

Harmonogram zadań systemu Windows.

Harmonogram zadań pełni istotną rolę każdego systemu operacyjnego. Za jego pomocą można wykonywać następujące czynności systemowe:

- uruchamiać programy o określonych godzinach i/lub okresach (np. każdego 5 dnia miesiąca wyrzucać komunikat przypominający o sprawdzeniu raportów)
- uruchamiać regularnie programy do sprawdzania kondycji systemu operacyjnego oraz określonego sprzętu komputerowego
- konfigurowania usług i innych zadań systemu operacyjnego
- wyłączenie/ponowne uruchamianie komputera przy określonych warunkach
- wykonywanie określonych skryptów przy spełnieniu postawionych wymagań

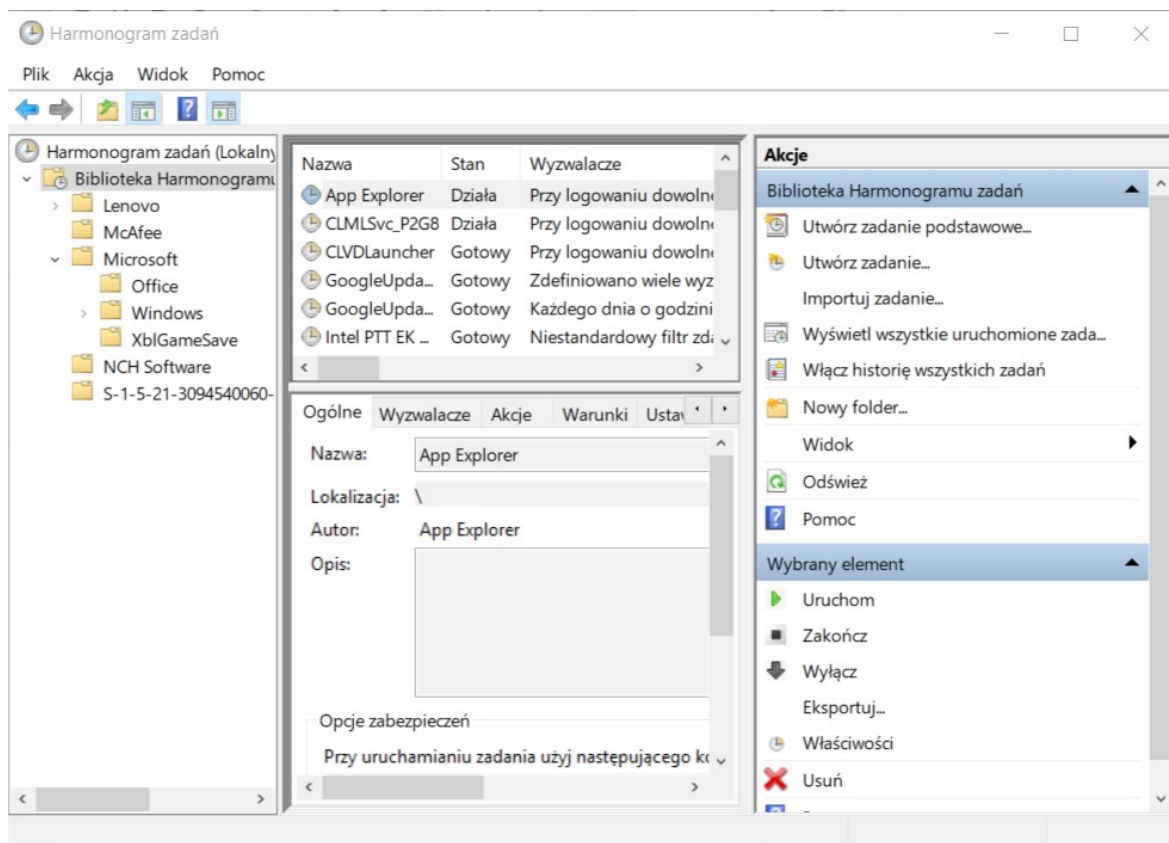
Harmonogram można także wykorzystać chociażby przy nadawaniu określonych uprawnień do uruchamiania programów i skryptów (alternatywa dla poświadczeń systemu Windows).

## 1. Obsługa harmonogramu zadań.

Najprostszą i najbardziej intuicyjną obsługą będzie zarządzanie przez:

- nazwę Harmonogram zadań (Scheduled Taks); wersja graficzna
- program schtasks.exe; wersja tekstowa
- przystawkę taskschd.msc; wersja graficzna
- polecenie PowerShell <akcja>-ScheduledTask\*, gdzie akcja to np. Get/Set/Show, ScheduledTask to człon uniwersalny każdego polecenia dotyczącego harmonogramu zadań, gwiazda natomiast zastępuje ostatni człon nazwy polecenia jak np. Principal będzie pozwalało na ustawienie właściciela zadania, Action wywołwaną/wywoływane akcje, zaś Trigger za wyzwalacze (opis i przykład tworzenia zadania przez PowerShell można znaleźć w materiale PowerShell2.pdf)

Poniżej, wraz z krótkim omówieniem, zostanie przedstawiona wersja GUI harmonogramu. Ogólny widok na harmonogram zadań:



Możliwe działania (menu po prawej stronie okna):

- Utwórz zadanie podstawowe... - tworzy zadanie poprzez bardzo proste narzędzie. Większość poleceń jest szczegółowo opisana:

Kreator tworzenia zadań podstawowych

Utwórz zadanie podstawowe

Ten kreator umożliwi szybkie zaplanowanie typowego zadania. Aby uzyskać bardziej zaawansowane opcje lub ustawienia, takie jak wiele akcji lub wyzwalaczy zadań, użyj polecenia Utwórz zadanie w okienku akcji.

**Utwórz zadanie podstawowe**

Wyzwalacz

Akcja

Zakończ

Nazwa: Przykład zadania

Opis: Opcjonalny opis

< Wstecz    Dalej >    Anuluj

Należy podać nazwę (obowiązkowo) oraz opis (który nie jest wymagany).

Kreator tworzenia zadań podstawowych

Wyzwalacz zadania

Utwórz zadanie podstawowe

Wyzwalacz

Codziennie

Akcja

Zakończ

Kiedy ma być uruchamiane zadanie?

Codziennie

Cotygodniowo

Comiesięcznie

Jeden raz

Przy uruchamianiu komputera

Przy logowaniu

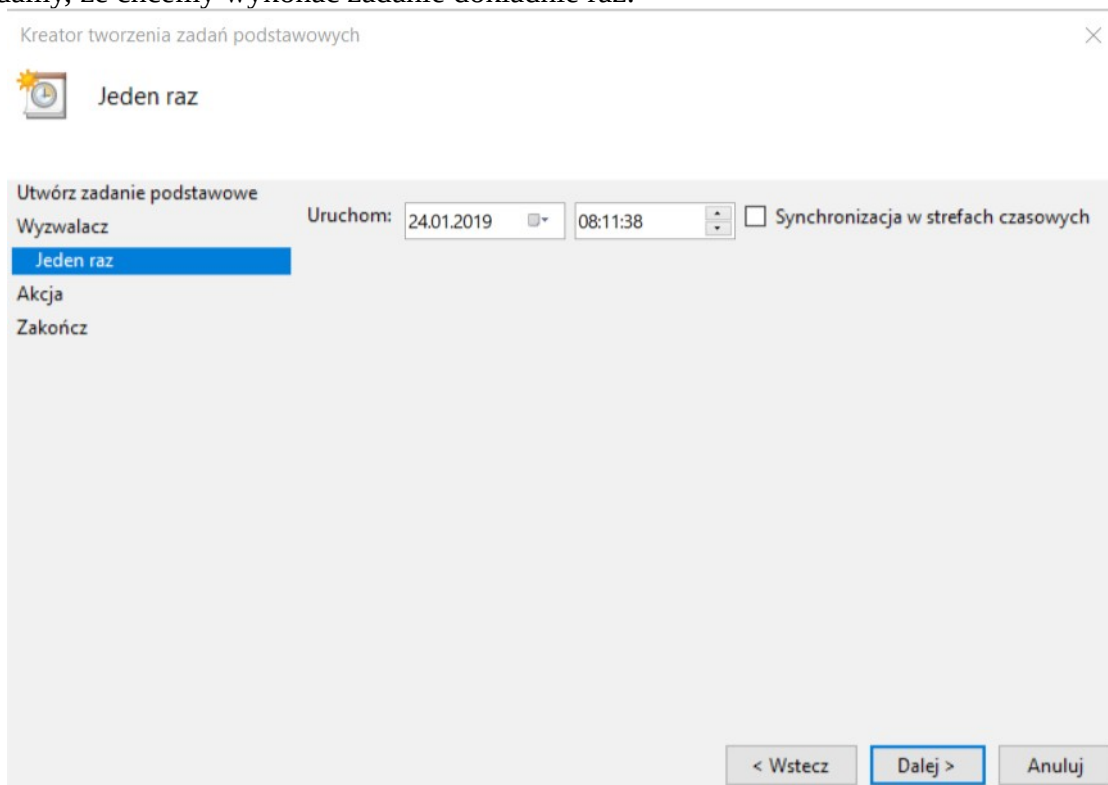
Przy rejestrowaniu określonego zdarzenia

< Wstecz    Dalej >    Anuluj

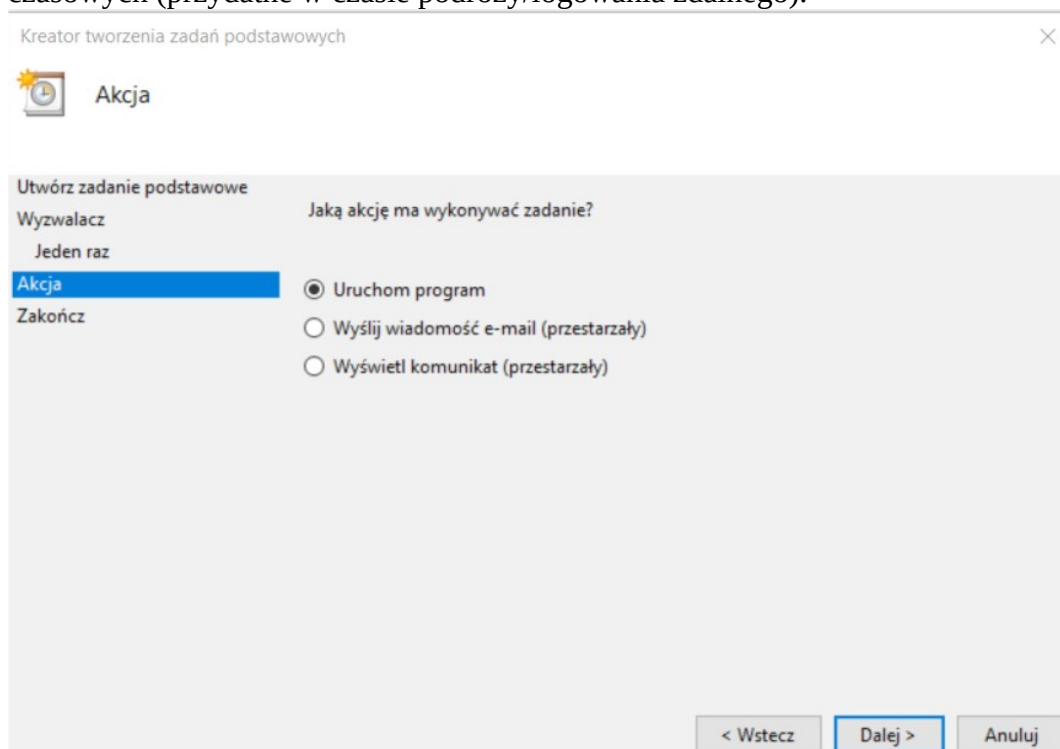
Wybieramy reakcję, która ma spowodować uruchomienie zdarzenia. W przypadku kreatora nie możemy wybrać więcej niż jednego rodzaju wzbudzenia. Dostępne opcje:

- **Codziennie** – zadanie będzie uruchamiane codziennie o określonej porze (godzinie)
- **Cotygodniowo** – zadanie będzie się uruchamiać w określone dni tygodnia o określonej godzinie; można wybrać częstotliwość powtarzania (domyślnie co tydzień)
- **Comiesięcznie** – zadanie będzie wywoływać się w określone dni miesiąca o określonej godzinie; można wybrać określone miesiące
- **Jeden raz** – zadanie uruchomi się dokładnie jeden raz

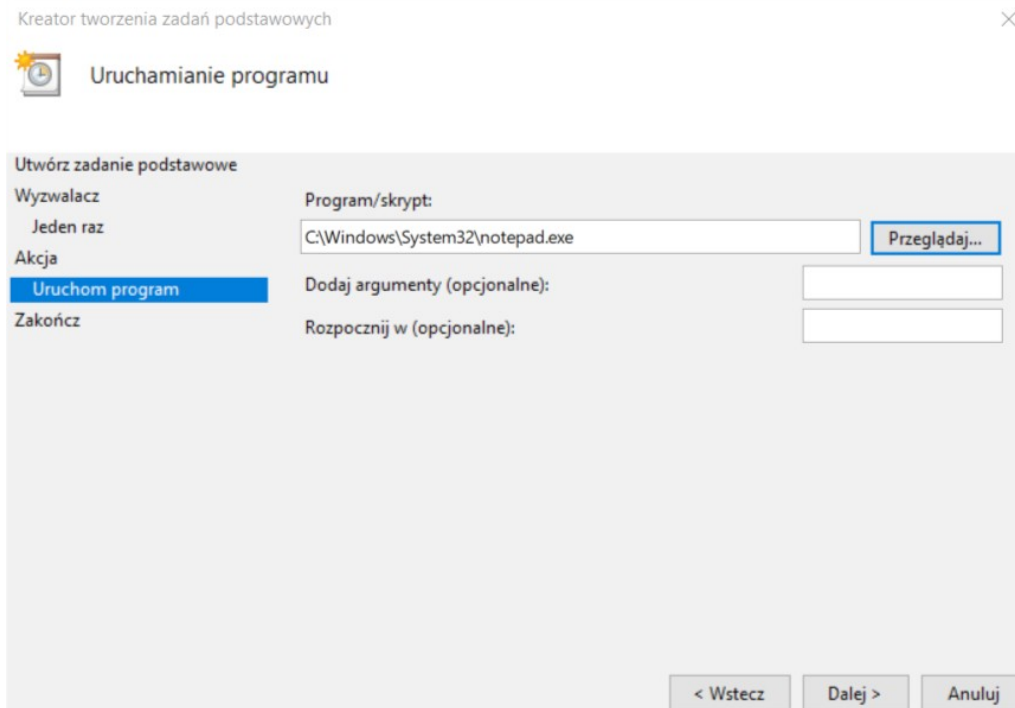
- **Przy uruchamianiu komputera** – zadanie będzie wywoływane zawsze przy starcie systemu operacyjnego
  - **Przy logowaniu** – zadanie aktywuje się w chwili logowania dowolnego bądź wskazanego użytkownika
  - **Przy rejestrowaniu określonego zdarzenia** – akcja wykonuje się w chwili zarejestrowania przez system określonego zdarzenia systemowego (np. otwarcie aplikacji, informacja od aplikacji, informacja od jądra systemu itp.)
- Zakładamy, że chcemy wykonać zadanie dokładnie raz:



Teraz można wybrać datę uruchomienia oraz godzinę. Ponadto możliwe jest synchronizowanie w strefach czasowych (przydatne w czasie podróży/logowania zdalnego).



