Zabbix. List of installed programs.

We will use Zabbix to keep track of the list of installed programs.

1. We collect data on installed programs using a script.

The script is written in Python 2.7, it collects data on installed programs and generates 2 files:

- directly the list of installed programs – **C:\Zabbix\scripts\soft_list\soft_list.lst**
- a list of changes in installed programs (what is removed / installed) – **C:\Zabbix\scripts\soft_list\soft_diff.lst**

The script is saved under the name **C:\Zabbix\scripts\soft_list\programmlist.py**.

Script text:

```
# -*- coding: utf-8 -*-
import os
import sys
import ctypes
import errno
import _winreg
import codecs

#
# Функция проверки наличия прав администратора
# Вход: нет
# Выход: true - есть права администратора,
#        false - нет прав администратора
#
def is_admin():
    try:
        return ctypes.windll.shell32.IsUserAnAdmin()
    except:
        return False
#
# Функция удаления дубликатов из списка
# Вход: список
# Выход: список без дубликатов
#
def DuplicateRemoval ( InboundList ):
    OutboundList = []
    for i in InboundList:
        if i not in OutboundList:
            OutboundList.append(i)
    return OutboundList

#
# Функция загрузки списка из файла
# Вход: путь к файлу со списком
# Выход: список
#
def ListLoad ( InboundFileName ):
    OutboundList = []

    # Если указанного файла не существует возвращаем пустой список
    if not os.path.exists( InboundFileName ):
```

```
        return OutboundList

    if not os.path.isfile( InboundFileName ):
        return OutboundList

    # Читаем файл построчно
    f = open( InboundFileName,'r')
    try:
        OutboundList = f.read().splitlines()
    except Exception:
        pass
    finally:
        f.close()

    return OutboundList

#
# Функция сохранения списка в файл
# Вход: список, путь к файлу
# Выход: true - успешная запись, false - ошибка при записи
#
def ListSave ( OutboundList, OutboundFileName ):

    # Записываем файл построчно
   f = open( OutboundFileName,'w')
   try:
     for line in OutboundList:
           f.write(line + '\n')
   except Exception:
       MyResult = False
   else:
       MyResult = True
   finally:
      f.close()

   return MyResult

#
# Функция получения списка установленных программ
# Вход: нет
# Выход: список установленных программ
#
def ProgramList():

    # Определим разрядность системы
    try:
        os.environ["PROGRAMFILES(X86)"]
        proc_arch = 64
    except:
        proc_arch = 32

    # Зададим режимы просмотра реестра
    if proc_arch == 32:
        arch_keys = []
    elif proc_arch == 64:
        arch_keys = [ _winreg.KEY_WOW64_32KEY, _winreg.KEY_WOW64_64KEY ]

    # Зададим ветки реестра, где нужно искать информацию об установленных
программах
    keys = [ r'hklm\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall',

r'hklm\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall',
```

```
            r'hkcu\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall']

    List = []

    for arch_key in arch_keys:
        for MyKey in keys:
            # ветки hkcu\...\Uninstall может не быть. Предусмотрим
вероятность срабатывания исключения
            try:
               if 'hkcu' not in MyKey:
                  key = _winreg.OpenKey(_winreg.HKEY_LOCAL_MACHINE,
MyKey[5:], 0, _winreg.KEY_READ | arch_key)
               else:
                  key = _winreg.OpenKey(_winreg.HKEY_CURRENT_USER, MyKey[5:],
0, _winreg.KEY_READ | arch_key)
            except Exception:
               pass
            else:
               for i in xrange(0, _winreg.QueryInfoKey(key)[0]):
                   skey_name = _winreg.EnumKey(key, i)
                   skey = _winreg.OpenKey(key, skey_name)
                   try:
                       List.append ( _winreg.QueryValueEx(skey,
'DisplayName')[0] )
                   except OSError as e:
                       if e.errno == errno.ENOENT:
                           # DisplayName не существует в этом разделе
                           pass
                   finally:
                       skey.Close()

    List.sort()
    NewList = DuplicateRemoval (List)
    return NewList

#############################################################################
###
#   Начало программы
#############################################################################
###
def main(argv=None):
   # Проверяем наличие прав админа.
   if not is_admin ():
      sys.exit ('Not enough permissions to run the script !!!')

   # Устанавливаем кодировку по умолчанию.
   reload(sys)
   sys.setdefaultencoding('utf8')

   # Зададим имена файлам, которые будем использовать
   MySoftListFile=r'C:\Zabbix\scripts\soft_list\soft_list.lst'
   MySoftDiffFile=r'C:\Zabbix\scripts\soft_list\soft_diff.lst'

   # Получаем список установленных программ
   List = ProgramList()

   # Получаем прежний список из файла
   OldList = ListLoad ( MySoftListFile )

   # Ищем установленные/удалённые программы
   DeletedList = []
   InstalledList = []
```

```
    # Получаем список свежеустановленных программ
    for i in List:
        if i not in OldList:
             InstalledList.append(i)

    # Получаем список удалённых программ
    for i in OldList:
        if i not in List:
             DeletedList.append(i)

    # Сохраняем список изменений в diff-файл построчно
    f = open( MySoftDiffFile,'w')
    try:
       f.write('Deleted:'+ '\n')
       for line in DeletedList:
          f.write(line + '\n')
       f.write('\n')
       f.write('Installed:'+ '\n')
       for line in InstalledList:
          f.write(line + '\n')
    except Exception:
       pass
    finally:
       f.close()

    # Сохраняем текущий список установленных программ в файл
    if not ListSave (List, MySoftListFile):
       sys.exit ('Unable to save installed program list!!!')

if __name__ == "__main__":
    sys.exit(main())
```

