

# **Notification Processor 0.24.0**

## **Dokumentacja techniczna**

**Katarzyna Władyszewska, Hadden Sp.J.**

---

## **Notification Processor 0.24.0: Dokumentacja techniczna**

by Katarzyna Władyszewska

Data wydania Kwiecień 2010

Copyright © 2003-2010 Hadden Sp.J.

HADDEN MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

All rights reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hadden Sp.J..

All trademarks included in this document are the property of their respective owners.

FIRMA HADDEN NIE PONOSI ŻADNEJ ODPOWIEDZIALNOŚCI ZA SKUTKI WYNIKAJĄCE Z UŻYWANIA NINIEJSZEJ DOKUMENTACJI.

Wszystkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszego dokumentu w jakiegokolwiek postaci jest zabronione.

Wszystkie nazwy handlowe i towarów występujące w niniejszej dokumentacji są znakami towarowymi zastrzeżonymi lub nazwami zastrzeżonymi odpowiednich firm odnośnych właścicieli.

---

---

---

---

## Spis treści

1. Konwencje typograficzne .....	1
2. Informacje ogólne o systemie David .....	2
2.1. Charakterystyka ogólna .....	2
2.2. Architektura systemu David .....	3
3. Terminologia .....	6
3.1. Autoryzacja dokonywana przez moduły systemu David .....	6
3.2. Terminy używane w systemie David i jego dokumentacji .....	6
4. Instalacja .....	8
4.1. Główny plik konfiguracyjny systemu David .....	8
4.2. Dedykowane konto dla obsługi systemu David .....	8
4.3. Układ katalogów systemu David .....	9
4.4. Konfiguracja demon-a syslogd .....	9
5. Wymagania dla Notification Processor-a .....	10
6. Instalacja .....	11
6.1. Instalacja z pakietu RPM .....	11
6.2. Instalacja za pomocą skryptu .....	11
7. Informacje ogólne .....	12
7.1. Funkcjonalność .....	12
7.2. Opis działania .....	12
7.3. Tematy pokrewne .....	12
8. Segregator Typów Zdarzeń (des) .....	14
8.1. Opis ogólny .....	14
8.2. Składnia .....	14
8.3. Znaczenie opcji w programie des .....	14
8.4. Format pliku konfiguracyjnego .....	15
8.5. Parametry przekazywane uruchamianym programom .....	16
8.6. Opis działania .....	16
8.7. Tematy pokrewne .....	16
9. Odbiorca Powiadomień SNMP-Trap (dtrapd) .....	17
9.1. Opis ogólny .....	17
9.2. Składnia .....	17
9.3. Znaczenie opcji w programie dtrapd .....	17
9.4. Format plików konfiguracyjnych .....	19
9.5. Parametry przekazywane uruchamianym programom .....	19
9.6. Opis działania .....	20
9.7. Tematy pokrewne .....	20
10. Rejestrator Wiadomości (dsi) .....	21
10.1. Opis ogólny .....	21
10.2. Składnia .....	21
10.3. Znaczenie opcji w programie dsi .....	21

10.4. Format zapisu danych .....	24
10.5. Opis działania .....	24
10.6. Tematy pokrewne .....	26
11. Edytor Plików Segregatora Typów Zdarzeń (xdesc) .....	27
11.1. Opis ogólny .....	27
11.2. Opis działania .....	27
11.2.1. Rozpoczynanie i kończenie pracy .....	27
11.2.2. Opis działania głównego okna aplikacji .....	27
11.3. Edycja poszczególnych elementów listy głównego okna dialogowego aplikacji .....	29
11.3.1. Znaczenie przycisków funkcyjnych w okienku dialogowym Traps range .....	29
11.3.2. Konfiguracja zaznaczonych programów i ich zakresów czasowych .....	30
11.4. Tematy pokrewne .....	33
12. Przyciski najczęściej występujące w aplikacjach WWW .....	34
12.1. Znaczenie przycisków .....	34
13. Przeglądarka Trapów (Trap Browser) .....	36
13.1. Opis ogólny .....	36
13.2. Opis działania .....	36
13.2.1. Specyfikacja kryteriów wyszukiwania .....	36
13.2.2. Wygenerowany raport .....	38
13.3. Tematy pokrewne .....	39

---

## Spis tabel

1.1. Konwencje typograficzne użyte w dokumencie .....	1
2.1. Produkty wchodzące w skład systemu David .....	3
8.1. Znaczenie opcji w programie des .....	14
9.1. Znaczenie opcji w programie dtrapd .....	17
9.2. Znaczenie argumentu próśb i zapytań .....	19
10.1. Znaczenie opcji w programie dsi .....	21
11.1. Aplikacja xdesc - opis przycisków z menu File i Edit .....	28
11.2. Opis przycisków okienka Traps range .....	30
11.3. Opis przycisków okienka Time ranges and programs .....	31
12.1. Przyciski najczęściej występujące w aplikacjach WWW .....	34
13.1. Przeglądarka Trapów - znaczenie pól widoku szczegółowego grupy Traps .....	37

---

# Rozdział 1. Konwencje typograficzne

Następujące konwencje typograficzne są użyte w niniejszym dokumencie:

**Tabela 1.1. Konwencje typograficzne użyte w dokumencie**

Czcionka	Znaczenie	Przykład
<i>Kursywa</i>	Nazwy zmiennych środowiskowych	Nazwa pliku przechowywana jest w zmiennej środowiskowej <code>\$DAVIDPRIVDIR...</code>
<i>Kursywa</i>	Opcje składni.	<code>[-l,--log-facility log_facility]</code>
<b>Pogrubiona</b>	Nazwy programów, aplikacji i produktów.	Program <b>damcsud</b> jest częścią <b>Operation Manager-a</b> .
Rozstrzelona	Nazwy opcji i menu.	W menu View znajduje się także opcja Show tool bar.
Rozstrzelona	Nazwy plików i katalogów.	... czyta swój plik konfiguracyjny <code>.damadbudrc</code> .
Rozstrzelona	Nazwy okienek i pól w okienkach dialogowych.	W okienku A sessions property w polu Sticking string podaje się tekst...
Rozstrzelona	Nazwy przycisków.	Przez naciśnięcie przycisku Tab z klawiatury możesz uzyskać focus.
<b>Rozstrzelona pogrubiona</b>	Wzory matematyczne.	<code>exp( -x )</code> , gdy $a = 0$ <code>1 / pow( a , a ) * pow( x , a ) * exp( -x + a )</code> , gdy $a > 0$ .
<b>Rozstrzelona pogrubiona</b>	Terminy użyte w terminologii systemu David.	<b>SNMP Data</b> - rodzaj danych występujących...
<b>Rozstrzelona pogrubiona</b>	Zawartość plików konfiguracyjnych.	<code>action</code> <code>{</code> <code>...</code> <code>}</code>

---

# Rozdział 2. Informacje ogólne o systemie David

## 2.1. Charakterystyka ogólna

**System David** to system zarządzania siecią komputerową. Jest on pakietem programów (modułów) pozwalającym zdalnie, tzn. poprzez rozległą sieć komputerową (np.: Internet), monitorować i zarządzać, w czasie rzeczywistym, urządzeniami pracującymi w sieciach komputerowych. Jedynym warunkiem, jakie muszą spełniać urządzenia jest, aby pracował na nich agent SNMP (Simple Network Management Protocol). Wobec faktu, że SNMP jest najbardziej rozpowszechnionym protokołem zarządzania na świecie, wymaganie to nie jest szczególnie trudne do spełnienia. Wiele urządzeń posiada oprogramowanie, które pozwala się z nimi komunikować poprzez protokół SNMP. Do urządzeń tych należą m.in.:

- routery IP,
- switche ATM-owe,
- zarządzalne switche ethernetowe,
- UPS-y wyposażone w adaptory SNMP,
- modemy telewizyjnych sieci kablowych pozwalające pracować urządzeniom IP w sieciach telewizji kablowej,
- stacje komputerowe.

Jedną z głównych cech **systemu David** jest fakt, że jest on złożony z wysoce konfigurowalnych i dalece niezależnych od siebie modułów. Staranność o utrzymanie takiego sposobu projektowania systemu jest widoczna od początku jego powstania. W konsekwencji, z tych samych modułów, można zbudować istotnie różniące się w działaniu konfiguracje **systemu David**. Jako jego główne cechy można więc wymienić:

- ogólność w podejściu do sterowania przepływem informacji wynikająca z wysokiej niezależności od siebie modułów systemu,
- wysoka konfigurowalność wszystkich modułów systemu pozwalająca maksymalnie zbliżyć się do oczekiwanego rezultatu podczas konfigurowania pracy systemu,
- skalowalność systemu, tzn. system można łatwo rozbudować dodając kolejne moduły, nawet nie wchodzące w skład **systemu David**, a także bez trudu można poszerzać listę monitorowanych urządzeń,
- wykorzystanie skryptów shell'owych w strumieniu przepływu informacji pozwala w łatwy sposób



formatować i wpływać na przetwarzane informacje,

- wszystkie pliki konfiguracyjne **systemu David**, a także pliki z danymi wejściowymi jak i wyjściowymi, pliki z istotnymi dla systemu komunikatami (log files), są plikami tekstowymi,
- komunikacja z monitorowanymi urządzeniami poprzez protokoły SNMPv1, SNMP 2C i SNMPv3.

## 2.2. Architektura systemu David

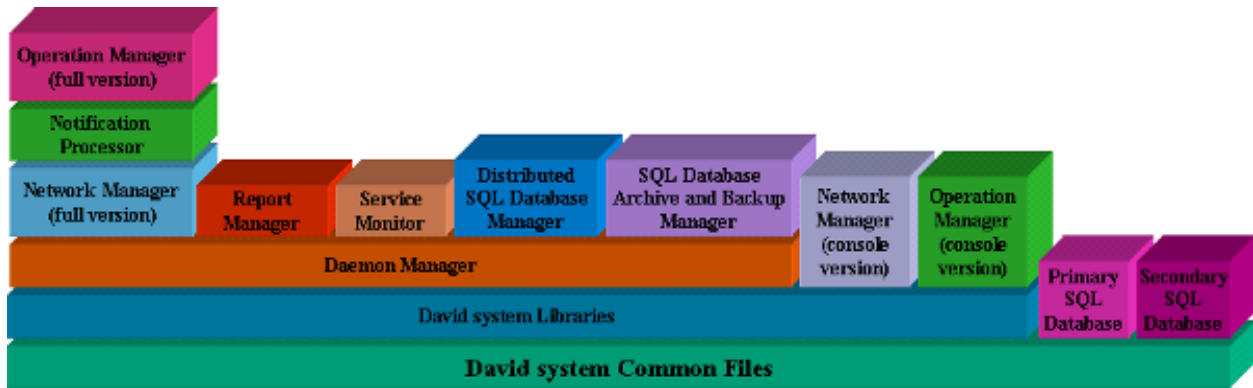
Tabela 2.1. Produkty wchodzące w skład systemu David