Możemy wybrać określoną akcję. Obecnie jedyną polecaną akcją jest uruchomić program (pozostałe są przestarzałe/nie obsługiwane).



Kreator tworzenia zadań podstawowych

U uruchamianie programu

Utwórz zadanie podstawowe

Wyzwalacz

Jeden raz

Akcja

Uruchom program

Zakończ

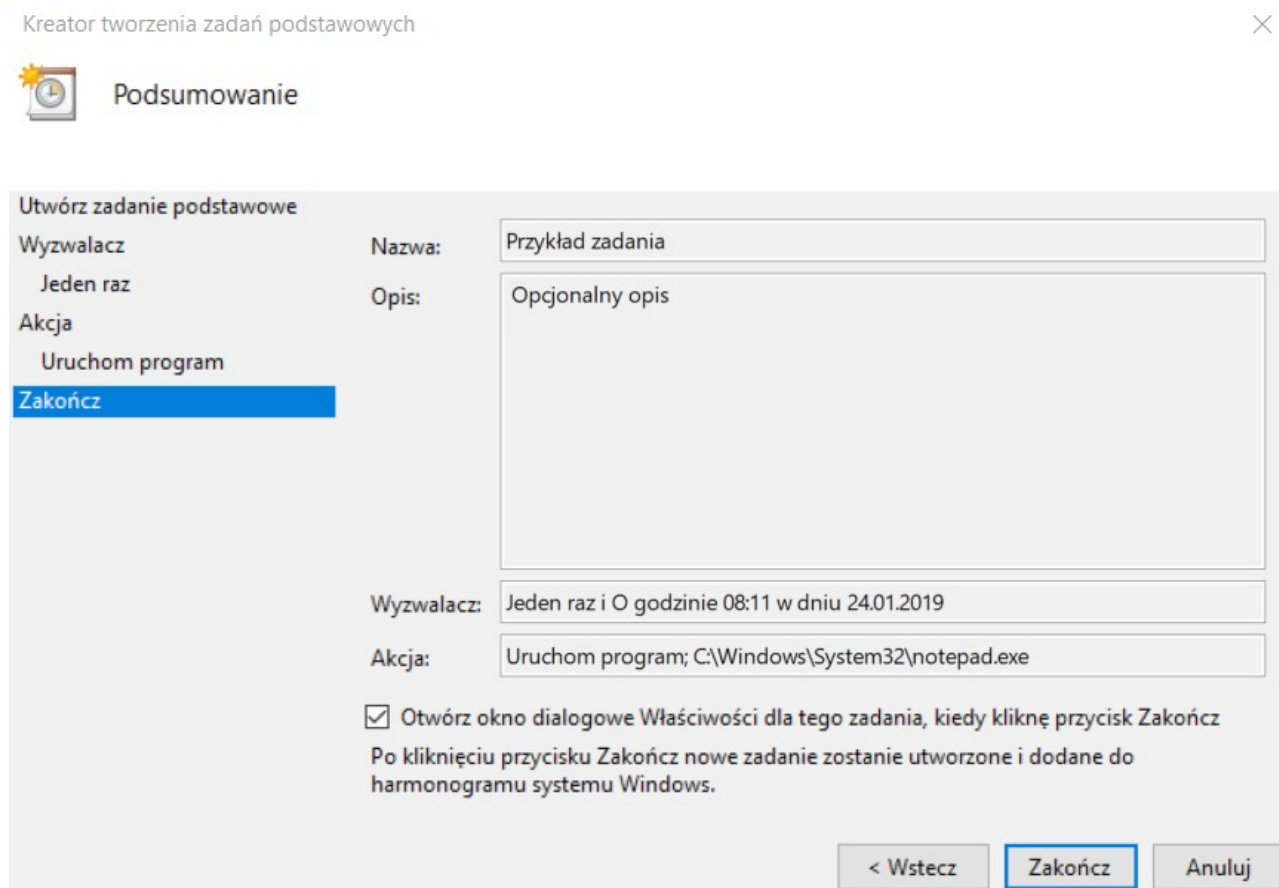
Program/skrypt: C:\Windows\System32\notepad.exe Przełóżaj...

Dodaj argumenty (opcjonalne):

Rozpocznij w (opcjonalne):

< Wstecz Dalej > Anuluj

Powyżej wybrano program notepad. Dodatkowo można podać argumenty uruchomienia (np. plik do otwarcia). 'Rozpocznij w' pozwala ustalić uruchamianej aplikacji ustawienie jej macierzystego folderu (tak by mogła czytać pliki jak by znajdowały się w jej folderze).



Kreator tworzenia zadań podstawowych

Podsumowanie

Utwórz zadanie podstawowe

Wyzwalacz

Jeden raz

Akcja

Uruchom program

Zakończ

Nazwa: Przykład zadania

Opis: Opcjonalny opis

Wyzwalacz: Jeden raz i O godzinie 08:11 w dniu 24.01.2019

Akcja: Uruchom program; C:\Windows\System32\notepad.exe

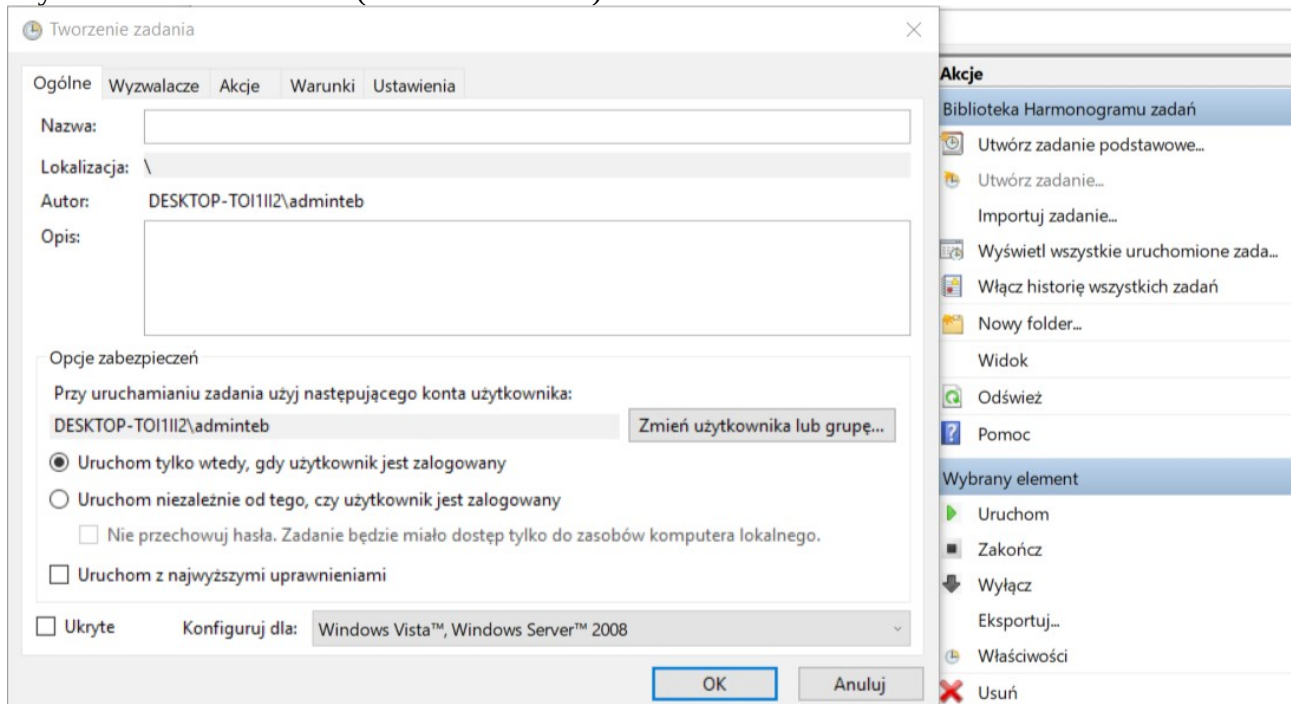
Otwórz okno dialogowe Właściwości dla tego zadania, kiedy kliknę przycisk Zakończ

Po kliknięciu przycisku Zakończ nowe zadanie zostanie utworzone i dodane do harmonogramu systemu Windows.

< Wstecz Zakończ Anuluj

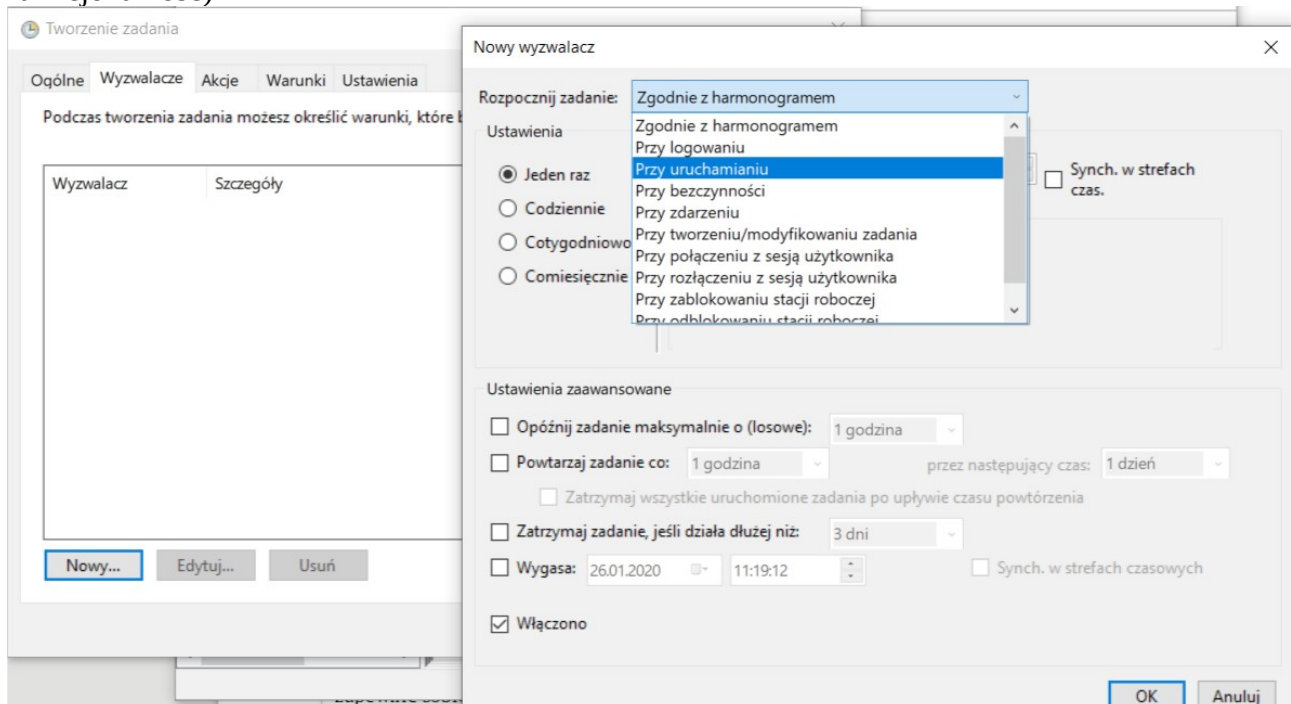
Okno podsumowuje utworzone zadanie. Można zaznaczyć dodatkową opcję by otworzyć okno właściwości utworzonego zadania.

Od tego momentu możemy edytować utworzone zadanie tym samym narzędziem, którego możemy użyć do tworzenia zadania (Utwórz zadanie...)



W tym widoku możemy ustawić więcej opcji niż w poprzednim. Poza nazwą i opisem możliwe jest też:

- zamienienie konta użytkownika, z którego uprawnieniami będzie działać zadanie
- wybranie sposobu, kiedy zadanie ma się uruchomić (przy zalogowaniu użytkownika bądź niezależnie od tego faktu)
- zachowanie (bądź nie) hasła użytkownika (w przypadku nie zachowania zadanie może się nie uruchomić poprawnie w przypadku, gdy potrzebne będą wyższe uprawnienia)
- zapewnić sobie uruchamianie zadania z najwyższymi uprawnieniami
- ukryć zadanie (dla pozostałych użytkowników)
- skonfigurować funkcjonalność zadania dla określonych systemów (domyślnie niższa funkcjonalność)



Druga zakładka pozwala dodać wyzwalacze, czyli zdarzenia, które mają aktywować zadanie. W przeciwieństwie do zadania podstawowego, można tutaj dodać więcej niż jeden wyzwalacz. Po kliknięciu na przycisk Nowy... wyskoczy nam nowe okno, w którym poszczególne opcje działają w następujący sposób:

- **Zgodnie z harmonogramem** – tutaj możemy decydować, czy zadanie ma się uruchomić dokładnie raz, codziennie, cotygodniowo, czy comiesięcznie.
- **Przy logowaniu** – uruchomienie następuje po zalogowaniu określonego użytkownika (bądź dowolnego)
- **Przy uruchamianiu** – proces wystartuje przy uruchomieniu systemu operacyjnego
- **Przy bezczynności** – zadanie wykona się w przypadku stwierdzonej przez system bezczynności użytkownika
- **Przy zdarzeniu** – analogicznie jak w zadaniu podstawowym, zadanie wykona się w przypadku określonego zdarzenia w dzienniku systemowym
- **Przy tworzeniu/modyfikowaniu zadania** – wywoła się zawsze, gdy ktokolwiek będzie tworzył lub modyfikował zadanie
- **Przy połączeniu z sesją użytkownika** – zadanie wykonuje się gdy określony użytkownik/użytkownicy logują się do systemu poprzez usługę RDP lub konsolę lokalną (zalogowanie/wznowienie sesji na pulpit)
- **Przy rozłączeniu z sesją użytkownika** – zadanie wykona się w chwili wylogowania użytkownika/użytkowników z sesji systemowej (lokalnej lub zdalnej)
- **Przy zablokowaniu stacji roboczej** – zadanie wykonuje się w przypadku gdy użytkownik bądź system zablokował komputer.
- **Przy odblokowaniu stacji roboczej** – zadanie wykona się gdy komputer zostanie przez kogoś odblokowany (zdalnie bądź lokalnie)

Opcja **Opóźnij zadanie o/Opóźnij zadanie maksymalnie o (losowo)** pozwala na wywołanie wskazanego zadania z określonym opóźnieniem. Dzięki temu można dać czas użytkownikowi bądź użytkownikom na określoną reakcję. Niekiedy tego typu działanie może być pożądane ze względu na nienadmierne obciążanie komputera uruchamianiem/zamykaniem określonych programów (bądź uruchamianiem ich w pewnej, określonej kolejności).

