

# 01. Import metadanych z formatu MARC



## Uwaga

Biblioteka programistyczna dla tego rozszerzenia to [dlibra-app-extension-mf-marc](#).

## Informacje ogólne

Import atrybutów z formatu komunikacyjnego MARC 21 w rozszerzeniu MARC domyślnie opiera się na konfiguracji wbudowanej w rozszerzenie. Zamiast domyślnej konfiguracji można używać innej, zdefiniowanej w zewnętrznych plikach tekstowych. Pliki te mają format prostego zbioru właściwości. Aby wskazać nowe pliki konfiguracyjne rozszerzenia należy wyświetlić jego konfigurację ([rysunek poniżej](#), wyświetlanie okna konfiguracji opisane jest w sekcji [konfiguracja](#)). Po wyświetleniu konfiguracji należy wskazać źródło nowej konfiguracji (plik na dysku lub adres URL) oraz zaznaczyć opcję Używaj konfiguracji z podanego źródła. Dodatkowo należy wyspecyfikować kodowanie plików MARC, które będą importowane. Kodowanie pliku MARC zależy od systemu informatycznego z którego taki plik pochodzi.

Import metadanych z formatu MARC - konfiguracja rozszerzenia

## Format pliku marcImport.properties

```
Title=245:${a} ${b} ${n};130;210;222;240;246;730;740;
en.Title=210;222;240;246;730;740;
Creator=100;110;111;
Subject=
Description=6XX;
Publisher=260a;260b;260f;
Contributor=700;710;711;
Date=260c;
Type=
Identifier=920;856u;
Source=
Language=041;546;008/35-37;
Relation=250;534;440;490;800;810;811;830;
Coverage=
Rights=506;540;"PAN";
```

Powyżej przedstawiona jest zawartość przykładowego pliku konfiguracyjnego `marcImport.properties`. W pliku tym można zdefiniować konfigurację MARC - przyporządkować wartości elementów z formatu MARC do atrybutów w systemie dLibra.

Każda linia w pliku konfiguracyjnym zawiera konfigurację importu wartości dla jednego atrybutu. W każdej linii po lewej stronie znaku równości wyspecyfikowana jest nazwa RDF atrybutu do którego zostaną przypisane wartości z elementów MARC znajdujących się po prawej stronie znaku równości. Jeśli nazwa RDF atrybutu poprzedzona jest nazwą języka i kropką (np. `en.Title=210;222;240;246;730;740;`) to następująca po nazwie reguła będzie użyta do importowania metadanych do tego języka. Jeśli brakuje nazwy języka to reguła wykorzystywana jest do importowania metadanych do wybranego przez użytkownika języka (na zakładce w edytorze metadanych). Nazwa języka musi być dwuliterowym skrótem według standardu ISO 639. Nazwy RDF atrybutów można znaleźć w aplikacji administratora (panel edycyjny dotyczący atrybutu). Elementy, które można importować z formatu MARC to m. in. wartość podpola, określone znaki z pól kontrolnych itp. Brak którejkolwiek z nazw RDF atrybutów w pliku konfiguracyjnym jest równoznaczny z pozostawieniem przy tym atrybucie pustej listy numerów pól. W tej sytuacji do danego atrybutu nie zostanie zaimportowana żadna wartość.

Podstawowy zapis numerów pól MARC (lista po prawej stronie znaku równości), z których mają zostać zaimportowane wartości atrybutów w aplikacji, ma składnię: `AAAb;`, gdzie `AAA` jest trzycyfrowym numerem pola, a `b` jest identyfikatorem podpola. Możliwy jest również zapis łączący podpola lub pobieranie zakresu znaków z pól kontrolnych (przypadki te opisane są poniżej). Należy zwrócić uwagę, że znak `;` (średnik) jest częścią niezbędną zapisu pola w konfiguracji.

Możliwe jest opuszczenie wartości podpola, jak również użycie identyfikatora wielowartościowego. Szczegóły i przykłady konfiguracji przedstawione są poniżej:

- `100;` - przykład użycia numeru pola.  
Taki zapis spowoduje w przypadku pola specjalnego import wartości tego pola (należy pamiętać, że pola specjalne w formacie MARC, czyli te o wartości mniejszej niż 010, nie posiadają nigdy identyfikatorów podpól) do danego atrybutu, a w przypadku pozostałych pól import wszystkich wartości występujących we wszystkich podpolach tego pola do danego atrybutu. Każda wartość podpola będzie zaimportowana jako oddzielna wartość atrybutu.
- `260c;` - przykład użycia numeru pola oraz identyfikatora podpola.  
Taki zapis spowoduje import do atrybutu tylko wartości konkretnego podpola (w tym przypadku `c` danego pola (w tym przypadku pola `260`).
- `6XX;` - przykład użycia identyfikatora wielowartościowego.  
Taki zapis spowoduje import do atrybutu wszystkich wartości pól i ich podpól z zakresu `600 - 699`. W tym przypadku nie można wyspecyfikować konkretnych podpól. Możliwy jest także przykładowo zapis `65X;`, który spowoduje analogiczny import wartości pól z zakresu `650 - 659`.
- `245:${a} ${b} ${n};` - przykład łączenia podpól pola MARC w jedną wartość.  
Zapis ten możemy podzielić na dwie części, które oddzielone są od siebie znakiem `:` (dwukropkiem):
  1. `245` - jest to numer pola, którego podpola będą łączone w jedną wartość
  2. `${a} ${b} ${n}` - jest to szablon, który definiuje w jaki sposób łączyć podpola.  
Zapis `${a}` oznacza, że w jego miejsce ma zostać wstawiona wartość podpola `a` z pola o numerze zapisanym przez znakiem `:` - w tym przypadku pola `245`. Zatem zapis ten spowoduje, że podpola `a`, `b` oraz `n` zostaną połączone w jedną wartość i będą oddzielone spacją. Przykładowo jeśli podpole `245a` ma wartość *wartość pierwsza*, podpole `245b` ma wartość *wartość druga* oraz podpole `245n` ma wartość *wartość trzecia* to wynikiem takiego zapisu będzie wartość *wartość pierwsza wartość druga wartość trzecia*. Jeżeli chcielibyśmy aby podpola te oddzielał jakikolwiek inny znak lub ciąg znaków, wystarczy je wpisać (np. `245:${a}-${b} podpole n: ${n};`). Wyjątkami są znaki `;` (średnik), ukośnik `\` oraz `$` - aby te znaki zostały poprawnie zinterpretowane należy poprzedzić je ciągiem dwóch ukośników `\\` (np. `245:${a}${b}\\;${n};`). Polskie znaki diakrytyczne, oraz inne znaki spoza standardowego zestawu ASCII, muszą zostać przekonwertowane do kodów utf-8 w formacie `\uXXXX`, gdzie `X` to cyfry szesnastkowe (konwerter jest dostępny np. pod adresem <https://itpro.cz/juniconv/>).
- `008/35-37` - dotyczy tylko pól kontrolnych - oznacza pobranie zakresu znaków z pola kontrolnego.  
Zapis ten składa się z dwóch części oddzielonych od siebie znakiem `/` (ukośnik):
  1. `008` - jest to numer pola kontrolnego z którego pobrane zostaną wartości
  2. `35-37` - jest to zakres znaków jaki zostanie pobrany z pola o numerze, który występuje przed znakiem `/`.  
Zapis ten oznacza, że znaki `35`, `36` oraz `37` z pola `008` będą wartością tego zapisu. Jeśli pole `008` na pozycji `35` ma znak `p`, na pozycji `36` ma znak `o` a na pozycji `37` ma znak `/` to wartością takiego zapisu będzie *pol*. Jeśli chcemy pobrać tylko jeden znak z określonej pozycji wystarczy ją wyspecyfikować po znaku `/`, np. `008/30`.
- `"PAN"` - jest wartość stała, dodawana do atrybutu niezależnie od zawartości wczytywanego pliku marc. Stałe wartości muszą być umieszczone w cudzysłowie `"`. Podobnie jak w przypadku [szablonów dla łączenia podpól](#), niektóre znaki muszą być poprzedzone dwoma ukośnikami `\\`: tym razem jest to cudzysłów `"` oraz ukośnik `\`. Również potrzebna jest konwersja polskich liter i innych niestandardowych znaków do kodów utf-8.

## Format pliku `marcImpRemChars.properties`

```
end-245b=a|b
begin-245a=OS/2
end-260c=c
```

Powyżej przedstawiony jest przykładowy plik konfiguracyjny `marcImpRemChars.properties`.

Plik ten umożliwia definiowanie ciągów znaków jakie mają być usuwane z konkretnych podpól MARC przed importowaniem ich do opisu bibliograficznego. Znaki (lub ciągi znaków) mogą być usuwane z początku (`begin`) lub z końca (`end`) podpola MARC. Ciągi znaków definiuje się używając wyrażeń regularnych. Wyrażenia regularne, które mogą być używane w mechanizmie usuwania znaków z wartości MARC muszą być zgodne z wyrażeniami regularnymi używanymi w języku Java (szczegóły można znaleźć [tutaj](#)).

Linia `end-245b=a|b` oznacza, że z końca (słowo `end`) podpole 245b (zapis 245b) zostanie usunięty znak *a* lub znak *b* (jeśli oczywiście któryś z tych znaków znajduje się na końcu wartości podpole 245b). Minus oddziela określenie miejsca z którego usuwamy znaki (w tym przypadku `end`) oraz podpole z którego te znaki będą usuwane (245b). Po znaku równości następuje specyfikacja wyrażenia regularnego (w tym przypadku `a|b`) definiującego jakie znaki mają być usuwane.

Przeanalizujmy następujący przykład: `begin-245a=ab`. Zapis ten spowoduje, że program redaktora usunie z początku podpole 245a ciąg znaków `ab` o ile taki ciąg zostanie znaleziony na początku wartości tego podpole. Jeśli zatem w pliku MARC pole 245a będzie miało wartość *abBajki* to po zastosowaniu mechanizmu usuwania znaków otrzymamy *Bajki* i taka właśnie wartość zostanie zaimportowana do opisu bibliograficznego.

## Domyślna konfiguracja

Domyślnie rozszerzenie skonfigurowane jest następującymi plikami:

- `marcImport.properties`:

```
Title=245;130;210;222;240;246;730;740;
Creator=100;110;111;
Subject=
Description=6XX;
Publisher=260a;260b;260f;
Contributor=700;710;711;
Date=260c;
Type=
Identifier=920;856u;
Source=
Language=041;546;
Relation=250;534;440;490;800;810;811;830;
Coverage=
Rights=506;540;
```

- `marcImpRemChars.properties`

Plik ten jest domyślnie pusty.