



**ЦИТАДЕЛЬ**

ООО «Цитадель»  
127015, г. Москва, ул. Новодмитровская, д. 2Б  
+7 (495) 666 2 333, info@ctdl.ru

**ПО ТС ОРМ «ОЛИМП-SDM»**  
*Руководство по установке и настройке*

## Содержание

<b>1</b>	<b>Общая информация о ПО ТС ОРМ «ОЛИМП-SDM»</b> .....	<b>3</b>
1.1	Область применения .....	3
1.2	Используемые протоколы .....	4
1.3	Обеспечение требований КПП .....	4
<b>2</b>	<b>Программное обеспечение ТС ОРМ «ОЛИМП-SDM»</b> .....	<b>7</b>
2.1	Компонент sorm_sdm .....	8
2.2	Компонент Installer .....	8
2.2.1	Назначение и область применения .....	8
2.2.2	Описание работы с компонентом. Запуск .....	8
2.2.3	Описание аргументов .....	8
2.2.4	Примеры инсталляции (установка, обновление, удаление).....	9
2.2.5	Структура инсталляции .....	9
2.2.6	Примеры управления модулем .....	10
2.2.7	Установка автозапуска .....	10
<b>3</b>	<b>Установка программных компонентов</b> .....	<b>11</b>
3.1	Установка ПО ТС ОРМ «ОЛИМП-SDM» .....	11
<b>4</b>	<b>Настройка ПО ТС ОРМ «ОЛИМП-SDM» и описание конфигурационных файлов</b> ....	<b>12</b>
4.1	Настройка компонента sorm_sdm .....	12
<b>5</b>	<b>Работа с ПО ТС ОРМ «ОЛИМП-SDM»</b> .....	<b>13</b>
5.1	Скрипт <имя_скрипта>.sh .....	13

# 1 Общая информация о ПО ТС ОРМ «ОЛИМП-SDM»

Программное обеспечение ТС ОРМ «ОЛИМП-SDM» (далее – ПО «ОЛИМП-SDM») в составе программно-аппаратного комплекса «ОЛИМП-SDM» (далее – ПАК «ОЛИМП-SDM») предназначено для сопряжения интерфейсов X1, X2 LI-интерфейса одного или нескольких SDM Huawei и интерфейсов COPM коммутационного оборудования с одним или несколькими ПУ ОРМ в соответствии с требованиями Приказа Минкомсвязи РФ №645 от 12.12.2016 (далее – Приказ №645).

К особенностям ПО ТС ОРМ «ОЛИМП-SDM» относятся следующие пункты:

- постановка объекта на контроль по MSISDN, IMSI;
- MSISDN/IMSI может быть поставлен на контроль на SDM Huawei только один раз (при этом можно одновременно поставить на контроль MSISDN и IMSI, принадлежащие одному абоненту). Таким образом на ПО ТС ОРМ «ОЛИМП-SDM» реализована функция многократной постановки на контроль одного MSISDN/IMSI.

## 1.1 Область применения

ПО ТС ОРМ «ОЛИМП-SDM» предназначено для согласования протокола ETSI LI, передаваемого от SDM производства Huawei, с протоколом передачи информации к одному или нескольким ТС ОРМ, обеспечивающим реализацию требований Приказа Минкомсвязи РФ №645 от 12.12.2016 с целью передачи информации об объектах контроля из домашнего регистра местоположения.

Общая схема работы ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM» приведена на рисунке 1.