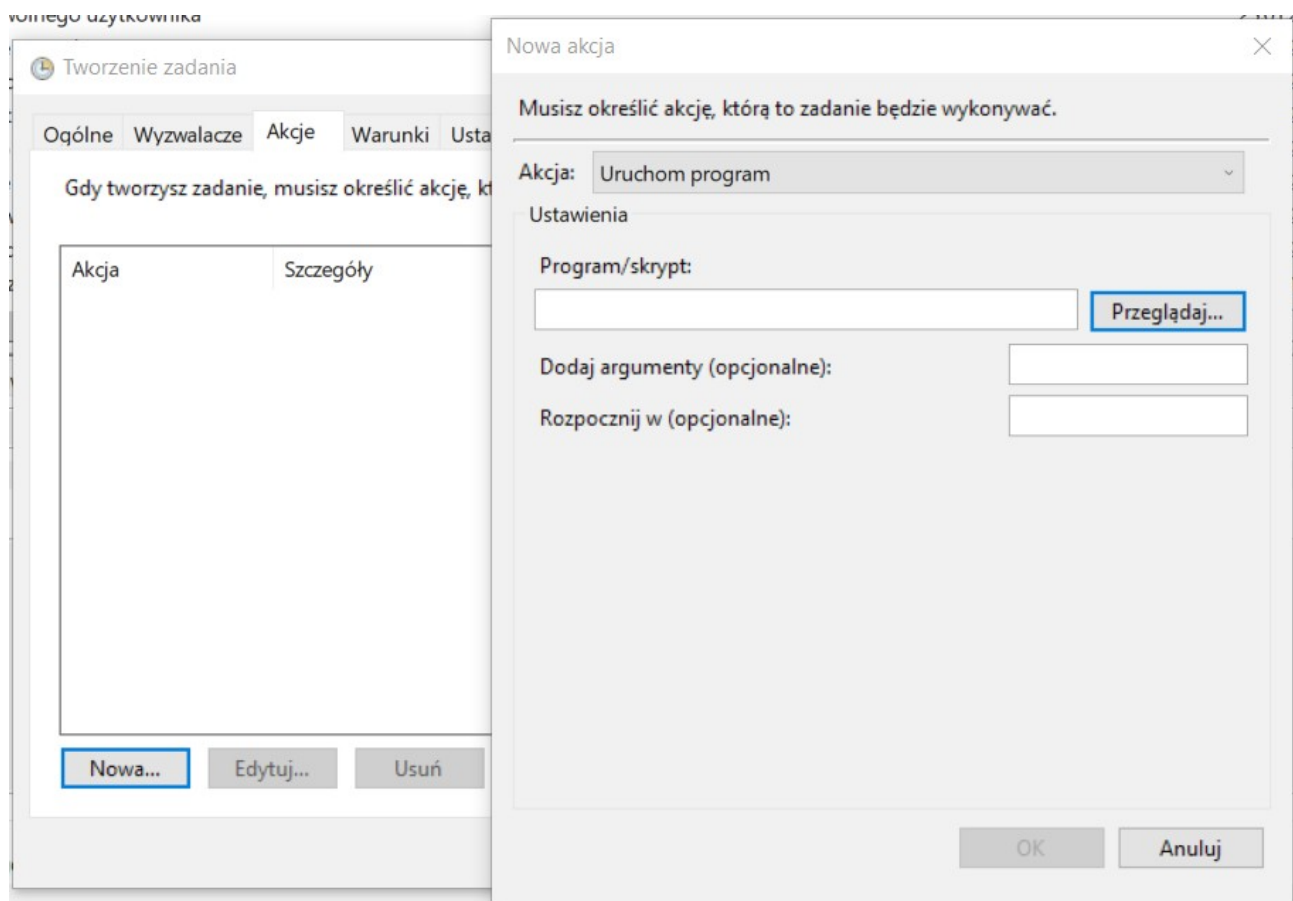
**Powtarzaj zadanie co** pozwala na wywołanie dokładnie tego samego zadania w kolejnych odstępach czasowych, powtarzanych przez określony czas. Przykładowo wywołanie zadania z określoną wiadomością dla użytkownika powinno być powtarzane co 1,5 godziny przez następne 10 godzin.

Dodatkowe pole, jakim jest **Zatrzymaj wszystkie uruchomione zadania po upływie czasu powtórzenia** pozwala zamknąć wszystkie zadania, które nadal pozostawały otwarte.

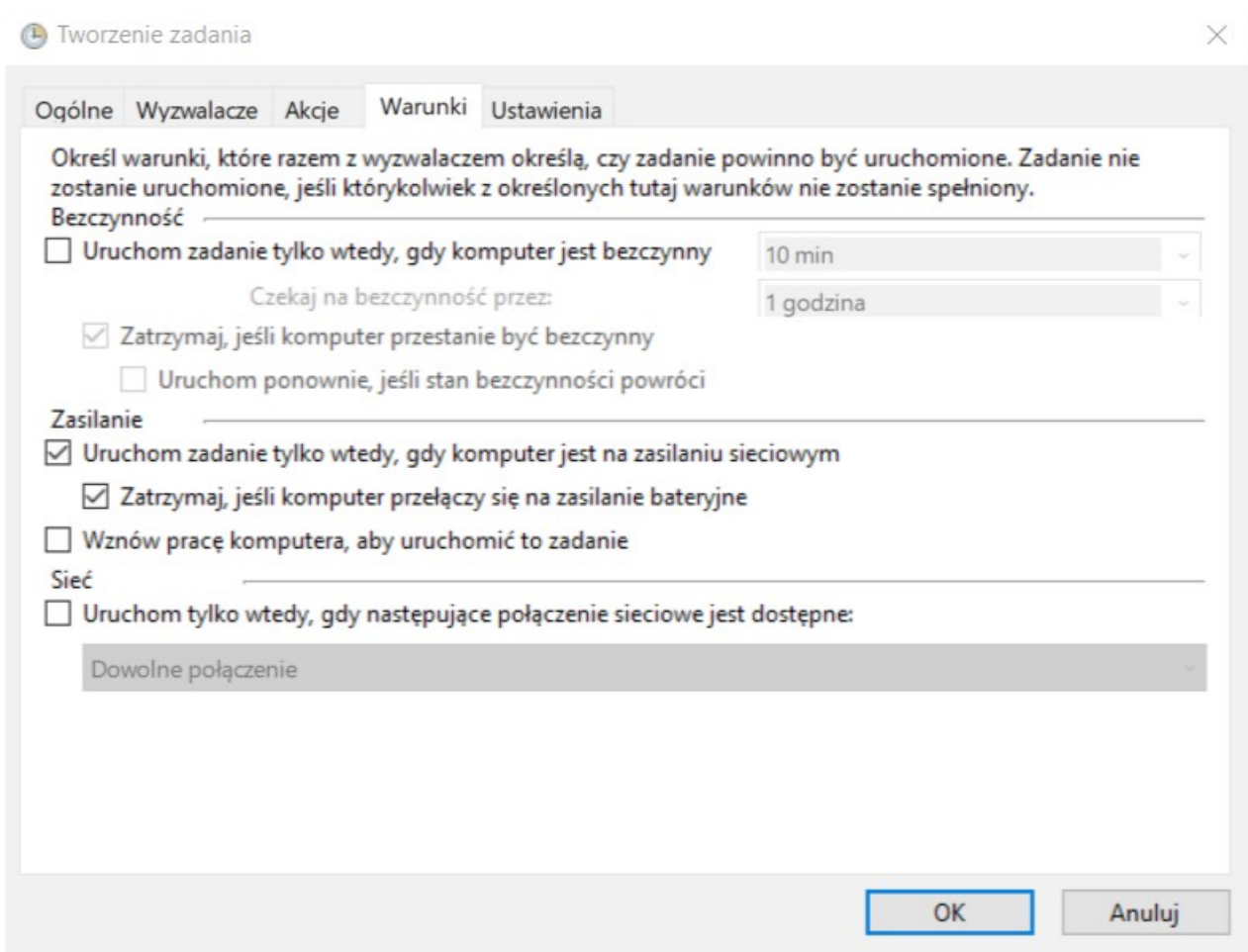
**Zatrzymaj zadanie, jeżeli działa dłużej niż** pozwala na bezwzględne zatrzymanie określonego zadania w przypadku, gdy działa ono dłużej niż przewidział jego twórca. Nierzadko zdarza się, że uruchomione zadanie widziane jest przez system Windows jako nadal aktywne, chociaż aplikacja/usługa, którą miało ono wywołać dawno przestała działać.

Pole **Wygasa** pozwala określić, po jakim czasie wygasa wskazany aktywator. Po tym czasie zadanie nie będzie więcej uruchamiane przez edytowane zdarzenie. Zaznaczenie opcji **Synch. w strefach czasowych** daje pewność, że zadanie wygaśnie o dokładnie wyznaczonej dacie (przydatne, gdy komputer będzie się przemieszczał, a jego właściciel będzie zmieniał datę na aktualną/system będzie o to dbał).

Zaznaczone pole **Włączono** gwarantuje, że wskazany aktywator będzie działać (inaczej zadanie nie będzie przez niego aktywowane).



Kolejną zakładką, Akcje, pozwala na wybranie uruchamianych programów i/lub skryptów w danym zadaniu. Można zmienić akcję na wysłanie poczty bądź wiadomości, jest to jednak przestarzała akcja (tego typu operacje powinny być realizowane poprzez odpowiednie skrypty). Wybrać można dowolny program i/lub skrypt uruchamiany w ramach zadania. Argumenty skryptu powinny być dodane w dodatkowym polu (np. uruchamiając polecenie shutdown parametry takie jak /f /s /r czy /t powinny być podane właśnie w tym polu). Rozpocznij w pozwala na zmianę katalogu domowego aplikacji. Przykładowo uruchamiamy konsolę cmd.exe, której katalog domowy to C:\Windows\System32; my jednak wywołujemy skrypt z folderu [C:\Skrypty](#) i potrzebujemy, by to on był katalogiem „domowym” dla cmd.exe, w związku z tym we wspomniane pole wpisujemy [C:\Skrypty](#).



Zakładka Warunki pozwala na ustalenie dodatkowych, wspólnych warunków uruchamiania i działania poszczególnych wyzwalaczy. Możemy np. zaplanować, że zadanie uruchomi się w przypadku, gdy komputer będzie stał beczynnie (nie będzie widział akcji od użytkownika) przez określony czas. Pole **Czekaj na beczynność przez** wskazuje, ile musi upłynąć czasu by system potraktował brak reakcji użytkownika jako beczynność. Dodatkowo można zaznaczyć **Zatrzymaj, jeżeli komputer przestanie być beczynny**. W tym momencie harmonogram przerwie działanie tworzonego zadania. Z kolei pole **Uruchom ponownie, jeżeli stan beczynności powróci** przywróci nam zadanie w przypadku, gdy system ponownie stwierdzi brak obecności aktywności jakichkolwiek aplikacji użytkownika (lub aplikacji zależnych).

**Uruchom zadanie tylko wtedy, gdy komputer jest na zasilaniu sieciowym** – aktywowanie tej opcji spowoduje, że system zainstalowany np. na laptopie, który nie będzie podłączony do sieci, zignoruje to zadanie. Zadanie jednak nie zostanie wstrzymane, jeżeli przy jego uruchamianiu komputer był na zasilaniu z sieci chyba, że zostanie zaznaczona opcja **Zatrzymaj, jeśli komputer przełączy się na zasilanie bateryjne**.

**Wznów pracę komputera, aby uruchomić to zadanie** wymusi na komputerze (poprzez ACPI) uruchomienie się poprzez wybudzenie tylko po to, by zadanie się wykonało. Opcja ta ma zastosowanie tylko do uśpionych i hibernowanych systemów operacyjnych.

Ostatnia opcja, **Uruchom tylko wtedy, gdy następujące połączenie sieciowe jest dostępne**, umożliwia zagwarantowanie, że zadanie zadziała tylko w przypadku, gdy system będzie połączony do określonej sieci. Opcja przydatna w przypadku, gdy komputer może być przenoszony, jednak zadanie wymaga uruchomienia skryptu bądź programu np. w biurze bądź wtedy, gdy jesteśmy w delegacji w innym oddziale.

Tworzenie zadania

Ogólne Wyzwalacze Akcje Warunki Ustawienia

Określ dodatkowe ustawienia wpływające na zachowanie zadania.

