

ARHAT Sp. z o.o. wszelkie prawa zastrzeżone.

FormAnalyzer DESIGNER

wersja: 6.21.0.0

Podręcznik użytkownika

Marzec 2018

Spis treści

1. WSTĘP	7
2. ROZPOZNAWANIE W FORMANALYZER ENGINE	9
2.1. ROZPOZNAWANIE TYPU DOKUMENTU.....	10
2.2. ROZPOZNAWANIE TYPU FORMULARZA	11
2.3. PRZETWARZANIE I ROZPOZNAWANIE ZAWARTOŚCI FORMULARZY	12
2.4. PRZETWARZANIE WSTĘPNE	14
2.5. POZYCJONOWANIE PÓL NA FORMULARZU	14
2.6. PRZETWARZANIE I ROZPOZNAWANIE PÓL	15
2.7. GRUPY PÓL.....	17
2.8. DOKUMENTY WIELOSTRONICOWE	17
3. PLIK KONFIGURACJI	19
4. PODSTAWY OBSŁUGI FORMANALYZER DESIGNER	20
4.1. URUCHOMIENIE PROGRAMU.....	20
4.2. PODŁĄCZENIE DO BAZY DANYCH.....	21
4.3. OKNO GŁÓWNE APLIKACJI FORMANALYZER DESIGNER.....	26
4.4. OKNO PODGLĄDU WYDRUKU	30
4.5. DOSTOSOWYWANIE MENU I PASKA NARZĘDZI	33
4.6. PLIKI KONFIGURACJI	36
4.7. OPERACJE NA ELEMENTACH FORMULARZA.....	43
4.7.1. Umieszczenie nowego elementu	44
4.7.2. Powielanie elementów.....	45
4.7.3. Zaznaczanie elementów	46
4.7.4. Przesuwanie elementu	47
4.7.5. Zmiana rozmiaru elementu	48
4.8. OPERACJE NA WIELU ELEMENTACH	48
4.9. WŁAŚCIWOŚCI WIELU WYBRANYCH OBIEKTÓW	52
4.10. ZMIANA ATRYBUTÓW WYŚWIETLANIA.....	53
4.11. PANEL WŁAŚCIWOŚCI	54
5. FORMULARZE.....	58
5.1. DODANIE NOWEGO FORMULARZA	58
5.2. AUTOMATYCZNE ROZPOZNANIE TYPU FORMULARZA.....	61
5.3. ROZPOZNANIE TYPU FORMULARZA - POLE ROZPOZNAWANIA TYPU	62
5.4. ROZPOZNANIE TYPU FORMULARZA - POLE NAZWY FORMULARZA	64
6. POZYCJONOWANIE PÓL NA FORMULARZU	66
6.1. ZNACZNIK POZYCJONOWANIA	66
6.2. OBSZAR POZYCJONOWANIA	69
6.3. TESTOWANIE POZYCJONOWANIA FORMULARZA	71
7. POLA ROZPOZNAWANE	74
7.1. KOLEJNOŚĆ PRZETWARZANIA PÓL	74
7.2. TYPY I NAZWY PÓL	76
7.3. POLE TYLKO OBRAZ	76
7.4. POLE PRZETWARZANIE WSTĘPNE.....	77
7.5. USUWANIE SZUMU	78
7.5.1. Usuwanie linii	79
7.5.2. Wygładzanie	80
7.5.3. Filtry IP.....	81
7.6. POLE WYBORU	82

7.7.	POLE KOD PASKOWY	82
7.8.	POLE DRUK.....	85
7.8.1.	<i>Przetwarzanie wstępne</i>	86
7.8.2.	<i>Rozpoznawanie treści pola</i>	86
7.8.3.	<i>Przetwarzanie końcowe</i>	89
7.8.4.	<i>Słowniki wartości</i>	95
7.8.5.	<i>Skrypty</i>	97
7.8.6.	<i>Funkcje weryfikacji z zewnętrznych bibliotek DLL</i>	99
7.8.7.	<i>Funkcje weryfikacji poprawności danych pola z wykorzystaniem WebService.</i> 101	
7.8.8.	<i>Uaktualnianie wartości pól i sprawdzanie poprawności funkcji weryfikacji.</i> ..	104
7.9.	POLE DRUK I PISMO RĘCZNE.....	105
7.10.	POLE RECOSTAR ENGINE.....	109
8.	GRUPOWANIE PÓŁ	114
8.1.	GRUPY PÓŁ WYBORU	115
8.2.	GRUPY PÓŁ TEKSTOWYCH	124
8.3.	WIERZ I SEKCJA DANYCH	126
9.	WŁAŚCIWOŚCI PÓŁ	133
9.1.	OPCJE PODSTAWOWE.....	134
9.2.	OPCJE WERYFIKACJI.....	138
9.3.	DODATKOWE OPCJE WERYFIKACJI	141
9.4.	OPCJE PROPAGACJI WARTOŚCI	144
9.5.	OPCJE EKSPORTU	145
9.6.	POLA INDEKSOWE	147
10.	KOLEJNOŚĆ PRZETWARZANIA PÓŁ.....	150
10.1.	KOLEJNOŚĆ ROZPOZNAWANIA PÓŁ	151
10.2.	KOLEJNOŚĆ WERYFIKACJI PÓŁ	152
10.3.	KOLEJNOŚĆ EKSPORTU PÓŁ.....	153
11.	SZABLONY DOKUMENTÓW	154
12.	DEFINIOWANIE WARUNKÓW POPRAWNOŚCI DANYCH	160
12.1.	OKNO GLOBALNYCH KRYTERIÓW POPRAWNOŚCI	161
12.2.	DEFINIOWANIE NOWEGO KRYTERIUM POPRAWNOŚCI	163
12.3.	NARZĘDZIE EDYTOR SKRYPTU.....	171
12.4.	IMPORTOWANIE KRYTERIÓW POPRAWNOŚCI.....	172
12.5.	TESTOWANIE ZAAWANSOWANE.....	176
12.6.	SKRYPTY W FORMANALYZER DESIGNER	178
12.6.1.	<i>Zmienne errorvar i errortxt.</i>	179
12.6.2.	<i>Zmienne przypisane polom szablonu</i>	179
12.6.3.	<i>Typy zmiennych</i>	180
12.7.	WBUDOWANE FUNKCJE ROZSZERZAJĄCE MOŻLIWOŚCI SKRYPTÓW	180
12.8.	PRZYKŁADY SKRYPTÓW	185
12.8.1.	<i>Przykład I – Skrypt weryfikujący wartość pól</i>	185
12.8.2.	<i>Przykład II - Skrypt modyfikujący wartość pól</i>	186
12.8.3.	<i>Przykład III – Wywołanie metody WebService z poziomu skryptu w języku JScript</i> 187	
12.8.4.	<i>Przykład IV – Pobranie wartości z bazy danych przypisanej do aktualnego dokumentu w języku JScript</i>	189
12.8.5.	<i>Przykład V – Wyrażenia regularne w języku JScript</i>	190
12.8.6.	<i>Przykład VI – Wyszukiwanie fraz tekstu oraz akceptacja dokumentu w języku JScript</i>	191
12.9.	SPRAWDZENIE DZIAŁANIA SKRYPTU	192

12.10.	TWORZENIE FUNKCJI WERYFIKACJI W BIBLIOTEKACH DLL	193
12.10.1.	<i>Modyfikacja wartości pola</i>	199
12.10.2.	<i>Integracja biblioteki</i>	204
12.11.	TWORZENIE FUNKCJI WERYFIKACJI PRZY UŻYCIU WEBSERVICE.	206
12.11.1.	<i>Definicja usługi WebService.</i>	206
12.11.2.	<i>Definicja zmiennych wyjściowych funkcji</i>	207
12.11.3.	<i>Tworzenie kryterium poprawności typu WebService</i>	207
12.11.4.	<i>Sprawdzanie poprawności usługi WebService</i>	211
13.	SŁOWNIKI DYNAMICZNE.	213
13.1.	CO TO SĄ SŁOWNIKI DYNAMICZNE?	213
13.2.	CZYNNOŚCI PRZEDWSTĘPNE	213
13.3.	UŻYWANIE SŁOWNIKÓW DYNAMICZNYCH W PROCESIE PRZETWARZANIA DOKUMENTÓW.....	214
13.3.1.	<i>Słowniki dynamiczne w procesie weryfikacji</i>	217
13.3.2.	<i>Słowniki dynamiczne w procesie rozpoznawania.</i>	219
14.	FORMAT DANYCH WYJŚCIOWYCH	221
14.1.	GRUPA OPCJI <i>PODSTAWOWE</i>	222
14.2.	GRUPA OPCJI <i>STRUKTURA</i>	224
14.3.	GRUPA OPCJI <i>OBRAZY STRON</i>	225
14.4.	GRUPA OPCJI <i>SKRYPT POCZĄTKOWY</i>	229
14.5.	GRUPA OPCJI <i>ARKUSZ XSLT</i>	230
14.5.1.	<i>Tworzenie arkusza XSLT</i>	231
14.5.2.	<i>Przykład arkusza XSLT</i>	231
14.5.3.	<i>Sprawdzanie poprawności arkusza XSLT</i>	232
14.6.	GRUPA OPCJI <i>SKRYPT KOŃCOWY</i>	234
15.	FILTRY	235
15.1.	FILTRY W PROCESIE SKANOWANIA I ROZPOZNAWANIA.	235
15.2.	ZASTOSOWANIE FILTRÓW W PROCESIE ROZPOZNAWANIA DOKUMENTÓW	235
16.	INNE FUNKCJE	240
16.1.	ZAMKNIĘCIE PROGRAMU	240
16.2.	INFORMACJE O PROGRAMIE	240

1. Wstęp

Dla wydajności systemu automatycznego rozpoznawania dokumentów największe znaczenie ma faza rozpoznawania treści dokumentu. Chodzi tu nie tylko o szybkość rozpoznawania, ale przede wszystkim o jego jakość. Mała liczba błędów pozwala znacząco skrócić czas weryfikacji treści dokumentów. Gdy można zastosować weryfikację automatyczną, jeszcze bardziej skraca się czas, w którym z dokumentu papierowego otrzymujemy sprawdzoną treść w postaci elektronicznej.

W systemie FormAnalyze za rozpoznawanie treści dokumentów jest odpowiedzialny moduł FormAnalyze Engine. Korzysta on z jednych z najlepszych pakietów rozpoznawania pisma ręcznego i drukowanego. Operacje pomocnicze wykonywane przez moduł (przetwarzanie wstępne, pozycjonowanie) zostały oparte na najnowszych osiągnięciach technik przetwarzania i rozpoznawania obrazu.

Przy rozpoznawaniu dokumentów *FormAnalyze Engine* korzysta z ich opisu zapisywanego w plikach konfiguracyjnych. Plik konfiguracji można traktować jako kompozycję wymienionych wyżej zaawansowanych technologicznie elementów składającą się na opis sposobu rozpoznawania formularzy (stron dokumentów) i dokumentów. Ten opis jest przygotowywany w module FormAnalyze Designer.

Plik konfiguracji ma pierwszorzędne znaczenie dla jakości rozpoznawania dokumentów. Jego tworzenie jest procesem złożonym, wykonywanym metodą kolejnych przybliżeń. Po przygotowaniu wstępnej wersji pliku konfiguracyjnego używamy go do rozpoznania testowej partii dokumentów. Analiza otrzymanych wyników pozwala na dokonanie odpowiednich poprawek w pliku. Kroki rozpoznawania testowej partii dokumentów i nanoszenia poprawek powtarzamy aż do uzyskania zadowalającego efektu.

Podręcznik użytkownika *FormAnalyze Designer*. ma za zadanie zapoznać użytkownika z procesem projektowania konfiguracji rozpoznawania dla *FormAnalyze Engine*.

Rozdział 2 - Rozpoznawanie w FormAnalyze Engine przedstawia sposób przetwarzania i rozpoznawania dokumentów w *FormAnalyze Engine*.

Rozdział 3 - Plik konfiguracji został poświęcony kwestii używania w systemie jednej bądź wielu konfiguracji.

Rozdział 4 - Podstawy obsługi FormAnalyze Designer zawiera krótkie omówienie interfejsu użytkownika FormAnalyze Designer.

Rozdział 5 - Formularze przedstawia zasady definiowania formularzy - także automatycznego rozpoznawania typu formularza.

Rozdział 6 - Pozycjonowanie pól na formularzu poświęcono pozycjonowaniu pól na formularzu.

Rozdział 7 - Pola rozpoznawane zawiera omówienie różnych typów pól rozpoznawanych używanych przy projektowaniu konfiguracji.

Rozdział 8 - Grupowanie pól jest poświęcony złożonym konstrukcjom pól rozpoznawanych – grupom pól tekstowych, wyboru oraz grupowaniu do celów wizualizacji danych w module FormAnalyze Verifier.

Rozdział 9 - Właściwości pól zawiera opis podstawowych opcji pól wykorzystywanych w przetwarzaniu dokumentu.

Rozdział 10 - Kolejność przetwarzania pól poświęcono definiowaniu kolejność przetwarzania pól w szablonie dokumentu.

Rozdział 11 - Szablony dokumentów omawia sposób składania zdefiniowanych formularzy w opis rozpoznawania dokumentu – szablon dokumentu.

Rozdział 12 - Definiowanie warunków poprawności danych omawia szczegółowo zagadnienia dotyczące skryptów weryfikacyjnych i funkcji WebService modyfikujących wartość pola.

Rozdział 13 - Słowniki dynamiczne. prezentuje użytkownikowi co to są i do czego służą słowniki dynamiczne.

Rozdział 14 - Format danych wyjściowych został poświęcony projektowaniu formatu danych wyjściowych.

Rozdział 15 - Filtry opisuje różnice w wykorzystywaniu filtrów w procesach skanowania i rozpoznawania dokumentów.

Rozdział 16 - Inne funkcje opisuje proces zamykania *FormAnalyze Designer* oraz okno informacji o programie.

2. Rozpoznawanie w FormAnalyze Engine

Program *FormAnalyze Engine* jest modułem systemu FormAnalyze odpowiedzialnym za rozpoznawanie zawartości dokumentów. Dokumenty są pobierane z bazy danych. Tam też są zapisywane wyniki rozpoznawania. Sposób przetwarzania dokumentów przez system FormAnalyze jest opisany w pliku konfiguracji. Plik ten jest przygotowywany w modułach *FormAnalyze Designer* i VTD.

Moduł *FormAnalyze Engine* rozpoznaje tylko te typy dokumentów, które zostały opisane w pliku konfiguracyjnym. W jednym pliku konfiguracyjnym może być opisanych wiele typów dokumentów. W systemie FormAnalyze typ dokumentu oznacza nie tylko rodzaj zawartości dokumentu, ale także sposób jej rozmieszczenia na stronach. Dokumenty, posiadające taką samą zawartość, ale inaczej rozmieszczoną na stronie (stronach), zostaną zdefiniowane, jako różne typy dokumentów. Można powiedzieć, że typ dokumentu oznacza unikalny sposób rozpoznawania danych znajdujących się w nim.

W pliku konfiguracji dla każdego typu dokumentu jest zdefiniowany **szablon dokumentu** - wzorzec jego rozpoznawania. W systemie FormAnalyze szablon dokumentu składa się z formularzy. Przez formularz rozumiemy tu sposób rozpoznawania **jednej strony** dokumentu. Szablon zawiera tyle formularzy, ile stron ma dokument. Szablony dokumentów jednostronicowych będą składać się tylko z jednego formularza.