Nazwa produktu	Opis
David system Common Files	Produkt podczas instalacji przygotowuje układ katalogów dla innych produktów <b>systemu David</b> . Zawiera też kilka podstawowych plików dla wszystkich innych produktów. Jest to podstawowy produkt <b>systemu David</b> wymagany przez resztę jego produktów.
Primary SQL Database	Produkt instaluje podstawową SQL-ową bazę danych <b>systemu David</b> . Każda instalacja <b>systemu David</b> musi mieć tylko jedną podstawową bazę danych.
Secondary SQL Database	Produkt instaluje dodatkową SQL-ową bazę danych <b>systemu David</b> . Może istnieć wiele dodatkowych baz danych w jednej instalacji <b>systemu David</b> . Pozwala to rozpraszać bazę danych <b>systemu David</b> na wiele serwerów.
David system Libraries	Dostarcza biblioteki <b>systemu David</b> potrzebne aplikacjom systemu. Produkt ten jest wymagany przez wiele innych produktów <b>systemu David</b> .
Daemon Manager	Zajmuje się uruchamianiem i zatrzymywaniem demonów <b>systemu David</b> oraz monitorowaniem ich pracy.
Network Manager (full version)	Produkt poprzez wykorzystanie protokołu SNMP umożliwia wizualizację topologii monitorowanych sieci i automatyczne odkrywanie urządzeń w zarządzanych sieciach. Wizualizacji podlega także stan monitorowanych obiektów. Produkt zajmuje się także zbieraniem danych dotyczących pracy monitorowanych urządzeń, wykorzystując do tego celu protokół SNMP oraz pozwala administrować kontami użytkowników.
Network Manager (console version)	Produkt poprzez graficzną aplikację umożliwia wizualizację topologii monitorowanych sieci oraz stanu monitorowanych obiektów. Pozwala także sterować pracą demonów monitorujących oraz kolekcjonujących dane. Obecnie większość funkcji tej aplikacji jest dostępna także poprzez aplikacje webowe.
Notification Processor	Zajmuje się głównie obsługą komunikatów typu SNMP Trap nadchodzących do stacji zarządzających od monitorowanych urządzeń. Otrzymane komunikaty mogą podlegać dowolnemu formatowaniu do

## Informacje ogólne o systemie David

Nazwa produktu	Opis
	postaci czytelnej dla człowieka, a następnie mogą być rejestrowane. Przetworzone w ten sposób komunikaty mogą być również przekazywane do dalszego przetwarzania.
Operation Manager (full version)	Na podstawie przekazywanych mu danych może on uruchamiać wyspecyfikowane akcje. Skomplikowany często sposób oceny sytuacji, dokonywany m.in. na podstawie otrzymywanych sygnałów od innych produktów <b>systemu David</b> wraz z ich korelacją, ma na celu, w sposób nieco bardziej inteligentny niż tylko prosta reakcja na nie, generowanie informacji dla operatora w odpowiedzi na zaistniałe zdarzenia. Graficzna aplikacja wyświetla powiadomienia o zdarzeniach oraz umożliwia odtwarzanie plików dźwiękowych oraz odczytywanie komunikatów przez zewnętrzny syntezytor mowy.
Operation Manager (console version)	Produkt zawiera graficzną aplikację wyświetlającą powiadomienia o zdarzeniach oraz umożliwiającą odtwarzanie plików dźwiękowych oraz odczytywanie komunikatów przez zewnętrzny syntezytor mowy.
Report Manager	Zajmuje się przetwarzaniem zarejestrowanych komunikatów SNMP Trap, wpisów o toczących się sprawach oraz wpisów dotyczących zmiany stanów monitorowanych obiektów (obiektów pingowanych, interfejsów sieciowych i sąsiadów BGP) i generuje na tej podstawie dane do raportów. Wizualizacja wygenerowanych raportów dokonywana jest przez aplikację webową.
Service Monitor	Monitoruje wskazane serwisy sieciowe na poziomie warstwy aplikacyjnej. W tym celu monitoruje wskazane porty TCP wyspecyfikowanych hostów. Sprawdza zarówno dostępność portów jak i poprawność reakcji dla kilku wybranych protokołów sieciowych (HTTP, SMTP, FTP). Może także weryfikować poprawność pracy serwisów poprzez weryfikację danych od nich uzyskanych. Wyniki jego pracy w postaci raportów i wykresów prezentuje aplikacja webowa.
SQL Database Archive and Backup Manager	Zajmuje się archiwizowaniem danych przechowywanych w SQL-owej Bazie Danych, z których korzystają aplikacje <b>systemu David</b> .
Distributed SQL Database Manager	Pozwala podzielić bazę danych <b>systemu David</b> na jedną bazę podstawową oraz wiele baz dodatkowych. Pozwala to zwiększyć szybkość pracy systemu poprzez rozproszenie obciążenia na wiele serwerów. Proces migracji odbywa się podczas normalnej pracy systemu a podział bazy danych może być modyfikowany wielokrotnie.

Zależności pomiędzy poszczególnymi produktami **systemu David** przedstawia poniższy schemat.



Funkcjonalność **systemu David** może być bardzo szeroka i w ogromnym stopniu zależy od konkretnej konfiguracji. Najważniejsze funkcje jakie system może dostarczać to:

- odkrywanie i wizualizacja topologii monitorowanych sieci wraz z wizualizacją stanów poszczególnych węzłów sieci;
- formatowanie i rejestrowanie komunikatów typu SNMP-Trap nadsyłanych przez monitorujące urządzenia;
- automatyczne reagowanie na wybrane komunikaty typu SNMP-Trap nadsyłane przez monitorowane urządzenia;
- możliwość identyfikacji operatora odbierającego od systemu zgłoszenie o problemie (awarii);
- kolekcjonowanie danych dotyczących pracy monitorowanych urządzeń;
- automatyczne reagowanie na wykryte podczas kolekcjonowania danych nieprawidłowe wartości danych;
- prowadzenie ewidencji aktualnych spraw prowadzonych przez system powstałych jako reakcja na zdarzenia mające miejsce w zarządzanej sieci i wykrytych przez system;
- monitorowanie serwisów sieciowych warstwy aplikacji.

---

# Rozdział 3. Terminologia

## 3.1. Autoryzacja dokonywana przez moduły systemu David

Moduły pracujące w ramach systemu David, które potrzebują dokonywać autoryzacji nadawców wiadomości (np. **damsnmpdaud**, **dnmmsd**, **dgnsd**), korzystają z biblioteki, która sprawdza, czy adres IP nadawcy pasuje do jakiegokolwiek wpisu w pliku `.known.host`. Biblioteka spodziewa się, że plik ten znajduje się w podkatalogu `.sec` katalogu, którego nazwa wskazywana jest przez zmienną `confdir` w pliku `/etc/system-david.conf`.

Wpisy w pliku `.known.host` mają postać wyrażeń regularnych specyfikujących adresy IP, które mają być akceptowane.

## 3.2. Terminy używane w systemie David i jego dokumentacji

Ponizej znajduje się wyjaśnienie części terminów, które są używane przez system David oraz w dokumentach opisujących jego pracę:

- **wiadomości (informacje)** - najczęściej są to dane otrzymywane przez interfejsy **Operation Manager-a**, jego analizatory danych oraz **Jednostkę Tworzącą Bazę Danych Aktywnych Spraw** należącą do tego produktu;
- **komunikaty** - termin ten występuje głównie w produktach: **Notification Processor**, **Operation Manager** i **Report Manager**; najczęściej są to dane, których źródłem są agenci SNMP pracujący na monitorowanych urządzeniach sieciowych;
- **zdarzenia (events)** - termin ten występuje najczęściej w produktach takich jak: **Operation Manager** i **Report Manager**; określa byt, którego źródłem jest pojedyncza dana typu SNMP Trap lub SNMP Data; **zdarzenie** zawsze wchodzi w skład **sprawy**;
- **sprawy (cases)** - termin ten występuje najczęściej w produktach takich jak: **Operation Manager** i **Report Manager**; określa zbiór zdarzeń skojarzonych ze sobą; w skład sprawy musi wchodzić przynajmniej jedno **zdarzenie**;
- **SNMP Trap** - rodzaj danych występujących w produkcie **Operation Manager**, których źródłem są komunikaty otrzymywane od agentów SNMP; komunikaty te nie są odpowiedzią na prośby wysyłane przez stację zarządzającą, lecz są samoistnie wysyłane przez agentów zarządzających urządzeniami sieciowymi i przetwarzane przez produkt **Notification Processor**;

- **SNMP Data** - rodzaj danych występujących w produkcie **Operation Manager**, których źródłem są odpowiedzi otrzymywane od agentów SNMP na prośby, wysyłane do nich przez stację zarządzającą za pomocą produktu **Network Manager**.

---

# Rozdział 4. Instalacja

## 4.1. Główny plik konfiguracyjny systemu David

Podstawowy plik konfiguracyjny systemu David to `/etc/david-system.conf`. Zawiera on wpisy konfiguracyjne jako pary: klucz = wartość. Poza wpisem `default_email_recipient` w zasadzie żadna inna pozycja nie musi być modyfikowana ręcznie. Wszystkie konieczne modyfikacje dokonywane są podczas instalowania poszczególnych produktów systemu David. Poniżej znajduje się lista możliwych wpisów wraz z ich opisem.

- `user` - nazwa użytkownika z prawami którego pracują demony systemu David;
- `default_email_recipient` - domyślny adres e-mail, gdzie są wysyłane wiadomości od aplikacji systemu David;
- `bindir` - katalog z aplikacjami systemu David (domyślnie: `/usr/bin/david-system`);
- `libdir` - katalog z bibliotekami systemu David (domyślnie: `/usr/lib/david-system`);
- `incdir` - katalog z plikami nagłówkowymi systemu David (domyślnie: `/usr/include/david`);
- `confdir` - katalog z plikami konfiguracyjnymi systemu David (domyślnie: `/etc/david-system`);
- `logdir` - katalog, gdzie są tworzone logi aplikacji systemu David (domyślnie: `/var/log/david-system`);
- `sharedir` - katalog z różnymi plikami (obrazki, pliki audio, serwis webowy itd.) systemu David (domyślnie: `/usr/share/david-system`);
- `docdir` - katalog z dokumentacją systemu David (domyślnie: `/usr/share/doc/david-system`);
- `vardir` - katalog z archiwami bazy SQL-owej systemu David (domyślnie: `/var/lib/david-system`);
- `is_sqldb_installed` - flaga, czy SQL-owa baza systemu David została zainstalowana.

## 4.2. Dedykowane konto dla obsługi systemu David

Nie ma potrzeby, aby jakikolwiek moduł systemu David pracował z prawami superużytkownika (zazwyczaj konto o nazwie `root` i UID równym 0). Nawet, jeśli dany daemon systemu David wymaga w chwili uruchomienia praw superużytkownika, to zawsze istnieje możliwość wyspecyfikowania jako argumentu uruchomienia demona nazwy użytkownika, którego prawa ma przyjąć.

Najwygodniej jest dodać nowego użytkownika do systemu operacyjnego, pod którego kontrolą ma

pracować system David.

## 4.3. Układ katalogów systemu David

Układ katalogów i ich zawartość może być zależna od konkretnej konfiguracji systemu David. W standardowej konfiguracji systemu poszczególne katalogi zawierają:

- `/usr/bin/david-system` - pliki binarne i skrypty;
- `/etc/david-system` - pliki konfiguracyjne;
- `/usr/share/doc/david-system` - dokumentację systemu David;
- `/usr/share/david-system` - pliki graficzne, dźwiękowe, portal webowy;
- `/usr/include/david` - pliki nagłówkowe systemu David;
- `/usr/lib/david-system` - biblioteki systemu David;
- `/var/log/david-system` - pliki z logami;
- `/var/lib/david-system` - archiwa bazy SQL-owej systemu David.