Zezwalaj na uruchamianie zadania na żądanie

Uruchom szybko zadanie, jeśli pominięto zaplanowane uruchomienie

Po błędzie uruchom ponownie co: 1 minut

Spróbuj uruchomić nie więcej niż: 3 razy

Zatrzymaj zadanie, jeśli działa dłużej niż: 3 dni

Jeśli zadanie nie zakończy się na żądanie, wymuś zatrzymanie

Jeśli zadanie nie jest ponownie zaplanowane, usuń je po: 30 dni

Jeśli zadanie jest już uruchomione, zastosuj następującą regułę:

Nie uruchamiaj nowego wystąpienia

OK Anuluj

Ostatnia zakładka pozwala na:

- Zezwalaj na uruchamianie zadania na żądanie – brak zaznaczenia tej opcji spowoduje, że zadanie nie będzie mogło być np. uruchamiane poprzez opcję Uruchom lub poprzez skrót do (możliwe jest robienie skrótów wyciągniętych np. na pulpit)
- Uruchom szybko zadanie, jeżeli pominięto zaplanowane uruchomienie – normalnie zadanie nie wykona się, jeżeli wyzwalacz, który miał je wywołać z jakiegoś powodu nie zadziałał. Po zaznaczeniu tej opcji zadanie wykona się w takim przypadku natychmiastowo
- Po błędzie uruchom ponownie co – można określić ile razy system ma podejmować próbę uruchomienia edytowanego zadania jeżeli pierwotne wywołanie spowodowało błąd.
- Zatrzymaj zadanie, jeżeli działa dłużej niż – opcja pozwala na niezależne zatrzymanie zbyt długo działającego zadania, niezależnie w jaki sposób zostały skonfigurowane poszczególne wyzwalacze.
- Jeżeli zadanie nie zakończy się na żądanie, wymuś zatrzymanie – gwarantuje, że zadanie zostanie bezwzględnie zatrzymane przez system
- Jeśli zadanie nie jest ponownie zaplanowane, usuń je po – jeżeli mamy pewność, że tworzone zadanie po użyciu nie będzie nam już więcej potrzebne, możemy zlecić jego usunięcie. Pozwoli to na lepsze uporządkowanie harmonogramu oraz zapobiegnie skopiowaniu zadania przez niepowołane osoby.
- Jeśli zadanie jest już uruchomione, zastosuj następującą regułę – pole pozwala zmienić zachowanie w przypadku wykrycia, że zadanie zostało już uruchomione. Domyślnie nowa instancja (nowe zadanie) nie zostanie uruchomione. Można wybrać kolejgowanie zadania – po zakończeniu uruchomionego zadania uruchomi się następne; zatrzymać obecnie wykonywane zadanie (zamiast niego uruchomi się nowe zadanie) bądź można uruchomić nowe zadanie równoległe (jako nową, niezależną instancję).

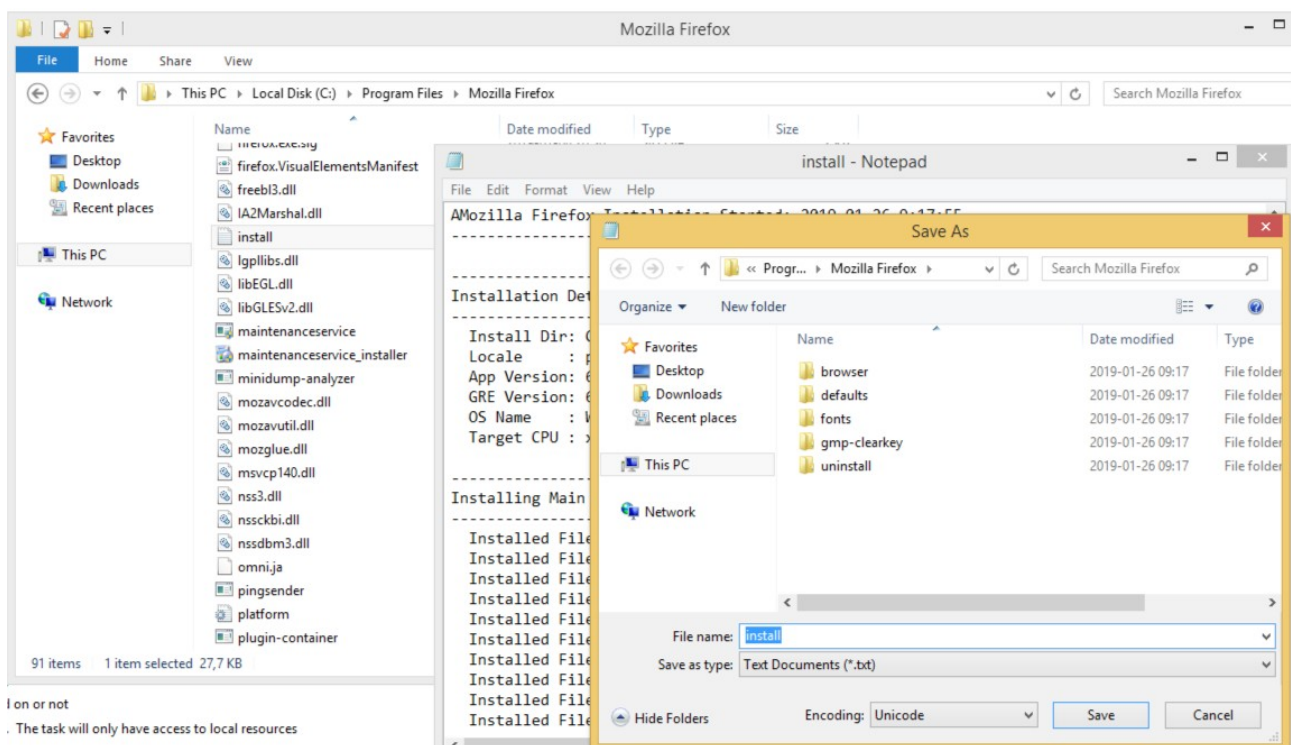
## 2. Przykłady wykorzystania harmonogramu zadań.

a) jako skrót do uruchamiania programów z innymi uprawnieniami

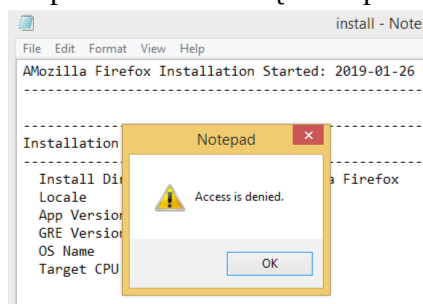
Jednym z ciekawszych zastosowań jest możliwość uruchamiania aplikacji z uprawnieniami innego użytkownika niż jest zalogowany w rzeczywistości. Rozwiązanie to pozwala na korzystanie np. z aplikacji w trybie administratora osobom, które nie znają odpowiedniego hasła (i nigdy nie powinny). Choć z pozoru może wydawać się to mało rozsądne to istnieją programy, które wymagają okresowej (nawet codziennej) aktualizacji, z czego każda z nich wymaga uprawnień administracyjnych. W tym wypadku, jeżeli jest to większa firma (posiadająca powyżej 10 komputerów) ręczne uruchamianie aplikacji przez administratora mogłoby być kłopotliwe.

**UWAGA!** Oczywiście powyższe rozważanie jest czysto hipotetyczne. W praktyce administrator powinien zadbać o aktualizacje z serwera zdalnego, odpowiednie skrypty lub instalację oprogramowania umieścić poza folderami systemowymi (instalować jako oprogramowanie dedykowane dla użytkownika).

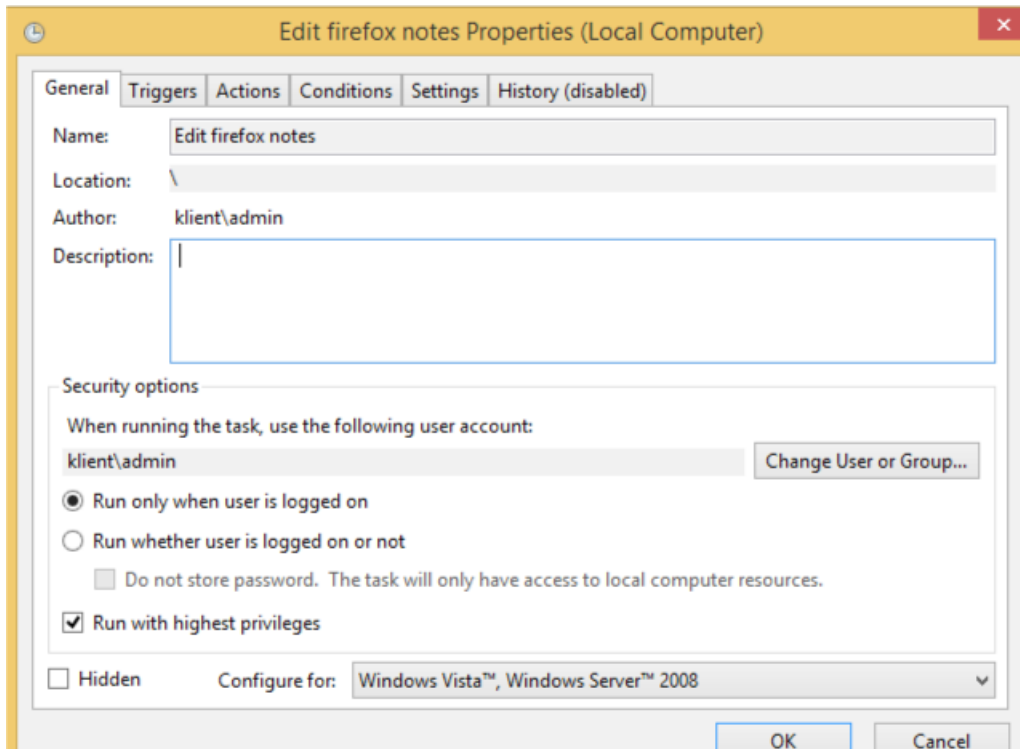
Domyślnie użytkownik nie ma możliwości edytowania plików dostępnych w Program Files (może je edytować tylko administrator). W związku z tym utworzymy zadanie, które będzie otwierać edycję pliku instalacyjnego Firefox. W normalnych warunkach, uruchamiając plik poprzez kliknięcie na niego (nawet jako użytkownik z grupy administratorzy) będziemy mieć odmowę zapisu zmian w oryginalnej lokalizacji:



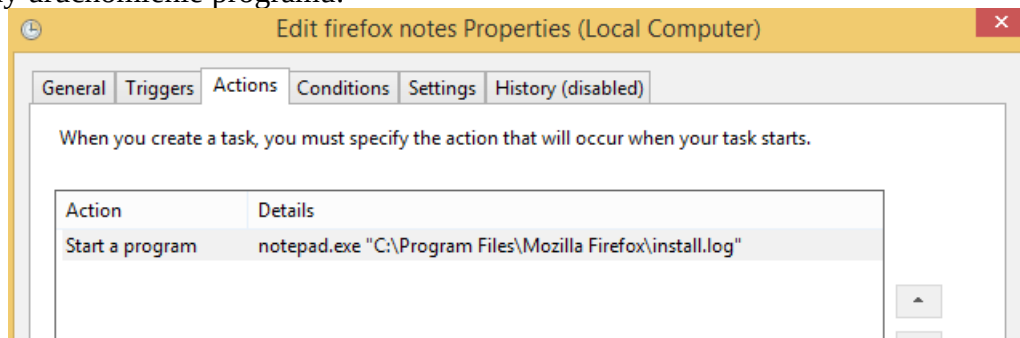
Jak widać ze zrzutu, nastąpi próba nadpisania pliku install.log w jego lokalizacji domowej (Program Files). Po kliknięciu Save oraz potwierdzeniu chęci nadpisania otrzymamy komunikat:



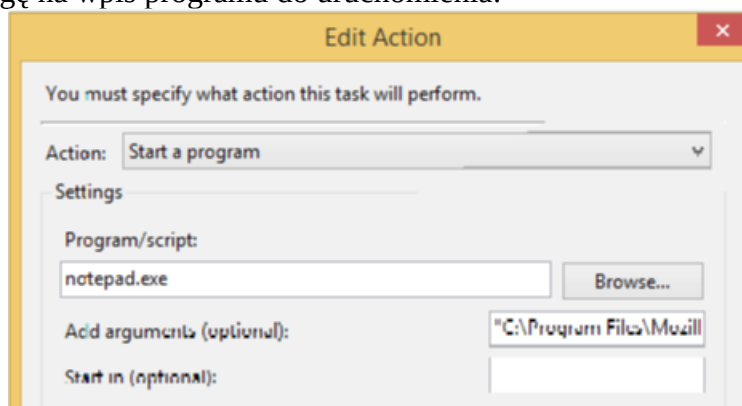
Teraz sprawdźmy jak poradzi sobie następujące zadanie:



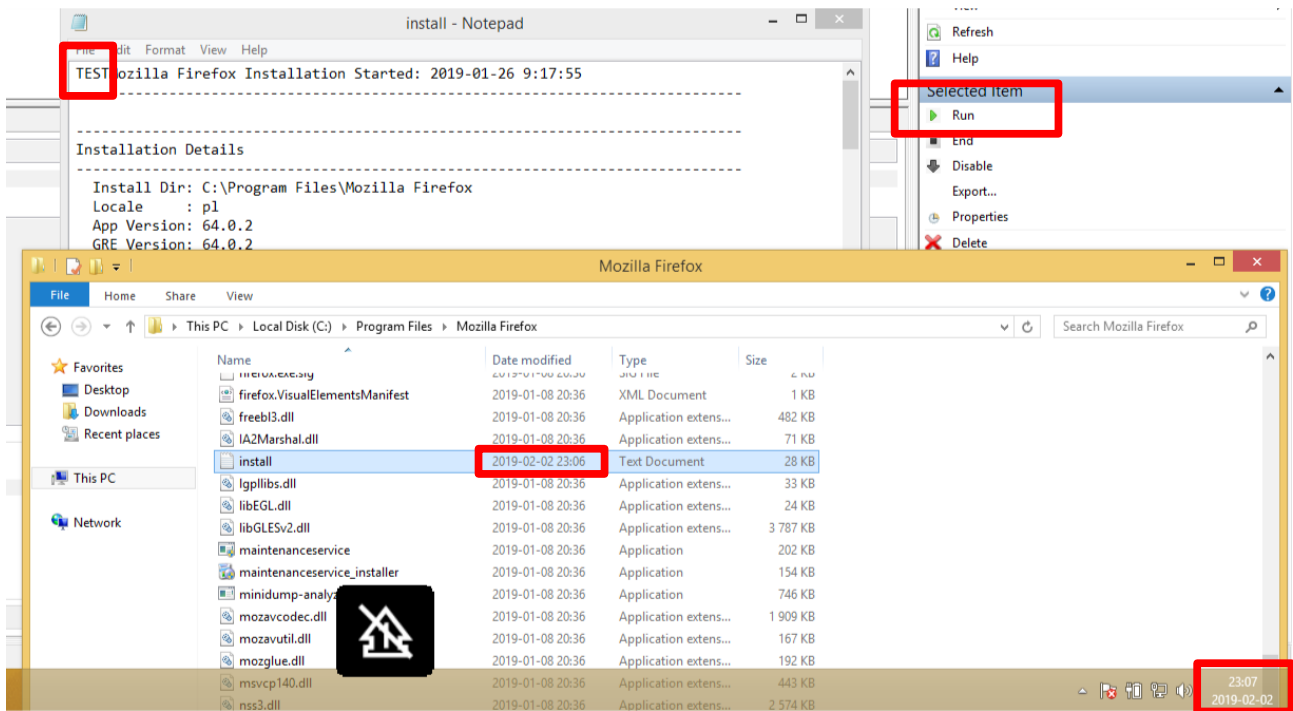
Najważniejsze to zaznaczenie opcji „Run with highest privileges”. Teraz z zakładki Actions dokładamy uruchomienie programu:



Należy zwrócić uwagę na wpis programu do uruchomienia:

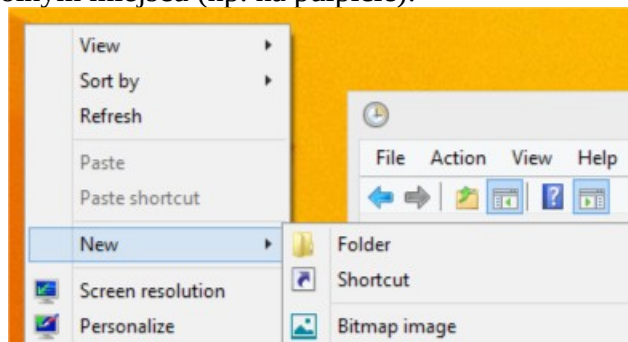


Ścieżka pliku do edycji jest podana jako parametr (nie zaś w nazwie programu). Teraz wystarczy zapisać zadanie i spróbować je uruchomić (opcja Run). Po dokonaniu zmian i kliknięciu Zapisz nasz plik powinien się nadpisać bez żadnego problemu:

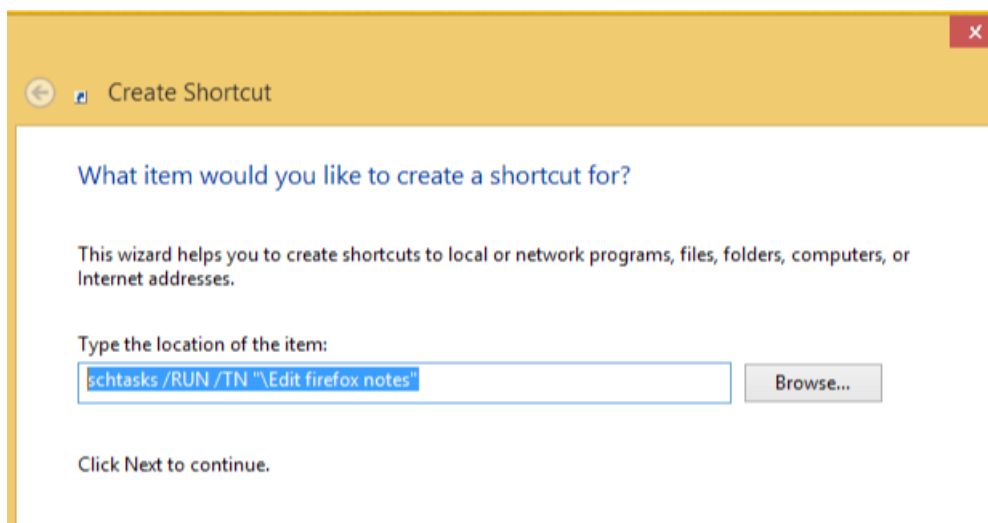


Jak widać na zrzucie powyżej po uruchomieniu zadania dodany został napis TEST i zapisany. Data modyfikacji jest aktualna (w przeciwieństwie do dat instalacji przeglądarki) co potwierdza, że udało się zapisać plik w lokalizacji pierwotnie niedostępnej.

W celu uruchamiania takiego zadania bez wchodzenia do harmonogramu można utworzyć odpowiedni skrót w dowolnym miejscu (np. na pulpicie):

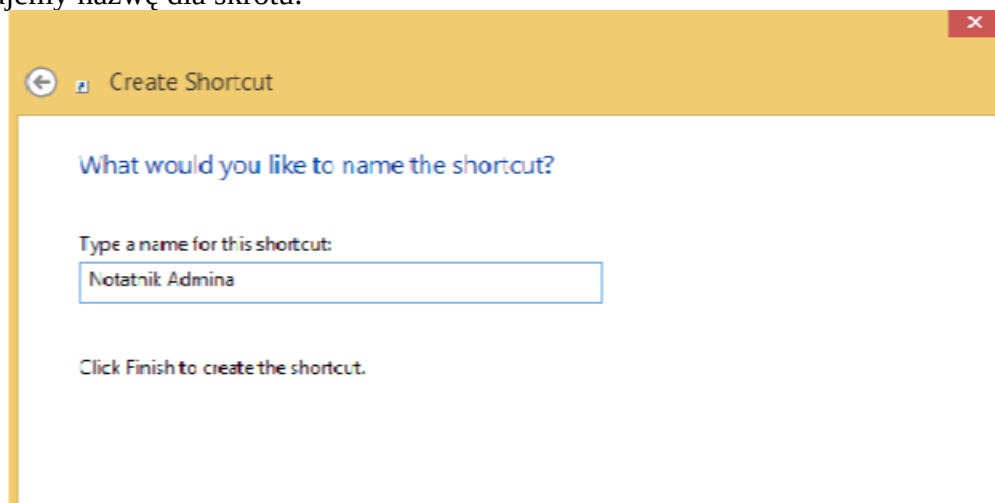


Wpisujemy odpowiednie polecenie:



Gdzie po /TN następuje pełna ścieżka do zadania (wraz z jego pełną nazwą). Ponieważ nasza nazwa zawiera spację należy otoczyć ją cudzysłowem (UWAGA! Stosowanie spacji NIE JEST POLECANIE – użyte tylko do przykładu!).

Teraz nadajemy nazwę dla skrótu:



Od tego momentu zadanie można uruchamiać z ze skrótu, bez wchodzenia do harmonogramu. Jak dla każdego skrótu możemy zmieniać nazwę, ikonę oraz pozostałe parametry (np. skrót klawiszowy).

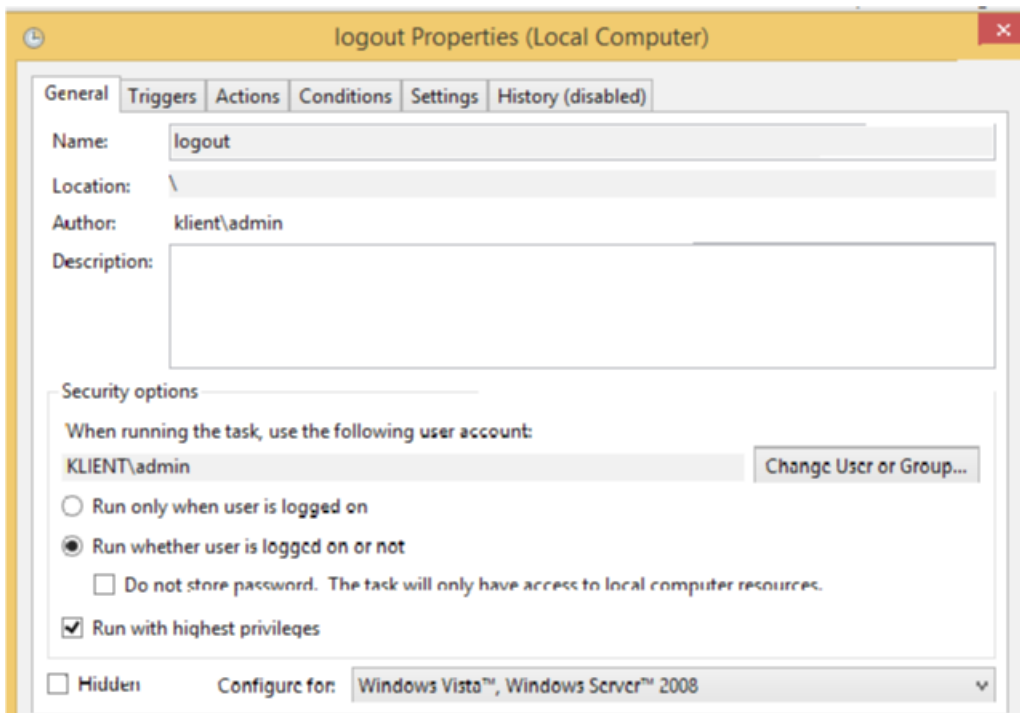
b) jako automatyzacja włączania/wyłączania komputera

Harmonogram można również wykorzystać jako narzędzie kontroli logowania/wylogowania użytkowników oraz restartu/wyłączania komputera. Takie działanie może być użyteczne chociażby w przypadku biur, gdzie pracownicy mogą zapominać o wyłączeniu swoich komputerów po zakończeniu pracy. Standardowo system Windows posiada odpowiednie narzędzia do wyłączenia komputera chociażby w stan hibernacji/uśpienia, jednak gdy użytkownik zostawi uruchomioną przeglądarkę lub inny program działający w ramach jego sesji Windows nie wykona uśpienia (chyba, że mechanizm wykrycia działań użytkownika zawiedzie – co nierzadko się zdarza). Aby wykonać to zadanie należy zapoznać się z następującym poleceniem:

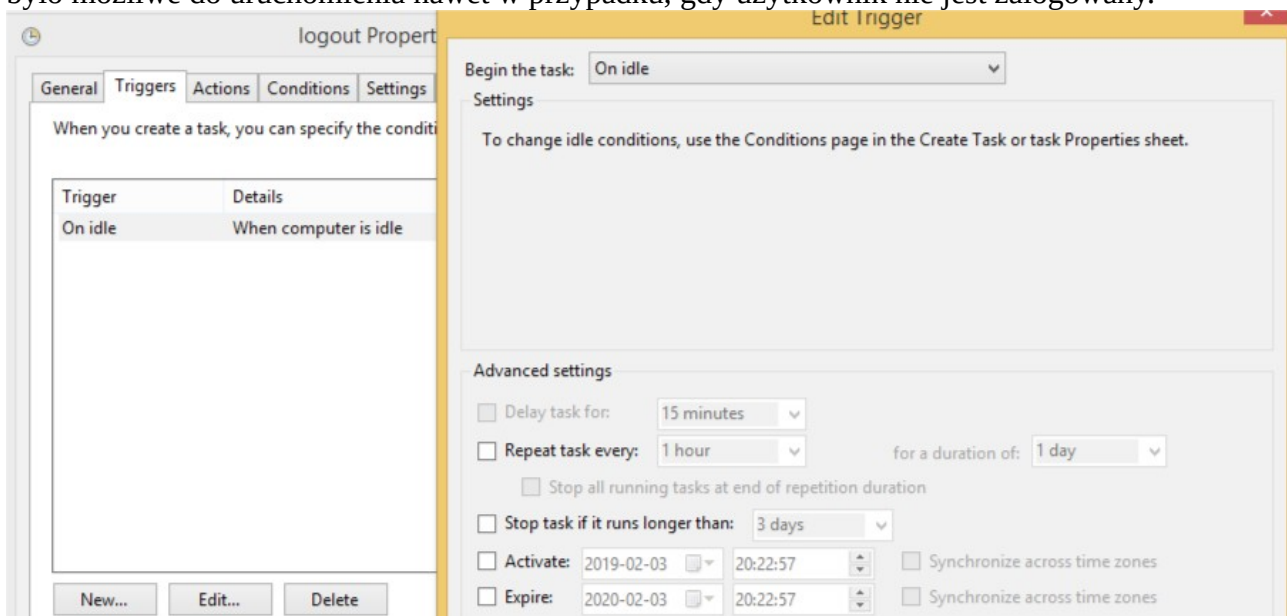
shutdown

Polecenie to pozwala na ponowne uruchamianie komputera (/r), uruchamianie go w trybie naprawy (/o), zamknięcia komputera (/s) czy też wylogowania użytkownika (/l). Narzędzie posiada więcej opcji, które można prześledzić w jego dokumentacji.

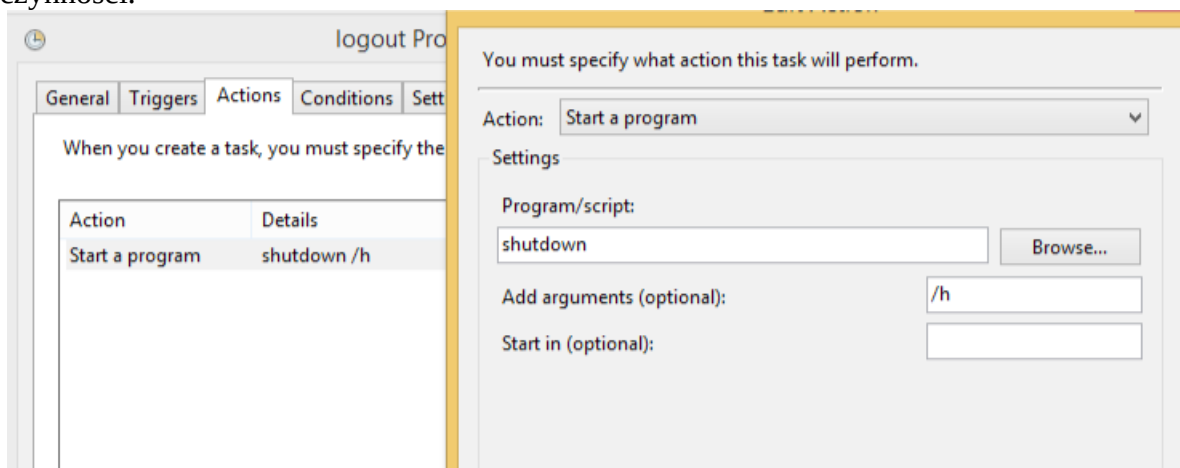
Poniższe zrzuty przedstawiają sytuację, w której komputer przejdzie w stan hibernacji gdy użytkownik nie będzie aktywny co najmniej godzinę i 10 minut



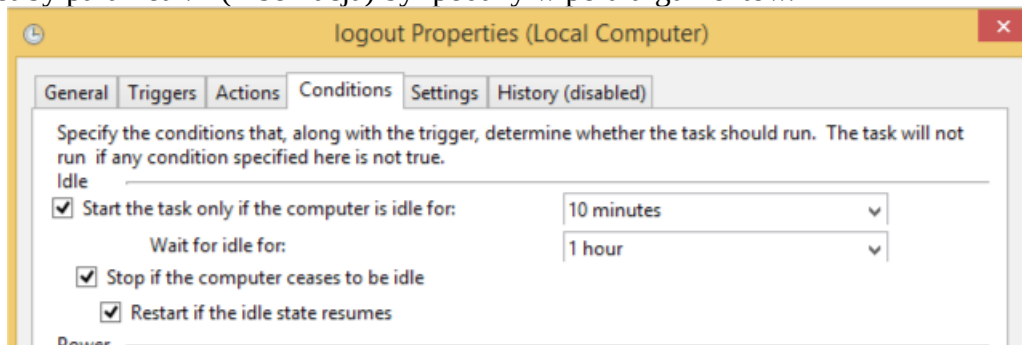
W powyższym przypadku należy zwrócić uwagę by zadanie miało najwyższe uprawnienia oraz było możliwe do uruchomienia nawet w przypadku, gdy użytkownik nie jest zalogowany.



