**Duża Hydra – pierwszy projekt**

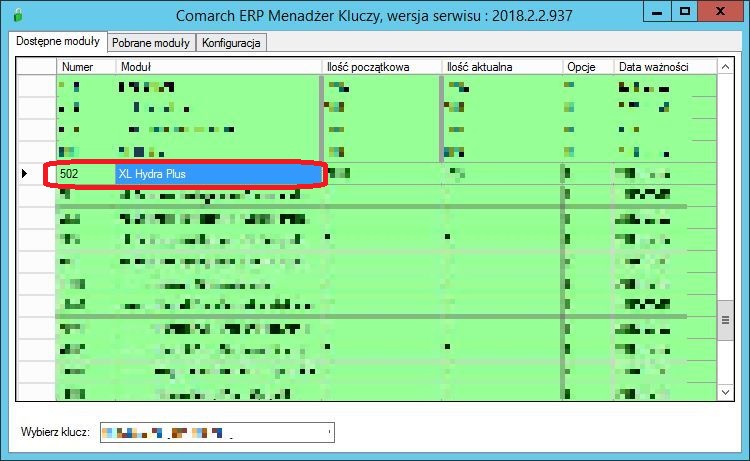
Większość kursów programistycznych zaczyna od czegoś banalnego typu wyświetlenie komunikatu Hello World! Jednakże na potrzeby tego rozwiązania proponuję, że wykonamy coś ciekawszego. Problem jaki rozwiążemy celowo przejaskrawiam, wiem, że w ten sposób lepiej zapadnie Ci w pamięci.

Załóżmy, że wśród pracowników przyjął się zwyczaj “śmieszkowania” poprzez umieszczania w nazwie towaru w kartotece towarowej opisów zawierających nieodpowiednie słowa np. kit, dupa, lipa.  
Ciężko się nie zgodzić, z faktem, że słabo będzie wyglądała pozycja na fakturze:  
Farba X – dupa nie farba  
Zróbmy coś z tym!  
  
Zakasaj rękawy, będzie kodowanie!

DH start!

Ale czekaj… Skoro czytasz ten artykuł pewnie robisz to pierwszy raz?  
Przygotujmy się, poniżej przedstawię jak sprawdzić czy posiadasz licencję Dużej Hydry (DH) w systemie Comarch ERP XL oraz jakie narzędzie developerskie będzie dla Ciebie niezbędne.

Aby praca nad dodatkiem Dużej Hydry była możliwa musisz posiadać moduł XL Hydra Plus, możesz zapytać o ten fakt swojego opiekuna/partnera Comarch lub samodzielnie sprawdzić w Menadżerze Kluczy poprzez odnalezienie pozycji XL Hydra Plus i zweryfikowanie jej aktywności.

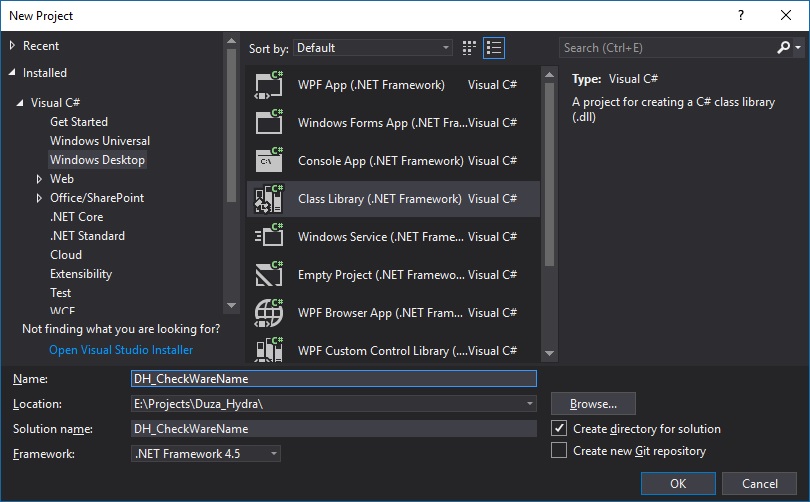
Menedżer Kluczy ERP XL – podgląd wystąpienia DH

Posiadasz dostęp do DH? Super! Możemy instalować środowisko developerskie. Jeszcze nie masz dostępu? Skontaktuj się z swoim opiekunem w tej sprawie i wróć do artykułu.

Aby wykonać dodatek Dużej Hydry będziesz potrzebował dwóch narzędzi:

* [Visual Studio](https://visualstudio.microsoft.com/pl/) – nie jest istotna wersja, chociaż zawsze zalecam Ci instalację najnowszej jeżeli tylko masz taką możliwość;
* Comarch ERP XL – po prostu wersja, którą posiadasz;

Uruchamiamy Visual Studio i tworzymy projekt typu Class Library, w którym utworzymy nasz dodatek DH.

