

# TrustedBitrix

Руководство разработчика



ООО «Цифровые технологии»

TrustedBitrix. Руководство разработчика. Версия 1.1.

Дата сборки документа 17.04.2013.

Этот документ является составной частью технической документации ООО «Цифровые технологии».

Сайт справки по продуктам ООО «Цифровые технологии» <http://www.trusted.ru>

© 2012-2013 ООО «Цифровые технологии». Все права защищены.

## **Контактная информация**

ООО «Цифровые технологии»

<http://www.trusted.ru>

## Содержание

<b>Введение .....</b>	<b>4</b>
<b>Раздел 1. Описание структуры хранилищ .....</b>	<b>5</b>
<b>Раздел 2. Описание API .....</b>	<b>7</b>
<b>КЛАСС CTB_CERTSTORE.....</b>	<b>7</b>
Методы класса CTB_CertStore.....	7
CTB_CertStore::hasClientCert.....	7
CTB_CertStore::GetClientCert.....	8
CTB_CertStore::isCertGOST.....	8
CTB_CertStore::parseClientData.....	8
CTB_CertStore::GetIDBlock.....	9
CTB_CertStore::GetPROP_STORE_ELEMENT.....	9
CTB_CertStore::SetPROP_STORE_ELEMENT.....	10
CTB_CertStore::Delete_STORE_ELEMENT.....	12
CTB_CertStore::AddCert.....	12
CTB_CertStore::DeleteCert.....	13
CTB_CertStore::ActivateCert.....	13
CTB_CertStore::DeactivateCert.....	13
CTB_CertStore::isValidityCert.....	14
CTB_CertStore::GetCountMappedCert.....	14
CTB_CertStore::GetIDCert.....	14
CTB_CertStore::GetIDUser.....	15
CTB_CertStore::AuthUserOnCert.....	15
<b>КЛАСС CTB_PROXY.....</b>	<b>16</b>
Методы класса CTB_Proxy.....	16
CTB_Proxy::GetStoreCode.....	16
CTB_Proxy::GetStoreType.....	16
CTB_Proxy::CTB_Proxy.....	16
CTB_Proxy::Delete.....	17
CTB_Proxy::GetID.....	17
CTB_Proxy::GetIP.....	17
CTB_Proxy::GetName.....	17
CTB_Proxy::GetProperty.....	17
CTB_Proxy::IsActive.....	18
CTB_Proxy::Save.....	18
CTB_Proxy::SetActive.....	18
CTB_Proxy::SetActiveTime.....	19
CTB_Proxy::SetID.....	19
CTB_Proxy::SetName.....	19
CTB_Proxy::SetProperty.....	20
Пример использования методов класса CTB_Proxy.....	20

## Введение

---

Руководство предназначено для администраторов порталов и разработчиков решений на базе систем *"1С-Битрикс: Управление сайтом"*, *"1С-Битрикс: Корпоративный портал"*, *"1С-Битрикс: Портал органов государственной власти"*, *"1С-Битрикс: Электронный магазин"*.

В руководстве представлены основные действия по установке и настройке продукта TrustedBitrix версии 1.1.0.

## Раздел 1. Описание структуры хранилищ

---

Для хранения сертификатов и информации о привязке сертификатов к пользователям используются хранилища на основе инфоблоков. Типы инфоблоков и сами инфоблоки создаются на этапе установки модуля.

**Важно!** В данной версии модуля не допускается изменение имен инфоблоков. Это может привести к некорректной работе процедуры авторизации.

Для хранилищ используются следующие инфоблоки:

Инфоблок CERT\_STORE – имеет тип TrustedBitrix\_cert и предназначен для хранения сертификатов по которым будет производиться авторизация пользователей.

Инфоблок помимо стандартных свойств имеет пользовательские свойства:

Название свойства	Тип	Код	Назначение
CERT_SERIAL	строка	CERT_SERIAL	Хранение серийного номера сертификата (Dec)
CERT_ISSUE	строка	CERT_ISSUE	Хранение информации об издателе сертификата
CERT_HASH	строка	CERT_HASH	Хранение хэш-суммы сертификата
CERT_USAGE	строка	CERT_USAGE	Хранение информации о назначении сертификата
CERT_VALID_FROM	строка	CERT_VALID_FROM	Хранение даты/времени начала действия сертификата
CERT_VALID_TO	строка	CERT_VALID_TO	Хранение даты/времени окончания действия сертификата
CERT_BLOB	HTML/текст	CERT_BLOB	Хранение тела сертификата (BASE64)

Инфоблок MAPPING\_STORE – имеет тип TrustedBitrix\_mapping и предназначен для хранения информации о пользователе (привязки сертификатов к пользователям сайта).