2. Install the script on the system.

It is better to run the script every hour - not too often, but not too rarely.

You can schedule script execution using the Windows Scheduler. The scheduler task can be created manually, or using a bat file.

I usually do this with the help of a "batch file", I gave the name to my batch file **C:\Zabbix\scripts\soft_list\INSTALL_get_soft_list.bat** .

Script:

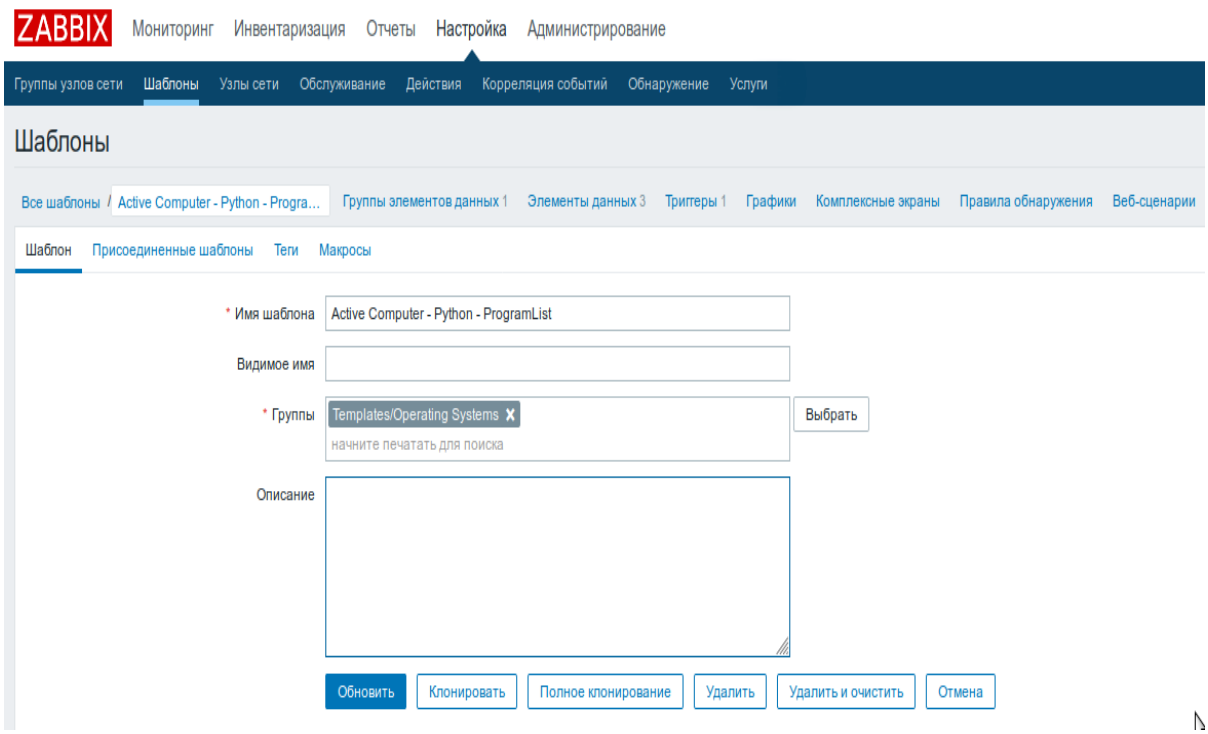
```
@echo off

Rem Предполагаем, что на Windows XP скрипт запускается администратором.
Rem Для более старших систем это неверно.

rem Получаем версию ОС
ver | find "5.1."

rem Windows XP ?
If %errorlevel%==0  (
```

```
        rem Пропускаем проверку админских прав
        GOTO SKIPADMIN
 )

SET HasAdminRights=0

FOR /F %%i IN ('WHOAMI /PRIV /NH') DO (
        IF "%%i"=="SeTakeOwnershipPrivilege" SET HasAdminRights=1
)