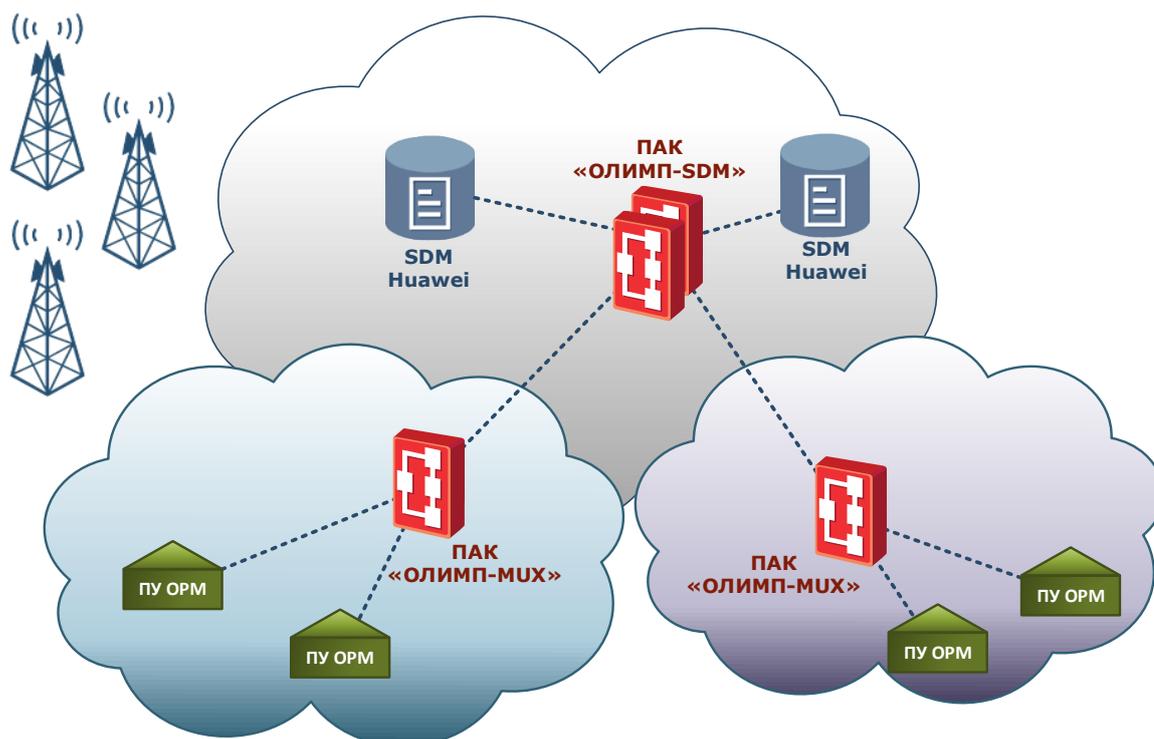


Рисунок 1. Общая схема работы ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM»

С помощью ПО ТС ОРМ «ОЛИМП-SDM» возможно организовать комплексное решение для передачи данных от SDM в схеме с глобальным резервированием (см. рисунок 2).

При этом ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM», непосредственно взаимодействующее с ТС ОРМ (в частности, ПАК «ОЛИМП-MUX»), помимо обеспечения постановки объектов на контроль на SDM производства Huawei в регионе присутствия, выполняет маршрутизацию команд от ТС ОРМ (ПУ ОРМ) на ПАК «ОЛИМП-SDM» в глобальном резерве и передаёт данные о контролируемых событиях с SDM из глобального резерва на региональный ПАК «ОЛИМП-SDM».