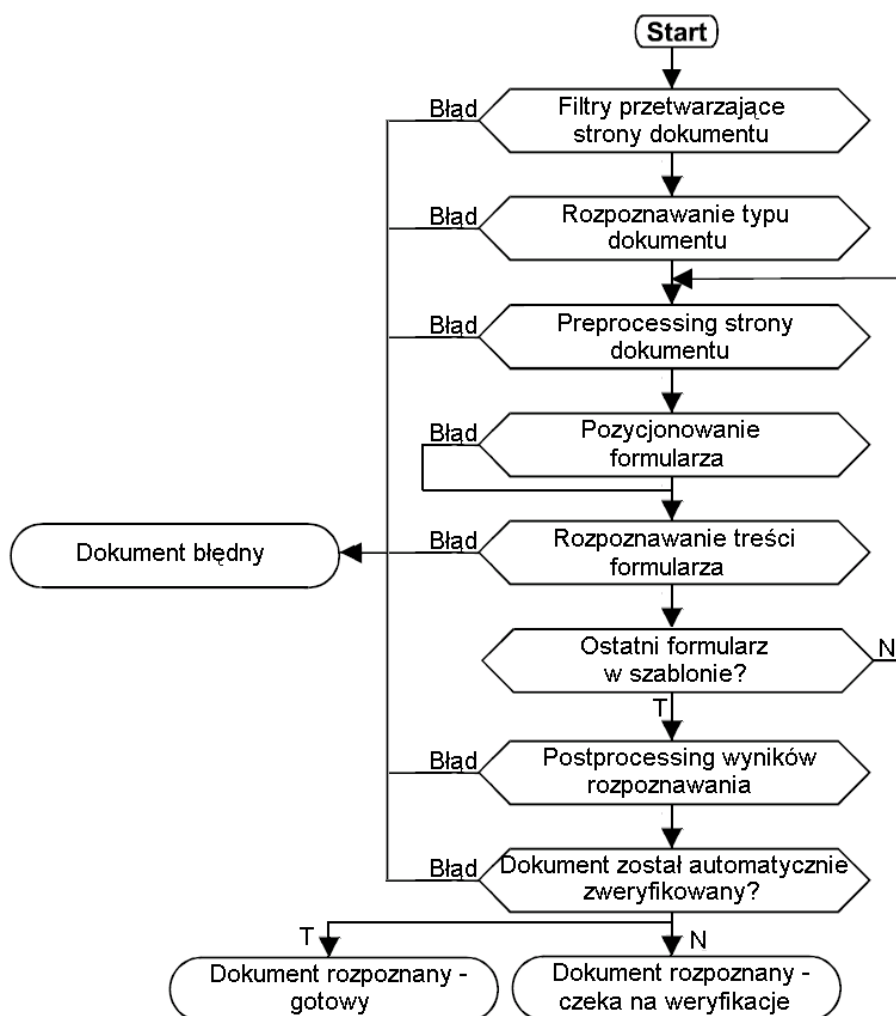
Formularze w szablonie są uporządkowane. Ich kolejność jest taka sama jak kolejność stron w dokumencie.

Pojedynczy formularz zawiera wszystkie informacje niezbędne do rozpoznania treści na nim zapisanych. Dodatkowo może zawierać informacje pozwalające na rozpoznanie typu formularza. Sposób rozpoznawania formularza omówiono dokładnie w kolejnych punktach.

Proces rozpoznawania dokumentu jest procesem złożonym. Schemat jego przedstawiono na Rys. 2-1, na stronie 8. Na początku FormAnalyze Engine pobiera z bazy danych dokument, który ma rozpoznać. Pierwszym krokiem jest ustalenie typu dokumentu, czyli szablonu, zgodnie, z którym będzie rozpoznawany dokument. Szczegóły dotyczące rozpoznawania typu dokumentu omówiono w dalszej części podręcznika. Jeżeli nie udało się rozpoznać typu dokumentu (szablonu), jest sygnalizowany błąd rozpoznania typu dokumentu, a sam zostanie dokument zaznaczony w bazie danych, jako rozpoznany niepoprawnie.

Po ustaleniu szablonu dokumentu FormAnalyze Engine przechodzi do rozpoznawania jego treści. Kolejne strony dokumentu są rozpoznawane zgodnie z formularzami zdefiniowanymi w szablonie.

W czasie rozpoznawania formularza, może wystąpić błąd, spowodowany, np. nie odnalezieniem elementów pozycjonujących na formularzu, lub błąd inicjacji pakietu rozpoznającego. W takiej sytuacji proces rozpoznawania jest przerywany.



Rys. 2-1 Rozpoznawanie dokumentu w FormAnalyzer Engine.

Wyniki rozpoznawania są zapisywane w bazie danych dopiero po rozpoznaniu wszystkich formularzy. Dokumenty te czekają na sprawdzenie przez operatorów.

2.1. Rozpoznawanie typu dokumentu

Jak już powiedzieliśmy pierwszym krokiem rozpoznawania dokumentu jest ustalenie jego typu. Ma to zasadnicze znaczenie dla poprawności rozpoznawania treści dokumentu.

W systemie FormAnalyzer dokumenty mają ściśle określoną liczbę stron. Na podstawie liczby stron są ustalane szablony, którym może odpowiadać rozpoznawany dokument. Możliwe są trzy przypadki:

W pliku konfiguracji nie ma szablonu o takiej liczbie stron. Przyczyną takiego błędu może być zaznaczenie do rozpoznawania dokumentu, który nie powinien być rozpoznawany, czyli nie odpowiada żadnemu ze zdefiniowanych szablonów.

Dokument może mieć nieprawidłową liczbę stron wskutek – pobrania z automatycznego podajnika skanera dwóch stron na raz.

Dokument, którego liczba stron nie zgadza się z liczbą formularzy w żadnym szablonie, nie może być rozpoznany. Sygnalizowany jest błąd rozpoznania typu dokumentu, a dokument jest zaznaczany w bazie danych jako rozpoznany niepoprawnie.

W pliku konfiguracji jest jeden szablon o odpowiedniej liczbie stron. W tym przypadku FormAnalyze Engine przyjmuje, że dokument jest zgodny tym właśnie szablonem.

W pliku konfiguracji jest więcej szablonów o takiej liczbie stron. W takiej sytuacji FormAnalyze Engine ustala typ dokumentu na podstawie typu formularza pierwszej strony. Rozpoznawanie typu formularza zostało omówione szczegółowo w punkcie 2.2.

Ustalenie typu dokumentu na podstawie tylko pierwszej strony wprowadza pewne ograniczenie. Dokumenty o takiej samej liczbie stron muszą mieć różne pierwsze strony (formularze). W innym przypadku, dla dokumentów wielostronicowych, jest rozpoznawany tylko pierwszy formularz, co znacznie przyspieszy proces rozpoznawania.

Po ustaleniu typu dokumentu można przystąpić do jego rozpoznawania. Z szablonu dokumentu są pobierane kolejne formularze i zgodnie z nimi są rozpoznawane kolejne strony dokumentu. Po skompletowaniu wyników rozpoznania wszystkich stron są one zapisywane w bazie danych.

W kolejnych punktach zostaną przedstawione poszczególne kroki przetwarzania formularza wykonywane przez FormAnalyze Engine.

2.2. Rozpoznawanie typu formularza

FormAnalyze Engine rozpoznaje typ formularza (jednej strony dokumentu) tylko wtedy, gdy jest to konieczne i nie zostało zdefiniowane uprzednio przez operatora. Pozostałe strony wielostronicowego dokumentu są domniemywane na podstawie rozpoznanego pierwszego formularza dokumentu.

FormAnalyze Engine stosuje trzy metody rozpoznawania typu formularza:

- Rozpoznanie nazwy formularza na stronie dokumentu.

Ta metoda jest wykorzystywana głównie dla dokumentów projektowanych z myślą o przetwarzaniu w systemach automatycznego rozpoznawania formularzy. Na formularzu jest umieszczane pole (np. z kodem paskowym) zawierające nazwę formularza.

- Rozpoznanie wyglądu formularza.

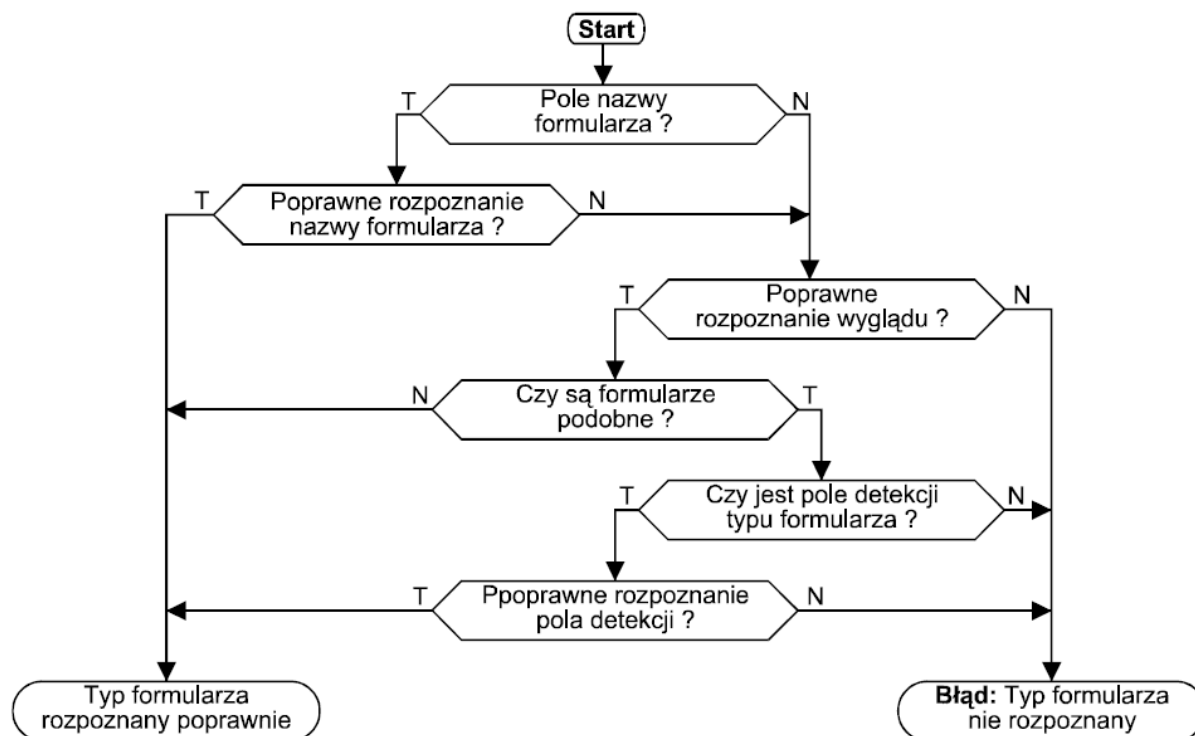
W tym przypadku są stosowane zaawansowane techniki klasyfikacji wzorców, które badają podobieństwo pomiędzy rozpoznawaną stroną a stronami formularzy zapamiętanymi w pliku konfiguracji. Ta metoda daje bardzo dobre wyniki, o ile formularze wystarczająco różnią się między sobą, tzn., gdy na pierwszy rzut oka możemy powiedzieć, że dwa formularze są różne.

- Rozpoznanie pola detekcji typu formularza

W pliku konfiguracji mogą znaleźć się formularze, których nie da się jednoznacznie odróżnić poprzednią metodą. Pole detekcji typu formularza służy do szczegółowego rozróżnienia podobnych z wyglądu formularzy.

Od wersji 4.0 w programie FormAnalyzer istnieje możliwość zdefiniowania szablonu dokumentu, co spowoduje przypisanie odpowiedniego szablonu do skanowanego dokumentu, oraz ominięcie całego procesu rozpoznawania. Kolejne strony dokumentu będą rozpoznawane zgodnie z odpowiednimi formularzami zapamiętanymi w szablonie. Szczegółowe informacje dotyczące wskazywania szablonów dokumentów znajdują się w podręczniku do modułu FormAnalyzer Scan&Administrator.

Algorytm rozpoznawania typu formularza przedstawiono na poniższym schemacie.



Rys. 2-2 Rozpoznawanie typu formularza w FormAnalyzer Engine.

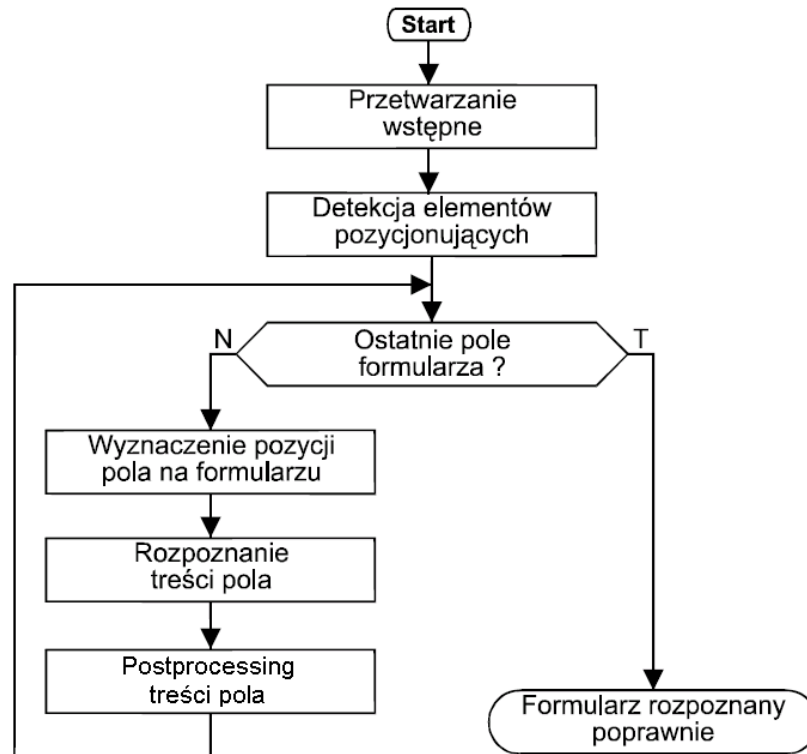
Jeżeli typu formularza nie udało się prawidłowo rozpoznać przetwarzanie dokumentu jest przerywane, a dokument zostaje zaznaczony w bazie danych jako błądny.

2.3. Przetwarzanie i rozpoznawanie zawartości formularzy

Obraz strony dokumentu, który powstał w wyniku skanowania dokumentu papierowego może znacznie różnić się od strony wzorcowej, dla której został zaprojektowany formularz. Te różnice wynikają z procesu tworzenia formularza oraz wprowadzania go do systemu przetwarzania. Formularze drukowane w różnych warunkach i na różnym sprzęcie mogą znacznie różnić się między sobą. Mogą być także skanowane z inną orientacją niż wymagana przy rozpoznawaniu. Dodatkowo, skanowanie wprowadza przekos obrazu dokumentu, czyli obrót obrazu o stosunkowo niewielki kąt w stosunku do oryginału.

Aby uzyskać odpowiednią jakość rozpoznawania FormAnalyzer Engine wykonuje procedurę, której schemat został pokazany poniżej.

Krok przetwarzania wstępnego ma za zadanie dostosować obraz dokumentu do wzorca strony, z którego korzysta formularz: dostosowanie orientacji strony oraz usunięcie przekosu (obrotu obrazu), który mógł powstać w trakcie skanowania.



Rys. 2-3 Rozpoznawanie treści formularza w FormAnalyzer Engine.

Detekcja elementów pozycjonujących, polega na odszukaniu na obrazie strony charakterystycznych elementów, względem których jest zapamiętane położenie pól rozpoznawanych. Odszukane elementy pozycjonujące umożliwiają wyznaczenie położenia interesujących nas treści na rozpoznawanej stronie dokumentu. W przypadku, gdy nie udało się ustalić położenia przynajmniej jednego elementu pozycjonującego, jest sygnalizowany błąd pozycjonowania. Taka strona nie może być rozpoznawana.