IF NOT %HasAdminRights%==1 (
        ECHO .
        ECHO Not enough permissions to run the script !!!
        ECHO .
        GOTO END
)

:SKIPADMIN

rem Получаем версию ОС
rem Windows XP ?

ver | find "5.1."

If %errorlevel%==0  (
        rem Windows XP
        SCHTASKS /Create /RU "NT AUTHORITY\SYSTEM" /SC HOURLY /ST 00:00:00 /TN
"InstalledSoftware" /TR "python
\"C:\zabbix\scripts\soft_list\programmlist.py\""
 ) else (
        rem HE Windows XP
        SCHTASKS /Create /RU "NT AUTHORITY\SYSTEM" /SC DAILY /ST 00:00 /RI 60
/DU 24:00 /TN "InstalledSoftware" /TR "python
\"C:\zabbix\scripts\soft_list\programmlist.py\"" /RL HIGHEST /F
 )

cd "C:\Zabbix\scripts\soft_list"

python "C:\zabbix\scripts\soft_list\programmlist.py"

:END

EXIT /B
```

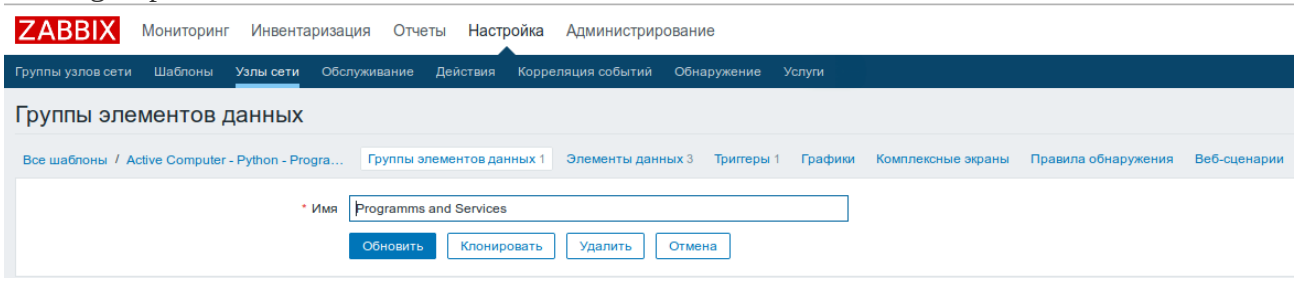
3. Forming a template in Zabbix.

At the same time, we add a simple template to Zabbix.