Następnie ustawiamy jeden wyzwalacz – zadanie ma się uruchomić w przypadku wykrycia bezczynności.



Ważne jest by parametr /h (hibernacja) był podany w polu argumentów.



W warunkach ustawiamy czas URUCHOMIENIA zadania od wykrycia bezczynności. Parametr CZEKANIA to okres, w którym ma nastąpić bezczynność. W powyższym przypadku oznacza to, że system będzie czekał na bezczynności godzinę, (zanim zacznie stosować okres uruchomienia). Zabezpieczenie to jest o tyle istotne, że użytkownicy często włączają komputer i na chwilę od niego odchodzą. Jeżeli czas ten byłby dłuższy od 10 minut za każdy razem musieliby ponownie uruchamiać komputer (co nie jest najrozsądniejszym rozwiązaniem).

Z kolei zaznaczenie Zatrzymaj jeżeli komputer nie będzie już bezczynny gwarantuje, że zadanie nie zadziała w chwili gdy użytkownik zacznie pracować na komputerze (oba czasy się resetują – zarówno czekania jak i uruchomienia).

Jeżeli zostanie zaznaczone Zrestartuje w przypadku powrotu bezczynności zmienia nieco działanie powyżej opisanego parametru; licznik bezczynności nie zostanie wyzerowany lecz zapisany i zostanie wznowiony po ponownym wykryciu stanu jałowego.

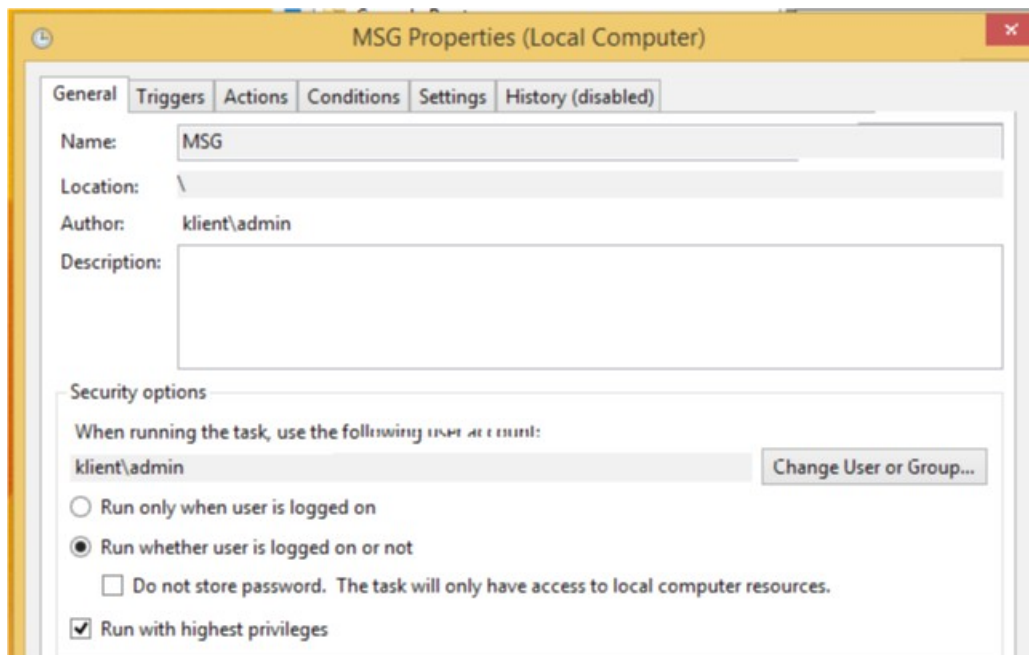
c) jako narzędzie przypomnienia o ważnych sprawach

System Windows pozwala na wysyłanie wiadomości do użytkowników systemu, którzy obecnie są zalogowani do swojej sesji. Narzędzie to można wykorzystać na różne sposoby, między innymi do przypomnienia o ważnych sprawach (przykładowo wysłania sprawozdania w określonych dniach albo wysłania komunikatu o planowanych pracach konserwacyjnych na urządzeniach). Do tego zadania wykorzystamy systemowy program o nazwie

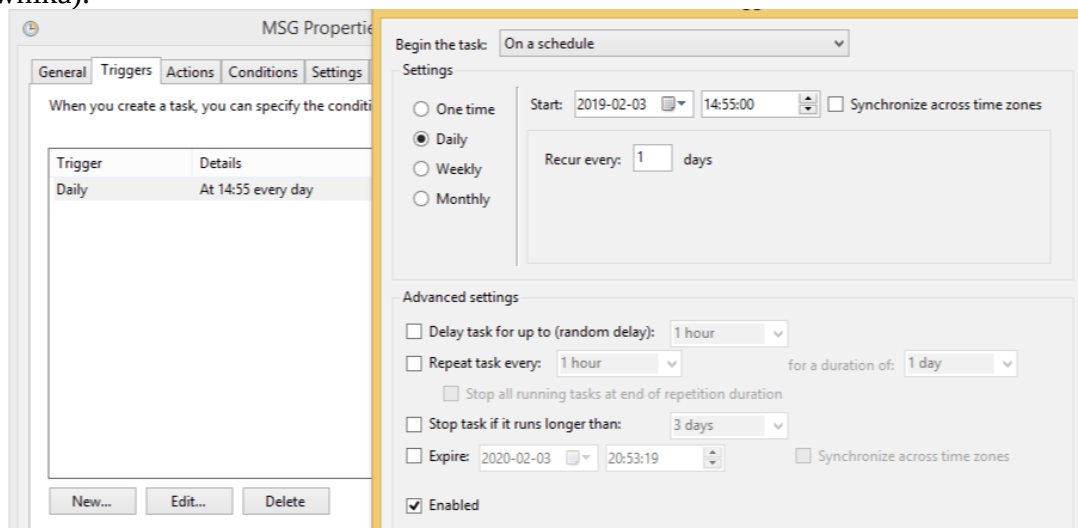
msg

Pozwala on na wysłanie wiadomości do określonego użytkownika, sesji użytkownika (nazwa bądź ID) lub do wszystkich użytkowników (\*).

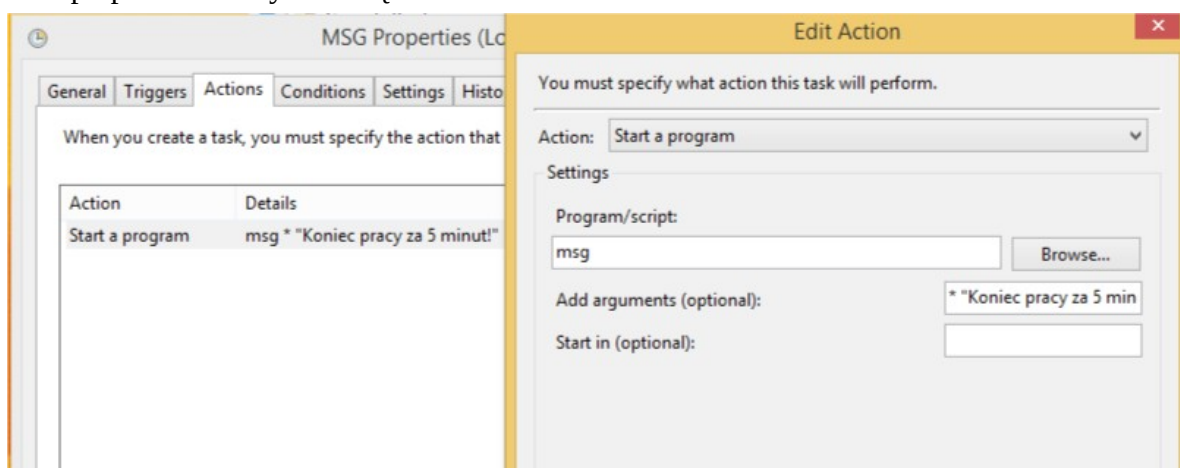
Nasze zadanie ma przypominać pracownikom iż praca kończy się za 5 minut (po tym czasie np. wkraczają prace konserwacyjne, w związku z tym pracownicy wiedzą, że po pojawieniu się tego komunikatu muszą zapisać swoje wszystkie aktywne dokumenty). Poniższe zrzuty pokazują utworzenie zadania:



Tak jak poprzednio, użytkownik wysyłający to admin, którego zalogowanie nie musi być wiążące. Ponadto zadanie uruchomi się z najwyższymi uprawnieniami (celem dostępu do każdej sesji użytkownika).



Ustawiamy zadanie tak by uruchamiało się zawsze (codziennie) o godzinie 14:55. Można by tutaj pokusić się o uruchamianie cotygodniowe (pn-pt) jednak jeżeli komputer nie będzie uruchomiony to zadanie po prostu nie wykona się.



Na koniec podajemy program do uruchomienia (msg) a jako parametr \* :Koniec pracy za 5 minut!''  
Znak gwiazdki jest obligatoryjny – bez niego wiadomość nie wyświetliłaby się nikomu (za mało

parametrów), zaś wyciągnięcie zmiennej środowiskowej %username% spowodowałyby wyświetlenie wiadomości jedynie użytkownikowi uruchamiającemu zadanie (admin).

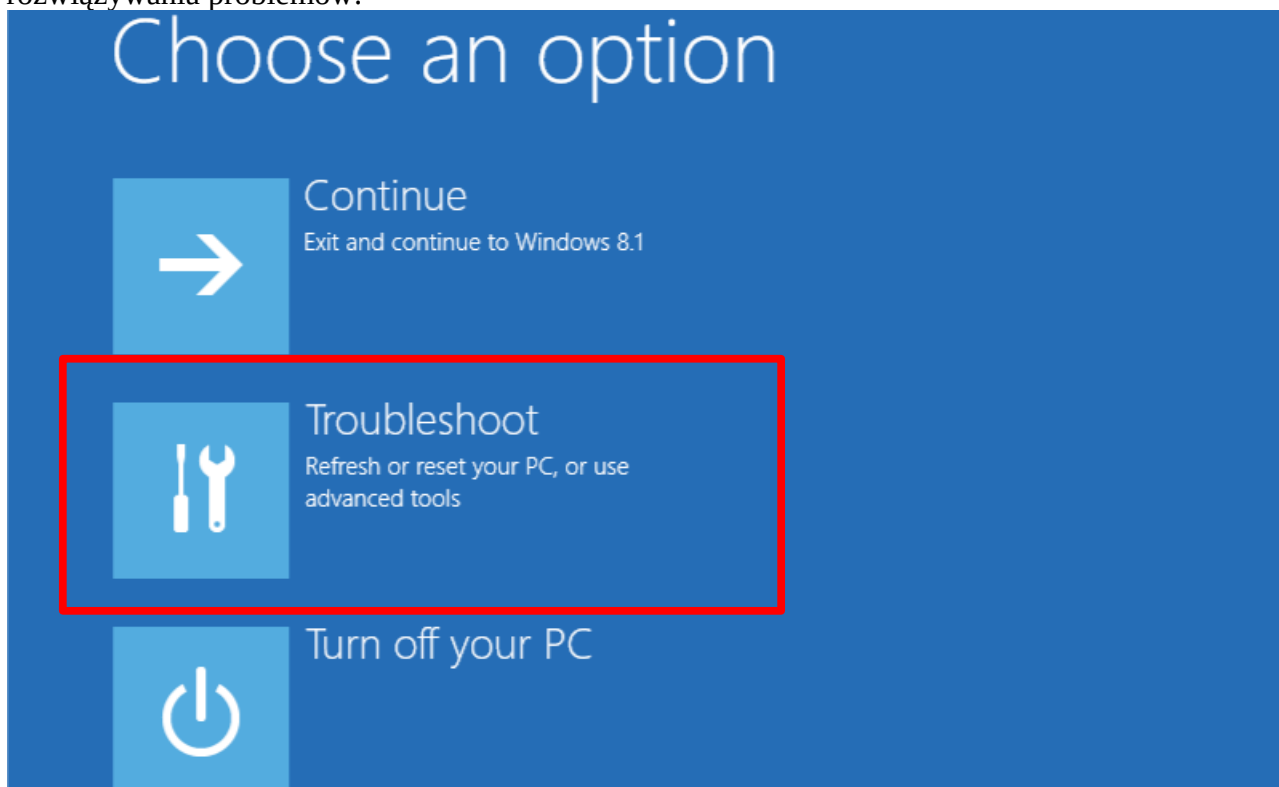
3. Usuwanie zadań w przypadku problemów z dostępem do komputera (np. usuwanie zdalne).

Czasami może się zdarzyć, że nie będziemy mogli zalogować się do komputera poprzez wadliwie działające zadanie (np. restartujące komputer co 1 minutę). Niezbędnym stanie się zalogowanie do systemu w trybie bezpiecznym (lub trybie naprawczym z płyty/pamięci instalacyjnej).