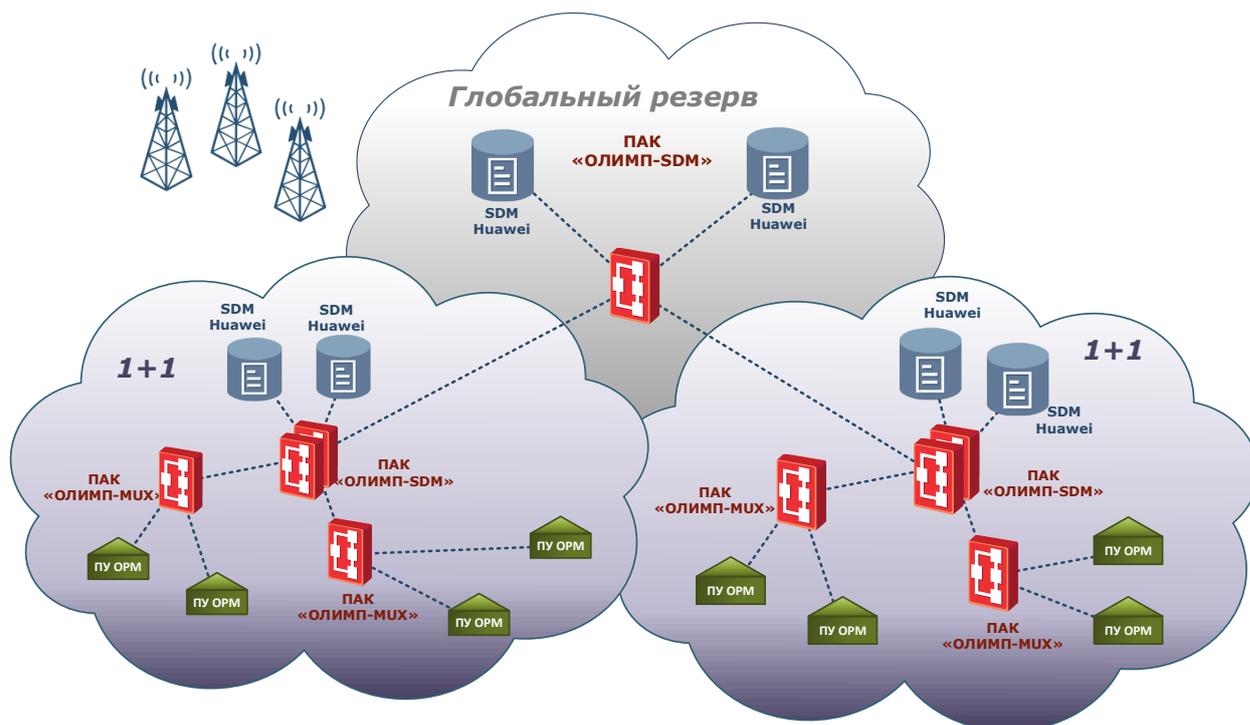


Рисунок 2. Общая схема работы ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM» в режиме суммирования интерфейсов COPM HLR

## 1.2 Используемые протоколы

ПО ТС ОРМ «ОЛИМП-SDM» и внешние системы взаимодействуют по протоколам, приведенным в таблице 1.

Таблица 1. Взаимодействие между ПО ТС ОРМ «ОЛИМП-SDM» и внешними системами

Элемент	Интерфейс	Протокол
ПУ ОРМ	КПД1, КПД2 (без КСЛ)	Модифицированный протокол Приказа №645
SDM	ETSI LI X1 и X2	ETSI LI Huawei

## 1.3 Обеспечение требований КПН

ПО ТС ОРМ «ОЛИМП-SDM» может обеспечивать требования комплекса повышенной надежности (далее - КПН). Для этого в состав ПАК «ОЛИМП-SDM» входит кластер OLIMP-SDM, состоящий из четырех серверов. Пример кластера приведен на рисунке 3.

Кластер OLIMP-SDM объединяет две физические серверные платформы OLIMP-SDM1 и OLIMP-SDM2. В каждой серверной платформе запущены по два аппаратно-независимых сервера – OLIMP-SDM1A, OLIMP-SDM1B, OLIMP-SDM2A, OLIMP-SDM2B.

При обеспечении требований КРН ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM» может принимать следующие роли:

- MASTER: реализует функционирование бизнес-логики, отправляет изменения в своей таблице объектов на SLAVE. Отсылает Alive-пакеты на SLAVE.
- EXMASTER: это промежуточное состояние, возникающее после потери сервером роли MASTER. При переходе из ExMaster в SLAVE не происходит запрос снимка таблицы объектов у MASTER (в отличие от перехода UNKNOWN → SLAVE). В этой роли ПО ТС ОРМ «ОЛИМП-SDM» ожидает получения Alive-пакетов (таймаут TimeoutBeSlave, секция [TIMERS]) или данных по X2 (таймаут anydata, секция [TIMEOUTS]), если этого не произойдет за определенный интервал времени, то произойдет переход в UNKNOWN.
- SLAVE: сохраняет полученные от MASTER изменения в таблице объектов контроля. Получает Alive-пакеты. Находится в ожидании переключения в состояние MASTER.
- UNKNOWN: ожидает получения Alive-пакетов или данных по X2 для принятия другой роли.