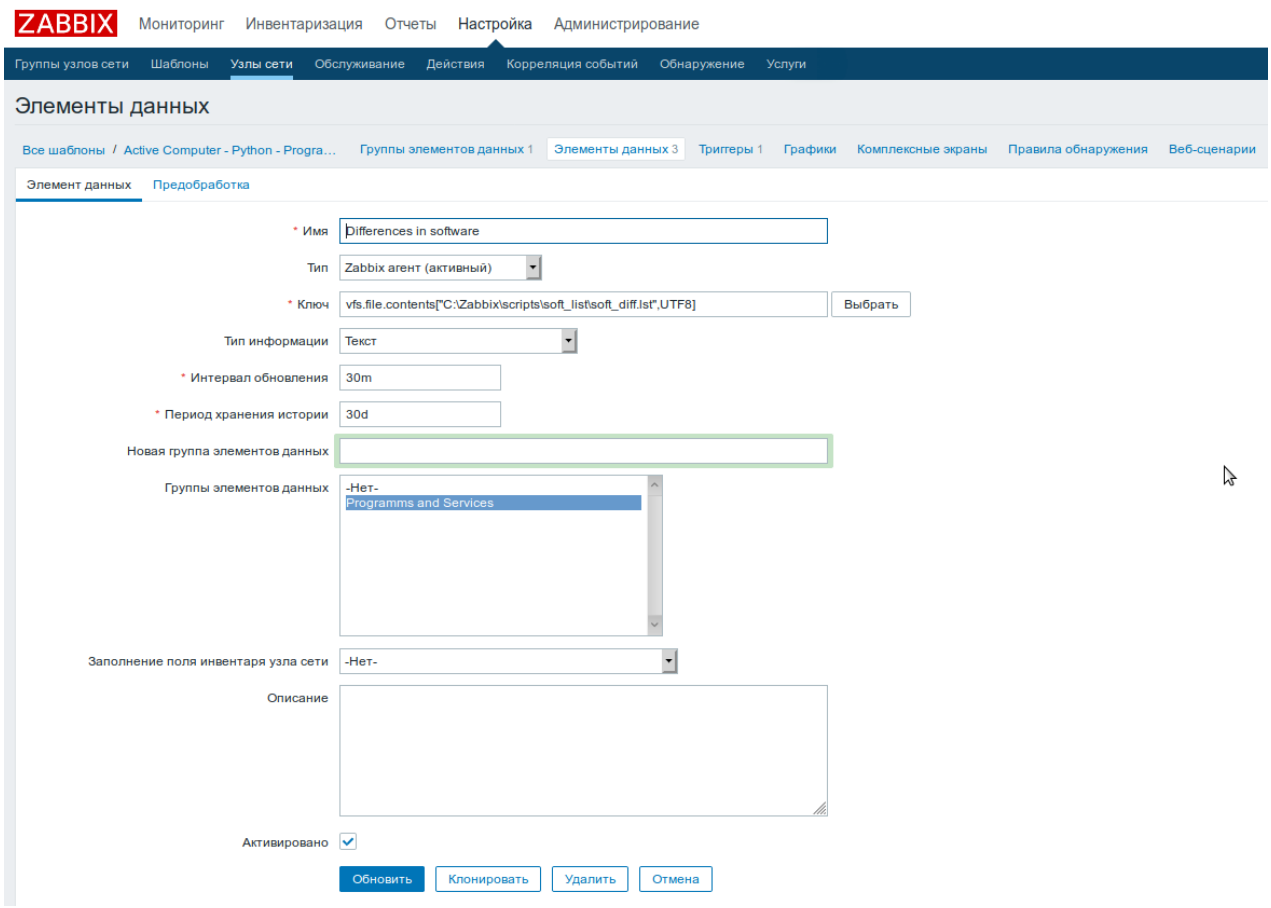
Template name: **Active Computer – Python – ProgramList**…