## 4.4. Konfiguracja demon-a syslogd

Moduły systemu David wykorzystują standardowy podsystem `syslog` dostępny na platformach UNIX-owych. Domyślne ustawienia modułów systemu David powodują, że informacje są przesyłane jako typ (`facility`) `local6`. Ustawienia te można oczywiście zmienić w momencie uruchamiania danego modułu. W związku z tym dobrym pomysłem wydaje się takie skonfigurowanie demona `syslogd`, aby wszystkie informacje przesyłane od modułów systemu David znalazły się w jednym miejscu (w jednym lub kilku plikach o charakterystycznej nazwie np.: `david.log`).

---

# Rozdział 5. Wymagania dla Notification Processor-a

Platforma zarządzania, na której ma pracować **Notification Processor** musi spełniać następujące wymagania:

- posiadać zainstalowaną kompatybilną wersję **Network Manager-a (full version)**.



---

# Rozdział 6. Instalacja

## 6.1. Instalacja z pakietu RPM

Instalacja produktu wymaga uruchomienia przez użytkownika posiadającego prawa `root-a`. Poniżej znajdują się kolejne kroki typowej instalacji:

- Zainstaluj produkt:

```
rpm -i david-xxx-np-yyy.rpm
```

## 6.2. Instalacja za pomocą skryptu

Instalacja produktu wymaga uruchomienia przez użytkownika posiadającego prawa `root-a`. Poniżej znajdują się kolejne kroki typowej instalacji:

- Rozkompresuj i rozpakuj archiwum:

```
gunzip david-xxx-np-yyy.i386.tar.gz  
tar xf david-xxx-np-yyy.i386.tar
```

Operacje te spowodują utworzenie katalogu `david-xxx-np-yyy.i386` w bieżącym katalogu.

- Zmień swój bieżący katalog na `david-xxx-np-yyy.i386`:

```
cd david-xxx-np-yyy.i386
```

- Przeczytaj plik `LICENSE` z bieżącego katalogu i **KONTYNUUJ INSTALACJĘ TYLKO WTEDY, KIEDY AKCEPTUJESZ WARUNKI TAM ZAWARTE.**
- Uruchom skrypt instalacyjny:

```
./install
```

---

# Rozdział 7. Informacje ogólne

## 7.1. Funkcjonalność

**Notification Processor** umożliwia:

- reagowanie na komunikaty SNMP-Trap otrzymywane od monitorowanych urządzeń zależnie od nadawcy i typu wiadomości;
- przekazywanie otrzymywanych komunikatów dowolnie wybranemu programowi (nie musi on być modulem systemu David) zależnie od nadawcy oraz typu wiadomości (można np. wysłać pocztę do operatora, sms na telefon komórkowy itp.);
- formatowanie otrzymanego komunikatu do dowolnej postaci;
- rejestrowanie otrzymywanych informacji;
- przeglądanie zarejestrowanych wiadomości;
- przekazywanie przetworzonej informacji do **Operation Manager-a**

## 7.2. Opis działania

**Notification Processor** zajmuje się przetwarzaniem komunikatów SNMP-Trap otrzymywanych od węzłów sieci komputerowej, które podlegają monitorowaniu. Efekty jego pracy mogą wpływać na produkt **Operation Manager**.

Produkt otrzymuje powiadomienia SNMP od odległych agentów i dla każdego otrzymanego komunikatu, uruchamia wyspecyfikowane programy i skrypty shell'owe zależnie od adresu IP nadawcy wiadomości lub typu urządzenia, na którym pracuje nadawca. Uruchamiane programy otrzymują informacje zawarte w komunikatach w postaci swoich argumentów wywołania.

**Notification Processor** rejestruje także otrzymywane komunikaty, które następnie mogą być przeglądane w postaci prostych raportów.

## 7.3. Tematy pokrewne

[Rejestrator Wiadomości \(dsi\)](#)

[Segregator Typów Zdarzeń \(des\)](#)

[Odbiorca Powiadomień SNMP-Trap \(dtrapd\)](#)

[Edytor Plików Segregatora Typów Zdarzeń \(xdesc\)](#)

[Przeglądarka Trapów \(Trap Browser\)](#)

---

# Rozdział 8. Segregator Typów Zdarzeń (des)

## 8.1. Opis ogólny

Program **des** jest Segregatorem Typów Zdarzeń i jest częścią **Notification Processor-a** systemu David. Uruchamia on podane programy ze względu na wartość jednego z otrzymanych parametrów: **-n**, **-p** lub **-g**. Został on specjalnie zaprojektowany z myślą o uruchamianiu go przez **dtrapd**. W ten sposób przekazywane są mu w postaci argumentów wywołania informacje zawarte w komunikacie SNMP-Trap otrzymanym przez **dtrapd**. Można zatem stwierdzić, że **dtrapd** dzieli strumień informacji ze względu na adresy IP nadawców komunikatów, zaś **des** może dalej dzielić ten strumień ze względu na rodzaj komunikatów. Należy jednak pamiętać, że z punktu widzenia **des** nie operuje on na komunikacie SNMP-Trap, lecz korzysta wyłącznie z danych przekazanych mu jako jego argumenty wywołania. Na podstawie tych argumentów oraz wczytanej konfiguracji **des** uruchamia wyspecyfikowane programy przekazując im w postaci argumentów wywołania prawie wszystkie informacje, które wcześniej otrzymał.

Aktualną konfigurację **des** otrzymuje w chwili uruchomienia z pliku podanego jako jeden ze swoich parametrów wywołania.

## 8.2. Składnia

Program **des** można uruchomić z następującymi opcjami: **[-o] [-l,--log-facility log\_facility] [-L,--log-level log\_level] [-v,--version] [-h,--help] (config\_file) (-n event\_oid | -p specific\_type | -g generic\_type) [others...]**

## 8.3. Znaczenie opcji w programie des

Tabela 8.1. Znaczenie opcji w programie des

Nazwa opcji	Opis
<b>-o</b>	Kończ przeszukiwanie swojej konfiguracji po napotkaniu pierwszego pasującego wzorca adresu IP nadawcy (domyślnie: szukaj wszystkich pasujących wpisów).
<b>-l,--log-facility log_facility</b>	Wybierz typ logowania (log facility) do syslogd: daemon   user   local0   ...   local7 (domyślnie: local6).
<b>-L,--log-level log_level</b>	Wybierz poziom logowania (na standardowe wyjście błędów i do daemona syslogd), tzn. będą wypisywane komunikaty wybranego poziomu oraz komunikaty poziomów ważniejszych: emerg   alert   crit   err   warning   notice   info   debug0   ...   debug2 (domyślnie: warning).
<b>config_file</b>	Nazwa pliku konfiguracyjnego.
<b>-n event_oid</b>	Indentyfikator (OID) komunikatu w bazie MIB.

Nazwa opcji	Opis
<i>-p specific_type</i>	Liczba określająca specyficzny typ komunikatu
<i>-g generic_type</i>	Liczba określająca ogólny typ komunikatu
<i>others...</i>	Pozostałe argumenty nie interpretowane przez des, lecz przekazywane uruchamianym programom.
<i>-v,--version</i>	Wypisz numer wersji na swoje standardowe wyjście błędów i zakończ.
<i>-h,--help</i>	Wypisz komunikat o użyciu na swoje standardowe wyjście błędów i zakończ.

## 8.4. Format pliku konfiguracyjnego

Plik konfiguracyjny składa się z linii. Każda z linii pliku konfiguracyjnego stanowi osobny wpis - jednostkę informacji, zaczynającą się od wyrażenia regularnego opisującego identyfikator OID lub typ komunikatu wyrażony liczbą całkowitą długą, ujętego w nawiasy kwadratowe. Format wpisu wygląda następująco:

```
[wyrażenie regularne określające numer powiadomienia] [dzień ...
GG:MM:SS GG:MM:SS] ... ścieżka_do_programu1 arg1 ... argN, ...,
ścieżka_do_programuN arg1 ... argN; [dzień ... GG:MM:SS GG:MM:SS] ...
ścieżka_do_programu1 arg1 ... argN, ..., ścieżka_do_programuN arg1 ...
argN; ...
```

Nie wszystkie elementy podane powyżej muszą wystąpić. Konieczne jest wyrażenie regularne określające numery akceptowanych powiadomień. Musi ono być ujęte w nawiasy kwadratowe. Kolejny element ujęty w takie same nawiasy może wystąpić zero lub większą ilość razy. Każdy z tych elementów określa pewien zakres czasowy. Jeśli informacja nadejdzie w porze wyspecyfikowanej w jednym z takich zakresów, to część linii do najbliższego średnika lub cała linia, jeśli go nie ma, jest akceptowana i dalej przetwarzana. Zakres czasowy podaje się w formie skróconych, angielskich nazw dni tygodnia (każdy dzień tygodnia zaczyna się wielką literą) oraz zakresu godzin wspólnego dla wszystkich wymienionych dni. Jeśli nie wymieni się żadnego dnia tygodnia, znaczy to, że zakres godzinowy dotyczy każdego dnia tygodnia. Jeśli nie poda się żadnego elementu określającego zakresu czasowego, oznacza to, że informacja ma być akceptowana zawsze.

Elementem, który musi zostać wyspecyfikowany jest ścieżka do programu, który ma zostać uruchomiony. Można podać także wymagane przez ten program argumenty. Wyspecyfikowanych programów może być więcej niż jeden, lecz muszą być one wtedy oddzielone od siebie przecinkami:

```
ścieżka_do_programu1 arg1 ... argN, ..., ścieżka_do_programuN arg1 ...
argN
```

Informacja zaczynająca się od ewentualnej specyfikacji zakresu czasowego, a kończąca się ostatnim programem wraz z jego argumentami, może być jedynym wpisem w danej linii i może, lecz nie jest to wymagane, kończyć się średnikiem. Jeśli takich wpisów jest więcej niż obowiązkowy jeden, to muszą być

oddzielone od siebie średnikami.

## 8.5. Parametry przekazywane uruchamianym programom

Poza parametrami, które mogą być wyspecyfikowane w pliku, każdy program otrzymuje również dane dodatkowe otrzymane przez **des** w postaci swoich argumentów wywołania. Do danych tych należą opcje **-n**, **-p**, **-g** z ich argumentami, które sam otrzymał (jeśli daną opcję wogóle otrzymał) oraz wszystkie parametry określone w [SKŁADNI](#) jako [others](#), które nie są interpretowane przez **des**.

## 8.6. Opis działania

Program **des** interpretuje każdą niepustą linię pliku konfiguracyjnego. Jako element wyróżniający rozumie parametr opcji **-n** lub jeśli go nie ma parametr opcji **-p**. Jeśli żadna z tych dwóch opcji nie występuje wtedy elementem wyróżniającym jest parametr opcji **-g**. Jeśli znaleziony element wyróżniający pasuje do wzorca (wyrażenia regularnego) w danej linii konfiguracji to ta linia uznawana jest za pasującą i jest dalej interpretowana. Dalsza część linii może być podzielona średnikami na elementy lub zawierać tylko jeden element zajmujący całą linię. Ostatni lub zajmujący całą linię element nie musi być zakończony średnikiem. Jeśli moment działania programu mieści się w zakresie czasowym danego elementu to uruchamiane są programy wyspecyfikowane w tym elemencie. Program **des** postępuje tak dla każdej linii, aż do napotkania końca pliku. Aby poprzestać na pierwszej pasującej linii program **des** musi być uruchomiony z opcją **-o**.