---

*Примечание:* Таблицу объектов контроля ПО ТС ОРМ «ОЛИМП-SDM» хранит в оперативной памяти, без записи в системные журналы, файлы и т.п.

---

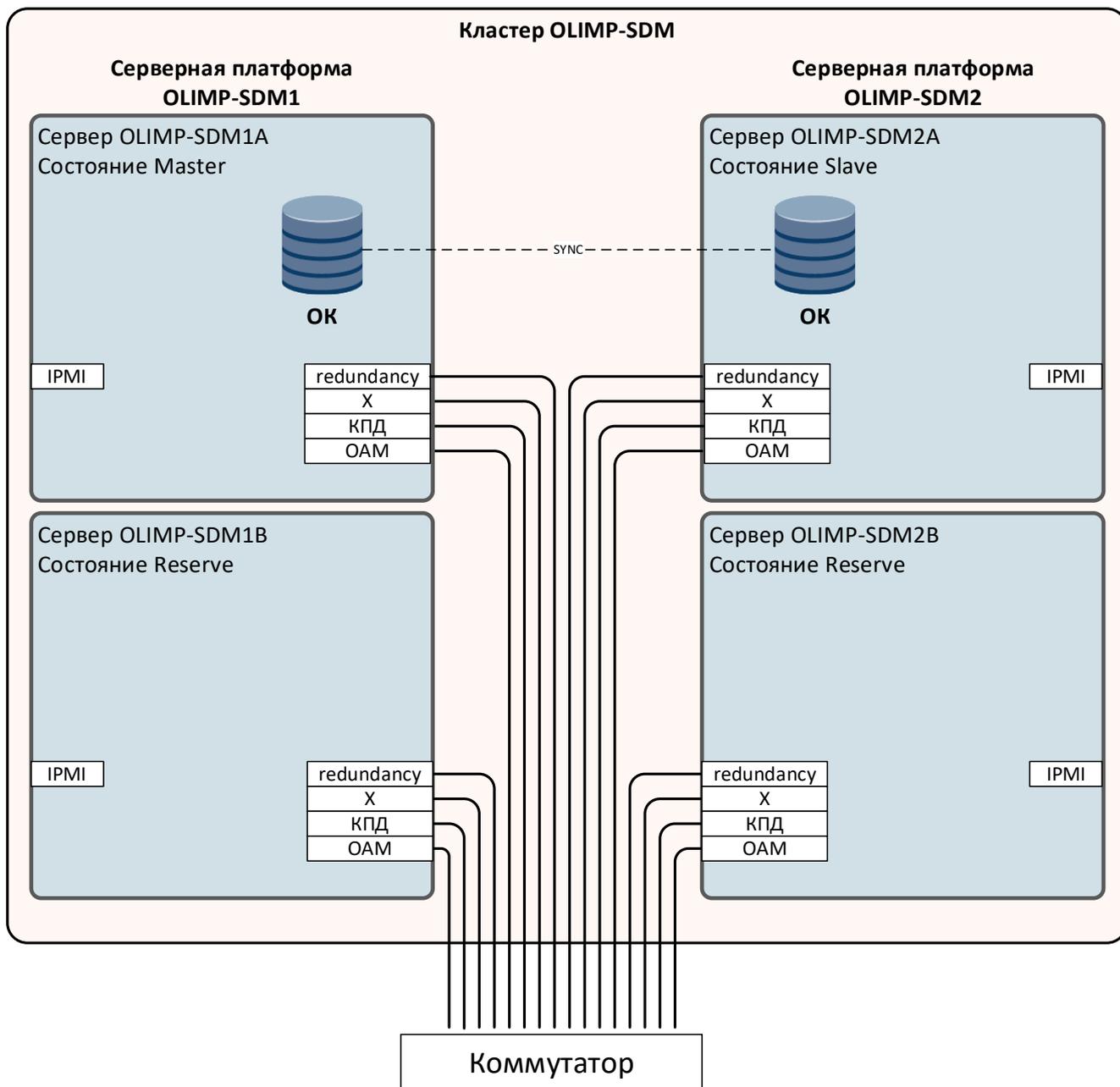


Рисунок 3. Схема работы ПО ТС ОРМ «ОЛИМП-SDM» в составе кластера из четырех серверов

## 2 Программное обеспечение ТС ОРМ «ОЛИМП-SDM»

ПО ТС ОРМ «ОЛИМП-SDM» включает в себя следующие программные компоненты:

- **sorm\_sdm**;
- **installer**.