Инфоблок помимо стандартных свойств имеет пользовательские свойства:

Название свойства	Тип	Код	Назначение
USR_ID	Привязка к пользователю	USR_ID	Хранение идентификатора пользователя сайта
CERT_ID	Привязка к элементам	CERT_ID	Хранение идентификатора сертификата

Инфоблок PROXY\_STORE – имеет тип TrustedBitrix\_proxy и предназначен для хранения информации о доверенных прокси-серверах.

Инфоблок помимо стандартных свойств имеет пользовательские свойства:

Название свойства	Тип	Код	Назначение
IP	строка	IP	ip-адрес прокси-сервера

## Раздел 2. Описание API

---

### Класс CTB\_CertStore

**CTB\_CertStore** – класс для работы с сертификатами из потока Apache и загружаемыми в хранилище.

### Методы класса CTB\_CertStore

Метод	Описание
hasClientCert	Возвращает флаг наличия/отсутствия клиентского сертификата, передаваемого от Apache
GetClientCert	Возвращает тело сертификата, передаваемого от Apache
isCertGOST	Выполняет проверку на соответствие ГОСТ-сертификату
parseClientData	Выполняет разбор RSA-сертификата
GetIDIBlock	Возвращает ID инфоблока хранилища
GetPROP_STORE_ELEMENT	Возвращает свойства элемента в хранилище
SetPROP_STORE_ELEMENT	Устанавливает свойства элемента в хранилище
Delete_STORE_ELEMENT	Удаляет элемент из хранилища
AddCert	Добавляет сертификат в хранилище с привязкой к пользователю
DeleteCert	Удаляет сертификат из хранилища
ActivateCert	Активирует сертификат в хранилище
DeactivateCert	Деактивирует сертификат в хранилище
isValidityCert	Возвращает флаг корректности/некорректности сертификата
GetCountMappedCert	Возвращает количество привязанных к пользователю личных сертификатов
GetCountCert	Возвращает общее число сертификатов в хранилище
GetIDCert	Возвращает идентификатор загруженного в хранилище сертификата
GetIDUser	Возвращает идентификатор пользователя, к которому привязан сертификат
AuthUserOnCert	Выполняет авторизацию пользователя по сертификату

### CTB\_CertStore::hasClientCert

```
bool CTB_CertStore::hasClientCert();
```

Возвращает true, если в переменной \$\_SERVER присутствует тело сертификата клиентской стороны, который передается от Apache и false – в противном случае.

Параметры функции

Параметр	Описание
Отсутствуют	

### **CTB\_CertStore::GetClientCert**

```
array CTB_CertStore::GetClientCert();
```

Возвращает тело сертификата клиентской стороны, который передается от Apache в переменной \$\_SERVER. Если тело сертификата отсутствует, то возвращается false.

#### Параметры функции

Параметр	Описание
Отсутствуют	

### **CTB\_CertStore::isCertGOST**

```
bool CTB_CertStore::isCertGOST(
    string x509
);
```

Возвращает true, если сертификата создан на ГОСТ-алгоритмах и false – в противном случае.

#### Параметры функции

Параметр	Описание
x509	Тело сертификата

### **CTB\_CertStore::parseClientData**

```
array CTB_CertStore::parseClientData(
    string x509
);
```

Возвращает массив следующей структуры:



CERT_DATA['CERT_OWNER']	Владелец сертификата (DN сертификата)
CERT_DATA ['CERT_ISSUE']	Издатель сертификата
CERT_DATA ['CERT_HASH']	Хэш-сумма сертификата
CERT_DATA ['CERT_SERIAL']	Серийный номер сертификата
CERT_DATA ['CERT_USAGE']	Назначение сертификата (OID)
CERT_DATA ['CERT_VALID_FROM']	Сертификат действителен со времени
CERT_DATA ['CERT_VALID_TO']	Сертификат действителен до времени

#### Параметры функции

Параметр	Описание
x509	Тело сертификата

### **CTB\_CertStore::GetIDIBlock**

```
int CTB_CertStore::GetIDIBlock(
    string TRUSTED_STORE
);
```

Возвращает номер инфоблока хранилища сертификатов или хранилища информации о пользователе. Если инфоблоков не существуют или они имеют иные имена, то возвращается 0.