## 8.7. Tematy pokrewne

[Rejestrator Wiadomości \(dsi\)](#)

[Odbiorca Powiadomień SNMP-Trap \(dtrapd\)](#)

[Edytor Plików Segregatora Typów Zdarzeń \(xdesc\)](#)

---

# Rozdział 9. Odbiorca Powiadomień SNMP-Trap (dtrapd)

## 9.1. Opis ogólny

Program **dtrapd** jest **Odbiorcą Powiadomień SNMP-Trap** i jest częścią **Notification Processor-a**. Jest to proces daemon, który pracuje bez przerwy w czasie działania systemu i oczekuje zgłoszeń (SNMP Trap), które nadchodzą poprzez sieć komputerową od odległych agentów SNMP. Na podstawie swojej konfiguracji dla każdego zgłoszenia uruchamia wyspecyfikowane programy, zależnie od zawartości komunikatu. Informacje zawarte w otrzymanym komunikacie przekazywane są uruchamianym programom jako ich parametry wywołania.

Podczas startu **dtrapd** czyta swoje pliki konfiguracyjne: `.dtrapdrc` i `.dtrapd-maybedropped-rc`. Program spodziewa się je znaleźć w katalogu `/etc/david-system`. Daemon **dtrapd** nie uruchomi się jeśli nie będzie mógł odczytać swojej konfiguracji z pliku `.dtrapdrc`. Sposób interpretacji konfiguracji może być modyfikowany poprzez parametry (opcje) podane programowi **dtrapd** w chwili uruchomienia.

## 9.2. Składnia

Program **dtrapd** można uruchomić z następującymi opcjami: `[-V,--verbose]` `[-d,--dump-packets]` `[-D,--debug-snmp-lib]` `[-o,--only-first-matching]` `[-l,--log-facility log_facility]` `[-L,--log-level log_level]` `[-P,--pid-file filename]` `[-p,--port port]` `[-r,--remote-ovevent-server host]` `[-a,--address-to-accept ipaddress]` `[-A,--address-to-deny address]` `[-c,--community-to-accept community]` `[-C,--community-to-deny community]` `[-u,--run-as-user username]`  `[--extended-distribution]`  `[--load-1level double]`  `[--load-2level double]`  `[--background]`  `[-v,--version]`  `[-h,--help]`

## 9.3. Znaczenie opcji w programie dtrapd

Tabela 9.1. Znaczenie opcji w programie dtrapd

Nazwa opcji	Opis
<code>-V,--verbose</code>	Wydrukuj informacje o każdym pojedynczym otrzymanym zdarzeniu na standardowe wyjście.
<code>-d,--dump-packets</code>	Wypisuj informacje o każdym pojedynczym komunikacie (SNMP-Trap) na swoje standardowe wyjście.
<code>-D,--debug-snmp-lib</code>	Włącz debugowanie podkładowej biblioteki SNMP, tzn. aktualnie wykorzystywanej przez pośredniczącą bibliotekę SNMP systemu David.
<code>-o,--only-first-matching</code>	Kończ przeszukiwanie swojej konfiguracji po napotkaniu pierwszego

## Odbiorca Powiadomień SNMP-Trap (dtrapd)

Nazwa opcji	Opis
	pasującego wpisu (domyślnie: szukaj wszystkich pasujących wpisów).
<i>-P,--pid-file filename</i>	Zapisz swój PID w wyspecyfikowanym pliku
<i>-l,--log-facility log_facility</i>	Wybierz typ logowania (log facility) do syslogd: daemon   user   local0   ...   local7 (domyślnie: local6).
<i>-L,--log-level log_level</i>	Wybierz poziom logowania (na standardowe wyjście błędów i do demona syslogd), tzn. będą wypisywane komunikaty wybranego poziomu oraz komunikaty poziomów ważniejszych: emerg   alert   crit   err   warning   notice   info   debug0   ...   debug2 (domyślnie: warning).
<i>-p,--port port</i>	Nasłuchuj bezpośrednio na podanym porcie UDP (domyślnie: poprzez OVEvent).
<i>-r,--remote-ovevent-server host</i>	Podłącz się do odległego demona OVEvent (domyślnie: do lokalnego OVEvent).
<i>-a,--address-to-accept ipaddress</i>	Akceptuj tylko komunikaty (SNMP-Trap) przychodzące z wyspecyfikowanym IP jako adresem źródłowym wiadomości. Opcja ta użyta wielokrotnie pozwoli zbudować listę akceptowanych adresów IP.
<i>-A,--address-to-deny address</i>	Odrzucaj komunikaty (SNMP-Trap) przychodzące z wyspecyfikowanym IP jako adresem źródłowym wiadomości. Opcja ta użyta wielokrotnie pozwoli zbudować listę nieakceptowanych adresów IP.
<i>-c,--community-to-accept community</i>	Akceptuj tylko komunikaty (SNMP-Trap) przychodzące z wyspecyfikowaną nazwą jako nazwą wspólnoty (community). Opcja ta użyta wielokrotnie pozwoli zbudować listę akceptowanych wspólnot (communities).
<i>-C,--community-to-deny community</i>	Odrzucaj komunikaty (SNMP-Trap) przychodzące z wyspecyfikowaną nazwą jako nazwą wspólnoty (community). Opcja ta użyta wielokrotnie pozwoli zbudować listę nieakceptowanych wspólnot (communities).
<i>-u,--run-as-user username</i>	Zrezygnuj z praw root'a i uruchom serwer z prawami podanego użytkownika.
<i>--extended-distribution</i>	Włącz rozszerzone rozsyłanie pakietów SNMP-Trap (pasujący numer wiadomości lub identyfikator producenta lub zmienne dołączone do pakietu) zamiast trybu domyślnego (pasujący adres nadawcy).
<i>--load-1level double</i>	Odrzucaj komunikaty pasujące do tych opisanych w pliku <b>.dtrapd-maybedropped-rc</b> kiedy obciążenie systemu operacyjnego jest większe od tej wartości i poniżej wartości <a href="#">--load-2level double</a> (domyślnie: 20.0).
<i>--load-2level double</i>	Odrzucaj wszystkie komunikaty kiedy obciążenie systemu operacyjnego jest większe od tej wartości (domyślnie: 30.0).
<i>--background</i>	Rozpocznij pracę w tle po uruchomieniu
<i>-v,--version</i>	Wypisz numer wersji na swoje standardowe wyjście błędów i zakończ.
<i>-h,--help</i>	Wypisz komunikat o użyciu na swoje standardowe wyjście błędów i



Nazwa opcji	Opis
	zakończ.

## 9.4. Format plików konfiguracyjnych

Plik konfiguracyjny `.dtrapd` składa się z linii. Każda z linii pliku konfiguracyjnego stanowi osobny wpis - jednostkę informacji, zaczynającą się od wyrażenia regularnego opisującego, zależnie od potrzeby, adres IP lub OID, ujętego w nawiasy kwadratowe. Następnie występują znaki spacji lub tabulacji, a dalej podana jest nazwa programu i ewentualnie jego argumenty, które należy mu przekazać w chwili uruchomienia.

### Przykładowe wpisy pliku konfiguracyjnego

Przykładowe wpisy mogą wyglądać następująco:

```
[^10\.0\.0\.[0-9]+$] $DAVIDDIR/bin/des -o $DAVIDDIR/conf/cisco-gw.esc
```

```
[.*] $DAVIDDIR/bin/des -o $DAVIDDIR/conf/rest.esc
```

Plik konfiguracyjny `.dtrapd-maybedropped-rc` jest uboższą wersją pliku `.dtrapd`. Każda z jego linii w całości zawiera wyrażenie regularne opisujące komunikaty, które mają w pierwszej kolejności być odrzucane, gdy obciążenie systemu operacyjnego wzrośnie powyżej zadanego poziomu.

## 9.5. Parametry przekazywane uruchamianym programom

Poza parametrami, które mogą być wyspecyfikowane w pliku konfiguracyjnym, każdy program otrzymuje również dane zawarte w komunikacie w postaci kolejnych argumentów wywołania. Są to:

**Tabela 9.2. Znaczenie argumentu próśb i zapytań**

Nazwa argumentu	Opis
-s address	Adres IP nadawcy komunikatu
-a address	Adres IP agenta - nadawcy komunikatu
-c community	Nazwa wspólnoty zawarta w komunikacie.
-e enterprise	Identyfikator (OID) producenta urządzenia w bazie MIB, z którego pochodzi komunikat.
-g generic_type	Liczba określająca ogólny typ komunikatu (tylko w SNMPv1).
-p specific_type	Liczba określająca specyficzny typ komunikatu (tylko w SNMPv1).
-n notify_oid	Identyfikator (OID) komunikatu w bazie MIB (tylko w SNMPv2).
-t timestamp	Liczba setnych części sekundy jaka upłynęła od inicjalizacji agenta, który

Nazwa argumentu	Opis
	wysłał tę wiadomość.

Zmienne zawarte w komunikacie przekazywane są na końcu listy argumentów bez poprzedzającego przełącznika. Każda zmienna jest ujęta w nawiasy kwadratowe oraz posiada następujący format: [OID: wartość], gdzie `OID` jest identyfikatorem zmiennej w bazie MIB zaś `wartość` jest jedną z możliwych typów wartości jakie mogą przyjmować zmienne bazy MIB.

## 9.6. Opis działania

W momencie otrzymania komunikatu daemon **dtrapd** sprawdza najpierw, czy adres IP nadawcy jest na liście adresów akceptowanych, jeśli wogóle istnieje taka lista. W przypadku, gdy nie ma go na tej liście wiadomość jest odrzucana. Natomiast jeśli nie ma listy akceptowanych adresów wówczas sprawdzane jest, czy adres IP nadawcy komunikatu nie jest na liście nieakceptowanych adresów. Jeśli znajduje się na tej liście wówczas wiadomość jest odrzucana. Komunikaty, które w ten sposób nie zostały odrzucone poddawane są analogicznej procedurze w odniesieniu do zawartości pola `community` w komunikacie.

Dla komunikatów, które nie zostały odrzucone uruchamiane są wyspecyfikowane programy zgodnie z wczytaną podczas startu konfiguracją. Jeśli program **dtrapd** został uruchomiony bez opcji [--extended-distribution](#) oraz adres IP nadawcy komunikatu pasuje do wyrażenia regularnego danej linii pliku konfiguracyjnego, to program z tej linii zostanie uruchomiony i jako argumenty uruchomienia zostaną mu przekazane informacje zawarte w tym komunikacie. Jeśli program **dtrapd** został zaś uruchomiony z opcją [--extended-distribution](#) oraz numer wiadomości lub identyfikator producenta, lub zmienne dołączone do pakietu pasują do wyrażenia regularnego danej linii pliku konfiguracyjnego, to program z tej linii zostanie uruchomiony i jako argumenty uruchomienia zostaną mu przekazane informacje zawarte w tym komunikacie.

Aby przestać na uruchomieniu programu tylko z pierwszej, pasującej linii należy uruchomić demona **dtrapd** z opcją [-o](#).