Структурная схема и взаимодействие ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM» с внешними системами представлена на рисунке 4.

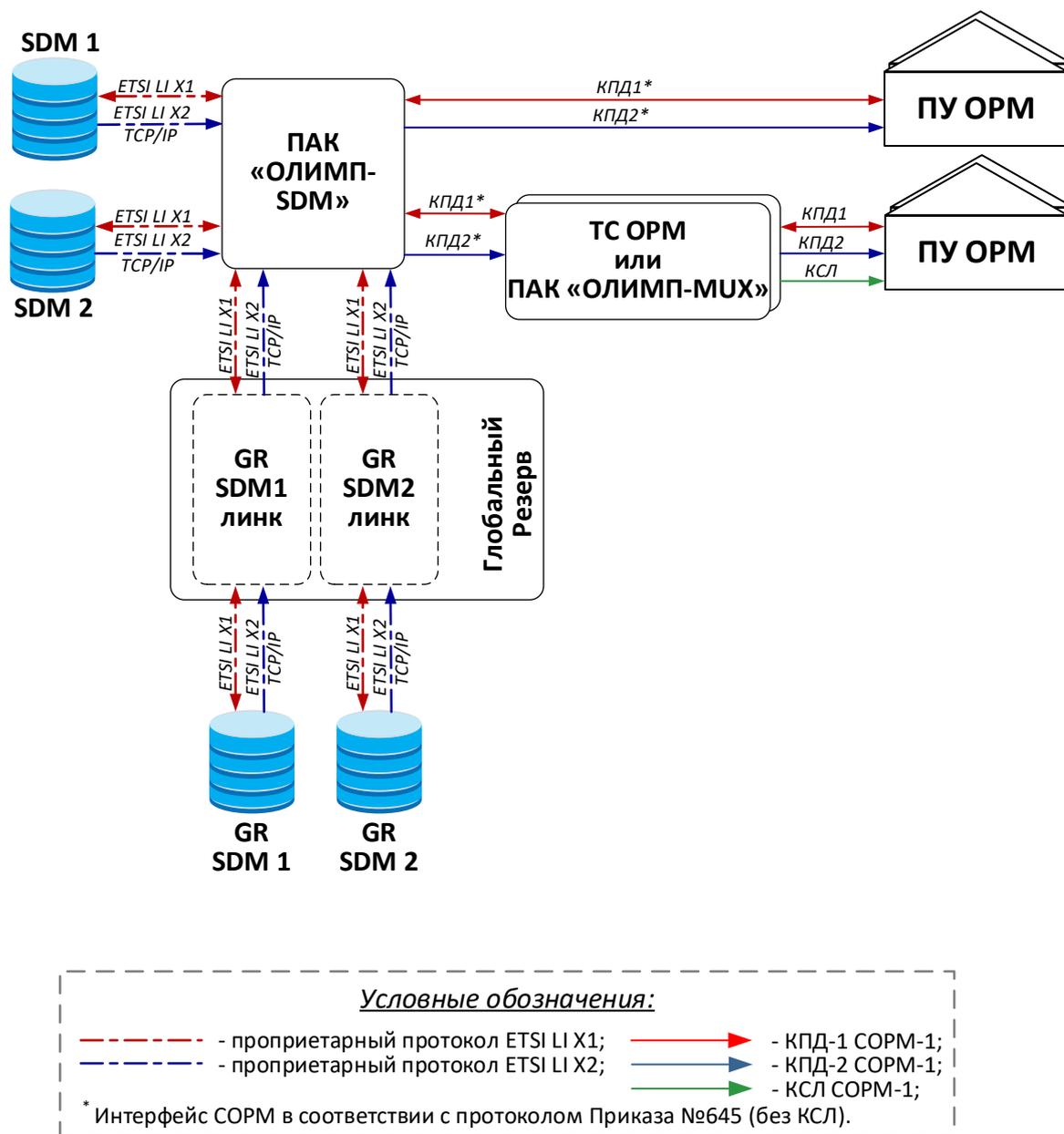


Рисунок 4. Структурная схема и взаимодействие ПО ТС ОРМ «ОЛИМП-SDM» в составе ПАК «ОЛИМП-SDM» с внешними системами