Dysponując elementami pozycjonującymi możemy przejść do rozpoznawania pól zawierających interesujące nas informacje. Dla każdego pola najpierw jest wyznaczane jego położenie na stronie. Korzysta się przy tym ze znanego położenia elementów pozycjonujących. Znając położenie pola FormAnalyzer Engine przystępuje do rozpoznawania jego treści. W zależności od sposobu wypełnienia pola (pismo drukowane, ręczne, kod paskowy) są stosowane różne metody rozpoznawania treści pola. FormAnalyzer Engine pozwala na rozpoznawanie pisma drukowanego i ręcznego, kodów paskowych oraz pól wyboru.

W kolejnych punktach przedstawiono bardziej szczegółowo poszczególne kroki przetwarzania.

2.4. Przetwarzanie wstępne

Krok przetwarzania wstępnego ma za zadanie doprowadzić rozpoznawaną stronę do postaci znormalizowanej, tzn. zgodnej z postacią strony wzorcowej, dla której była tworzona formuła rozpoznawania. Rozbieżność między obrazem strony w bazie danych, a stroną wzorcową wynika głównie z procesu skanowania.

Dokument może być skanowany w innej orientacji niż strona(y) wzorcowe. Takie skanowanie może być wymuszone rozmiarami dokumentu. Np. strony formatu A3 są skanowane w orientacji portretowej, podczas gdy informacje na nich są zapisane najczęściej w orientacji krajobrazowej. Dla bardzo małych dokumentów może się okazać, że tylko w jednej orientacji dokumentu (niezgodnej z wzorcem) skaner pracuje poprawnie. Często stosuje się również zmianę orientacji dla przyspieszenia samego skanowania: strony formatu A4 są skanowane na ogół szybciej w orientacji krajobrazowej niż portretowej.

Uzyskanie właściwej orientacji dokumentu przed jego rozpoznawaniem uzyskuje się zadając wykonanie operacji obrotu o odpowiedni kąt, będący wielokrotnością 90°. Skanowanie dokumentów - niezależnie od jakości podajnika automatycznego - wiąże się zawsze z wprowadzeniem do obrazu większego lub mniejszego przekosu. Zaznaczenie opcji usuwania przekosu powoduje automatyczne wyznaczenie odpowiedniego kąta i obrotu strony, aby go usunąć.

Po wykonaniu przetwarzania wstępnego formularz jest gotowy do odszukania w nim elementów umożliwiających określenie współrzędnych elementów pozycjonujących.

2.5. Pozycjonowanie pól na formularzu

Położenia pól rozpoznawanych, określone w pliku konfiguracji, nie odpowiadają położeniom pól na zeskanowanych formularzach. Różnice w położeniu wynikają z różnych przyczyn. Na samych formularzach drukowanych na różnym sprzęcie i w różnych warunkach mogą występować niewielkie różnice w położeniu pól. Użycie innego typu skanera, zmiana parametrów skanowania, a przede wszystkim prowadzenie papieru poprzez mechanizm skanera, mogą dawać znaczące dla jakości rozpoznawania przemieszczenia pól w stosunku do wzorca.

FormAnalyze Engine przed przystąpieniem do rozpoznawania pól formularza ustala na nim położenie elementów pozycjonujących. W pliku konfiguracyjnym jest zapamiętane położenie pól rozpoznawanych względem pól pozycjonujących. Złożenie tych dwóch informacji - współrzędnych elementu pozycjonującego na formularzu rozpoznawanym i względnego położenia pola z pliku konfiguracji - pozwala dokładnie obliczyć położenie pola rozpoznawanego na formularzu.

FormAnalyze Engine korzysta z dwóch typów elementów pozycjonujących:

- Znacznik pozycjonowania

Jest to wzorec geometryczny utworzony przez przecinające się linie. Tego typu wzorce można spotkać niemal na każdym formularzu (np. narożniki tabel).

- Obszar pozycjonowania

Jest wyróżnionym elementem graficznym (innego typu niż znacznik), względem, którego pola rozpoznawane mają stałą pozycję.

Elementy pozycjonujące są wykrywane na formularzu w ograniczonym sąsiedztwie pozycji zaznaczonej na wzorcu. Może się zdarzyć, że na formularzu nie zostanie wykryty z odpowiednim poziomem zaufania żaden element pozycjonujący. W tym przypadku jest sygnalizowany błąd pozycjonowania formularza.

W przypadku, gdy na formularzu nie zostaną wykryte niektóre z elementów pozycjonujących, ich rolę przejmują pozostałe elementy.

Gdy stosowane są złożone formularze, w których wymagana jest duża precyzja pozycjonowania można skorzystać z pozycjonowania dwupoziomowego. Najpierw na formularzu są odszukiwane podstawowe elementy pozycjonujące. Dopiero względem odzyskanych współrzędnych elementów podstawowych oblicza się obszary poszukiwania pozostałych elementów pozycjonujących.

2.6. Przetwarzanie i rozpoznawanie pól

Kolejnym etapem - po wyznaczeniu współrzędnych elementów pozycjonujących - jest rozpoznawanie pól.

Pierwszym krokiem wykonywanym w tym etapie jest obliczenie położenia pola na formularzu. Na podstawie współrzędnych najbliższych elementów pozycjonujących (oczywiście tylko tych, które zostały wykryte z odpowiednim poziomem zaufania w poprzednim etapie) są obliczane współrzędne środka pola rozpoznawanego. Dalej jest wyznaczany obszar zajmowany przez dane pole na formularzu.

Ze względu na różne typy pól używanych w FormAnalyze Engine dalsze przetwarzanie może się istotnie różnić w zależności od typu pola. FormAnalyze używa pól następujących typów:

- Pole obrazu

Ten typ pola jest stosowany, gdy chcemy zaprezentować w weryfikacji fragment obrazu formularza niepokrywający się z żadnym polem rozpoznawanym lub nie rozpoznawany w ogóle (np. podpis). Dla tego typu pola wyznacza się jedynie współrzędne obrazu na formularzu.

- Pole przetwarzania wstępnego

W obszarze, który zajmuje pole tego typu, są wykonywane operacje przetwarzania obrazu. Mogą to być: usuwanie szumu, usuwanie linii i wygładzanie. Rodzaj i parametry tych operacji są właściwościami danego pola przetwarzania wstępnego.

- Pole wyboru

W polu wyboru jest rozpoznawane, czy pole wyboru zostało zakreślone czy też nie.

- Pole kodu paskowego

W polu tego typu jest rozpoznawany tekst reprezentowany jako kod paskowy. Można określić konkretny rodzaj kodu paskowego (wybierając ewentualnie

dodatkowe opcje) lub pozostawić wykrycie typu kodu procedurze rozpoznawania kodu paskowego.

- Pole drukowane

Pole drukowane jest rozpoznawane z wykorzystaniem modułu rozpoznawania znaków drukowanych. Przed przekazaniem obrazu pola do modułu rozpoznawania jest na nim wykonywane przetwarzanie wstępne, którego parametry można zdefiniować. Poprawność rozpoznawania pola można osiągnąć podając dodatkowe informacje, dotyczące zawartej w nim treści: czy może to być dowolny napis czy też np. data lub liczba; w jakim języku jest wypełniane pole, czy pole jest jedno- czy wiele liniowe itd.

W wynikach otrzymanych z modułu rozpoznawania jest wykonywane przetwarzanie końcowe. Jego zadaniem jest wygenerowanie wyniku rozpoznania w określonym formacie. W przypadku, gdy dla pola nie rozpoznano żadnych znaków (wynik rozpoznania jest pusty), jako wynik jest używana wartość domyślna.

- Pole wypełnione drukiem lub ręcznie

Podstawową różnicą w stosunku do poprzedniego typu pola jest użycie innego modułu rozpoznawania, przeznaczonego do rozpoznawania pisma ręcznego. Jest to podstawowa funkcja tego modułu - oprócz tego potrafi on także rozpoznawać pismo drukowane. Podobnie jak poprzednio, można podać dodatkowe informacje dotyczące spodziewanej treści pola. Moduł rozpoznawania pisma ręcznego pracuje jedynie w języku polskim, natomiast jest możliwość wyboru jednego z kilku klasyfikatorów: m.in. pisma drukowanego, pisma ręcznego, cyfr pisanych ręcznie lub drukowanych.

Jak poprzednio przed rozpoznawaniem jest wykonywane przetwarzanie wstępne, natomiast po nim przetwarzanie końcowe.

- Pole RecoStar Engine

Charakterystyka tego pola jest bardzo podobna do przedstawionego wyżej pola pismo wypełnione drukiem lub ręcznie. Do rozpoznawania pisma ręcznego i drukowanego jest tu używany moduł RecoStar.

- Pole rozpoznawaniu typu formularza

To pole jest używane do rozróżniania formularzy na podstawie charakterystycznych elementów graficznych znajdujących się na formularzu. (Patrz punkt 5.3 - Rozpoznanie typu formularza - pole rozpoznawania typu.)

Z polami, których wynikiem rozpoznania jest tekst, można łączyć słowniki bazodanowe. Mogą one być wykorzystane do poprawienia jakości rozpoznawania, przez wybranie wartości ze słownika znajdującej się „najbliżej” rozpoznanej treści (w przypadku, gdy różnica między wartością rozpoznaną a ze słownika jest zbyt duża, rozpoznana treść zostaje niezmienną).

Dla niektórych typów pól można zdefiniować skrypty modyfikujące wartość pola. Takie skrypty są wykonywane w czasie rozpoznawania treści dokumentu - zapewnia to, że skrypt modyfikujący wartość zostanie wykonany tylko raz dla danego dokumentu. Tzw. skrypty lokalne, korzystające z wartości tylko jednego pola, są wykonywane bezpośrednio po rozpoznaniu treści pola. Skrypty globalne, korzystające z rozpoznanych wartości wielu pól dokumentu, są wykonywane dopiero po rozpoznaniu wszystkich pól dokumentu.

Drugą metodą wykonania procedur modyfikujących wartość pola jest łączyć z polem funkcji z zewnętrznej biblioteki DLL. Wykorzystanie funkcji modyfikujących wartość jest

bardzo podobne do skryptów: są one wykonywane w czasie rozpoznawania treści dokumentu, funkcje lokalne wykonuje się bezpośrednio po rozpoznaniu treści pola, a funkcje globalne po rozpoznaniu wszystkich pól dokumentu.

2.7. Grupy pól

W niektórych przypadkach korzystnie jest podzielić treść pola na mniejsze fragmenty, rozpoznawane niezależnie od siebie. Takie podejście stosuje się, gdy dla tych fragmentów można uzyskać lepsze wyniki rozpoznawania. Po rozpoznaniu fragmenty należy połączyć w jedną wartość.

Przykładem może być tutaj seria i numer dowodu osobistego. Jeśli podzielimy to pole na dwa: serię i numer, to dla serii będą rozpoznawane tylko litery, a dla numeru tylko cyfry. W ten sposób osiąga się na ogół lepszą jakość rozpoznawania.

Do połączenia wartości pól rozpoznawanych są używane grupy pól. W zależności od typu pól wchodzących w skład grupy, różne jest ich wykorzystanie.

W przypadku grup pól wyboru zyskujemy możliwość kontroli poprawności rozpoznanych danych. Polega to na określeniu dla grupy minimalnej i maksymalnej liczby zaznaczonych pól wyboru. Jeżeli liczba zaznaczonych pól wyboru nie mieści się w podanym przedziale, oznacza to błąd rozpoznawania (lub niepoprawne wypełnienie formularza).

Bardziej złożone przetwarzanie ma miejsce w przypadku grup pól tekstowych. (Przez pola tekstowe rozumiemy pola, których wartością może być tekst, czyli pola drukowane, wypełnione drukiem lub ręcznie i kodów paskowych). Grupy pól tekstowych są wykorzystywane na dwa sposoby:

- Składanie wartości rozpoznanej
W tym przypadku, wartość rozpoznana jest składana z fragmentów rozpoznanych dla poszczególnych pól oraz separatorów. Najważniejszym zastosowaniem tego trybu grupowania jest dokładniejsze określenie rodzaju danych, które powinny się pojawić w pojedynczych polach.
- Wybór lepszej wartości
W wielu przypadkach, przy rozpoznawaniu danego pola, nie da się z góry określić, jaka metoda rozpoznawania da najlepszy wynik. Co więcej, dla różnych dokumentów, stosując różne metody rozpoznawania tego samego pola, możemy ostatecznie uzyskać o wiele lepsze wyniki w procesie rozpoznawania. W takim przypadku grupuje się pola rozpoznawane z zastosowaniem różnych metod (i/lub różnych parametrów rozpoznawania). Po rozpoznaniu spośród tych pól, które spełniają kryteria poprawności jest wybierane to, które ma najwyższy współczynnik zaufania.

2.8. Dokumenty wielostronicowe

Jak już wcześniej zostało powiedziane, wszystkie typy dokumentów są w systemie FormAnalyzer definiowane jako szablony dokumentów. Szablony dokumentów składają się z formularzy definiujących przetwarzanie i rozpoznawanie poszczególnych stron dokumentów.

Bardzo istotne jest, że szablon dokumentu zawiera ściśle określoną liczbę formularzy. Zatem szablon może być stosowany do rozpoznawania dokumentów o takiej właśnie

liczbie stron. System FormAnalyze nie pozwala na rozpoznawanie dokumentów o nieokreślonej liczbie stron.

Z opisu rozpoznawania typu dokumentu wynika, że szablon dokumentu jest w pierwszej kolejności rozpoznawany na podstawie liczby stron dokumentu. Dopiero, gdy mamy więcej niż jeden szablon o tej samej liczbie stron, jest rozpoznawany typ formularza dla pierwszej strony dokumentu. Odpowiedni typ formularza decyduje o wyborze szablonu dla rozpoznawanego dokumentu.

Dla kolejnych stron dokumentu wielostronicowego typ formularza nie jest rozpoznawany - znamy szablon dokumentu, w którym są podane typy formularzy wszystkich stron.

Takie rozwiązanie przyspiesza znacznie proces rozpoznawania dokumentów składających się z większej liczby stron. Wprowadza jednak pewne ograniczenie: nie możemy zdefiniować dwóch lub więcej szablonów mających tyle samo stron i pierwszą stronę o tym samym typie formularza.

Z punktu widzenia algorytmów przetwarzania i rozpoznawania najbardziej istotne operacje są wykonywane na pojedynczej stronie (formularzu). Dokument wielostronicowy jest zatem zbiorem formularzy. Dla każdego formularza z tego zbioru są wykonywane operacje opisane w poprzednich punktach. Wyniki rozpoznawania kolejnych formularzy są akumulowane, aż do momentu rozpoznania wszystkich formularzy dokumentu. Ma to kilka istotnych konsekwencji:

- Wyniki rozpoznawania są zapisywane dla dokumentu jako całości, bez sztucznego - z punktu widzenia treści dokumentu - podziału na strony.
- Wystąpienie poważnego błędu w trakcie rozpoznawania dowolnej strony dokumentu powoduje zaznaczenie całego dokumentu jako rozpoznanego niepoprawnie.

3. Plik konfiguracji

FormAnalyze Designer jest używany do stworzenia pliku konfiguracji zawierającego informacje dotyczące dokumentów i formularzy, które rozpoznaje FormAnalyze Engine.

Plik konfiguracji składa się z szablonów dokumentów. Każdy szablon dokumentu odpowiada jednemu typowi dokumentu, który może być rozpoznawany przez FormAnalyze Engine.

Szablon dokumentu jest uporządkowaną listą formularzy (stron) dokumentów. Jest on niezbędny nawet dla dokumentów jednostronicowych - składa się wtedy z jednego formularza. Tworzenie i modyfikacja szablonów dokumentów została omówiona w rozdziale - Szablony dokumentów.