Tworzenie projektu DH – Class Library

Usuwamy klasę Class1 i w tworzymy klasę *CheckWareNameWidget,* która będzie przechowywała definicję naszego dodatku.

Dodajemy do projektu referencje do dll z “Hydrą” **CdnHydra.dll**, plik ten znajduje się w katalogu z instalacją XL-a, przeważnie jest to coś w stylu:  
C:\Program Files (x86)\Comarch ERP XL 2018.2\CdnHydra.dll

Dodajemy

using Hydra;

następnie dziedziczymy klasę abstrakcyjną

Callback

oraz nadpisujemy metody abstrakcyjne

Init()

oraz

Cleanup()

Powinieneś mieć przed oczyma taki kod:

1. using Hydra;
2. namespace DH\_CheckWareName
3. {
4. public class CheckWareNameWidget : Callback
5. {
6. public override void Init()
7. {
8. }
9. public override void Cleanup()
10. {
11. }
12. }
13. }

nad słowem kluczowym namespace dodajemy definicję dodatku DH, zgodnie z poniższym schematem:

1. [assembly: CallbackAssemblyDescription(Name: "Kontroluj nazwę towaru",
2. Description: "Sprawdzaj czy na kartotece towarowej nie użyto zabronionych słów.",
3. Author: "Szymon Liszewski",
4. Version: "1.0",
5. SystemVersion: "2018.2",
6. Created: "14-11-2018")]

Pamiętaj aby poświęcić chwilę na wypełnienie powyższego opisu assembly, gdyż na jego podstawie będziesz odnajdywał plik w systemie ERP. W przypadku gdy wykonasz 20 i więcej dodatków DH, niezwykle istotne jest prawidłowe ich opisanie, dzięki czemu od razu jesteś w stanie odnaleźć dodatek realizujący daną funkcjonalność jeżeli będzie istniała potrzeba jego edycji/poprawy/deaktywacji.

Kolejnym krokiem jest dodanie atrybutu z subskrypcją procedury bezpośrednio nad nazwą klasy – atrybut ten określa w jakim miejscu w programie chcemy podłączyć się z naszym dodatkiem DH. W kolejnych wpisach opiszę dokładniej na czym polega owa subskrypcja, w tej chwili zaufaj mi i wprowadź kod:

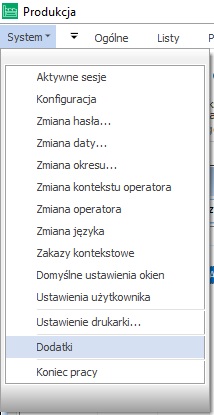
1. [SubscribeProcedure(Procedures.TwrEdycja, "Edycja kartoteki towarowej")]

Tym sposobem utworzyliśmy definicję dodatku DH. Czas na to co tygryski lubią najbardziej czyli mięsko. Dodajemy zatem takiego potworka w ciele metody *Init()*:

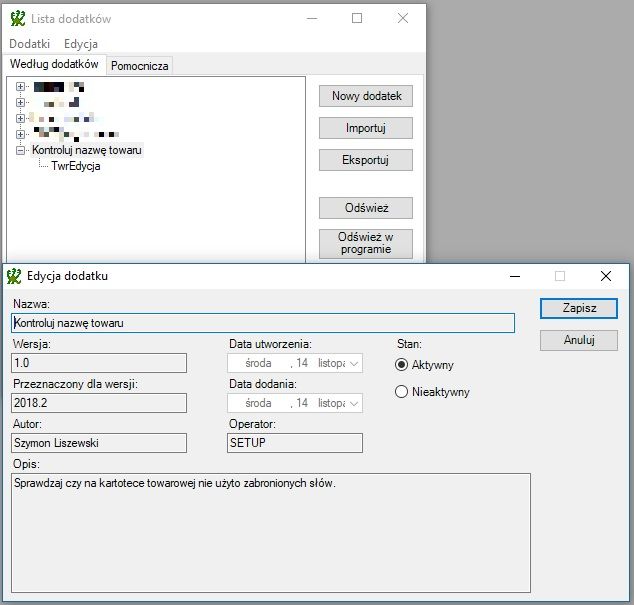
1. AddSubscription(true, 0, Events.OpenWindow, (procID, ctrlID, eveID) =>
2. {
3. AddSubscription(true, GetWindow().Children["?Cli\_Zapisz"].Id, Events.Accepted, (procedureID, controlID, eventID) =>
4. {
5. ClaWindow wareNameControl = GetWindow().Children["?Twr:Nazwa"];
6. string wareName = wareNameControl.ScreenTextRaw;
7. if (wareName.ToLower().Contains("lipa")
8. || wareName.ToLower().Contains("dupa")
9. || wareName.ToLower().Contains("kit"))
10. {
11. MessageBox.Show("Nazwa towaru zawiera niedozwoloną frazę. Niedozwolone słowa: lipa, dupa, kit.", "Blokada zapisu kartoteki", MessageBoxButtons.OK, MessageBoxIcon.Stop);
12. return false;
13. }
14. return true;
15. });
16. return true;
17. });