#### Параметры функции

Параметр	Описание
TRUSTED_STORE	Принимает два значения: 'CERT_STORE' – хранилище сертификатов; 'MAPPING_STORE' – хранилище информации о пользователях.

### **CTB\_CertStore::GetPROP\_STORE\_ELEMENT**

```
array CTB_CertStore::GetPROP_STORE_ELEMENT(
    int ID_ELEMENT,
    string TRUSTED_STORE
);
```

Возвращает массив свойств элемента инфоблока. Если инфоблоков не существуют или они имеют иные имена, то возвращается 0.

Свойства элемента хранилища сертификатов имеют структуру:

PROP_STORE['NAME']	Имя элемента (DN сертификата)
PROP_STORE['ACTIVE']	Флаг активности элемента инфоблока
PROP_STORE['DATE_ACTIVE_FROM']	Дата начала активности элемента
PROP_STORE['DATE_ACTIVE_TO']	Дата окончания активности элемента
PROP_STORE['CERT_ISSUE']	Издатель сертификата
PROP_STORE['CERT_HASH']	Хэш-сумма сертификата
PROP_STORE['CERT_SERIAL']	Серийный номер сертификата
PROP_STORE['CERT_USAGE']	Назначение сертификата (OID)
PROP_STORE['CERT_VALID_FROM']	Сертификат действителен со времени
PROP_STORE['CERT_VALID_TO']	Сертификат действителен до времени
PROP_STORE['CERT_BLOB']	Тело сертификата

Свойства элемента хранилища информации о пользователях имеют структуру:

PROP_STORE['NAME']	Имя элемента (Имя пользователя)
PROP_STORE['ACTIVE']	Флаг активности элемента инфоблока
PROP_STORE['DATE_ACTIVE_FROM']	Дата начала активности элемента
PROP_STORE['DATE_ACTIVE_TO']	Дата окончания активности элемента
PROP_STORE['USR_ID']	Идентификатор пользователя сайта (таблица User)
PROP_STORE['CERT_ID']	Идентификатор сертификата, привязанного к пользователю

#### Параметры функции

Параметр	Описание
ID_ELEMENT	Идентификатор элемента хранилища для которого выводятся свойства
TRUSTED_STORE	Принимает два значения: 'CERT_STORE' – хранилище сертификатов; 'MAPPING_STORE' – хранилище информации о пользователях.

### **CTB\_CertStore::SetPROP\_STORE\_ELEMENT**

```
bool CTB_CertStore::SetPROP_STORE_ELEMENT(
    int ID_ELEMENT,
```

```
string TRUSTED_STORE,  
array PROP_ELEMENT  
);
```

Возвращает true, если свойства элемента хранилища успешно изменены и false – в противном случае.

Свойства элемента хранилища сертификатов имеют структуру:

PROP_ELEMENT['NAME']	Имя элемента (DN сертификата)
PROP_ELEMENT ['ACTIVE']	Флаг активности элемента инфоблока
PROP_ELEMENT ['DATE_ACTIVE_FROM']	Дата начала активности элемента
PROP_ELEMENT ['DATE_ACTIVE_TO']	Дата окончания активности элемента
PROP_ELEMENT ['CERT_ISSUE']	Издатель сертификата
PROP_ELEMENT ['CERT_HASH']	Хэш-сумма сертификата
PROP_ELEMENT ['CERT_SERIAL']	Серийный номер сертификата
PROP_ELEMENT ['CERT_USAGE']	Назначение сертификата (OID)
PROP_ELEMENT ['CERT_VALID_FROM']	Сертификат действителен со времени
PROP_ELEMENT ['CERT_VALID_TO']	Сертификат действителен до времени
PROP_ELEMENT ['CERT_BLOB']	Тело сертификата

Свойства элемента хранилища информации о пользователях имеют структуру:

PROP_ELEMENT ['NAME']	Имя элемента (Имя пользователя)
PROP_ELEMENT ['ACTIVE']	Флаг активности элемента инфоблока
PROP_ELEMENT ['DATE_ACTIVE_FROM']	Дата начала активности элемента
PROP_ELEMENT ['DATE_ACTIVE_TO']	Дата окончания активности элемента
PROP_ELEMENT ['USR_ID']	Идентификатор пользователя сайта (таблица User)
PROP_ELEMENT ['CERT_ID']	Идентификатор сертификата, привязанного к пользователю

### Параметры функции

Параметр	Описание
ID_ELEMENT	Идентификатор элемента хранилища для которого выводятся свойства
TRUSTED_STORE	Принимает два значения: 'CERT_STORE' – хранилище сертификатов; 'MAPPING_STORE' – хранилище информации о пользователях.