Każdy formularz zawiera wszystkie informacje niezbędne do poprawnego przetwarzania i rozpoznawania pojedynczej strony dokumentu. Sposób definiowania formularzy oraz ich elementów został szczegółowo omówiony w następnych rozdziałach. (Obok formularzy dopełnienie szablonu dokumentu stanowią tzw. globalne kryteria poprawności, które konfrontują wartości odczytane z różnych formularzy).

Przed przystąpieniem do projektowania poszczególnych formularzy i szablonów dokumentów należy podjąć kilka decyzji o dużym wpływie na działanie całego systemu.

Podstawową z nich jest kwestia liczby plików konfiguracji, które będą rutynowo używane w systemie. Można przyjąć dwa rozwiązania:

- Wszystkie rozpoznawane szablony dokumentów (a zatem i formularze) są zgrupowane w jednym pliku konfiguracji. Takie rozwiązanie jest często stosowane w konfiguracjach systemu FormAnalyze obsługujących niedużą liczbę typów dokumentów. Jego zalety to:
 - Praca FormAnalyze Engine w ustalonej, niezmiennionej konfiguracji;
 - Wszystkie dokumenty (zdefiniowanych typów) mają szansę być rozpoznane.

Przy większej liczbie zdefiniowanych formularzy trudność przy takim rozwiązaniu może stanowić rozróżnienie typów formularzy. Nawet przy optymalnym zdefiniowaniu metod rozpoznawania typu formularza mogą pojawiać się błędy polegające na niepoprawnej klasyfikacji formularza.

- Jest wiele plików konfiguracji zawierających formularze pogrupowane według pewnych kryteriów. To rozwiązanie jest stosowane w przypadku, gdy można przetwarzać dokumenty grupami zawierającymi małe liczby typów formularzy.

Podstawową zaletą tego rozwiązania jest uproszczenie fazy projektowania sposobu rozpoznawania typu formularza. Ze względu na mniejszą liczbę typów uzyskujemy większą pewność klasyfikacji formularzy. Wadami tego rozwiązania są:

- FormAnalyze Engine rozpoznaje tylko typy dokumentów zdefiniowane w aktualnym pliku konfiguracji
- Operatorzy obsługujący FormAnalyze Engine mogą pomylić pliki konfiguracyjne FormAnalyze Engine.

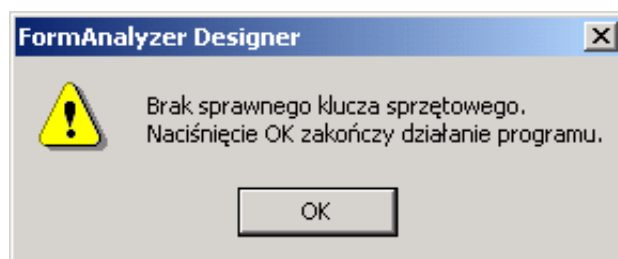
4. Podstawy obsługi FormAnalyze Designer

W tym rozdziale zostaną przedstawione podstawowe zasady posługiwania się programem FormAnalyze Designer.

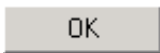
4.1. Uruchomienie programu

FormAnalyze Designer uruchamiamy klikając dwukrotnie na ikonie programu , jeżeli znajduje się ona na pulpicie systemu Windows. Drugą metodą uruchomienia programu jest wybranie z grupy programów FormAnalyze programu FormAnalyze Designer.

Program jest chroniony kluczem sprzętowym podłączonym do portu równoległego komputera. Jeśli w trakcie uruchamiania programu nie zostanie wykryty klucz sprzętowy FormAnalyze Designer to na ekranie wyświetlony zostanie poniższy komunikat.



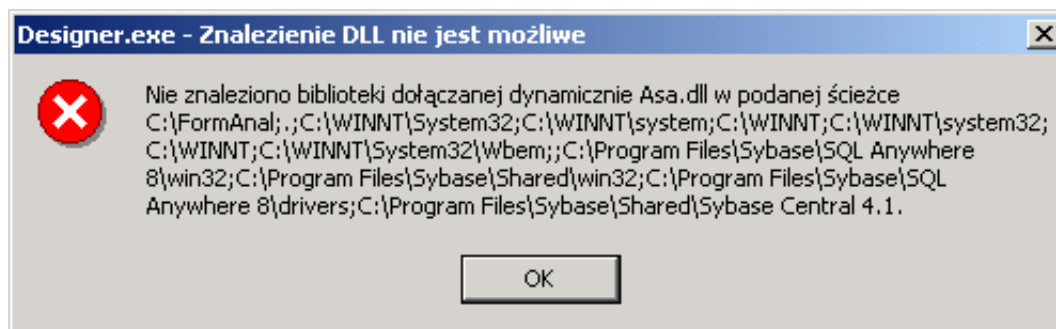
Rys. 4-1 Komunikat o nie wykryciu klucza sprzętowego.

Kliknięcie przycisku  w oknie dialogowym zakończy działanie programu.

Powyższa sytuacja może być spowodowana trzema przyczynami:

- Do portu równoległego komputera nie włożono klucza sprzętowego. Przed ponowną próbą uruchomienia programu należy umieścić klucz w porcie.
- Klucz sprzętowy jest niedokładnie włożony do gniazda portu. Trzeba sprawdzić sposób włożenia klucza do portu i - po ewentualnym poprawieniu umieszczenia - ponownie uruchomić program.
- Jeżeli wykluczylismy obie poprzednie możliwości, może to wskazywać na uszkodzenie portu równoległego.

FormAnalyze Designer korzysta z kilku dynamicznie dołączanych bibliotek. W przypadku, gdy nie jest możliwe załadowanie którejś z nich, jest wyświetlany standardowy komunikat systemu Windows. Poniżej zaprezentowano przykład takiego komunikatu dla pliku *STAT.DLL*



Rys. 4-2 Komunikat o braku wymaganego pliku: Asa.dll.

Wystąpienie takiego błędu oznacza, że instalacja programu uległa uszkodzeniu. Należy powtórzyć instalację programu FormAnalyze Designer.

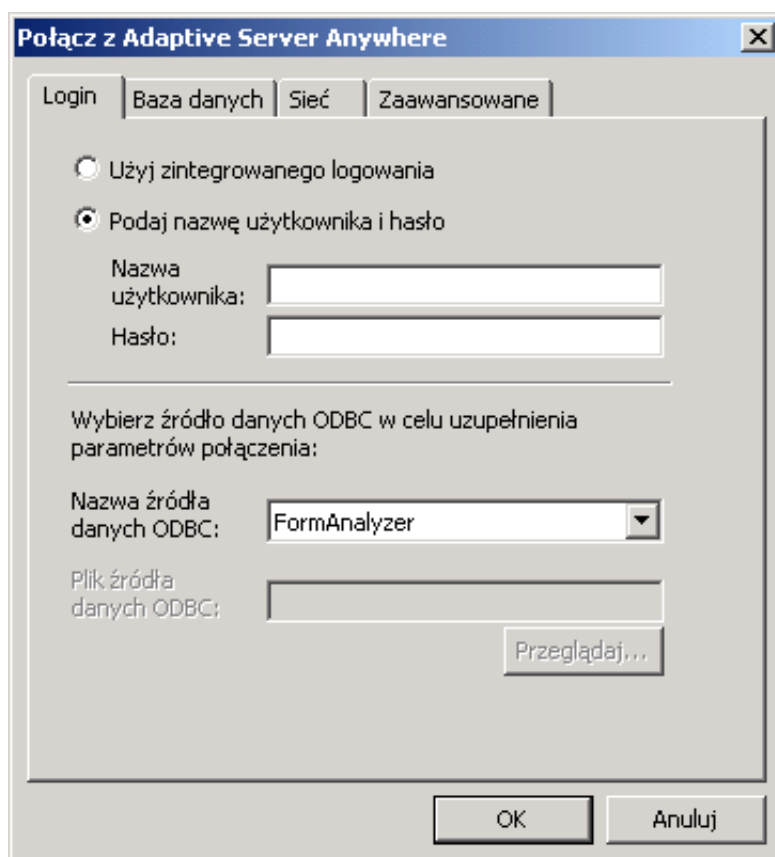
Kiedy wszystkie wymagane do uruchomienia programu składniki zostaną odnalezione, program przechodzi do nawiązania połączenia z bazą danych.

4.2. Podłączenie do bazy danych

Pierwszą czynnością, jaką należy wykonać po uruchomieniu programu FormAnalyze Designer jest podłączenie do bazy danych. Do podłączenia się do bazy danych służy okno dialogowe *Połącz z Adaptive Server Anywhere*.

Dane, które należy wprowadzić w tym oknie, mają umożliwić identyfikację bazy danych, jej użytkownika, sposobu podłączenia do niej i opcje tego połączenia.

Parametrami identyfikującymi użytkownika są: jego nazwa i hasło w bazie danych. Należy je wpisać w pierwszej zakładce *Login*, odpowiednio w pola *Nazwa użytkownika* i *Hasło*.

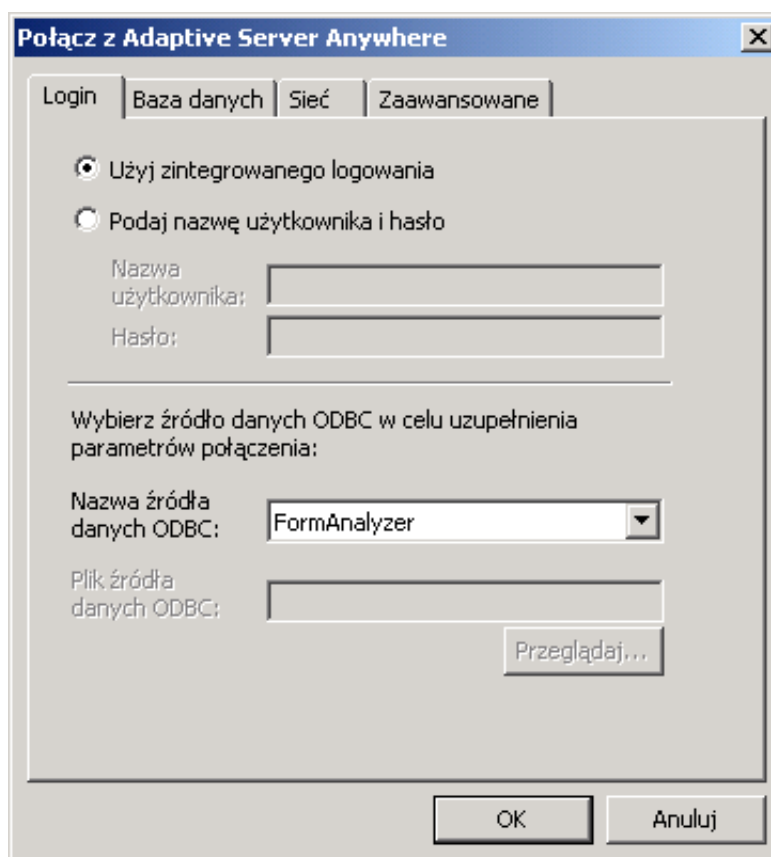


Rys. 4-3 Okno podłączenia do bazy danych – pierwsza zakładka.

W niektórych systemach, parametry te ustalane są na podstawie identyfikatora użytkownika w systemie Windows. Na poniższym rysunku pokazano okno z zaznaczoną opcją *Użyj zintegrowanego logowania*. W tym przypadku, pola *Nazwa użytkownika* i *Hasło* są nieaktywne. Stosowna zawartość tych pól jest automatycznie wyznaczana w serwerze bazy danych, na podstawie nazwy użytkownika, z którą połączyliśmy się do serwera.

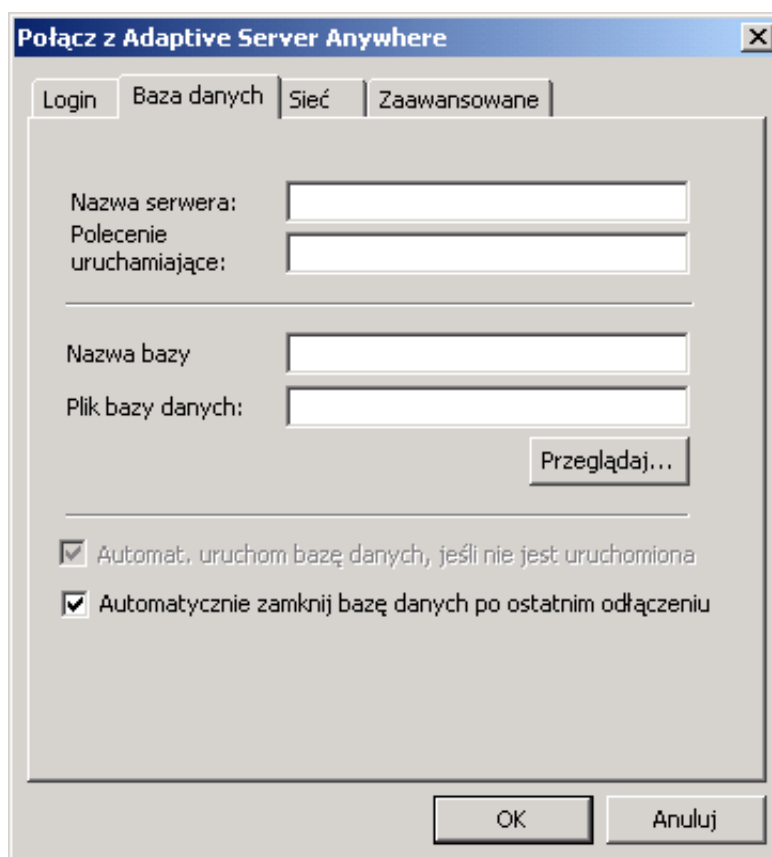
W dolnej części zakładki podaje się informacje związane ze źródłem danych ODBC.

Źródło danych ODBC służy do przechowywania parametrów połączenia z konkretną bazą danych. Z reguły potrzebne źródła danych ODBC już są zainstalowane w systemie operacyjnym i wystarczy wybrać odpowiadające nam źródło z listy rozwijalnej przy napisie *Nazwa źródła danych ODBC*. Z reguły będzie to *FormAnalyze*. W momencie wyboru pozycji z listy rozwijalnej pole *Plik źródła danych ODBC* przestaje być dostępne.