Na pierwszy rzut oka kod może wydawać się skomplikowany, jednakże jest bardziej banalny niż Ci się wydaje.  
Metoda ***AddSubscription*** dodaje subskrypcję na daną kontrolkę lub okno.  
Pierwsze wywołanie AddSubscription dodaje subskrypcję na okno “Karta towaru”, która jest podniesiona do trybu edycji. Następne wywołanie, zagnieżdżone, AddSubscription dodaje zdarzenie w momencie kliknięcia buttona zapisz “dyskietki” w oknie “Karty towaru”.  
***ClaWindow*** jest klasą, która reprezentuje każdy pojedynczy element GUI, zarówno główne okno GetWindow() jak również np. kontrolkę textbox z nazwą towaru *GetWindow().Children[“?Twr:Nazwa”]*.  
W przyszłości opiszę skąd możesz pobierać nazwy dzieci w oknie dla poszczególnych kontrolek, np. *?Twr:Nazwa* lub *?Cli\_Zapisz*.  
Jak również schemat działania dodatków DH. Na ten moment musi wystarczyć Ci informacja, że AddSubscription jest zdarzeniem, które oczekuje zaistnienia akcji na wybranej kontrolce, ***return true***– oznacza, że zdarzenie zostaje wykonane, a ***return false*** – powoduje przerwanie zdarzenia.

Dysponujemy poprawnym kodem aplikacji, czas podłączyć ją w systemie, w tym celu wykonujemy builda aplikacji w trybie Release oraz w dowolnym module ERP XL wybieramy z głównego menu aplikacji *System > Dodatki.*

Uruchomienie okna zarządzające dodatkami DH w programie

Importujemy dodatek z za pomocą buttona *Importuj* – wskazujemy utworzoną przez nas bibliotekę DLL. Pojawia się na liście nowy dodatek o nazwie *“Kotroluj nazwę towaru”* wybieramy go oraz podnosimy do *Edycji*, w oknie edycji dodatku zmieniamy *Stan* dodatku ustawiając go na *Aktywny*.

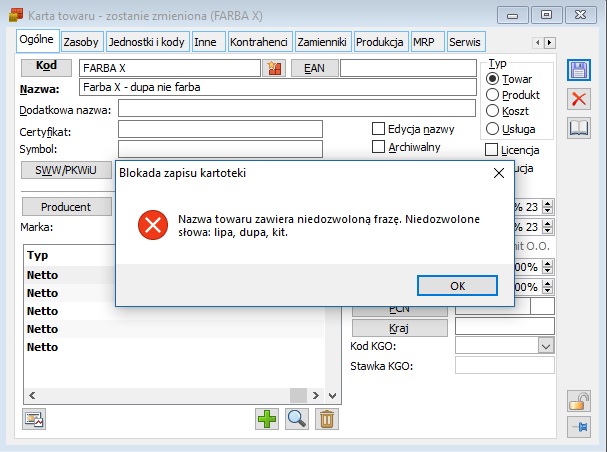
Zaimportowanie dodatku oraz jego aktywowanie.

Ostatnim krokiem jest uruchomienie modułu ***Administrator*** i podpięcie dodatku pod konkretne centrum w programie. W tym celu w module Administrator wybieramy ***Firma > Struktura Firmy***, po czym na zakładce ***Struktura praw*** podnosimy do edycji wybrane centrum (może to być główne centrum lub wybrane podcentrum). Przechodzimy do zakładki ***Dodatki***, w której dodajemy nowo utworzony dodatek.

Od tej pory dodatek będzie działał w serwisie.  
Jeżeli zdarzyłoby się, że jakiemuś operatorowi dodatek by nie działał sprawdź:

* czy dodatek na pewno jest aktywny;
* czy dodatek został przypisany do aktywnej struktury praw;
* czy kod dodatku działa poprawnie;
* czy operator na pewno jest przypisany do centrum, dla którego przypisałeś dodatek;
* spróbuj w oknie Lista dodatków wybrać opcję “Odśwież w programie”;
* uruchom ponownie moduł ERP XL;

Tak prezentuje się działająca aplikacja, po wciśnięciu buttona zapisz na kartotece towarowej w przypadku gdy nazwa kartoteki zawiera nieodpowiedni tekst:

Zablokowanie zapisu kartoteki towarowej w chwili gdy nazwa kartoteki zawiera niestosowne słowo

Zrealizowaliśmy dodatek DH, część rzeczy nie została wyjaśniona dokładnie, chciałem abyś mógł wykonać całą aplikację jak najszybciej. Znajdziesz opisy zaawansowanych zagadnień w kolejnych wpisach.  
Jeżeli potrzebujesz, poniżej znajduje się realizowane zadanie, wskaż jedynie referencje na plik CdnHydra.dll.

Gratuluję właśnie wykonałeś swój pierwszy dodatek DH!