## 9.7. Tematy pokrewne

[Segregator Typów Zdarzeń \(des\)](#)

# Rozdział 10. Rejestrator Wiadomości (dsi)

## 10.1. Opis ogólny

Program **dsi** jest **Rejestratorem Wiadomości** i jest częścią **Notification Processor-a**. Zadaniem **dsi** jest zapisywanie danych, które otrzymuje jako swoje parametry wywołania w odpowiednim formacie do wskazanych plików lub na standardowe wyjście. Program **dsi** rejestruje informacje otrzymane jako swoje parametry wywołania w odpowiednim formacie. Typowym źródłem tych danych są powiadomienia SNMP, które otrzymuje **Notification Processor**, a które wcześniej mogły być przetwarzane przez inne programy tego podsystemu (np. **des** lub **dtrapd**). Innym źródłem może być **Operation Manager**. Jeżeli **dsi** uruchomi się z opcją **-A**, to wówczas będzie się starał poprosić serwer okienek graficznych **dgnsd** z produktu **Operation Manager**, aby wyświetlił powiadomienie dla operatora. Jeśli serwer będzie mógł wyświetlić okienko z informacją to jako informację zwrotną zwróci identyfikator operatora i czas po jakim zareagował on na tę wiadomość. Te informacje również zostaną zapisane do plików. Jeśli żaden plik nie zostanie wyspecyfikowany lub do żadnego z podanych plików nie można było zapisać informacji to są one wypisywane na standardowe wyjście. Program **dsi** jest również przygotowany do obsługi (rejestracji i próby wyświetlenia) spraw, które powstają jako efekt pracy produktu **Operation Manager**.

## 10.2. Składnia

Program **dsi** można uruchomić z następującymi opcjami: `[-A,--ask-server]` `[-L,--log-facility log_facility]` `[-L,--log-level log_level]` `[-F,--server-file socket_file]` `[-H,--server-host host]` `[-P,--server-host-tcp-port port]` `[-f,--file file]` `[-C,--color-foreground red green blue]` `[-B,--color-background red green blue]` `[-S,--font-size fontsize]` `[-i,--severity level]` `[-r,--display-time seconds]` `[-s,--source-ip IP_address]` `[-a,--agent-ip IP-address]` `[-c community]` `[-n,--number information_nr]` `[-e,--enterprise enterprise_oid]` `[-t,--time-stamp time-stamp]` `[-g,--generic generic_type]` `[-p,--specific specific_type]` `[-d,--message-id string]`  `[--community community]`  `[-m,--message message]`  `[--ctime unsignedtype_number]`  `[--mtime unsignedtype_number]`  `[--active-value signedtype_number]`  `[--passive-value signedtype_number]`  `[--closing-reason description]`  `[--closed-by username]`  `[--event-ctime unsignedtype_number]`  `[--event-mtime unsignedtype_number]`  `[--event-id unsignedtype_number]`  `[--event-hits-number unsignedtype_number]`  `[--event-state (active | passive | not-managed-here)]`  `[--event-msg text]`  `[--event-successors text]`  `[-v,--version]`  `[-h,--help]`

## 10.3. Znaczenie opcji w programie dsi

Tabela 10.1. Znaczenie opcji w programie dsi

Nazwa opcji	Opis
<code>-A,--ask-server</code>	Przed zapisaniem informacji do pliku poprosz serwer o wyświetlenie okienka graficznego z informacjami o otrzymanym komunikacie (domyślnie: nie kontaktuj się z serwerem, tylko zapisz informacje do

## Rejestrator Wiadomości (dsi)

Nazwa opcji	Opis
	plików).
<i>-l,--log-facility log_facility</i>	Wybierz typ logowania (log facility) do syslogd: daemon   user   local0   ...   local7 (domyślnie: local6).
<i>-L,--log-level log_level</i>	Wybierz poziom logowania (na standardowe wyjście błędów i do daemona syslogd), tzn. będą wypisywane komunikaty wybranego poziomu oraz komunikaty poziomów ważniejszych: emerg   alert   crit   err   warning   notice   info   debug0   ...   debug2 (domyślnie: warning).
<i>-F,--server-file socket_file</i>	Połącz się z serwerem poprzez wyspecyfikowany plik typu socket (domyślnie: /tmp/dgns.ic.socket).
<i>-H,--server-host host</i>	Połącz się z serwerem pracującym na wyspecyfikowanym komputerze (domyślnie: połącz się poprzez plik typu socket).
<i>-P,--server-host-tcp-port port</i>	Połącz się z serwerem pracującym na wyspecyfikowanym komputerze używając wyspecyfikowanego portu TCP (domyślnie: 6677, ale domyślne połączenie jest realizowane poprzez plik typu socket: /tmp/dgns.ic.socket).
<i>-f,--file file</i>	Zapisz informacje w wyspecyfikowanym pliku zamiast przekazywać je na standardowe wyjście. Użycie tej opcji wielokrotnie spowoduje stworzenie listy plików do zapisu (domyślnie: wypisz informacje do zapisu na standardowe wyjście).
<i>-C,--color-foreground red green blue</i>	Wybierz kolor czcionki (format RGB) wiadomości pokazywanej przez serwer w okienku dialogowym (np.: -C "220 0 20" znaczy, że poszczególne składowe RGB będą miały następujące wartości: czerwony=220, zielony=0, niebieski=20, domyślnie: R=0, G=0, B=160).
<i>-B,--color-background red green blue</i>	Wybierz kolor tła (format RGB) wiadomości pokazywanej przez serwer w okienku dialogowym (np.: -B "0 0 100" znaczy, że poszczególne składowe RGB będą miały następujące wartości: czerwony=0, zielony=0, niebieski=100, domyślnie: R=0, G=220, B=0).
<i>-S,--font-size fontsize</i>	Wybierz rozmiar czcionki wiadomości pokazywanej przez serwer w okienku dialogowym (domyślnie: 12).
<i>-i,--severity level</i>	Poziom ważności tej informacji wyrażony jest jako liczba zmiennoprzecinkowa podwójnej precyzji - double (domyślnie: 0.0).
<i>-r,--display-time seconds</i>	Maksymalny czas (w sekundach max: 2 <sup>32</sup> -1) przez jaki wiadomość będzie wyświetlana przez serwer w postaci okienka dialogowego (domyślnie: 60 sekund).
<i>-s,--source-ip IP_address</i>	Adres IP nadawcy wiadomości.
<i>-a,--agent-ip IP-address</i>	Adres IP agenta, który wysłał wiadomość.
<i>-c community</i>	Nazwa wspólnoty SNMP tej wiadomości.
<i>-n,--number information_nr</i>	Identyfikator (OID) wiadomości.
<i>-e,--enterprise enterprise_oid</i>	Identyfikator (OID) producenta urządzenia (w bazie MIB), z którego pochodzi komunikat.

## Rejestrator Wiadomości (dsi)

Nazwa opcji	Opis
<i>-t,--time-stamp time-stamp</i>	Liczba sekund jaka upłynęła od inicjalizacji agenta, który wysłał tę wiadomość.
<i>-g,--generic generic_type</i>	Liczba określająca ogólny typ komunikatu
<i>-p,--specific specific_type</i>	Liczba określająca specyficzny typ komunikatu
<i>-d,--message-id string</i>	Łańcuch znaków identyfikujący tę informację w <b>Operation Manager</b> .
<i>--community community</i>	Community przyznane tej wiadomości.
<i>-m,--message message</i>	Czytelna dla człowieka wiadomość opisująca ten komunikat.
<i>--ctime unsignedtype_number</i>	Czas utworzenia sprawy (liczba sekund od 01.01.1970).
<i>--mtime unsignedtype_number</i>	Czas modyfikacji sprawy (liczba sekund od 01.01.1970).
<i>--id unsignedtype_number</i>	Unikalny w skali dnia identyfikator sprawy.
<i>--active-value signedtype_number</i>	Wartość, która powinna być zwrócona dla dowolnego zdarzenia, kiedy operator podczas wyświetlania okienka przez serwer zdecyduje, że to zdarzenie powinno być aktywne (domyślnie: 1).
<i>--passive-value signedtype_number</i>	Wartość, która powinna być zwrócona dla dowolnego zdarzenia, kiedy operator podczas wyświetlania okienka przez serwer zdecyduje, że to zdarzenie powinno być pasywne (domyślnie: -1).
<i>--closing-reason description</i>	Powód zamknięcia sprawy.
<i>--closed-by username</i>	Identyfikator użytkownika zamykającego sprawę.
<i>--event-ctime unsignedtype_number</i>	Czas utworzenia kolejnego zdarzenia w tej sprawie (liczba sekund od 01.01.1970).
<i>--event-mtime unsignedtype_number</i>	Czas modyfikacji kolejnego zdarzenia w tej sprawie (liczba sekund od 01.01.1970).
<i>--event-id unsignedtype_number</i>	ID kolejnego zdarzenia w tej sprawie.
<i>--event-hits-number unsignedtype_number</i>	Liczba wystąpień kolejnego zdarzenia w tej sprawie.
<i>--event-state (active   passive   not-managed-here)</i>	Stan kolejnego zdarzenia w tej sprawie (aktywny znaczy, że coś jest jeszcze nie w porządku, pasywny, że wszystko jest OK, not-managed-here - tylko do twojej wiadomości).
<i>--event-msg text</i>	Czytelna dla człowieka wiadomość opisująca kolejne zdarzenie w tej sprawie.
<i>--event-successors text</i>	Następcy kolejnego zdarzenia w tej sprawie (znak \n nowej linii oddziela kolejnych następców dla tego zdarzenia).
<i>-v,--version</i>	Wypisz numer wersji na swoje standardowe wyjście błędów i zakończ.
<i>-h,--help</i>	Wypisz komunikat o użyciu na swoje standardowe wyjście błędów i

Nazwa opcji	Opis
	zakończ.

## 10.4. Format zapisu danych

Wszystkie informacje otrzymane jako parametry wywołania programu **dsi** oraz ewentualne dane otrzymane od serwera okienek graficznych są zapisywane w postaci pojedynczej linii składającej się z porcji informacji ujętej w kwadratowe nawiasy i ewentualnie poprzedzonej odpowiednią literą z myślnikiem. Ogólna postać takiej pojedynczej linii wygląda następująco:

*-T* [czas wywołania programu] *-U* [czas wywołania programu w formacie UNIX (liczba sekund od 01.01.1970)] *-i* [argument wywołania **dsi** opcji *-i*] *-r* [argument wywołania **dsi** opcji *-r*] *-s* [argument wywołania **dsi** opcji *-s*] *-a* [argument wywołania **dsi** opcji *-a*] *-c* [argument wywołania **dsi** opcji *-c*] *-g* [argument wywołania **dsi** opcji *-g*] *-p* [argument wywołania **dsi** opcji *-p*] *-n* [argument wywołania **dsi** opcji *-n*] *-e* [argument wywołania **dsi** opcji *-e*] *-t* [argument wywołania **dsi** opcji *-t*] *-d* [argument wywołania **dsi** opcji *-d*] *-M* [argument wywołania **dsi** opcji *--community*] *-q* [argument wywołania **dsi** opcji *--closing-reason*] *-b* [argument wywołania **dsi** opcji *--closed-by*] *-m* [czytelny dla człowieka opis sprawy lub zdarzenia] *-u* [identyfikator operatora otrzymany od serwera] *-R* [czas wyświetlania okienka otrzymany od serwera] reszta danych do zapisania (*other data to store*), które są nierozpoznanymi opcjami wywołania **dsi**.