Rys. 4-4 Okno podłączenia do bazy danych z wybranym zintegrowanym logowaniem

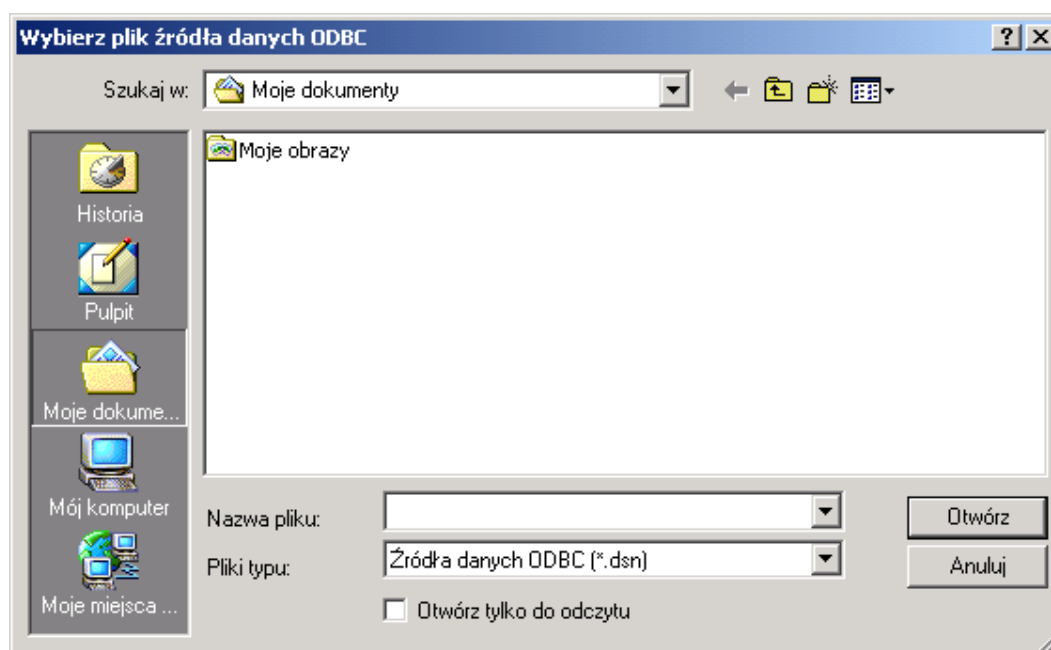
Jeżeli interesujące nas źródło danych ODBC nie zostało zainstalowane w systemie możemy podać plik, w którym są zapisane ustawienia. W tym celu, należy kliknąć przycisk **Przeglądaj...**. Na ekranie zostanie wyświetlone standardowe okno otwierania pliku, ze zdefiniowanym rozszerzeniem plików *.dsn (w oknie widoczne będą wyłącznie foldery potrzebne do nawigacji oraz pliki źródeł danych ODBC). Po odnalezieniu odpowiedniego pliku, należy zaznaczyć go, a następnie kliknąć przycisk **Otwórz**. W polu *Plik źródła danych ODBC* pojawia się ścieżka dostępu do wskazanego pliku.



Rys. 4-5 Okno połączenia do bazy danych – druga zakładka.

Druga zakładka okna Połącz z Adaptive Server Anywhere służy do wprowadzenia informacji o bazie danych, do której chcemy się podłączyć.

Ważnym elementem przy wprowadzaniu parametrów w odpowiednie pola okna dialogowego jest fakt, czy chcemy połączyć się z bazą lokalną, czy sieciową.

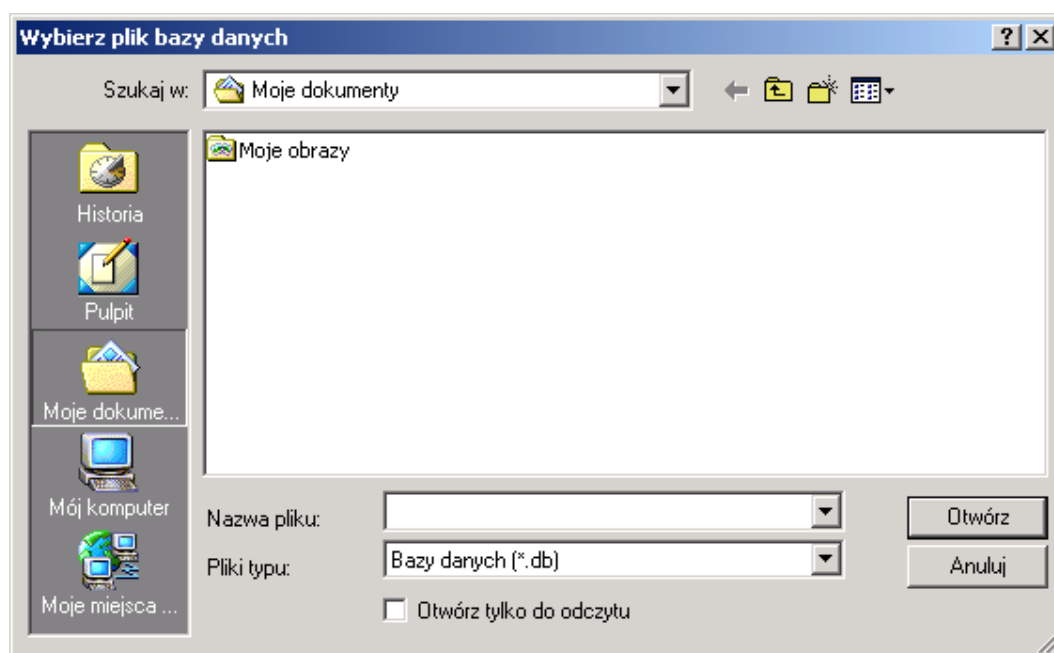


Rys. 4-6 Okno otwierania pliku źródła danych ODBC

W systemie FormAnalyze najczęściej pracuje się z sieciową bazą danych. W tym przypadku, należy wypełnić pole *Nazwa Serwera*, wpisując tam nazwę serwera obsługującego bazę danych FormAnalyze. Najczęściej stosuje się konfigurację, w której nazwa serwera jest taka sama jak nazwa bazy danych. W takim przypadku nie trzeba wypełniać pola *Nazwa bazy*. Jeżeli jednak obie te nazwy różnią się od siebie, pole *Nazwa bazy* będzie służyło do wprowadzenia nazwy bazy danych, do której chcemy się podłączyć.

W polu *Polecenie uruchamiające* możemy wpisać nazwę i parametry programu obsługi bazy danych (lub połączenia z bazą danych). Informacje na temat programów obsługi i ich parametrów można znaleźć w dokumentacji Sybase SQL Anywhere.

Lokalną bazę danych stosuje się znacznie rzadziej. W takim przypadku z bazy danych może korzystać tylko jeden użytkownik, na którego komputerze uruchomiono połączenie lokalne. Identyfikacja użytkownika jest taka sama jak w połączeniu sieciowym: należy podać nazwę i hasło konta użytkownika bazy danych. Identyfikacja bazy danych polega na wskazaniu pliku bazy danych: w polu *Plik bazy danych* podajemy pełną ścieżkę, nazwę oraz rozszerzenie do pliku bazy danych. W tym celu można także posłużyć się standardowym oknem wyszukiwania plików, klikając uprzednio przycisk **Przeglądaj...**. Po wskazaniu odpowiedniego pliku, należy zaznaczyć go, a następnie kliknąć przycisk **Otwórz**. W polu *Plik bazy danych* pojawia się ścieżka dostępu do tego pliku.



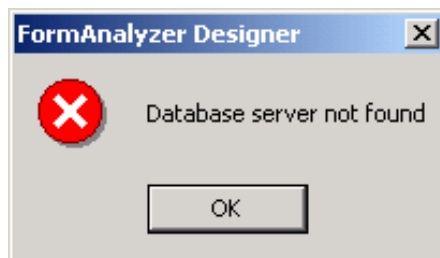
Rys. 4-7 Okno otwierania pliku bazy danych

Dodatkowo na tej zakładce możemy zaznaczyć opcję *Automatycznie zamknij bazę danych po ostatnim odłączeniu*, co spowoduje zamknięcie bazy danych po odłączeniu się od niej ostatniego zalogowanego użytkownika.

Jeżeli podłączamy się do bazy, która nie jest aktualnie włączona trzeba koniecznie zaznaczyć opcję *Automat. Uruchom bazę danych, jeśli nie jest uruchomiona*, co spowoduje uruchomienie bazy danych, do której chcemy się podłączyć.

W czasie nawiązywania łączności z bazą danych mogą wystąpić błędy. Są one sygnalizowane wyświetleniem okna komunikatu zawierającego ogólne określenie błędu.

Na poniższym rysunku przedstawiono przykładowe okno komunikatu sygnalizujące błąd związany z bazą danych



Rys. 4-8 Okno komunikatu bazy danych.

Najczęstszą przyczyną powstawania błędów przy podłączaniu się do bazy danych są nieprawidłowo wypełnione poszczególne pola okna dialogowego Połącz z Adaptive Server Anywhere. Na ogół właściwy jest następujący tryb postępowania: powtórne wprowadzenie parametrów w poszczególne pola okna dialogowego i druga próba podłączenia do bazy danych. Jeżeli połączenia nie udało się uzyskać trzeba sprawdzić, czy przyczyna błędu jest spowodowana niepoprawną konfiguracją systemu (np. plik bazy danych nie jest plikiem SQL Anywhere, nie został uruchomiony serwer bazy danych, lub wystąpił brak połączenia sieciowego itd.). Poniżej przedstawiono opis najczęściej występujących błędów przy podłączeniu do bazy danych:

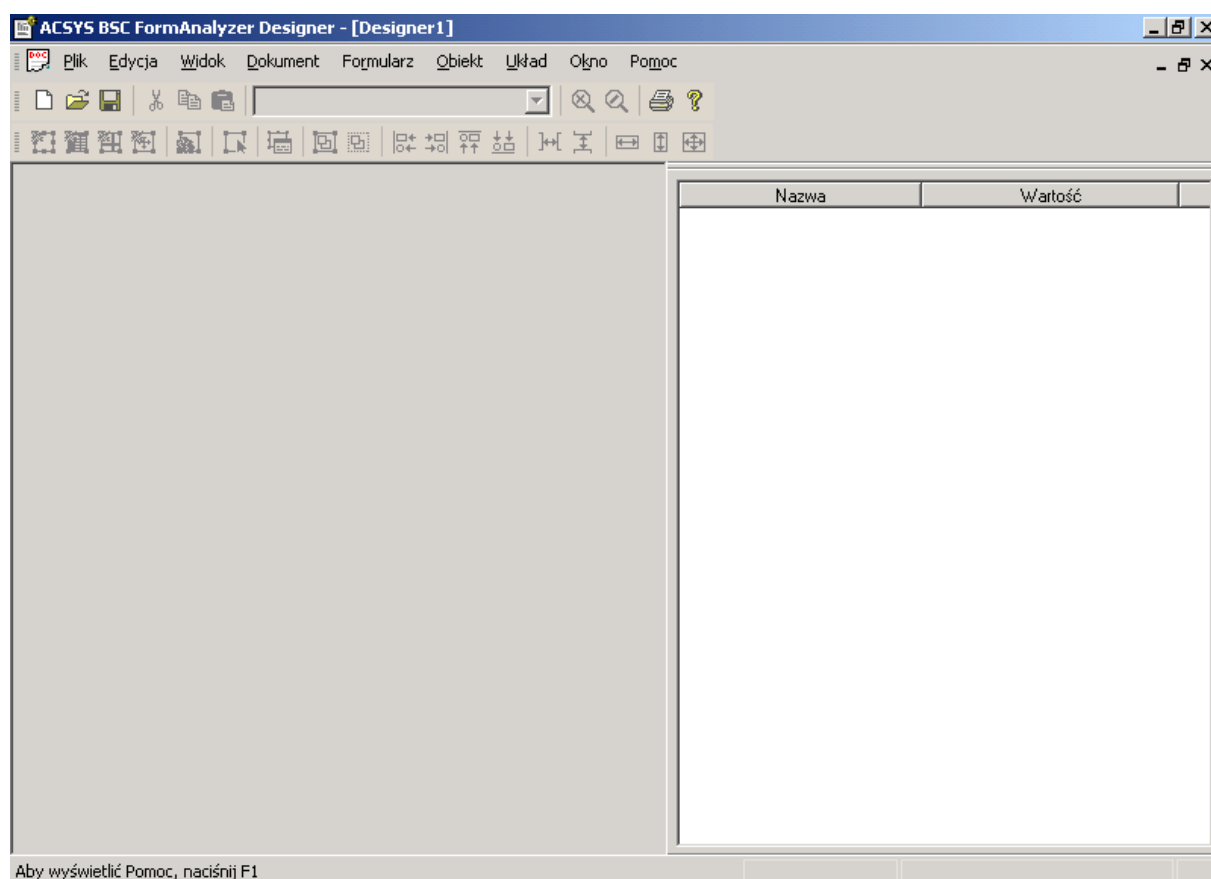
- invalid userid or password (-103)
Niepoprawna nazwa lub hasło użytkownika bazy danych - przyczyną tego błędu jest najczęściej błąd przy wpisywaniu nazwy i/lub hasła użytkownika. Może się również zdarzyć, że w bazie danych nie został zdefiniowany taki użytkownik, lub, że użytkownik ma zmienione hasło.
- unable to start specified database (-82)
Baza danych nie może być uruchomiona - przyczyną jest najczęściej niepoprawnie wpisana nazwa serwera bazy danych. Taki sam błąd pojawi się, gdy nie pracuje serwer bazy danych.
- specified database not found (-83)
Bazy danych nie odnaleziono - błąd w ścieżce dostępu do lokalnej bazy danych lub zaznaczenie opcji *Lokalna* przy łączeniu z bazą danych sieciową.

Designer w aktualnej wersji pozwala na podłączenie do jednej bazy danych z jednej instancji programu. Aby podłączyć się do drugiej bazy danych trzeba uruchomić drugą instancję modułu FormAnalyze Designer.

4.3. Okno główne aplikacji FormAnalyze Designer.

W momencie sprawdzenia klucza sprzętowego i prawidłowym podłączeniu się do bazy danych na ekranie wyświetlone zostanie główne okno aplikacji.

W pasku tytułu okna programu znajduje się jego nazwa - *FormAnalyze Designer* – oraz nazwa aktywnego pliku konfiguracji (na przykładzie jest to *Designer1*). Nazwa tego pliku jest również powtórzona w pasku tytułu okna pliku konfiguracji. Przy każdorazowym uruchomieniu aplikacji FormAnalyze Designer tworzony jest nowy, pusty plik konfiguracji o nazwie *Untitled*.



Rys. 4-9 Okno aplikacji FormAnalyzer Designer po uruchomieniu.

Poniżej znajdują się opisy poszczególnych poleceń pogrupowanych w rozwijalne listy menu głównego programu. W celu wybrania konkretnej pozycji, należy kliknąć nagłówek menu, a następnie wskazać ją na wyświetlonej liście poleceń.

<i>Plik</i>	Grupuje polecenia dotyczące plików konfiguracji, połączenia z bazą danych, drukowaniem i zamykaniem aplikacji.
<i>Edycja</i>	Obejmuje polecenia dotyczące edycji pól, powielania ich, a także usuwania elementów nałożonych na formularz..
<i>Widok</i>	Znajdują się tu polecenia dotyczące sposobu wyświetlania informacji przez FormAnalyzer Designer
<i>Dokument</i>	Grupuje polecenia dotyczące szablonów dokumentów, kryteria ich poprawności, oraz importu szablonów dokumentów.
<i>Formularz</i>	Obejmuje polecenia dotyczące formularza: tworzenia nowego formularza, wyświetlania jego właściwości i usuwania. Dodatkowo zawiera pozycję włączającą wyświetlanie kolejności przetwarzania pól formularza.
<i>Obiekt</i>	Grupuje polecenia dotyczące tworzenia, właściwości i grupowania obiektów
<i>Układ</i>	Obejmuje polecenia dotyczące rozmieszczenia obiektów na formularzu.

<i>Okno</i>	Grupuje polecenia dotyczące okna lub okien pliku konfiguracji, ich rozmieszczenie w panelu.
<i>Pomoc</i>	Grupuje polecenia związane z pomocą programu

W paskach narzędzi są zebrane najczęściej używane polecenia z menu. Przycisk na pasku narzędzi odpowiada poleceniu z menu.

Użytkownik może sterować wyświetlaniem pasków narzędzi i stanu. Opcja *Pasek stanu* znajdująca się w menu *Widok* służy do włączania i wyłączania paska stanu. Jeżeli przed pozycją znajduje się znak ✓ to pasek jest wyświetlany. Kliknięcie tej pozycji spowoduje zmianę wyświetlania paska. W pasku statusu są wyświetlane informacje pomocnicze. Z lewej strony pojawia się objaśnienie dotyczące zaznaczonego polecenia w menu lub na pasku narzędzi.