## 2.1 Компонент `sorm_sdm`

Компонент `sorm_sdm` выполняет следующие функции:

- сопряжение одного или нескольких LI-интерфейсов SDM Huawei с одним или несколькими интерфейсами ПУ ОРМ в соответствии с требованиями Приказа №645;
- контроль доступа на сервер с ПО ТС ОРМ «ОЛИМП-SDM» по SSH или доступа напрямую;
- взаимодействие с «Подсистемой защиты от НСД» (не входит в состав ПО ТС ОРМ «ОЛИМП-SDM») для получения информации о целостности защитного контура ПАК «ОЛИМП-SDM».

## 2.2 Компонент `Installer`

### 2.2.1 Назначение и область применения

Компонент `Installer` – это установщик компонентов. Основное преимущество: возможность установки N независимых инстансов на сервер.

### 2.2.2 Описание работы с компонентом. Запуск

---

**Внимание!** Название инстансов и компонентов является примером и не является обязательным. При доработке компонента `installer` в будущем это будет происходить автоматически.

---

Запуск осуществляется исполняемым файлом, находящимся в каталоге установщика `./installer`, и передачей ему аргументов.

### 2.2.3 Описание аргументов

#### Аргумент `help/h`

Используется для вывода справки. Пример использования:

```
./installer -h
./installer --help
```

Также возможно использование после аргумента `in/un/up`.

#### Аргумент `install/in`

Используется для установки компонента и всех зависимостей в системе. Пример использования:

```
./installer in
./installer install
```

#### Аргумент `uninstall/un`

Используется для удаления компонента и всех зависимостей в системе.

```
./installer un  
./installer uninstall
```

### Аргумент `update/up`

Используется для обновления компонента и всех зависимостей в системе.

```
./installer up  
./installer update
```

## 2.2.4 Примеры инсталляции (установка, обновление, удаление)

Установка:

```
./installer in /home/sorm_sdm-x.x-x.x.zip /olimp/ sdm_install  
./installer install /home/sdm-x.x-x.x.zip /olimp/ sdm_install
```

Удаление:

```
./installer un /olimp/ sdm_install  
./installer uninstall /olimp/ sdm_install
```

Обновление:

```
./installer up /home/sorm_sdm-x.x-x.x.zip /olimp/ sdm_install  
./installer update /home/sorm_sdm-x.x-x.x.zip /olimp/ sdm_install
```

## 2.2.5 Структура инсталляции

Все компоненты **Installer** по умолчанию расположены в каталоге **/home**. Рекомендуется создавать директории по следующему правилу:

```
/  
|- home  
  |- product_name  
    |- module_name  
      |- instance_name_1  
        |- bin  
        |- cfg  
        |- lib  
        |- temp  
      ...  
      |- instance_name_N  
        |- bin  
        |- cfg  
        |- lib  
        |- temp
```

Директории **bin/cfg/lib/temp** создаются инсталлятором автоматически.

Пример иерархического списка с описанием подкаталогов и файлов приведен ниже:

```
/
|- olymp
  |- sdm_install
    |- bin
    |- cfg
    |- log
    |- temp
  ...
```

## 2.2.6 Примеры управления модулем

Запуск модуля осуществляется командой **start**, пример ниже:

```
systemctl start sdm_install.service
```

Остановка модуля осуществляется командой **stop**, пример ниже:

```
systemctl stop sdm_install.service
```

Перезапуск модуля осуществляется командой **stop/start**, пример ниже:

```
systemctl stop sdm_install.service
systemctl start sdm_install.service
```

Запрос версии осуществляется командой **version**, пример ниже:

```
sdm_install version
```

Запрос статуса осуществляется командой **status**, пример ниже:

```
sdm_install status
```