Informacje zapisywane jako wartość opcji *-m* zależą od tego, czy **dsi** został uruchomiony z opcją *--id* i jej argumentem różnym od -1. Jeśli nie została podana ta opcja (**dsi** nie został więc uruchomiony przez żaden z programów produktu **Operation Manager**) wówczas zapisywany jest argument wywołania **dsi** opcji *-m*. Jeśli opcja *--id* została podana (tzn. **dsi** został uruchomiony przez jakiś program produktu **Operation Manager**, czyli została także podana reszta informacji o sprawie oraz wchodzących w jej skład zdarzeniach) to zapisywane są za opcją *-m* takie informacje jak: argument wywołania **dsi** opcji *-m*, argument wywołania **dsi** opcji *-i* oraz opisy dla każdego zdarzenia oddzielone od siebie średnikami. W skład poszczególnego opisu wchodzi: stan zdarzenia, gdzie (A) znaczy aktywne, (H) znaczy pasywne, czyli historyczne, zaś (NM) znaczy, że w tym przypadku nie można było zmienić jego stanu; argument wywołania *--event-ctime* dla danego zdarzenia; argument wywołania *--event-mtime* dla danego zdarzenia; argument wywołania *--event-hits-number* dla danego zdarzenia; argument wywołania *--event-successors* dla danego zdarzenia (znaki nowej linii oddzielające poszczególnych następców zdarzenia są zamieniane na średniki).

## 10.5. Opis działania

Podstawowym zadaniem programu jest zapisywanie w odpowiednim formacie danych przekazanych mu w postaci parametrów wywołania. Dodatkową funkcją **dsi** może być ewentualne powiadomienie operatora o zdarzeniu lub sprawie, o których informacje zostaną **dsi** przekazane. Powiadomienie operatora następuje poprzez wysłanie prośby do serwera okienek graficznych wraz z całą porcją informacji o sprawie. Serwer wyświetlając informacje dla operatora przekaże z powrotem informacje o czasie wyświetlania okienka oraz identyfikatorze operatora, jeśli ten potwierdzi swoją tożsamość. Serwer

przekazuje także stany dla poszczególnych zdarzeń danej sprawy, zaakceptowane przez operatora (operator może zmienić stan zdarzenia z aktywnego na pasywny lub odwrotnie albo pozostawić je bez zmian). Informacje o stanach zdarzeń, otrzymane od serwera, zostaną jako ostatnie informacje wypisane zawsze na standardowe wyjście. Każda linia tych informacji opisuje osobne zdarzenie. Składa się ona z identyfikatora zdarzenia będącego nieujemną liczbą całkowitą (argumenty wejściowe opcji [--event-id](#)) oraz z liczby określającej czy zdarzenie to ma pozostać aktywne czy pasywne (patrz: opcje [--active-value](#) i [--passive-value](#)). Jeśli operator dokona zmiany stanu przynajmniej jednego zdarzenia, powodując, że wszystkie, które mogą, staną się pasywne, to dodatkowo, także na standardowe wyjście, zostanie wypisana informacja closed-by po której nastąpi identyfikator operatora.

Sprawa może mieć więcej niż jedno zdarzenie. W związku z tym wszystkie opcje postaci `--event-*` (np.: [--event-ctime](#), [--event-id](#), [--event-state](#)) mogą wystąpić więcej niż jeden raz. Dla każdego zdarzenia muszą wystąpić wszystkie opcje postaci `--event-*`, ale ich kolejność jest dowolna. Każda z tych opcji ma wewnętrzny licznik wystąpień dzięki czemu może wystąpić przemieszanie zdarzeń. Program **dsi** stara się w ten sposób łączyć minimalną ilość w pełni zdefiniowanych zdarzeń np. dwa poniższe zdarzenia, choć podane w różny sposób (w innej kolejności), są tożsame:

1)

[--event-ctime 15](#)

[--event-mtime 20](#)

[--event-id 1](#)

[--event-hits-number 2](#)

[--event-state active](#)

[--event-msg "zdarzenie 1"](#)

[--event-ctime 16](#)

[--event-mtime 21](#)

[--event-id 2](#)

[--event-hits-number 3](#)

[--event-state active](#)

[--event-msg "zdarzenie 2"](#)

2)

[--event-ctime 15](#)

[--event-ctime 16](#)

[--event-mtime 20](#)

[--event-mtime 21](#)

[--event-id 1](#)

[--event-id 2](#)

[--event-hits-number 2](#)

[--event-hits-number 3](#)

[--event-state active](#)

[--event-state active](#)

[--event-msg "zdarzenie 1"](#)

[--event-msg "zdarzenie 2"](#)

Informacje przekazane **dsi** w postaci argumentów wywołania zapisywane są do plików (dzięki wielokrotnemu użyciu opcji **-f** można stworzyć ich całą listę). Jeśli żaden plik nie został wyspecyfikowany, albo jeśli do żadnego pliku nie udało się zapisać danych, to informacje zostaną wypisane na standardowe wyjście. Zapisywanie tych informacji odbywa się przed wypisaniem informacji o stanach poszczególnych zdarzeń sprawy na standardowe wyjście, o którym była wyżej mowa. Informacje są też zawsze zapisywane w bazie danych SQL, z której korzystają prawie wszystkie moduły systemu David.

## 10.6. Tematy pokrewne

[Odbiorca Powiadomień SNMP-Trap \(dtrapd\)](#)

[Notification Processor: Segregator Typów Zdarzeń \(des\)](#)

**Operation Manager:** Serwer Powiadomień Graficznych (dgnsd)

**Operation Manager:** Prezentor Powiadomień Graficznych (xdgpn)



---

# Rozdział 11. Edytor Plików Segregatora Typów Zdarzeń (xdesc)

## 11.1. Opis ogólny

Aplikacja **xdesc** jest **Edytorem Plików Segregatora Typów Zdarzeń** i jest częścią **Notification Processor-a**. Jest ona programem narzędziowym pełniącym rolę graficznego interfejsu, pozwalającym przeglądać oraz edytować pliki konfiguracyjne, które są następnie interpretowane przez program [des](#). Proces edycji został zbudowany na zasadzie rozbierania całości na coraz mniejsze jednostki, którymi można w dowolny sposób manipulować. Elementy podstawowe można już edytować bezpośrednio. Powyższy sposób edycji pozwala zachować przejrzystość operacji specyfikacji całego pliku konfiguracyjnego.

## 11.2. Opis działania

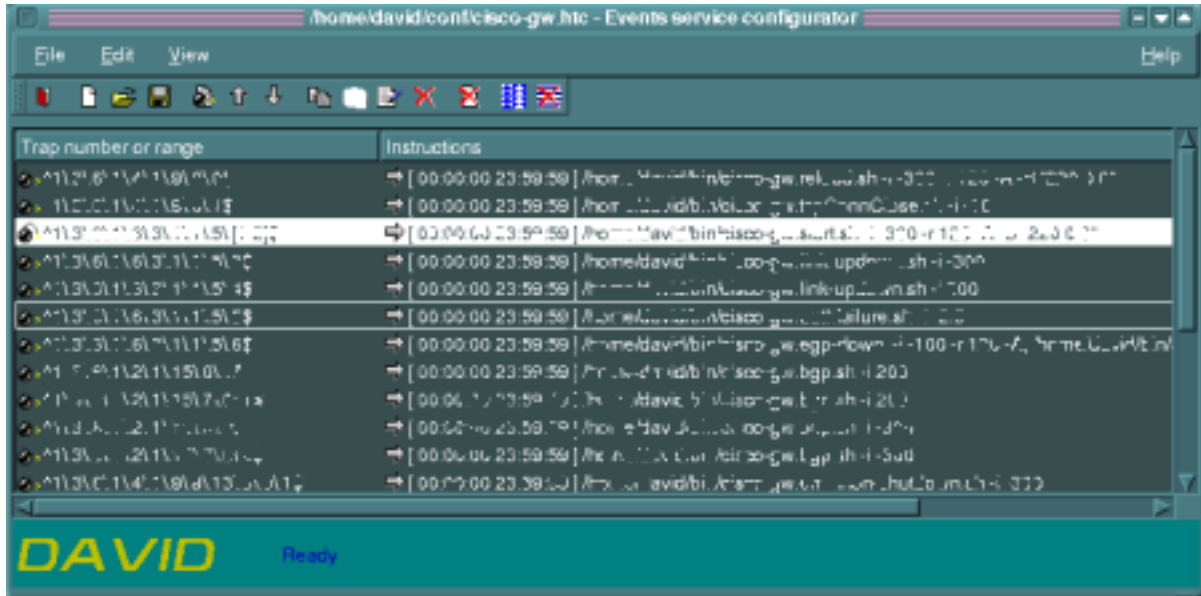
### 11.2.1. Rozpoczynanie i kończenie pracy

Aplikacja **xdesc** podczas startu odczytuje swoje parametry konfiguracyjne, dotyczące swojego wyglądu jak również niektórych parametrów pracy z pliku `.xdescrc`. Aplikacja spodziewa się znaleźć ten plik w katalogu, którego nazwa przechowywana jest w zmiennej środowiskowej `$DAVIDPRIVDIR`. W przypadku kiedy taki plik nie istnieje aplikacja rozpoczyna pracę z domyślnymi ustawieniami. Podczas kończenia swojej pracy program zapisuje w pliku `.xdescrc` bieżące ustawienia i parametry swojej pracy. Informacje o ewentualnych błędach, podczas pracy aplikacji, mogą być wypisywane, w niektórych przypadkach, na standardowe wyjście błędów, zamiast wyświetlania okienka z informacją o błędzie

### 11.2.2. Opis działania głównego okna aplikacji










Główny widok aplikacji pokazuje listę linii pliku konfiguracyjnego. Lista jest podzielona na dwie kolumny. Pierwsza kolumna pokazuje wyrażenia regularne określające zakres komunikatów. Druga natomiast pokazuje instrukcje, które mają być przetwarzane, jeśli identyfikator danego komunikatu pasuje do wzorca opisanego wyrażeniem regularnym widocznym w pierwszej części.






#### 11.2.2.1. Znaczenie przycisków funkcyjnych w głównym oknie aplikacji



Wyświetlone w okienku informacje można edytować za pomocą przycisków znajdujących się na pasku narzędzi. Pierwsze cztery przyciski znajdujące się w lewej części paska narzędziowego odpowiadają opcjom z menu **File**, a kolejne z opcjom z menu **Edit**. Opis funkcjonalności przycisków został opisany w tabeli poniżej:

**Tabela 11.1. Aplikacja xdesc - opis przycisków z menu File i Edit**


Przycisk	Opis
	Umożliwia wyjście z programu. Użytkownik jednak wcześniej zostanie zapytany czy chce zapisać zmiany jakich dokonał w edytowanym pliku (jeśli jakiegokolwiek zmiany zostały dokonane).
	Umożliwia tworzenie nowego pliku konfiguracyjnego.
	Służy do otwierania wcześniej utworzonego pliku konfiguracyjnego.
	Pozwala zapisać edytowany plik konfiguracyjny pod tą samą nazwą (w przypadku nowoutworzonego pliku użytkownik zostanie poproszony o podanie jego nazwy).
Save as	Nie jest wyszczególniona na pasku narzędzi, ale znajduje się w menu <b>File</b> . Pozwala zapisać informacje do pliku o wskazanej przez użytkownika nazwie.
	Pozwala dodać nową linię pliku konfiguracyjnego.
	Pozwala zmienić kolejność wskazywanych linii w pliku, czyli przesunąć je do góry względem siebie.
	Pozwala zmienić kolejność wskazywanych linii w pliku, czyli przesunąć je do dołu względem siebie.
	Umożliwia skopiować zaznaczone linie.
	Pozwala wkleić skopiowane linie na koniec listy.