Paski narzędzi użytkownik może także dowolnie wyświetlać, w tym celu należy z menu *Widok* wybrać polecenie *Paski narzędzi*, a następnie na rozwijalnej liście zaznaczyć bądź odznaczyć odpowiednie pozycje.

Na pasku narzędzi znajduje się rozwijalna lista zawierająca nazwy szablonów opisanych w pliku konfiguracyjnym.

Oprócz rozwijalnej listy, na pasku narzędzi znajdują się przyciski umożliwiające m.in.: Otwarcie nowego pliku konfiguracji, Zapisywanie obecnie otwartego pliku konfiguracji, Kopiowanie, Wycinanie, Wklejanie elementów, Drukowanie dokumentu itd.

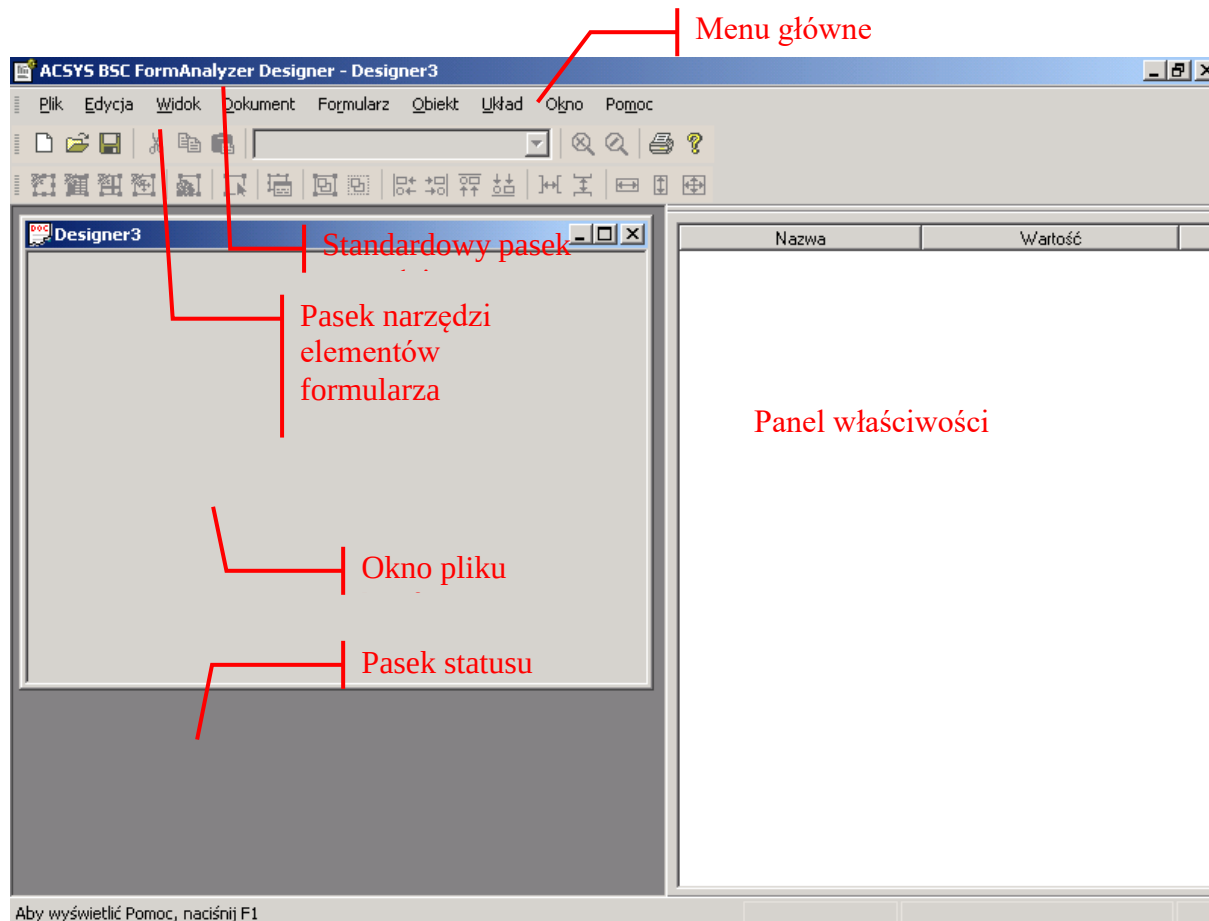
Każdemu z poszczególnych poleceń przyporządkowane są odpowiednie skróty klawiszowe.

Poniżej standardowego paska narzędzi znajduje się pasek narzędzi elementów formularza. Umieszczone są na nim następujące elementy:

Symbol przycisku na pasku narzędzi	Dostępny w menu		Nazwa i funkcjonalność w programie
	Obiekt Nowy	→	Nowe pole – tworzy nowe pole na formularzu
	Obiekt Nowy	→	Nowy znacznik pozycjonowania – tworzy nowy znacznik pozycjonowania
	Obiekt Nowy	→	Nowy obszar pozycjonowania - tworzy nowy obszar pozycjonowania
	Obiekt Nowy	→	Wyszukaj pola wyboru – wyszukuje pole wyboru. W menu Obiekt funkcja ta występuje pod nazwą Automat dla pól wyboru
	Obiekt Nowy	→	Zmień format pola - zmienia wewnętrzny format pola. W menu Obiekt→ Nowy funkcja ta występuje pod nazwą Format wewnętrzny.

	Obiekt	Podaj właściwości obiektu – Powoduje wyświetlenie okna dialogowego Właściwości obiektu, dla zaznaczonego elementu na formularzu.
	Obiekt	Grupuj – łączy wybrane obiekty w grupę.
	Obiekt	Rozgrupuj – rozgrupowuje wybrany obiekt.
	Układ Wyrównaj	→ Wyrównaj w lewo – lewe krawędzie zaznaczonych obiektów są wyrównywane do lewej krawędzi elementu odniesienia.
	Układ Wyrównaj	→ Wyrównaj w prawo – prawe krawędzie zaznaczonych obiektów są wyrównywane do prawej krawędzi elementu odniesienia.
	Układ Wyrównaj	→ Wyrównaj w górę – górne krawędzie zaznaczonych obiektów są wyrównywane do górnej krawędzi elementu odniesienia
	Układ Wyrównaj	→ Wyrównaj w dół – dolne krawędzie zaznaczonych obiektów są wyrównywane do dolnej krawędzi elementu odniesienia
	Układ Wycentruj	→ Wyrównaj w pionie – środki zaznaczonych obiektów są wyrównywane do linii pionowej środka elementu odniesienia.
	Układ Wycentruj	→ Wyrównaj w poziomie – środki zaznaczonych obiektów są wyrównywane do linii poziomej środka elementu odniesienia.
	Układ Dopasuj rozmiar	→ Nadaj taką samą szerokość – szerokość zaznaczonych elementów dopasowana jest do szerokości elementu odniesienia
	Układ Dopasuj rozmiar	→ Nadaj taką samą wysokość – wysokość zaznaczonych elementów dopasowana jest do wysokości elementu odniesienia.
	Układ Dopasuj rozmiar	→ Nadaj takie same rozmiary - rozmiar zaznaczonych elementów dopasowany jest do rozmiarów elementu odniesienia.

Okno główne programu podzielone jest na dwa panele. W lewym znajduje się okno pliku konfiguracji. W nim wykonywana jest większość operacji dotycząca edycji formularza. W prawym panelu wyświetlane są parametry aktualnie wybranego pliku konfiguracji obiektu (obiektów). Korzystając z tego panelu można także szybko zmodyfikować wartości niemal wszystkich ustawień obiektów. Stanowi to bardzo wygodny zamiennik okien właściwości obiektów. Granica pomiędzy obydwojema panelami jest nieruchoma i nie można zmieniać ich szerokości.

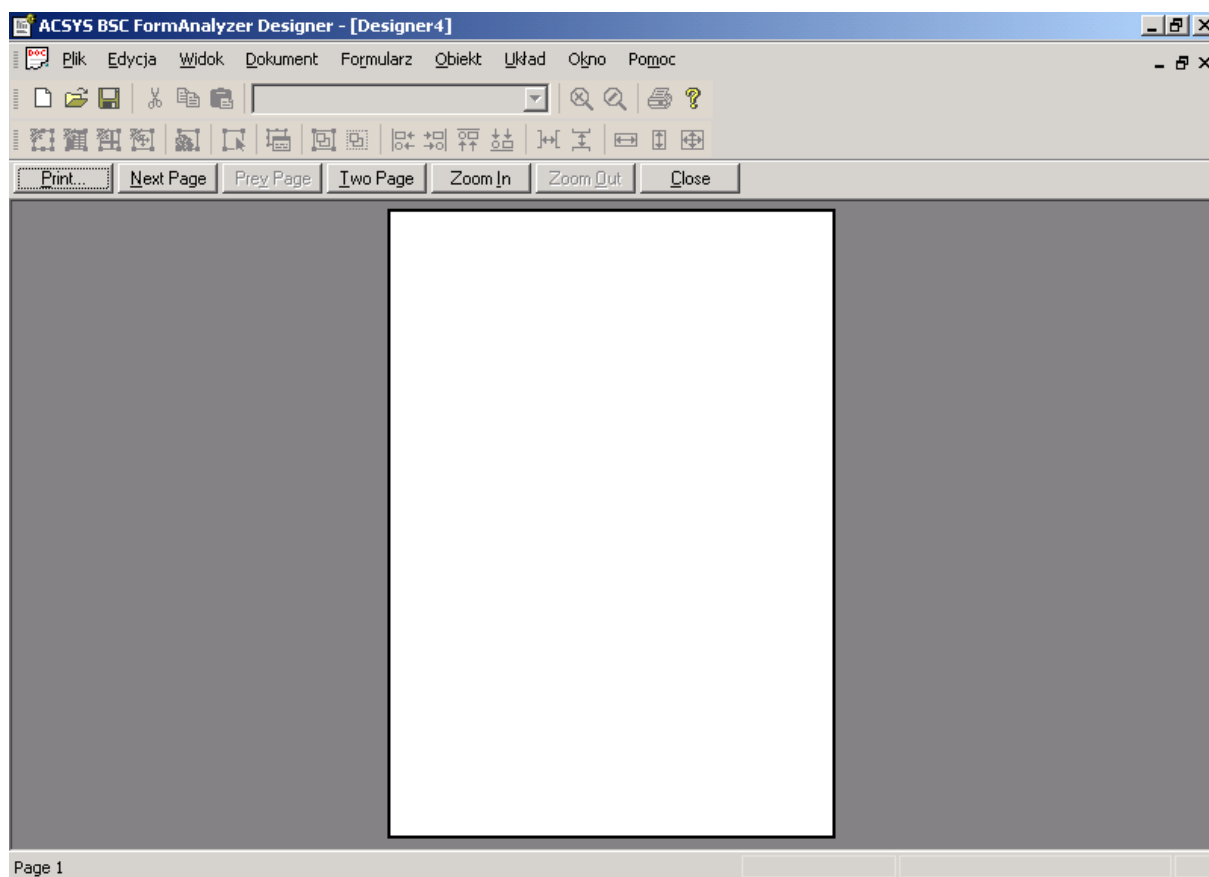


Rys. 4-10 Okno główne aplikacji FormAnalyze Designer bez otwartego pliku konfiguracyjnego.

Powyżej zaprezentowano okno główne programu FormAnalyze Designer w momencie, gdy wszystkie pliki konfiguracyjne są zamknięte.

4.4. Okno podglądu wydruku

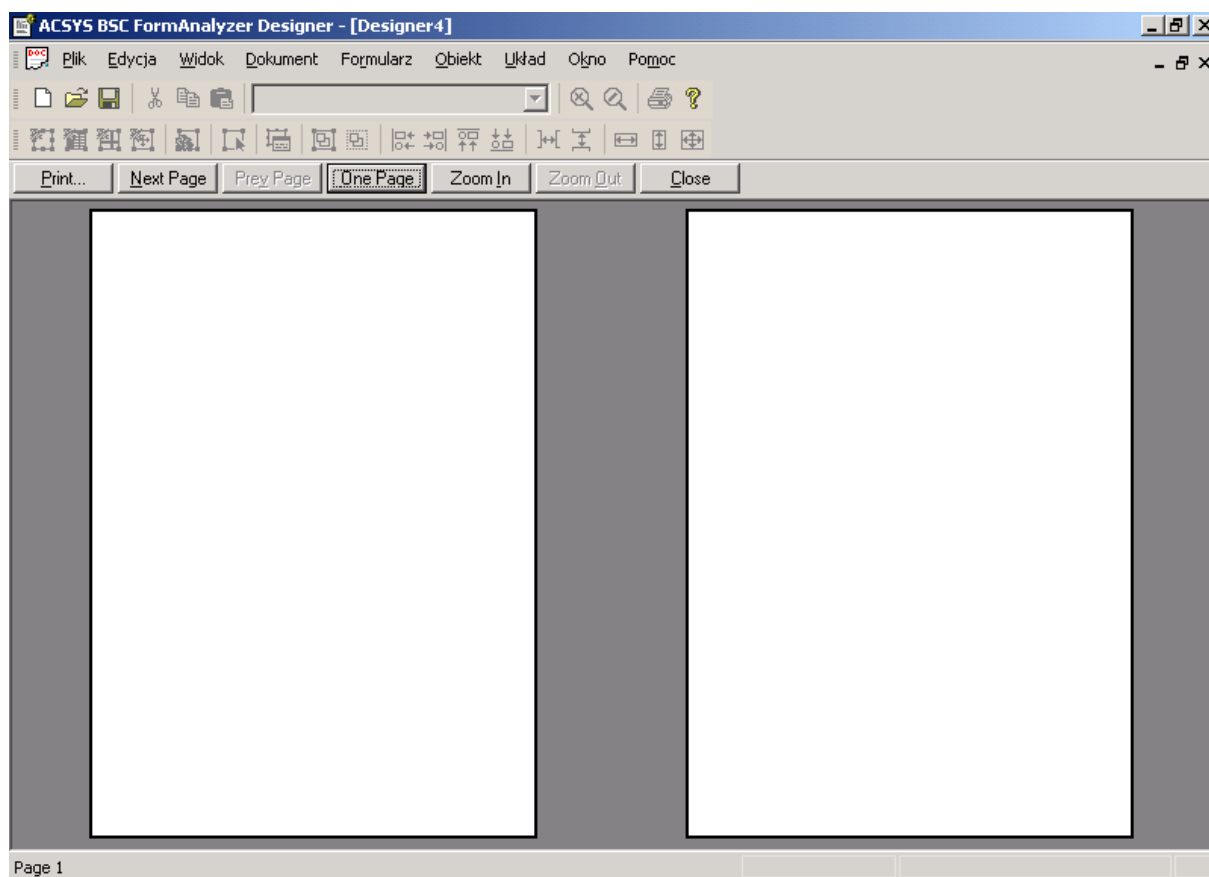
Aby wywołać okno dialogowe *Podgląd wydruku*, należy wybrać z menu *Plik* polecenie *Podgląd wydruku*. Możemy w nim zobaczyć, w jaki sposób zostaną wydrukowane strony danego dokumentu. W pasku tytułu okna podana jest nazwa pliku konfiguracyjnego.