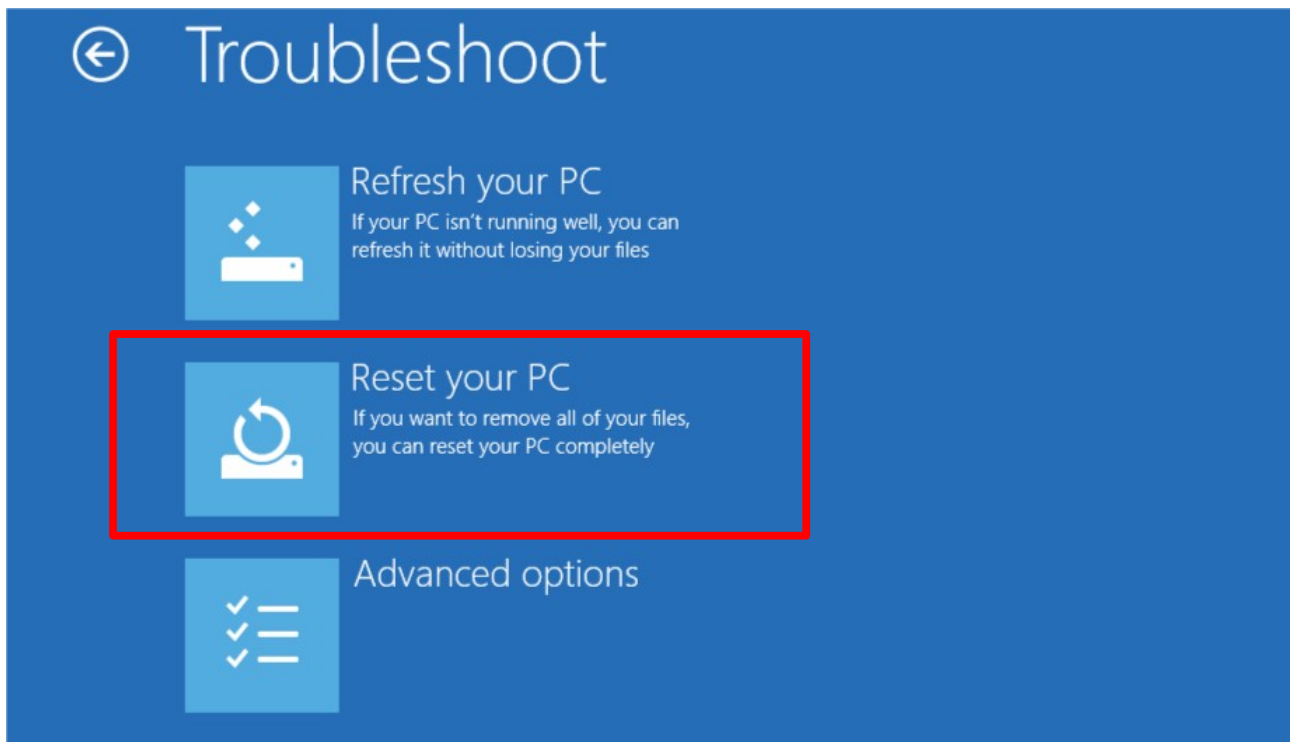
W zależności od możliwości musimy wprowadzić system w tryb naprawy/tryb bezpieczny. W zależności od uszkodzenia systemu możemy tego dokonać poprzez:

- 1) polecenie shutdown /o /r
- 2) przy wybranej opcji ponownego uruchamiania przytrzymać klawisz [SHIFT]
- 3) podczas uruchamiania przerwać ten proces (np. odłączyć kabel zasilający) minimum 3 razy
- 4) przejść do opcji Menu Start → Ustawienia → Aktualizacja i zabezpieczenia → Odzyskiwanie i wybrać przycisk Uruchom ponownie teraz
- 5) uruchomić instalator systemu i wybrać opcję naprawy (ręczna, poprzez konsolę)

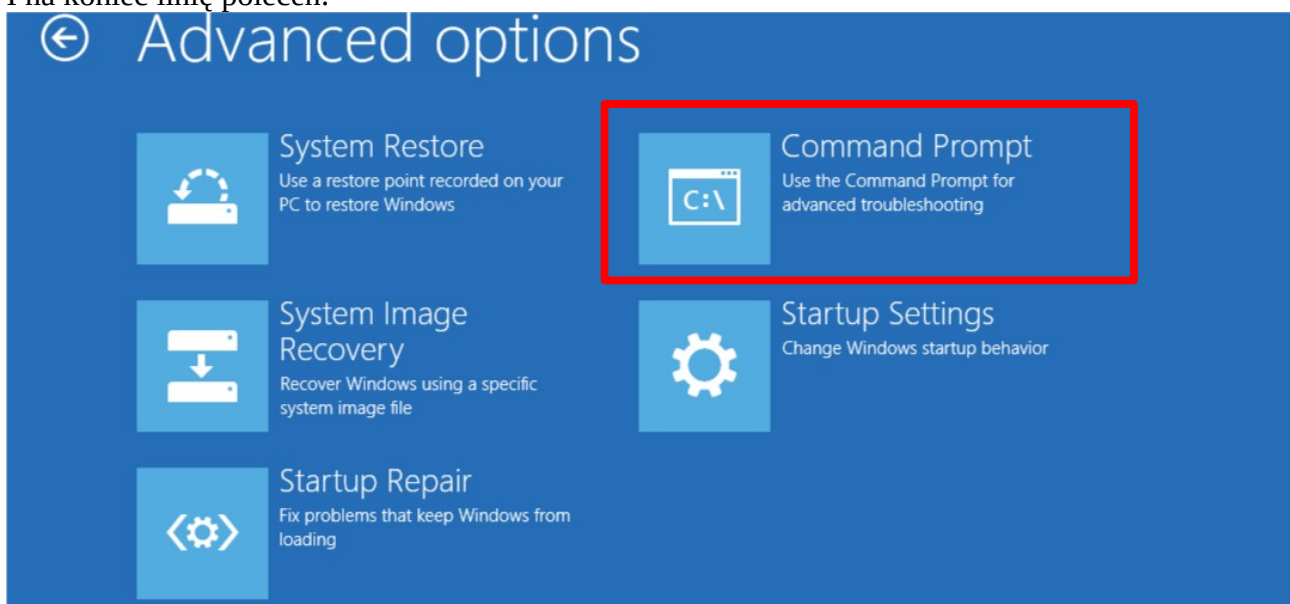
Jakkolwiek nie osiągniemy naszego celu, po wejściu do ekranu naprawy powinniśmy wybrać opcję rozwiązywania problemów:



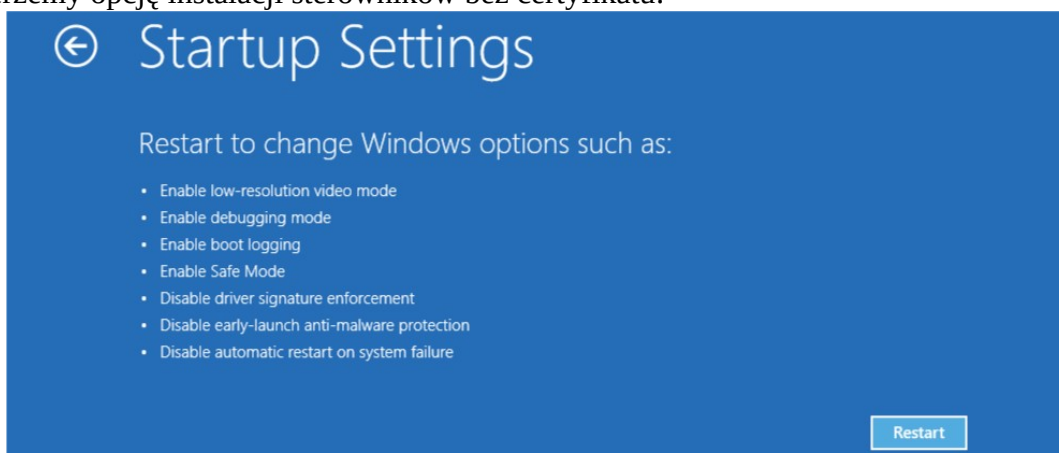
Na następnym ekranie wybieramy opcje zaawansowane:



I na koniec linie poleceń:



Tutaj należy zaznaczyć, że dostęp do linii poleceń uzyskamy jedynie poprzez podanie nazwy użytkownika administracyjnego oraz hasła. Ciekawostką jest, że nie będziemy pytani o hasło jeżeli wymusimy bezpieczne uruchomienie, na ekranie powyżej wybierzemy Opcje startowe (zębátka) i wybierzemy opcję instalacji sterowników bez certyfikatu:



Po ponownym uruchomieniu powinna widnieć opcja Instalacji sterowników systemowych. Po przerwaniu tego procesu otrzymamy linię poleceń administratora bez uprawnień (działa do Windows 10 wersji 1803, na najnowszej nie testowane).

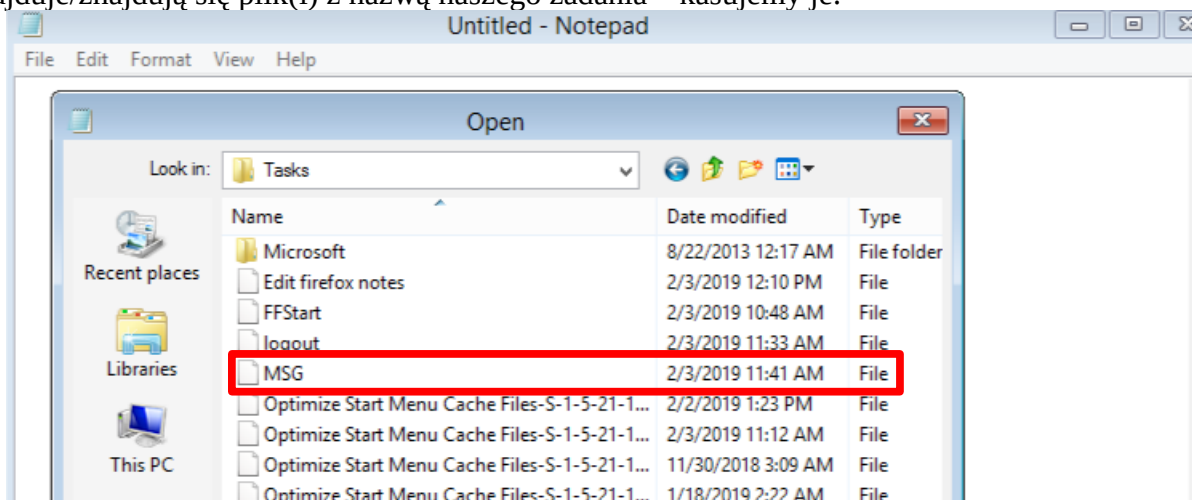
W jakikolwiek sposób nie otrzymamy linii poleceń możemy rozpocząć proces usuwania wadliwego zadania systemowego. Założmy, że będzie nim zadanie wiadomości (utworzone w poprzednim punkcie MSG).

Przed wszystkim upewniamy się, że zadanie nie zostało zapisane jako podręczne w rejestrze systemowym. W tym celu uruchamiamy edytor rejestru (wpisujemy polecenie regedit [ENTER]). Wybieramy gałąź

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\

I sprawdzamy, czy nie ma w niej naszego zadania. Jeżeli jest, kasujemy cały klucz. Jeżeli nie ma, wykonujemy następną czynność.

Przechodzimy do ścieżki instalacji systemu operacyjnego (zmienna %systemroot%), po czym wybieramy katalog Tasks ORAZ System32\Tasks. Jeżeli w którymś w wymienionych katalogów znajduje/znajdują się plik(i) z nazwą naszego zadania – kasujemy je.



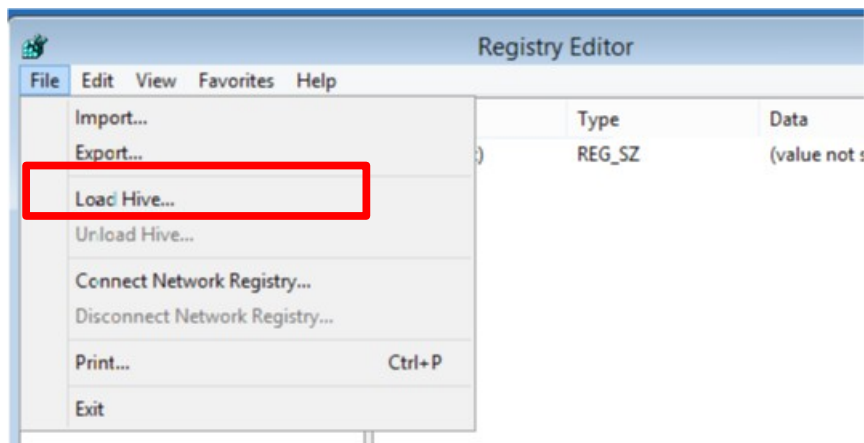
Należy przy tym pamiętać, że plik można znaleźć przez opcję Plik → Otwórz w Notatniku (jest dostępny w wersji tzw. Core systemu) jak i poprzez linię poleceń/PowerShell (PowerShell można uruchomić w linii cmd poprzez wpisanie polecenia powershell).