Przycisk	Opis
	Pozwala konfigurować zaznaczoną linię pliku; otwiera okienko Traps range.
	Umożliwia usunąć zaznaczone linie.
	Pozwala usunąć wszystkie linie wyświetlone w głównym widoku aplikacji.
	Umożliwia odwrócić zaznaczenie linii pliku.
	Pozwala odznaczyć wszystkie wcześniej zaznaczone linie.

W menu View znajdują się jeszcze dwie opcje - Show tool bar oraz Show status bar - za pomocą których można wyświetlić lub schować pasek narzędziowy oraz pasek statusu.

Poprzez menu Help można dowiedzieć się o wersji i czasie powstania aplikacji.

## 11.3. Edycja poszczególnych elementów listy głównego okna dialogowego aplikacji

W głównym oknie aplikacji można edytować poszczególne elementy listy. Kiedy w głównym widoku aplikacji klikniemy podwójnie myszką na zaznaczony element listy lub skorzystamy z przycisku , wówczas spowodujemy otworzenie się okienka dialogowego Traps range, w którym można dokonać konfiguracji zapisanych danych. Okienko jest podzielone na dwie części. Górne pole służy edycji wyrażenia regularnego. Pole dolne podzielone jest na dwie kolumny. Pierwsza pokazuje zakres czasowy, a druga programy z ich argumentami.

### 11.3.1. Znaczenie przycisków funkcyjnych w okienku dialogowym Traps range

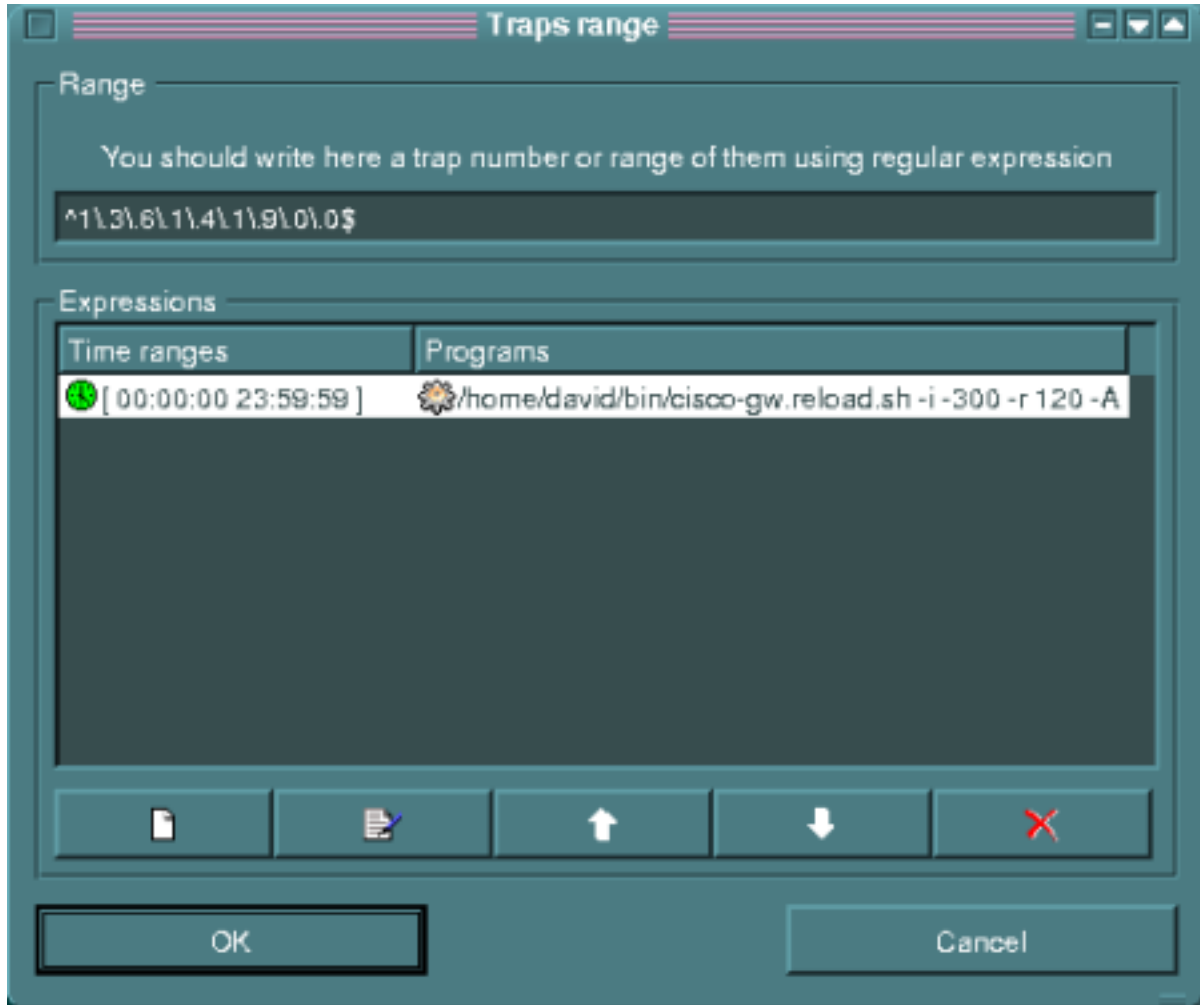







Tabela 11.2. Opis przycisków okienka Traps range

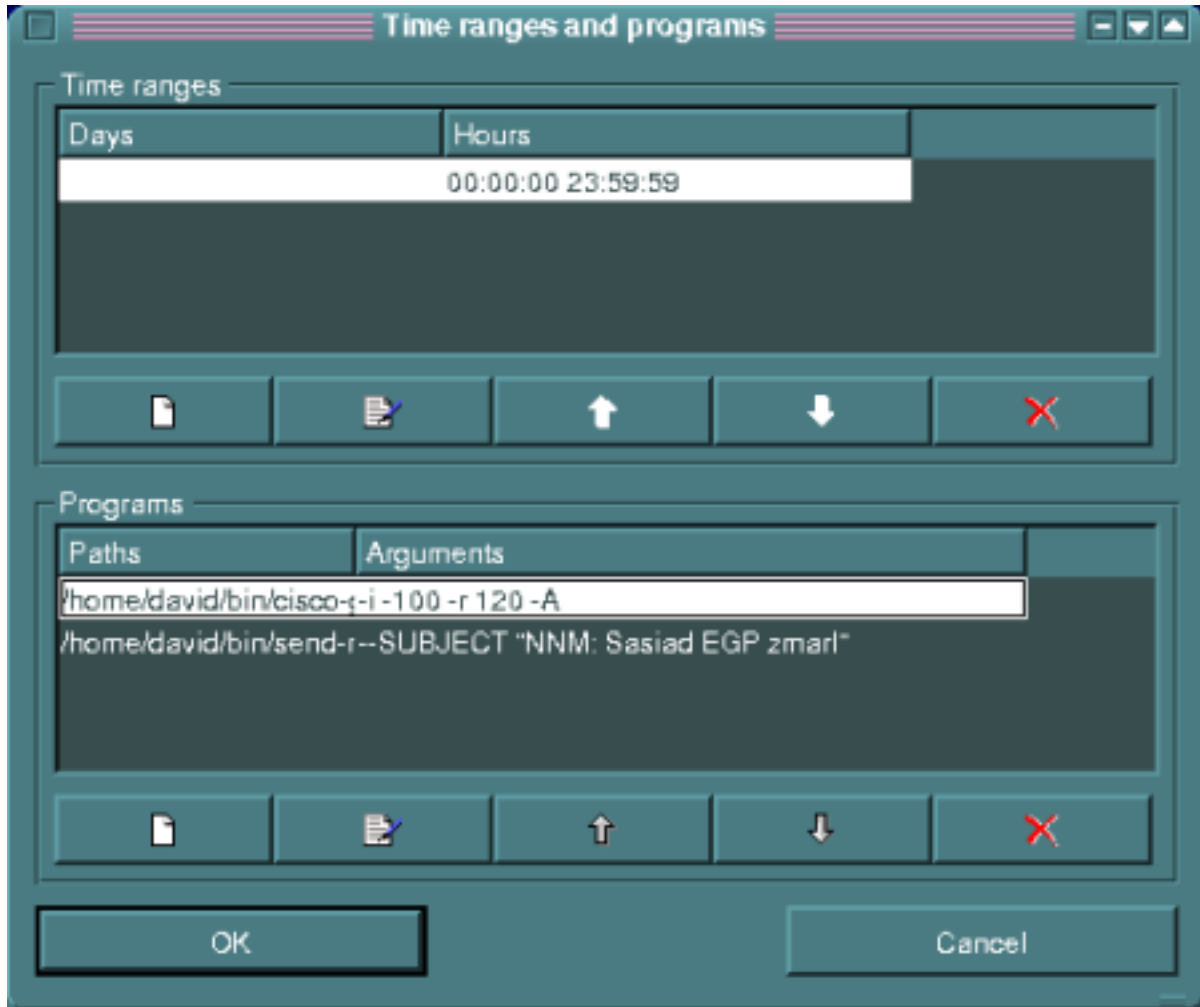
Przycisk	Opis
	Pozwala dodać nowy element do listy.
	Pozwala na konfigurację zaznaczonego programu lub jego zakresu czasowego.
	Pozwala zmienić kolejność wskazywanego elementu (przesunąć go do góry).
	Pozwala zmienić kolejność wskazywanego elementu (przesunąć go do dołu).
	Umożliwia usunięcie wskazanego elementu listy..

### 11.3.2. Konfiguracja zaznaczonych programów i ich zakresów czasowych

Za pomocą przycisku można dokonać konfiguracji programów i ich zakresów czasowych. Do tego służy





okienko dialogowe `Time ranges and programs`, które podzielone jest na dwie części. Każda z nich zawiera listę elementów, które z kolei podzielone są na dwie kolumny. Górna lista zawiera dni tygodnia i zakres godzin. Dolna zaś zawiera ścieżki do programów i ich argumenty.