Rys. 4-11 Okno podglądu wydruku

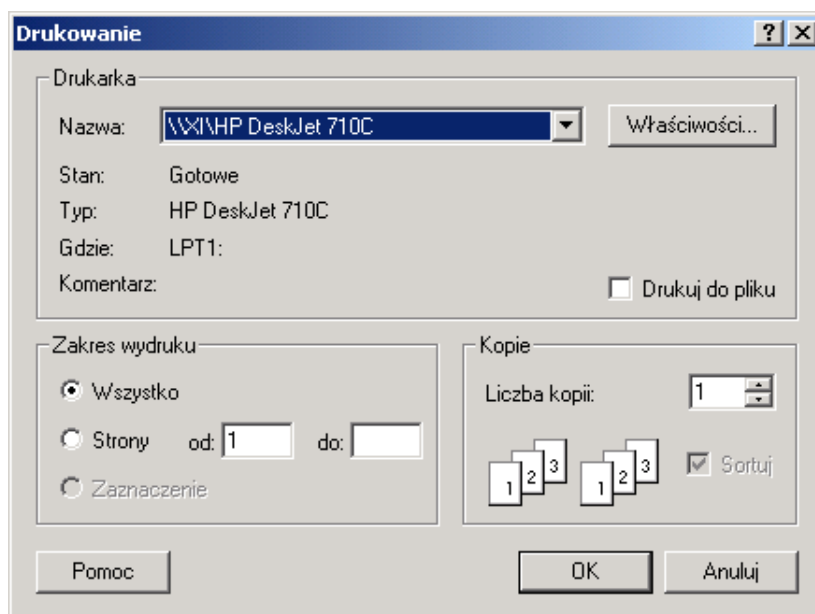
W górnej części okna umieszczone są przyciski umożliwiające: wydrukowanie bieżącego dokumentu - **Print...**; przejścia do następnej strony dokumentu - **Next Page**; przejścia do poprzedniej strony dokumentu - **Prev Page**; możliwość wyświetlania pojedynczo stron dokumentu - **One Page**; możliwość wyświetlenia dwóch stron dokumentu jednocześnie - **Iwo Page**; powiększenie rozmiaru obrazu strony - **Zoom In**; zmniejszenia rozmiaru obrazu strony - **Zoom Out**; zamknięcie okna dialogowego Podgląd wydruku - **Close**.

Funkcje **Next Page**, **Prev Page**, **One Page**, oraz **Iwo Page** stają się aktywne tylko w przypadku dokumentów wielostronicowych. Okno podglądu wydruku może wyświetlić maksymalnie dwie strony na raz. Jeżeli obraz zajmuje więcej niż dwie strony, w celu obejrzenia kolejnych stron dokumentu, należy skorzystać z przycisków **Next Page** lub **Prev Page**. Można także skorzystać z paska przewijania znajdującego się z prawej strony okna dialogowego. Poniżej znajduje się przykład podglądu wydruku dokumentu wielostronicowego.



Rys. 4-12 Okno podglądu wydruku – dwie strony.

Jeżeli podgląd wydruku dokumentu jest prawidłowy, klikając przycisk  wyświetlone zostanie standardowe okno drukowania, gdzie można zdefiniować opcje drukowania dokumentu.

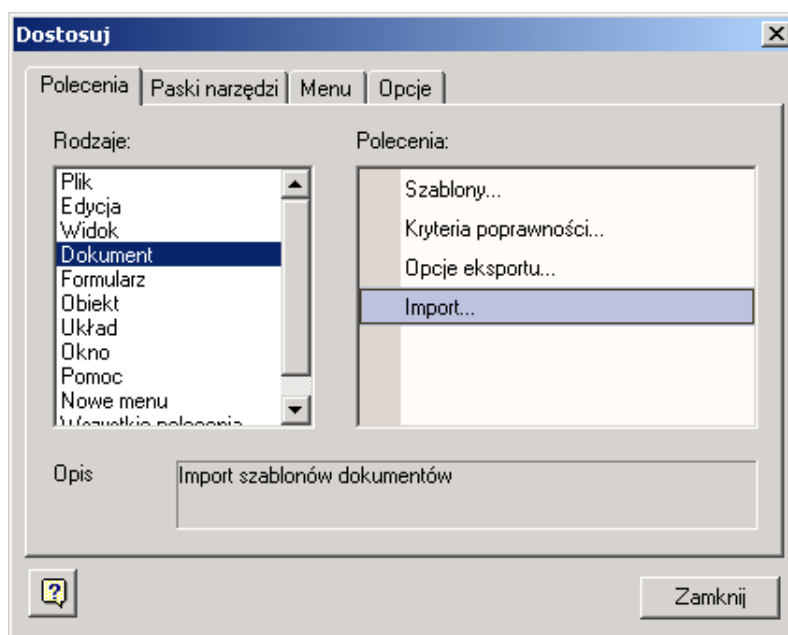


Rys. 4-13 Standardowe okno opcji wydruku.

4.5. Dostosowywanie menu i paska narzędzi.

FormAnalyzer Designer umożliwia dostosowanie do indywidualnych potrzeb użytkownika wyglądu pasków narzędzi oraz pozycji menu.

Aby dostosować pasek narzędzi do własnych potrzeb, należy wybrać z menu *Widok* → *Paski narzędzi* polecenie *Dostosuj*. Można także kliknąć prawym przyciskiem myszy w dowolnym miejscu paska narzędzi, a następnie z menu kontekstowego wybrać polecenie *Dostosuj*. Spowoduje to wyświetlenie okna dialogowego *Dostosuj* zaprezentowanego poniżej.



Rys. 4-14 Okno dostosowywania menu – pierwsza zakładka.

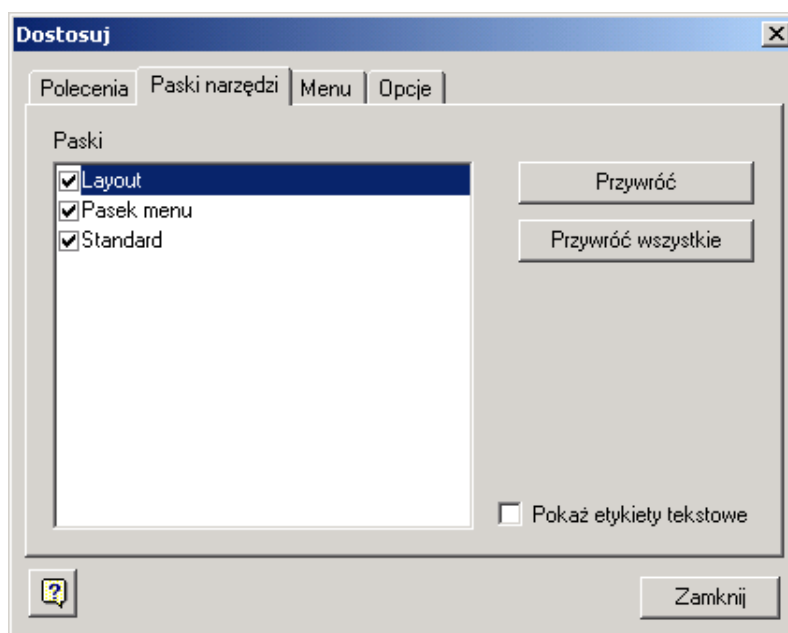
Zakładka *Polecenia* okna dialogowego *Dostosuj* składa się z dwóch obszarów. W lewym obszarze *Rodzaje* zaprezentowane są pozycje menu FormAnalyzer Designer, w prawym *Polecenia* – wyświetlana jest zawartość poszczególnych pozycji menu, zaznaczonej w lewym obszarze okna. W polu *Opis* wyświetlony zostanie dostępny opis elementu wskazanego w obszarze *Polecenia*.

Korzystając z tej zakładki można umieścić dowolny przycisk na pasku narzędzi. W tym celu, należy w prawym obszarze okna dialogowego kliknąć i przytrzymać lewy przycisk myszki na poleceniu, które chcemy dodać, a następnie przeciągnąć je na pasek narzędzi.

W celu usunięcia przycisku z paska narzędzi, należy klikając żądany przycisk na pasku narzędzi i przytrzymując wciśnięty lewy przycisk myszki przyciągnąć go, a następnie upuścić w prawym obszarze zakładki *Polecenia*.

Druga zakładka *Paski narzędzi* służy do włączania i wyłączania wyświetlania poszczególnych pasków. W tym celu, należy odznaczyć pole wyboru przy nazwie paska - ☐ -, aby go ukryć i zaznaczyć je - ☒ - aby go ponownie wyświetlić. Nie można ukryć paska menu.

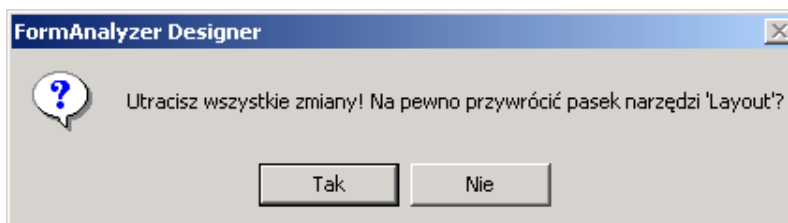
Istnieje możliwość umieszczenia etykiet tekstowych w paskach narzędzi. W tym celu, należy w zakładce *Paski narzędzi*, kliknąć lewym przyciskiem myszki na nazwie dowolnego paska, a następnie zaznaczyć pole wyboru *Pokaż etykiety tekstowe*. Funkcja ta jest niedostępna dla paska menu.



Rys. 4-15 Okno dostosowywania menu – druga zakładka.

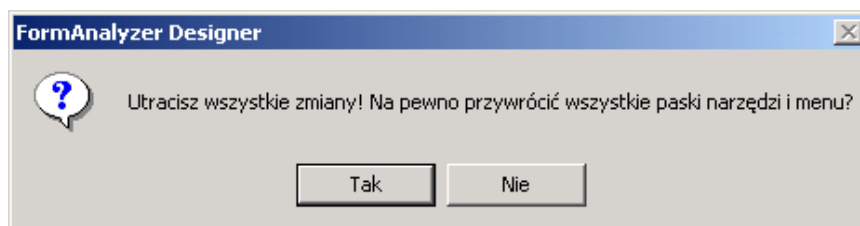
W celu przywrócenia domyślnego wyglądu wybranemu paskowi narzędzi, należy wybrać dowolny pasek, a następnie kliknąć przycisk **Przywróć**. Po wykonaniu powyższej czynności na ekranie zostanie wyświetlone okno komunikatu z zapytaniem o przywrócenie domyślnych ustawień wskazanego paska narzędzi.

Kliknięcie przycisku **Tak** powoduje przywrócenie ustawień, kliknięcie przycisku **Nie** spowoduje zachowanie zmian i zamknięcie okna dialogowego Dostosuj.



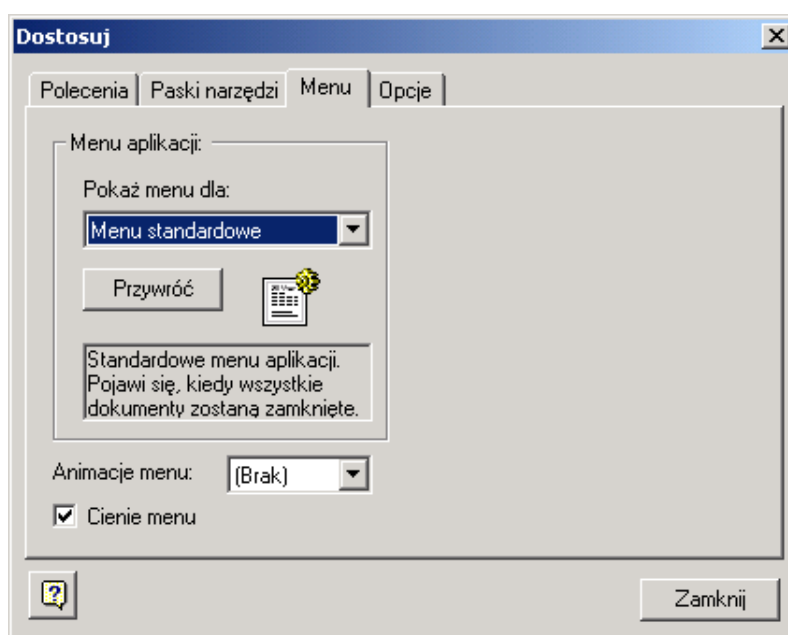
Rys. 4-16 Okno potwierdzenia przywrócenia domyślnych ustawień dla paska narzędzi *Layout*.

Istnieje możliwość przywrócenia domyślnych ustawień dla wszystkich pasków narzędzi. W tym celu, należy kliknąć przycisk **Przywróć wszystkie**. Spowoduje to wyświetlenie okna komunikatu ze stosownym zapytaniem. W tym przypadku również kliknięcie przycisku **Tak** powoduje przywrócenie domyślnych ustawień.



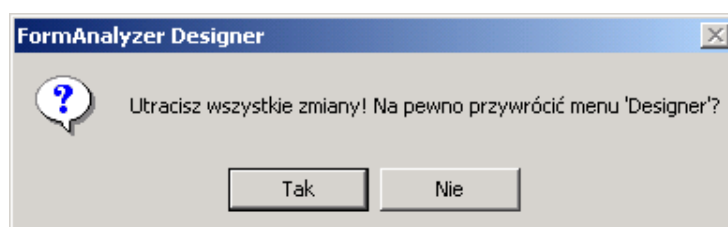
Rys. 4-17 Okno potwierdzenia przywrócenia domyślnych ustawień dla wszystkich pasków.

Zakładka Menu pozwala zdefiniować opcje związane ze sposobem animacji wyświetlania okien menu aplikacji. Z rozwijalnej listy *Pokaż menu dla*, należy wybrać jedną z dwóch aktualnie dostępnych opcji: *Menu standardowe* oraz *Designer*. Z rozwijalnej listy *Animacje menu*, należy wybrać efekt, który będzie towarzyszył wyświetlaniu pozycji aplikacji: *brak animacji*, *rozwiń*, *slajd* lub *zanikanie*. Na samym dole można włączyć lub wyłączyć generowanie cienia przez menu.



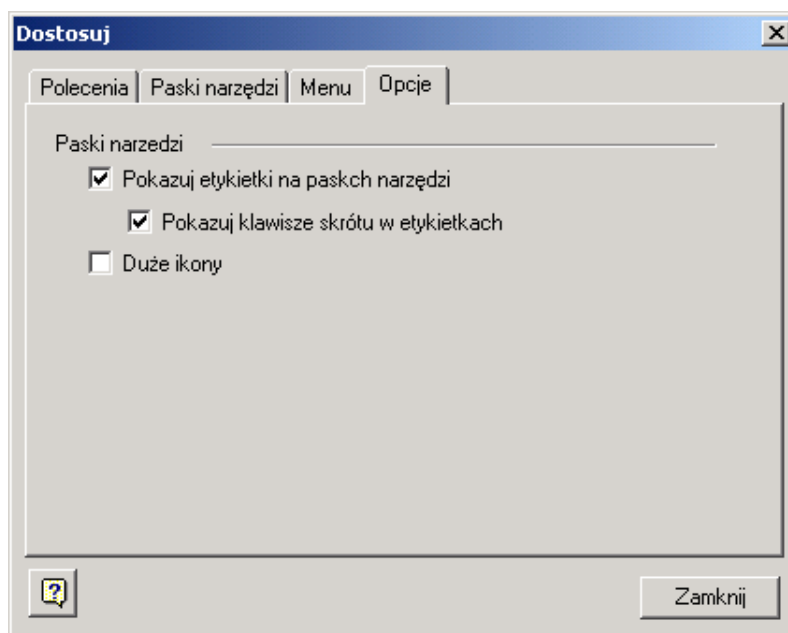
Rys. 4-18 Okno dostosowywania menu – trzecia zakładka.

W celu powrotu do domyślnych ustawień, należy uprzednio wskazując żądane menu kliknąć przycisk **Przywróć**. Na ekranie pojawi się okno komunikatu ze stosownym zapytaniem. Kliknięcie przycisku **Tak** spowoduje przywrócenie domyślnych ustawień.