PROP_ELEMENT	Массив со структурой для хранения сертификатов или информации о пользователе (Описание структур см.выше).
--------------	---

### **CTB\_CertStore::Delete\_STORE\_ELEMENT**

```
bool CTB_CertStore::Delete_STORE_ELEMENT(
    int ID_ELEMENT,
    string TRUSTED_STORE
);
```

Возвращает true, если элемента инфоблока удален из хранилища и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
ID_ELEMENT	Идентификатор элемента хранилища для которого выводятся свойства
TRUSTED_STORE	Принимает два значения: 'CERT_STORE' – хранилище сертификатов; 'MAPPING_STORE' – хранилище информации о пользователях.

### **CTB\_CertStore::AddCert**

```
bool CTB_CertStore::AddCert(
    string x509,
    int USER_ID
);
```

Возвращает true, если сертификата был успешно добавлен в хранилище и привязан к пользователю и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
x509	Тело сертификата
USER_ID	Идентификатор пользователя на сайте (таблица User).

### **CTB\_CertStore::DeleteCert**

```
bool CTB_CertStore::DeleteCert(  
    int ID_CERT  
);
```

Возвращает true, если сертификата был успешно удален из хранилища и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
ID_CERT	Идентификатор сертификата в хранилище 'CERT_STORE'

### **CTB\_CertStore::ActivateCert**

```
bool CTB_CertStore::ActivateCert(  
    int ID_CERT  
);
```

Возвращает true, если сертификат был успешно активирован и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
ID_CERT	Идентификатор сертификата в хранилище 'CERT_STORE'

### **CTB\_CertStore::DeactivateCert**

```
bool CTB_CertStore::DeactivateCert(  
    int ID_CERT  
);
```

Возвращает true, если сертификат был успешно деактивирован и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
----------	----------

ID_CERT	Идентификатор сертификата в хранилище 'CERT_STORE'
---------	--

### **CTB\_CertStore::IsValidityCert**

```
bool CTB_CertStore::IsValidityCert(
    int ID_CERT
);
```

Возвращает true, если сертификат корректен (не вышел срок действия) и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
ID_CERT	Идентификатор сертификата в хранилище 'CERT_STORE'

### **CTB\_CertStore::GetCountMappedCert**

```
int CTB_CertStore::GetCountMappedCert(
    int USER_ID
);
```

Возвращает количество сертификатов, привязанных к пользователю и false – в противном случае.

#### **Параметры функции**

Параметр	Описание
USER_ID	Идентификатор пользователя на сайте (таблица CUser)

### **CTB\_CertStore::GetIDCert**

```
int CTB_CertStore::GetIDCert(
    string x509
);
```

Возвращает идентификатор сертификата, если он присутствует в хранилище и false – в противном случае.

#### Параметры функции

Параметр	Описание
x509	Тело сертификата

### **CTB\_CertStore::GetIDUser**

```
int CTB_CertStore::GetIDUser(  
    string x509  
);
```

Возвращает идентификатор пользователя, к которому привязан сертификат и false – в противном случае.

#### Параметры функции

Параметр	Описание
x509	Тело сертификата

### **CTB\_CertStore::AuthUserOnCert**

```
bool CTB_CertStore::AuthUserOnCert(  
    string x509  
);
```

Выполняет авторизацию пользователя на сайте по предъявленному клиентскому сертификату.

#### Параметры функции

Параметр	Описание
x509	Тело сертификата

## **Класс CTB\_Proxy**

Класс **CTB\_Proxy** предназначен для работы с объектами «прокси-сервер».