### 11.3.2.1. Znaczenie przycisków funkcyjnych



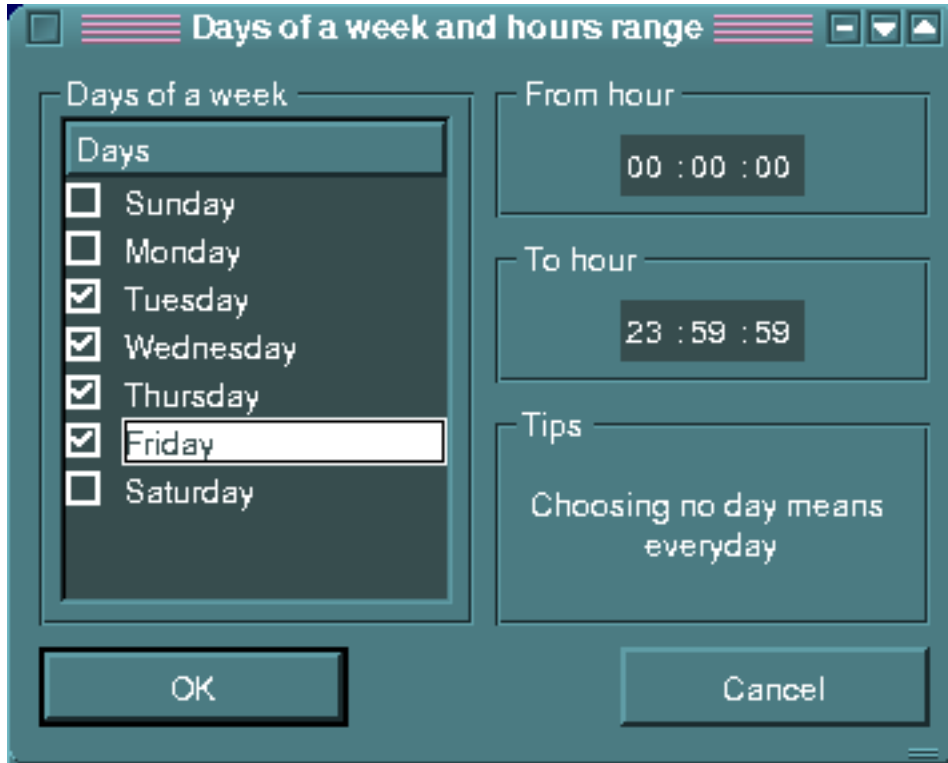
Każdy z elementów obu list można edytować za pomocą niżej opisanych przycisków.

**Tabela 11.3. Opis przycisków okienka `Time ranges and programs`**

Przycisk	Opis
	Pozwala dodać nowy element do listy.
	Pozwala na konfigurację specyfikującego zakres czasowy oraz na edycję elementu określającego program.
	Pozwala zmienić kolejność wskazywanego elementu (przesunąć go do góry).
	Pozwala zmienić kolejność wskazywanego elementu (przesunąć go do dołu).

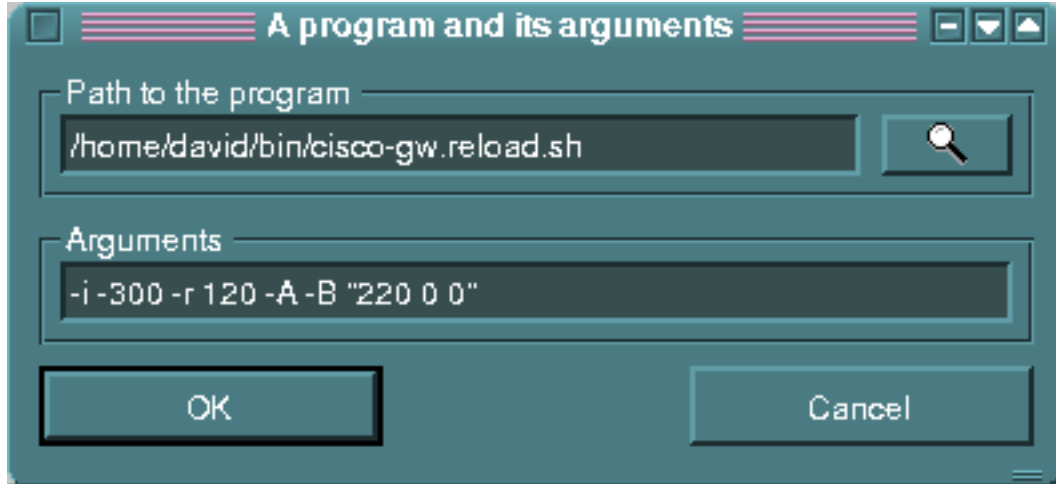
Przycisk	Opis
	Umożliwia usunięcie wskazanego elementu listy.


### 11.3.2.2. Edycja elementu specyfikującego zakres czasowy



Wybierając edycję elementu specyfikującego zakres czasowy, uruchamiamy następane okienko dialogowe Days of a week and hours range, gdzie możemy zaznaczyć, w które dni tygodnia i między którymi godzinami mają być uruchamiane podane programy. Jeżeli chcemy, aby zakres czasowy miał znaczenie w przypadku edytowanego wyrażenia to przynajmniej musimy wybrać zakres godzinowy. Brak jakiegokolwiek zakresu czasowego spowoduje, że wskazane programy będą mogły być uruchamiane każdego dnia o dowolnej porze. Natomiast nie zaznaczenie żadnego dnia tygodnia jest równoważne z zaznaczeniem ich wszystkich.

### 11.3.2.3. Edycja elementu określającego program



Edytując element określający program , który ma być uruchamiany, mamy możliwość podania ścieżki do niego oraz argumentów, które zostaną mu przekazane podczas wywołania. Przycisk  umożliwia wybranie pliku zamiast bezpośredniego wpisywania w miejscu edycji.

## 11.4. Tematy pokrewne

[Segregator Typów Zdarzeń \(des\)](#)










---

# Rozdział 12. Przyciski najczęściej występujące w aplikacjach WWW

## 12.1. Znaczenie przycisków

W poniższej tabeli zostały zebrane przyciski, które w aplikacjach WWW występują najczęściej. Ich funkcja w poszczególnych aplikacjach jest zbliżona, a czasem nawet identyczna. Niektóre z nich mogą jednak pełnić dodatkowe funkcje, które przy okazji omawiania poszczególnych aplikacji.

**Tabela 12.1. Przyciski najczęściej występujące w aplikacjach WWW**





Przycisk	Opis
	Powoduje powrót do widoku poprzedniej strony.
	Generalnie powoduje usunięcie elementu tzn. np: powoduje zamknięcie sprawy (case), ustawienie zdarzenia (event) w stan pasywny itp.
	Podwołuje przejście do edycji danego elementu.
	Najczęściej oznacza zatwierdzenie operacji i przejście do jej wykonania (np.: wygenerowanie raportu używając wybranych kryteriów).
	Powoduje przejście do widoku szczegółowego.
	Pozwala przejść do wyższego poziomu w hierarchii elementów.
	Otwiera nowe okno z danymi przygotowanymi do wydruku.
	Powoduje przejście do prezentacji wykresu z danymi dla danego elementu ( <a href="#">Przeglądarka Kolekcji</a> ).
	Odświeża widok strony.
	Powoduje akceptację zmienionych wartości jako aktualnych.



---

## Przyciski najczęściej występujące w aplikacjach WWW

---

Przycisk	Opis
	Powoduje przejście do raportu dla danego elementu ( <a href="#">Przeglądarka Raportów o Węzłach</a> ).
	Powoduje przejście do przeglądarki raportów o Trapach dla danego elementu ( <a href="#">Przeglądarka Trapów</a> ).
	Powoduje przejście do przeglądarki raportów o sprawach (cases) dla danego elementu ( <a href="#">Przeglądarka Zarejestrowanych Spraw</a> ).
	Zachowuje zmiany dokonane przez użytkownika.

---

# Rozdział 13. Przeglądarka Trapów (Trap Browser)

## 13.1. Opis ogólny

**Przeglądarka Trapów** jest aplikacją WWW oraz częścią **Notification Processor-a**. Pozwala ona przeglądać otrzymane komunikaty SNMP Trap od monitorowanych urządzeń.


## 13.2. Opis działania

### 13.2.1. Specyfikacja kryteriów wyszukiwania

#### 13.2.1.1. Widok domyślny aplikacji

**Przeglądarka Trapów** jest dostępna poprzez zakładkę Reports. Jest to grupa Traps tej zakładki. Podając kryteria wyboru w grupie Traps dostajemy listę zarejestrowanych powiadomień.

#### 13.2.1.2. Grupa Traps

**Traps** 

Source IP

Agent IP

SNMP community

Generic

Specific

OID

Enterprise

Text

ID

Community

Source device

Show/hide matched

Time range

Date and period

Results per page

Each bar occupies

Graph's width

Graph's height

Pola zawarte w grupie oraz ich znaczenie zostały przedstawione w poniższej tabeli.

**Tabela 13.1. Przeglądarka Trapów - znaczenie pól widoku szczegółowego grupy Traps**

Pole	Znaczenie
Source IP	IP urządzenie - nadawcy komunikatu
Agent IP	IP agenta - nadawcy komunikatu
SNMP community	SNMP community komunikatu
Generic	Ogólny numer komunikatu (dotyczy tylko komunikatów SNMPv1).
Specyfic	Szczegółowy numer komunikatu (dotyczy tylko komunikatów SNMPv1).
OID	Numer komunikatu (dotyczy tylko komunikatów SNMPv2C).
Enterprise	Identyfikator dostawcy urządzenia.

## Przeglądarka Trapów (Trap Browser)

<b>Pole</b>	<b>Znaczenie</b>
Text	Znaczenie komunikatu czytelne dla człowieka.
ID	Identyfikator tekstowy komunikatu czytelny dla człowieka.
Community	Etykieta nadana komunikatowi w trakcie przetwarzania (etykiety tekstowe oddzielone znakiem ':').
Source device	Urządzenie będące nadawcą komunikatu.
Show/hide matched	Pole to oznacza, czy komunikaty spełniające kryteria wyszukiwania mają zostać pokazane, czy ukryte.
Time range	Specyfikacja zakresu czasowego poprzez wybranie odcinka czasu trwającego do chwili obecnej.
Date and period	Specyfikacja zakresu czasowego poprzez wybranie konkretnej daty i czasu w połączeniu z określonym odcinkiem czasu (np.: 1 dzień, 2 dni, 1 tydzień itp.).
Results per page	Określa maksymalną ilość wpisów pokazanych za jednym razem.
Each bar occupies	Wskazanie ile czasu ma obejmować pojedynczy słupek na wykresie rozkładu występowania trapów w czasie.
Graph's width	Szerokość wykresu.
Graph's height	Wysokość wykresu.

### 13.2.2. Wygenerowany raport



Widok raportu jest podzielony na dwie części. Górna część zawiera listę wpisów, zaś dolna część pokazuje wykres słupkowy będący wizualizacją rozkładu występowania wybranych komunikatów w czasie.

Lista jest podzielona na kolumny oznaczające czas otrzymania komunikatu, urządzenie źródłowe komunikatu, wiadomość oraz identyfikator komunikatu (OID). Kolumna *Source* zamiast adresu IP nadawcy może zawierać link do **Przeglądarki Węzłów** ukazującej nadawcę komunikatu. Kolumny *Message* oraz *OID* zawierają linki pozwalające przejść do widoku szczegółowego danego wpisu. Nad wierszem z opisem kolumn mogą znajdować się przyciski pozwalające przeglądać całą listę wybranych komunikatów SNMP Trap.

## 13.3. Tematy pokrewne

[Segregator Typów Zdarzeń \(des\)](#)

[Odbiorca Powiadomień SNMP Trap \(dtrapd\)](#)

[Rejestrator Wiadomości \(dsi\)](#)

**Network Manager:** [Przeglądarka Węzłów \(Node Browser\)](#)