## 2.2.7 Установка автозапуска

Указание опции автозапуска и удаление программы из списка автозапуска осуществляется с помощью утилиты **systemctl**:

```
# Добавить экземпляр список автозапуска
sudo systemctl enable <имя_экземпляра>.service

# Удалить экземпляр из списка автозапуска
sudo systemctl disable <имя_экземпляра>.service

# Проверить статус автозапуска экземпляра
sudo systemctl is-enabled <имя_экземпляра>.service

# Просмотреть список всех имеющихся сервисов на машине
sudo systemctl list-unit-files
```

## 3 Установка программных компонентов

### 3.1 Установка ПО ТС ОРМ «ОЛИМП-SDM»

Дистрибутив представляет собой папку вида **X.XX**, где **X.XX** - номер релиза, содержащую архивы дистрибутивов для их установки или обновления.

Папка с дистрибутивами копируется на сервер. Сначала распаковывается дистрибутив компонента **installer** в заранее созданный каталог.

Пример команды распаковки:

```
tar xaf installer-x.x-xx.xxxxx.tar.gz
```

Далее производится установка программных компонентов комплекса, примеры описаны выше в пункте 2.2.4 описания компонента **installer**.

Структура инсталляции и примеры управления модулями так же описаны выше в пунктах 2.2.5 и 2.2.6 соответственно.

## 4 Настройка ПО ТС ОРМ «ОЛИМП-SDM» и описание конфигурационных файлов

### 4.1 Настройка компонента `sorm_sdm`

Основные параметры настройки компонента `sorm_sdm` содержатся в файле `config.example.cfg`. Допустимо переименовать конфигурационный файл.

## 5 Работа с ПО ТС ОРМ «ОЛИМП-SDM»

Для работы с ПО ТС ОРМ «ОЛИМП-SDM» используются три инструмента:

- 1) Управление модулем через **systemd** после установки с помощью модуля **installer**.  
Примеры описаны выше в подразделе 2.2.6 описания компонента **installer**.
- 2) скрипт `<имя_скрипта>.sh` ( см. раздел 5.1);

### 5.1 Скрипт `<имя_скрипта>.sh`

В архиве компонента **installer** идёт скрипт-алиас для сервисов **generate\_alias.sh**.

При запуске скрипт выводит список установленных ранее на сервер сервисов и предлагает выбрать N сервисов, далее для N сервисов создаётся скрипт `<имя_скрипта>.sh` управляющий через system всеми вложенными сервисами.

Пример:

- 1) Запуск скрипта:

```
# ./generate_alias.sh
Installed services:
0: sorm_sdm
Enter services' numbers you wish to control in start order (space separated):
```

- 2) Выбор номеров сервисов в том порядке, в каком они будут запускаться (через пробел):

```
Enter services' numbers you wish to control in start order (space separated): 0
```

- 3) Выбор номеров сервисов в том порядке, в каком они будут останавливаться (через пробел):

```
Enter services' numbers you wish to control in stop order (space separated): 0
```

- 4) Пишем название скрипта, после чего появляется скрипт `<имя_скрипта>.sh` для управления ранее установленными сервисами:

```
Enter name for alias script: example
# ls
example.sh
```

Скрипт `<имя_скрипта>.sh` в ПО ТС ОРМ «ОЛИМП-SDM» возможно использовать для следующих компонентов:

- **sorm\_sdm**.

С помощью скрипта поддерживаются следующие команды работы перечисленных выше компонентов:

**{start|stop|restart|version|status}**

- Запуск

Для запуска всех модулей с помощью скрипта необходимо выполнить команду:

```
# ./example.sh start
```

- Остановка

Для остановки всех модулей с помощью скрипта необходимо выполнить команду:

```
# ./example.sh stop
```

- Перезапуск

Для перезапуска всех модулей с помощью скрипта необходимо выполнить команду:

```
# ./example.sh restart
```

- Проверка версии

Для вывода версий всех модулей с помощью скрипта необходимо выполнить команду:

```
# ./example.sh version
```

- Проверка статуса

Для вывода статуса работы всех модулей с помощью скрипта необходимо выполнить команду:

```
# ./example.sh status
```