wchar_t* OutField4, wchar_t* OutField5, wchar_t* OutField6, wchar_t* OutField7, wchar_t* OutField8, wchar_t* NewRow, wchar_t* NewForm)

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **eventhandle (IN)** - дескриптор события, возникшего в результате запроса от терминала
- **Field1 : Field8 (IN)** - поле базы данных терминала, указатель должен указывать на выделенную область памяти, достаточную для получения как минимум 256 символов
- **NewRow (IN)** - номер новой строки формы, если необходимо перейти. Если значение равно NULL или задана пустая строка, работа формы продолжится в соответствии с настройками в Генераторе Приложений
- **NewForm (IN)** - номер новой формы, если необходимо перейти. Если значение равно NULL или задана пустая строка, работа формы продолжится в соответствии с настройками в Генераторе Приложений

Выходные параметры: код ошибки

Описание: Отправка данных в терминал, в обработчике события. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

16. **int cipherlabtcpGetAddr(void *handle, void *eventhandle, wchar_t *AddrTSD)
int cipherlabtcpGetAddrA(void *handle, void *eventhandle, char *AddTSD)**

Входные параметры:

- **handle (IN)** - дескриптор сервера-компоненты
- **eventhandle (IN)** - дескриптор события, возникшего в результате запроса от терминала
- **AddrTSD (OUT)** - адрес устройства должен указывать на выделенную область памяти, достаточную для получения как минимум 256 символов; если NULL - значение не формируется

Выходные параметры: код ошибки

Описание: Получение адреса устройства. Функция с суффиксом А работает с кодировкой 1251, без суффикса в UNICODE.

Коды ошибок:

- 0 - нет ошибки;
- 1 - внутренняя (системная) ошибка, ее код можно получить вызовом `cipherlabtcpGetLastError`;
- 2 - сервер уже запущен;
- 3 - от терминала нет событий;
- 4 - терминал не подключен;
- 5 - неверный дескриптор события;
- 6 - неверный параметр (скорее всего NULL);
- 7 - операция уже была выполнена ранее;
- 8 - недостаточно памяти;
- 9 - компонента не инициализирована;
- 10 - неверный дескриптор.