**UWAGA!!**

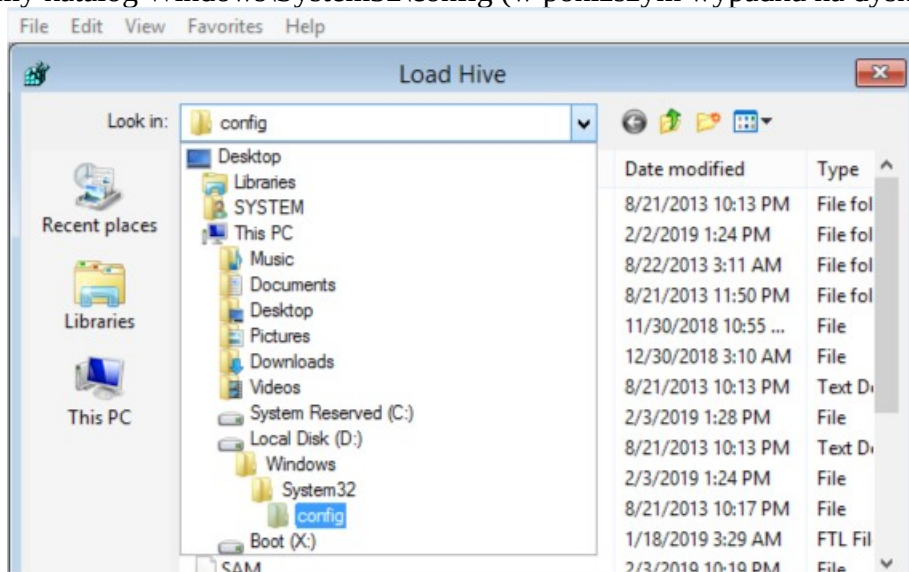
Powyższe zadanie da się wykonać JEDYNIEM W PRZYPADKU gdy system ma uruchomioną konsolę np. w trybie bezpiecznym. Jeżeli uruchamiamy tryb naprawy, tryb przywracania bądź narzędzie naprawy z nośnika instalacyjnego NIE BĘDZIEMY mieli dostępu np. do REJESTRU NASZEGO SYSTEMU (a jedynie do rejestru minisystemu naprawy). To samo tyczy się zmiennej %systemroot%. Aby wszystko zadziałało poprawnie:

1) w rejestrze trzeba załadować gałąź naszego systemu operacyjnego. W tym celu, w narzędziu regedit:

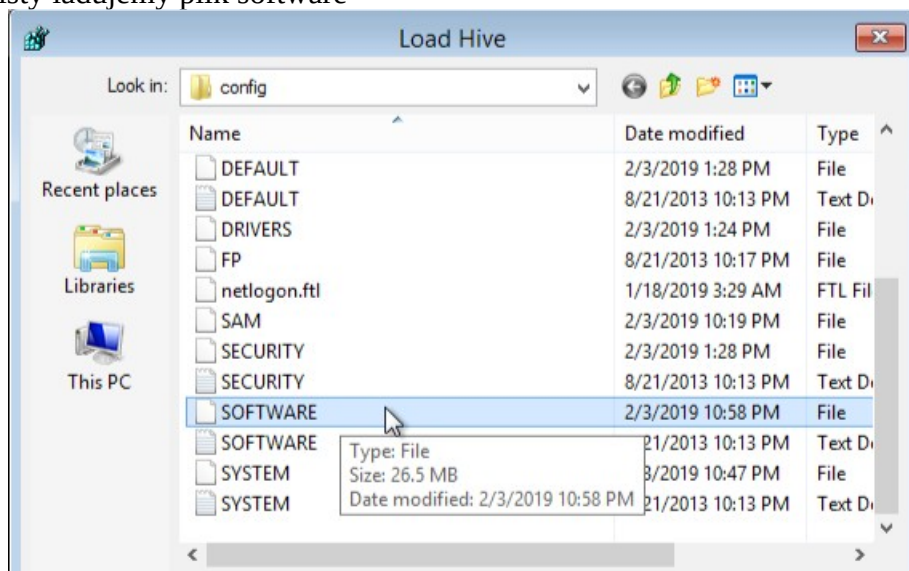
- wybieramy gałąź HKEY\_USERS (musi być zaznaczona – inaczej poniższa opcja będzie niedostępna)
- wybieramy opcję Załaduj rój



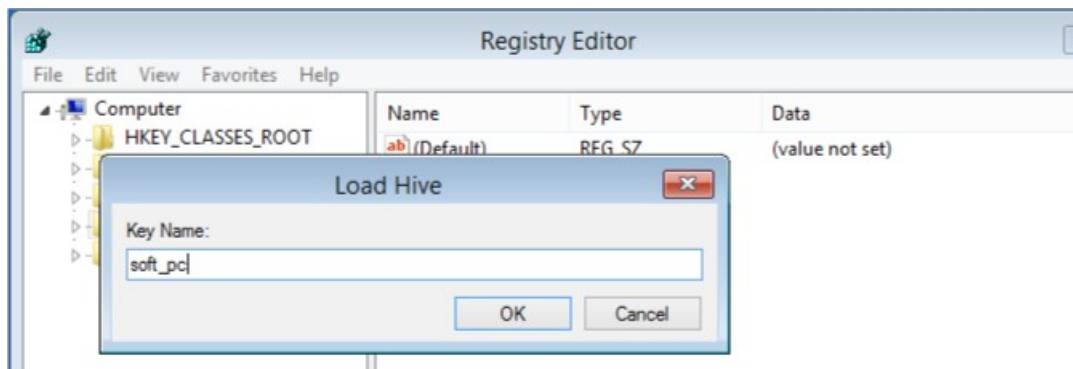
- odnajdujemy katalog Windows\System32\config (w poniższym wypadku na dysku D)



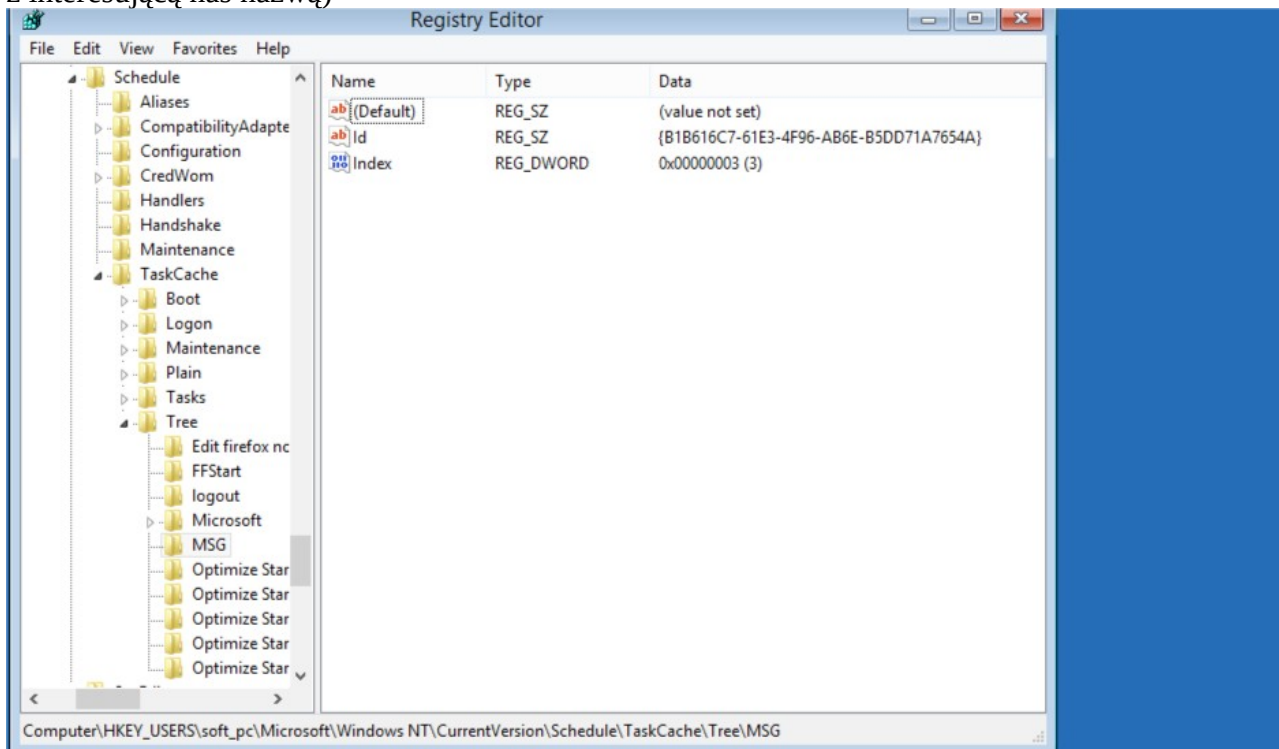
- z listy ładujemy plik software



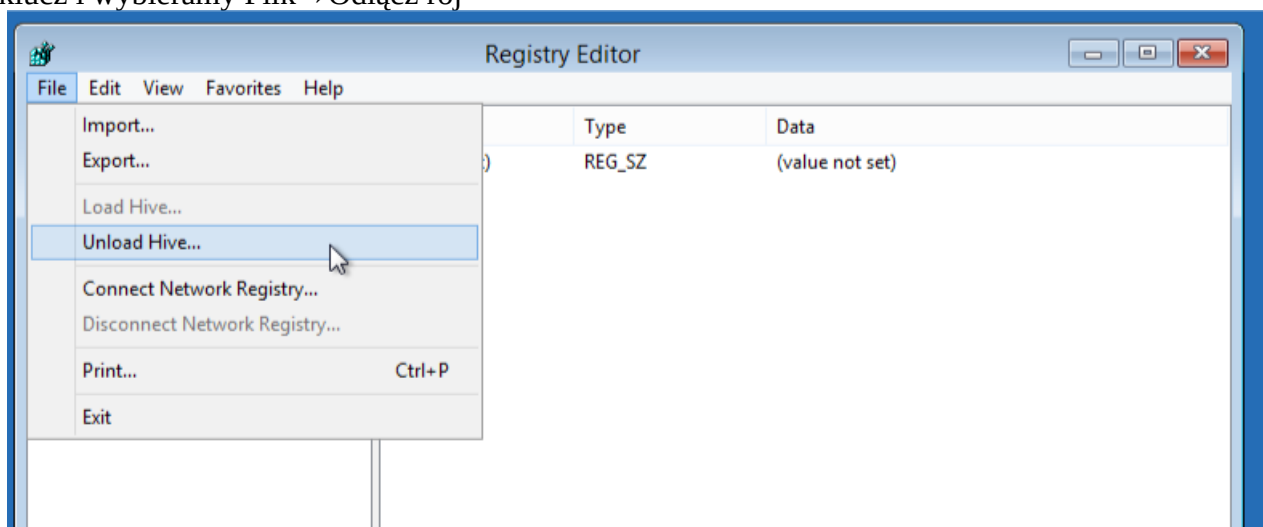
- podajemy nazwę klucza, pod którym ma znaleźć się zawartość ładowanego pliku

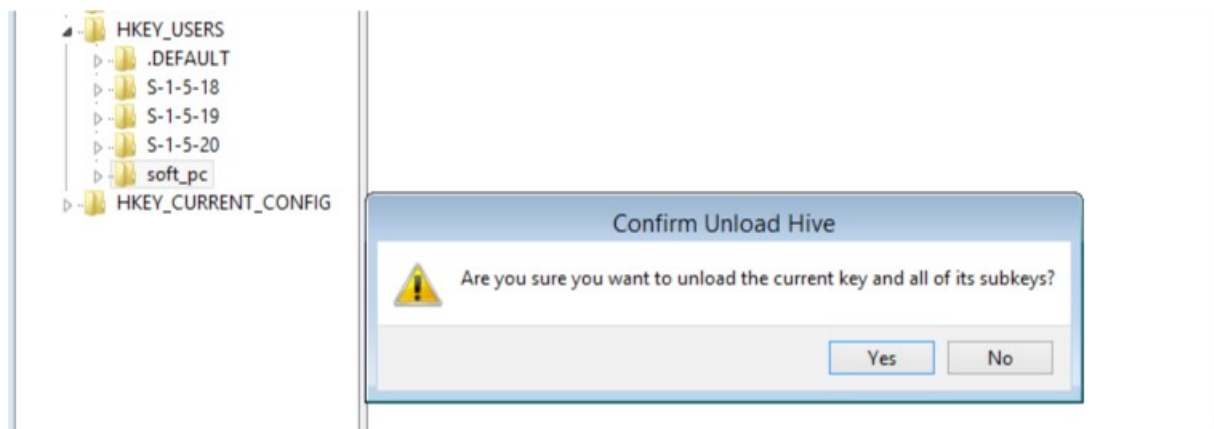


- w dalszej części poruszamy się po załadowanym rejestrze tak, jak poprzednio (i kasujemy zadanie z interesującą nas nazwą)



Na koniec dobrze jest odłączyć załadowany w ten sposób plik. Klikamy na utworzony wcześniej klucz i wybieramy Plik → Odłączyć rój





Jeżeli chodzi o skasowanie pliku/plików zadania przechodzimy do katalogu systemu, z którego załadowaliśmy plik rejestru. Po restarcie skasowanego zadania nie powinno być w harmonogramie, a co za tym idzie nie powinno przeszkadzać w uruchomieniu systemu i/lub użytkowaniu go.

#### 4. Zadania do wykonania

- a) Prześledzić w jaki sposób włączane są programy interakcyjne (z powłoką GUI) w przypadku uruchamiania gdy użytkownik jest zalogowany lub gdy uruchomienie ma nastąpić niezależnie od zalogowania użytkownika tworzącego zadanie. Należy krótko scharakteryzować wskazane zachowanie.
- b) należy zaproponować zadanie, które po okresie 2 minut bezczynności będzie blokowało sesję obecnie zalogowanego użytkownika
- c) utworzyć zadanie, które w każdą sobotę oraz niedzielę będzie wylogowywało każdego użytkownika nie-administratora z systemu
- d) spróbować utworzyć zadanie, które w przypadku uruchomienia konkretnego programu będzie uruchamiało określoną usługę systemową (np. Windows Search). W przypadku zamknięcia tejże aplikacji usługa powinna być zatrzymywana. W celu realizacji tego zadania należy wykorzystać Dziennik zdarzeń systemu Windows (rejestrację zdarzeń wskazanej aplikacji).
- e) Spróbować odtworzyć sytuację opisaną w punkcie 3. W tym celu należy utworzyć konto użytkownika BEZ HASŁA i ustawić na jego logowanie zadanie restartu systemu Windows. Następnie zalogować się na niego. Sprawdzić opisany sposób naprawy harmonogramu. Czy zadanie jest wykonalne?

Materiały i dodatkowe informacje:

<https://superuser.com/questions/199738/windows-stuck-in-reboot-loop-due-to-existing-scheduled-task>

<https://serverfault.com/questions/319701/how-to-delete-scheduled-task-from-command-line-without-confirmation-schtasks-d>

<http://www.idsecuritysuite.com/blog/how-to-lock-your-computer-using-command-line>

<https://superuser.com/questions/130400/how-to-switch-user-from-a-command-line>

<https://blogs.technet.microsoft.com/askperf/2015/02/18/help-my-scheduled-task-does-not-run/>

<https://superuser.com/questions/770420/schedule-a-task-with-admin-privileges-without-a-user-prompt-in-windows-7>

<https://www.makeuseof.com/tag/boot-safe-mode-windows-8/>

<https://www.thewindowsclub.com/where-are-the-windows-registry-files-located-in-windows-7>