### **Методы класса CTB\_Proxy**

Метод	Описание
<a href="#">GetStoreCode</a>	Возвращает код инфоблока для хранения прокси-серверов
<a href="#">GetStoreType</a>	Возвращает тип инфоблока для хранения прокси-серверов
<a href="#">CTB_Proxy</a>	Создает (конструктор) элемент «прокси-сервер»
<a href="#">Delete</a>	Удаляет элемент из базы данных
<a href="#">GetID</a>	Возвращает ID элемента инфоблока
<a href="#">GetIP</a>	Возвращает ip-адрес элемента
<a href="#">GetName</a>	Возвращает наименование элемента
<a href="#">GetProperty</a>	Возвращает значения полей и свойств элемента
<a href="#">isActive</a>	Возвращает статус активности элемента и его временные границы активности
<a href="#">Save</a>	Сохраняет значения полей и свойств элемента в базе данных
<a href="#">SetActive</a>	Устанавливает статус активности элемента
<a href="#">SetActiveTime</a>	Устанавливает временные границы активности элемента
<a href="#">SetID</a>	Устанавливает ID элемента
<a href="#">SetName</a>	Устанавливает наименование элемента
<a href="#">SetProperty</a>	Устанавливает значения полей и свойств элемента

### **CTB\_Proxy::GetStoreCode**

```
string CTB_Proxy::GetStoreCode();
```

Метод класса возвращает код инфоблока для хранения элементов «прокси-сервер».

### **CTB\_Proxy::GetStoreType**

```
string CTB_Proxy::GetStoreType();
```

Метод класса возвращает тип инфоблока для хранения элементов «прокси-сервер».

### **CTB\_Proxy::CTB\_Proxy**

```
CTB_Proxy::CTB_Proxy(  
    string ip_addr="127.0.0.1",  
    string name_proxy=""
```



```
) ;
```

Создает экземпляр класса «прокси-сервер».

#### Параметры метода

Параметр	Описание
ip_addr	ip-адрес прокси-сервера
name_proxy	Наименование прокси-сервера

#### **CTB\_Proxy::Delete**

```
bool CTB_Proxy::Delete() ;
```

Метод экземпляра класса удаляет элемент из инфоблока.

#### **CTB\_Proxy::GetID**

```
string CTB_Proxy::GetID() ;
```

Метод экземпляра класса возвращает ID элемента инфоблока.

#### **CTB\_Proxy::GetIP**

```
string CTB_Proxy::GetIP() ;
```

Метод экземпляра класса возвращает ip-адрес элемента «прокси-сервер».

#### **CTB\_Proxy::GetName**

```
string CTB_Proxy::GetName() ;
```

Метод экземпляра класса возвращает наименование элемента «прокси-сервер».

#### **CTB\_Proxy::GetProperty**

```
array CTB_Proxy::GetProperty() ;
```

Метод экземпляра класса возвращает значения полей и свойств элемента «прокси-сервер». Возвращаемый массив состоит из пар - код (наименование) поля и его значение. Элемент массива с кодом «PROPERTY\_VALUES» содержит аналогичный по структуре массив, но уже со значениями свойств элемента.

### **CTB\_Proxy::IsActive**

```
string CTB_Proxy::IsActive(
    string &start_time,
    string &end_time
);
```

Метод экземпляра класса возвращает статус активности и временные границы активности элемента «прокси-сервер».

#### **Параметры метода**

Параметр	Описание
start_time	Время начала активности
end_time	Время завершения активности

Возвращаемое методом значение – статус активности - принимает значение «Y» или «N».

### **CTB\_Proxy::Save**

```
string CTB_Proxy::Save();
```

Метод экземпляра класса возвращает значение false, если сохранение элемента не произведено, иначе возвращает значение ID элемента инфоблока.

### **CTB\_Proxy::SetActive**

```
CTB_Proxy::SetActive(
    string active_proxy="Y"
);
```

Метод экземпляра класса изменяет статус активности элемента.

#### **Параметры метода**

Параметр	Описание
active_proxy	Статус активности. Если значение = «Y», то элемент активируется, иначе деактивируется.

### **CTB\_Proxy::SetActiveTime**

```
CTB_Proxy::SetActiveTime(  
    string start_time="",  
    string end_time=""  
);
```

Метод экземпляра класса изменяет временной интервал активности элемента.

#### **Параметры метода**

Параметр	Описание
start_time	Время начала активности. Соответствует полю «DATE_ACTIVE_FROM» элемента инфоблока.
end_time	Время окончания активности. Соответствует полю «DATE_ACTIVE_TO» элемента инфоблока.

### **CTB\_Proxy::SetID**

```
CTB_Proxy::SetID(  
    string id  
);
```

Метод экземпляра класса изменяет ID элемента.

#### **Параметры метода**

Параметр	Описание
id	Соответствует полю «ID» элемента инфоблока.