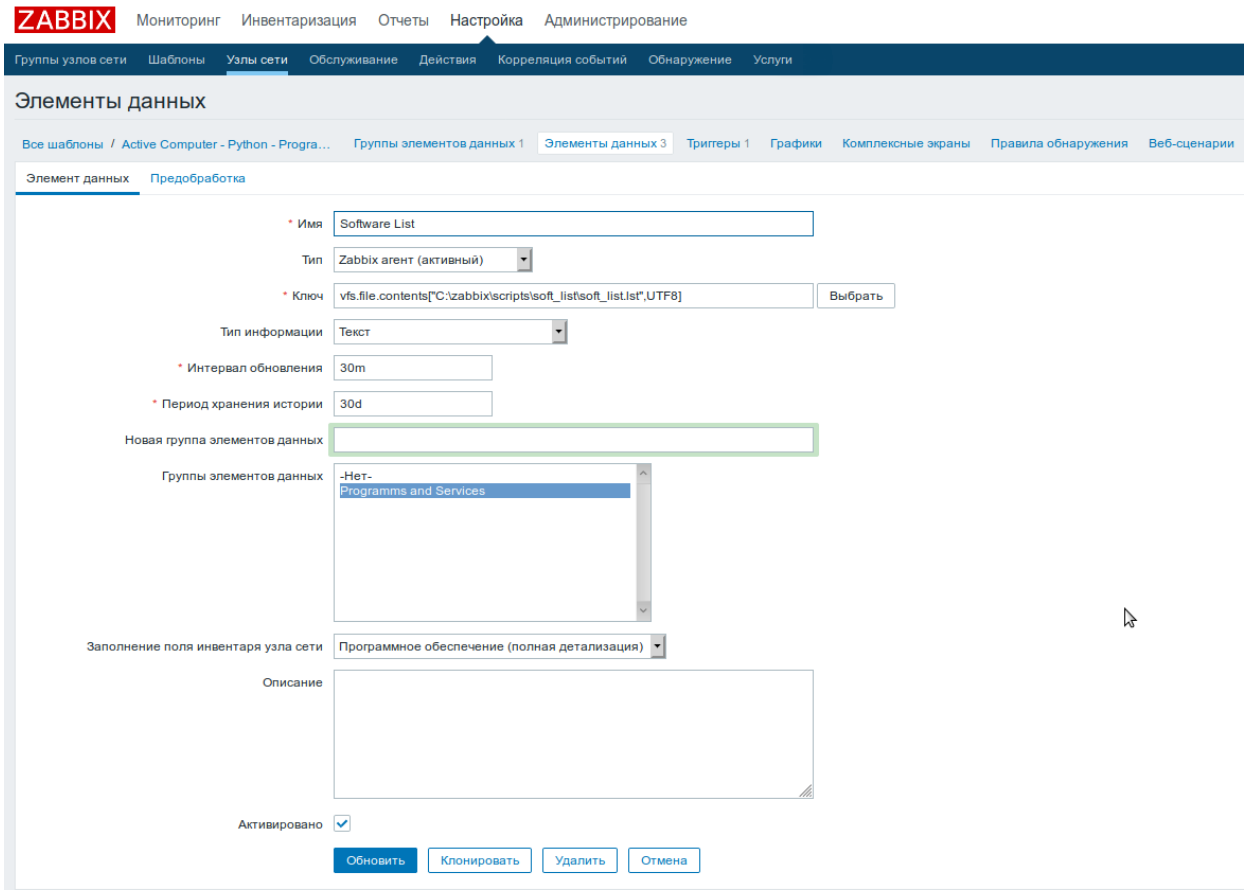


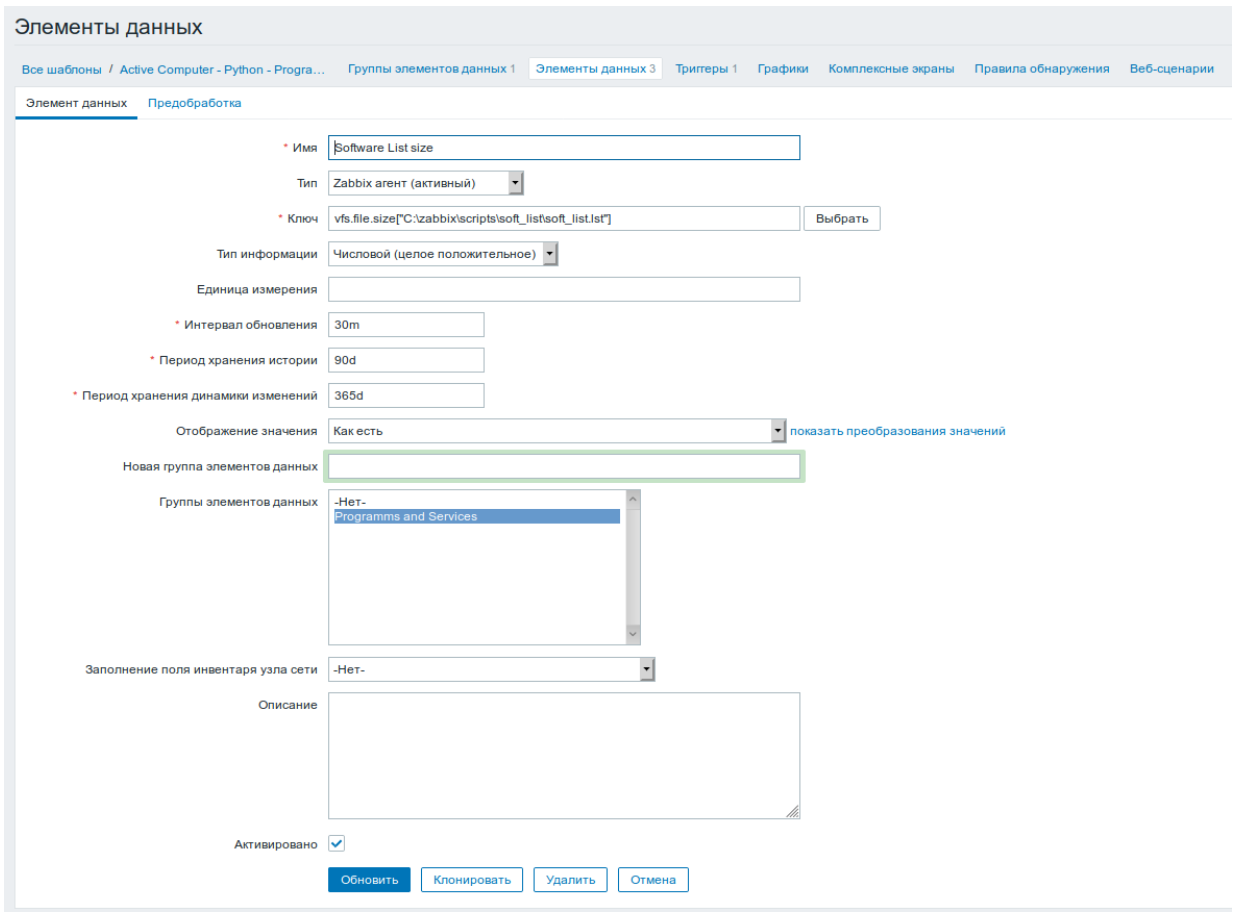
Add a group of data items …



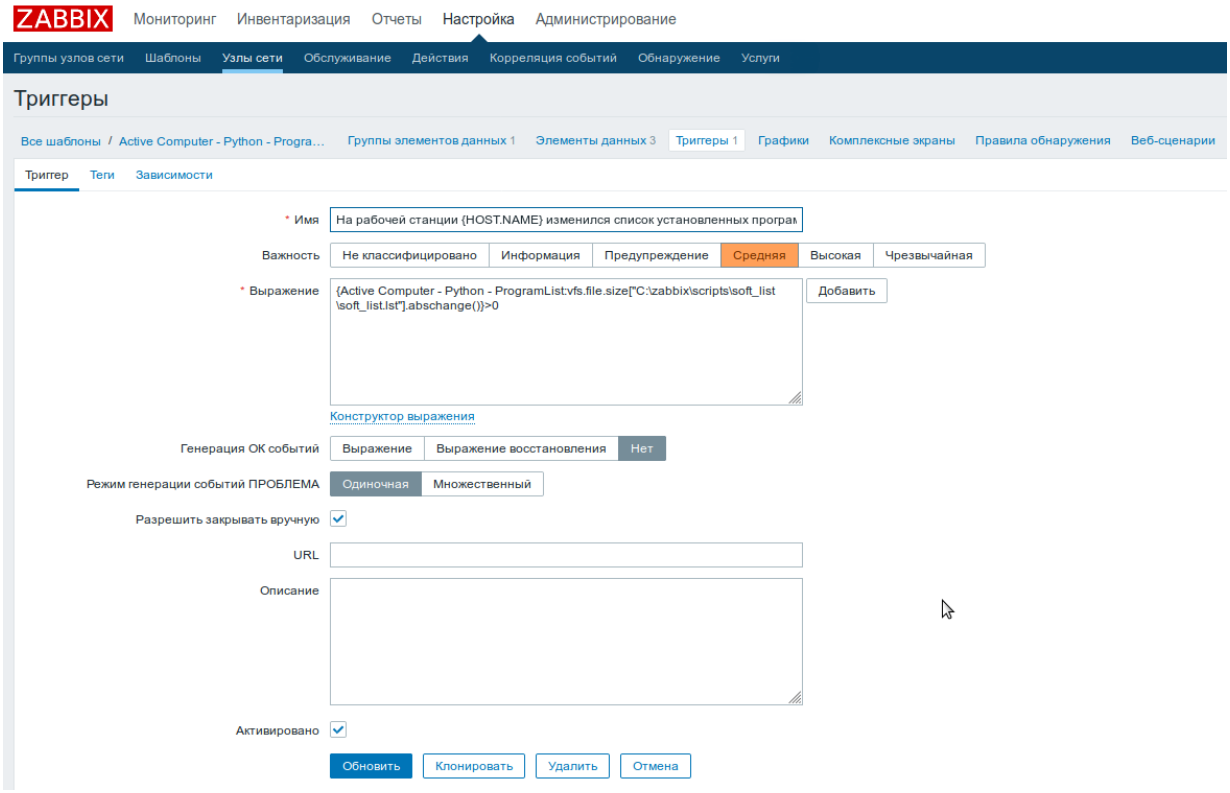
Now add 3 data items …




ZABBIX
Мониторинг
Инвентаризация
Отчеты
Настройка
Администрирование
Группы узлов сети
Шаблоны
Узлы сети
Обслуживание
Действия
Корреляция событий
Обнаружение
Услуги
Элементы данных
Все шаблоны / Active Computer - Python - Progra...
Группы элементов данных 1
Элементы данных 3
Триггеры 1
Графики
Комплексные экраны
Правила обнаружения
Веб-сценарии
Элемент данных
Предобработка
* Имя
Software List
Тип
Zabbix агент (активный)
* Ключ
vfs.file.contents["C:\zabbix\scripts\soft_list\soft_list.lst",UTF8]
Выбрать
Тип информации
Текст