### **CTB\_Proxy::SetName**

```
CTB_Proxy::SetID(  
    string name_proxy  
);
```

Метод экземпляра класса изменяет наименование элемента.

#### **Параметры метода**

Параметр	Описание
name_proxy	Наименование элемента «прокси-сервер». Соответствует полю «NAME» элемента инфоблока.

### **CTB\_Proxy::SetProperty**

```
bool CTB_Proxy::SetProperty(
    array property
);
```

Метод экземпляра класса устанавливает значения полей и свойств элемента.

#### **Параметры метода**

Параметр	Описание
property	Массив устанавливаемых значений полей и свойств.

Передаваемый массив состоит из пар - код (наименование) поля и его значение. Элемент массива с кодом «PROPERTY\_VALUES» содержит аналогичный по структуре массив, но уже со значениями свойств элемента. Метод возвращает значение false, если обнаруживается дублирование элементов по их ip-адресам. В этом случае установка значений не производится.

### **Пример использования методов класса CTB\_Proxy**

Приведем небольшой пример по использованию методов класса:

```
CModule::IncludeModule("trusted.tbstartPRO");
echo "GetStoreType() = ". CTB_proxy::GetStoreType(). "<br>";
echo "GetStoreCode() = ". CTB_proxy::GetStoreCode(). "<br>";
$p=new CTB_proxy("127.0.0.1");
$d_s="";
$d_e="";
$a=$p->isActive ($d_s, $d_e);

echo "<br> id = ".$p->GetID();
echo "<br> ip = ".$p->GetIp();
echo "<br> Name = ".$p->GetName();
echo "<br> active = ".$a;
echo "<br> active from = ".$d_s;
echo "<br> active to = ".$d_e;
echo "<br>";
```

```
echo "<pre>";
print_r($p->GetProperty());
echo "</pre>";
```

Вот пример вывода данного скрипта:

```
GetStoreType() = TrustedBitrix_proxy
GetStoreCode() = PROXY_STORE

id = 201
ip = 127.0.0.1
Name = vme-1cbitrix
active = Y
active from =
active to =
Array
(
    [ID] => 201
    [TIMESTAMP_X] => 27.03.2013 12:51:22
    [TIMESTAMP_X_UNIX] => 1364374282
    [MODIFIED_BY] => 2
    [DATE_CREATE] => 26.03.2013 16:05:11
    [DATE_CREATE_UNIX] => 1364299511
    [CREATED_BY] => 2
    [IBLOCK_ID] => 105
    [IBLOCK_SECTION_ID] =>
    [ACTIVE] => Y
    [ACTIVE_FROM] =>
    [ACTIVE_TO] =>
    [DATE_ACTIVE_FROM] =>
    [DATE_ACTIVE_TO] =>
    [SORT] => 500
    [NAME] => vme-1cbitrix
    [PREVIEW_PICTURE] =>
    [PREVIEW_TEXT] =>
    [PREVIEW_TEXT_TYPE] => text
    [DETAIL_PICTURE] =>
    [DETAIL_TEXT] =>
    [DETAIL_TEXT_TYPE] => text
    [SEARCHABLE_CONTENT] => VME-1CBITRIX

    [WF_STATUS_ID] => 1
    [WF_PARENT_ELEMENT_ID] =>
    [WF_LAST_HISTORY_ID] =>
    [WF_NEW] =>
    [LOCK_STATUS] => green
```

```

[WF_LOCKED_BY] =>
[WF_DATE_LOCK] =>
[WF_COMMENTS] =>
[IN_SECTIONS] => N
[SHOW_COUNTER] =>
[SHOW_COUNTER_START] =>
[CODE] =>
[TAGS] =>
[XML_ID] => 201
[EXTERNAL_ID] => 201
[TMP_ID] =>
[USER_NAME] => (vib) Иван Петров
[LOCKED_USER_NAME] =>
[CREATED_USER_NAME] => (vib) Иван Петров
[LANG_DIR] => /s2/
[LID] => s2
[IBLOCK_TYPE_ID] => TrustedBitrix_proxy
[IBLOCK_CODE] => PROXY_STORE
[IBLOCK_NAME] => PROXY_STORE
[IBLOCK_EXTERNAL_ID] =>
[DETAIL_PAGE_URL] =>
[LIST_PAGE_URL] =>
[CREATED_DATE] => 2013.03.26
[BP_PUBLISHED] => Y
[PROPERTY_VALUES] => Array
    (
        [IP] => 127.0.0.1
    )
)

```