* Интервал обновления
30m
* Период хранения истории
30d
Новая группа элементов данных
Группы элементов данных
-Нет-
Programms and Services
Заполнение поля инвентаря узла сети
Программное обеспечение (полная детализация)
Описание
Активировано
Обновить
Клонировать
Удалить
Отмена


Элементы данных
Все шаблоны / Active Computer - Python - Progra...
Группы элементов данных 1
Элементы данных 3
Триггеры 1
Графики
Комплексные экраны
Правила обнаружения
Веб-сценарии
Элемент данных
Предобработка
* Имя
Software List size
Тип
Zabbix агент (активный)
* Ключ
vfs.file.size["C:\zabbix\scripts\soft_list\soft_list.lst"]
Выбрать
Тип информации
Числовой (целое положительное)
Единица измерения
* Интервал обновления
30m
* Период хранения истории
90d
* Период хранения динамики изменений
365d
Отображение значения
Как есть
показать преобразования значений
Новая группа элементов данных
Группы элементов данных
-Нет-
Programms and Services
Заполнение поля инвентаря узла сети
-Нет-
Описание
Активировано
Обновить
Клонировать
Удалить
Отмена

And finally, a simple trigger that will fire when the list of programs changes …



And here is the zipped file ready for import with the template described above: zbx_export_template_programlist.xml

The scripts described earlier can be found in this archive: soft_list

4. Data collection

It remains to assign the created template to the corresponding host and wait for the start of data arrival. ☺