Rys. 4-19 Okno potwierdzenia przywrócenia domyślnych ustawień menu.

Ostatnia zakładka *Opcje* umożliwia włączenie i wyłączenie wyświetlania etykiet oraz wybór wielkości wyświetlanych ikon przycisków na paskach narzędzi. Jeżeli zostanie zaznaczone pole wyboru Pokazuj etykiety na paskach narzędzi, aktywnym stanie się polecenie Pokazuj klawisze skrótu w etykietkach. Jego zaznaczenie spowoduje wyświetlenie klawiszy skrótu obok nazwy przycisku w etykietce.

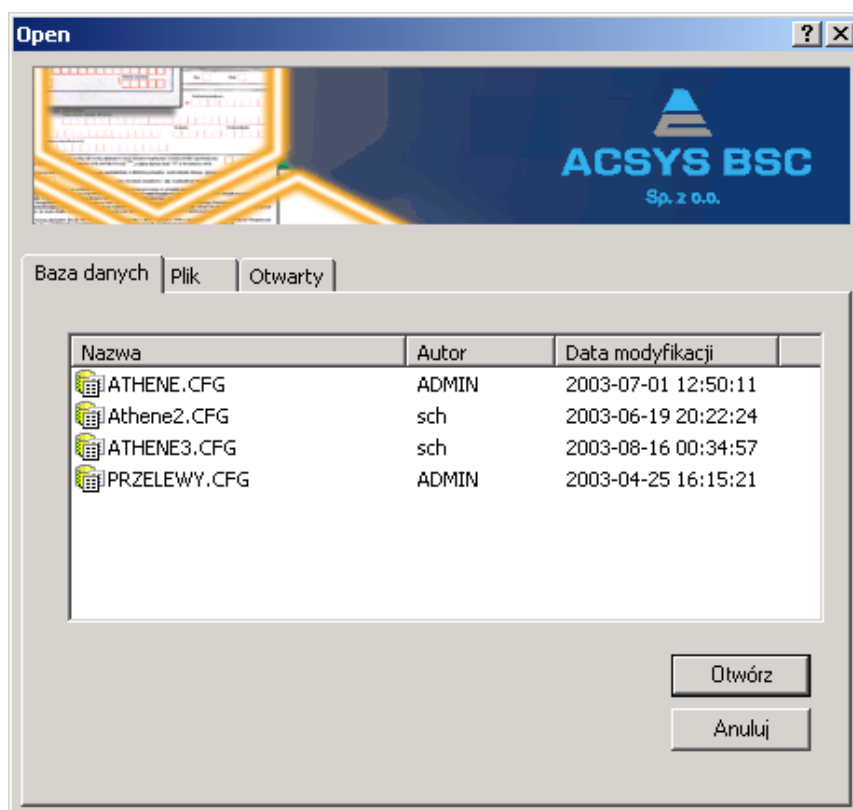


Rys. 4-20 Okno dostosowywania menu –czwarta zakładka.

4.6. Pliki konfiguracji.

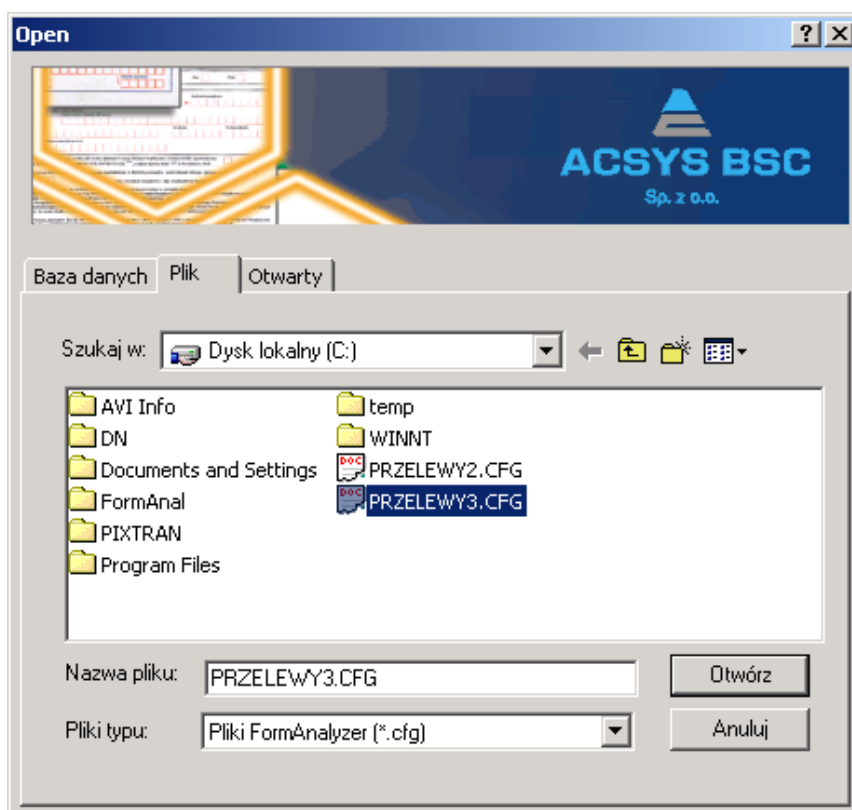
W celu utworzenia nowego, pustego pliku konfiguracji, można posłużyć się jedną z metod: kliknąć przycisk Nowy  na pasku narzędzi, wybrać polecenie Nowy z menu Plik, bądź posłużyć się kombinacją klawiszy Ctrl + N. Każdemu nowemu plikowi konfiguracji zostanie nadana nazwa Untitle.

W celu otworenia istniejącego pliku konfiguracji, można posłużyć się jedną z metod: kliknąć przycisk Otwórz  na pasku narzędzi, wybrać polecenie Otwórz z menu Plik, bądź posłużyć się kombinacją klawiszy Ctrl + O. Na ekranie wyświetlone zostanie okno służące do otwierania plików, pokazane poniżej.



Rys. 4-21 Okno otwierania pliku konfiguracji – pierwsza zakładka.

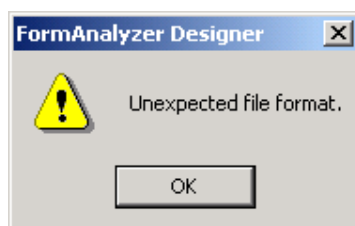
Okno dialogowe *Open* składa się z trzech zakładek: *Baza danych*, *Plik* oraz *Otwarty*. W zakładce *Baza danych*, znajduje się lista plików konfiguracyjnych zarejestrowanych w bazie danych. Wyświetlana jest również informacja o autorze pliku konfiguracyjnego oraz dacie ostatniej modyfikacji.



Rys. 4-22 Okno otwierania pliku konfiguracji – druga zakładka

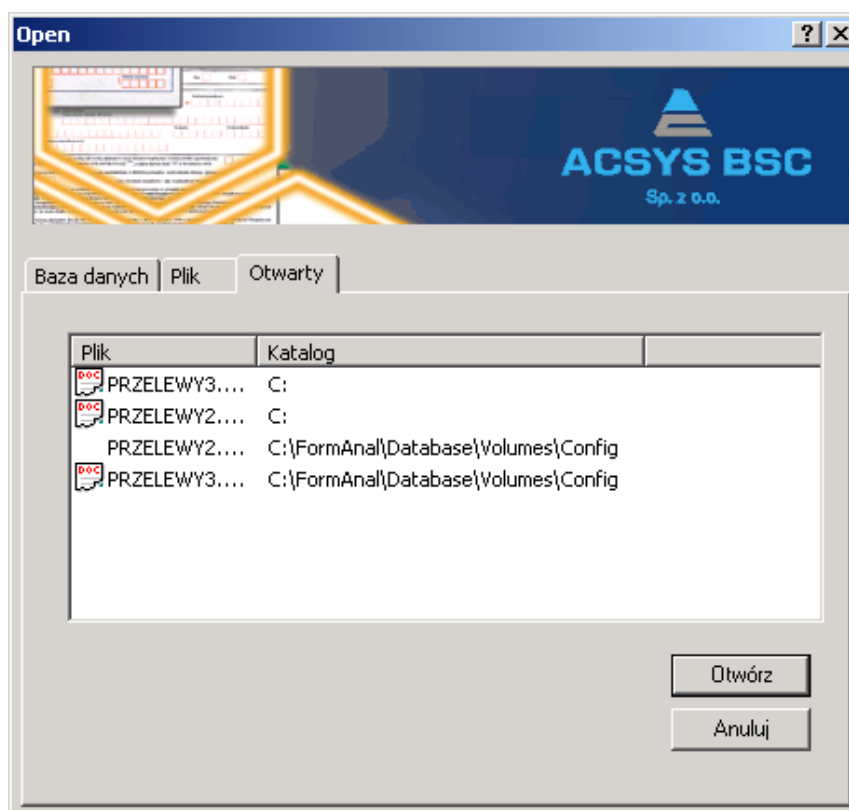
W zakładce Plik można wyszukać, a następnie otworzyć żądany plik konfiguracji zapisane na dysku. Pliki konfiguracji posiadają rozszerzenie *.cfg. Typ rozszerzenia pliku, w razie potrzeby należy wybrać z rozwijalnej listy Pliki typu. Wybranie z tej listy opcji *All files* (*.*) umożliwi dostęp do wszystkich plików zapisanych na dysku.

Jeżeli wybrany plik konfiguracji posiada nieprawidłowe rozszerzenie, przy próbie jego otwarcia na ekranie pojawi się okno komunikatu sygnalizujące niewłaściwy format pliku Rys. 4-23.



Rys. 4-23 Komunikat o złym formacie otwieranego pliku konfiguracyjnego

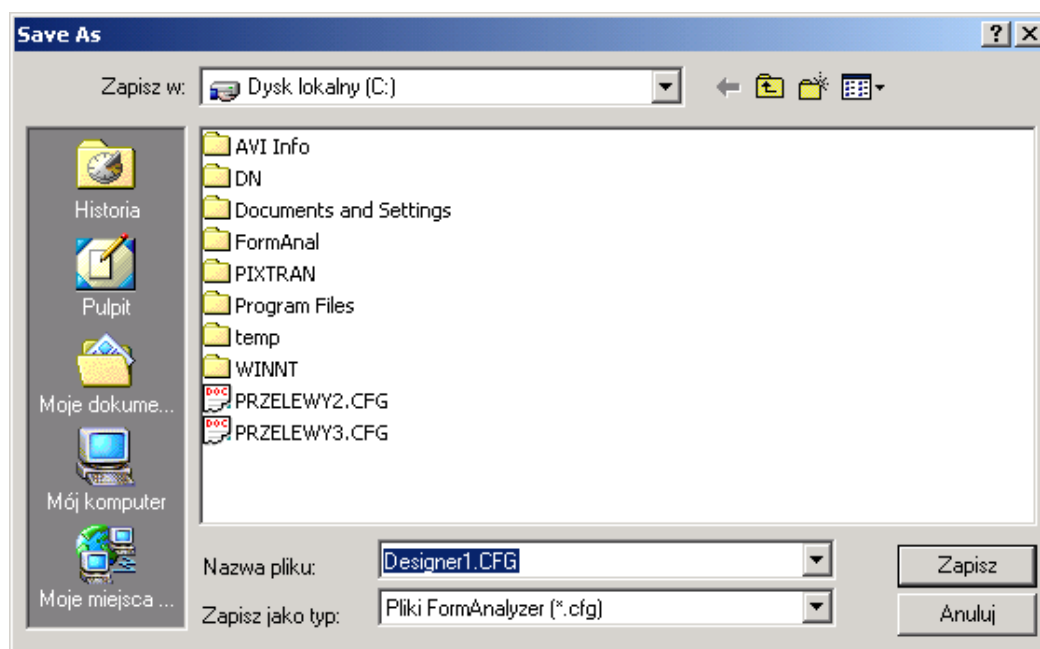
W zakładce *Otwarty*, okna dialogowego *Open* znajdują się ostatnio otwierane pliki oraz ich ścieżki dostępu.



Rys. 4-24 Okno otwierania pliku konfiguracji – trzecia zakładka

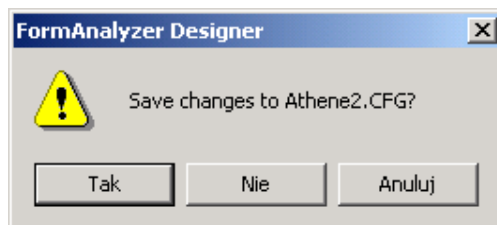
W celu otwarcia wskazanego pliku konfiguracji, w jednej z trzech zakładek omówionych powyżej, należy dwukrotnie kliknąć lewym przyciskiem myszki na nazwie żadanego pliku, lub zaznaczyć go, a następnie kliknąć przycisk **Otwórz**. Kliknięcie przycisku **Anuluj** spowoduje anulowanie rozpoczętej operacji i zamknięcie okna dialogowego *Open*.

W celu zapisania pliku konfiguracji można posłużyć się jedną z trzech dostępnych metod: kliknąć przycisk **Zapisz** na pasku narzędzi, wybrać polecenie **Zapisz** z menu **Plik** lub posłużyć się kombinacją klawiszy **Ctrl + S**. W przypadku pliku konfiguracji, który nie był jeszcze zapisywany na dysku, można posłużyć się poleceniem **Zapisz jako** znajdującym się w menu **Plik**. Wybranie tego polecenia spowoduje wyświetlenie okna dialogowego **Save As**. W okno to umożliwia określenia lokalizacji i nazwy zapisywanego pliku konfiguracji.

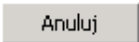


Rys. 4-25 Okno zapisu pliku konfiguracji ze zmianą nazwy (położenia).

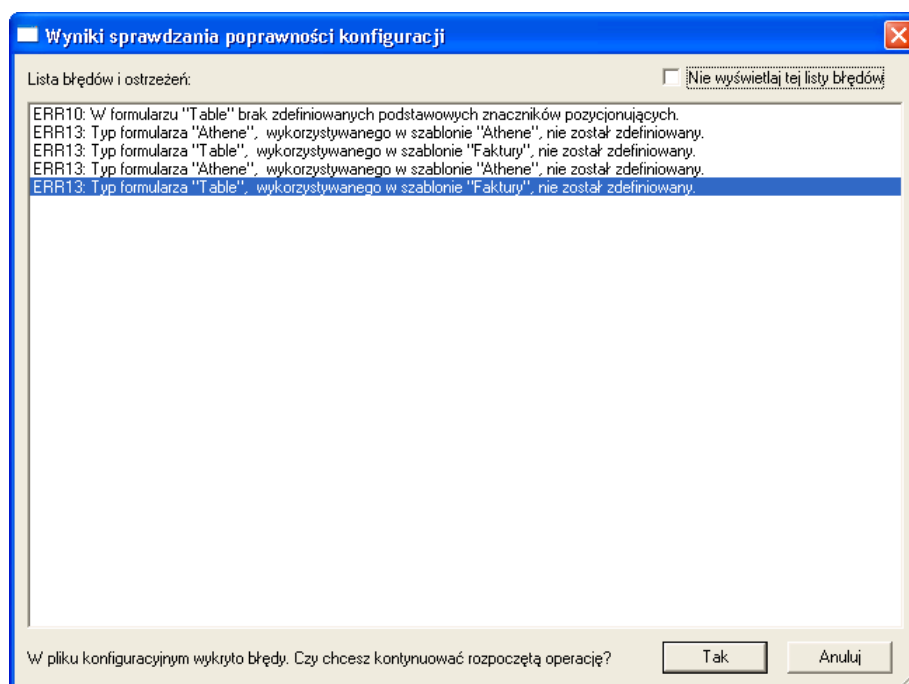
W celu zamknięcia pliku konfiguracji, należy posłużyć się jedną z czterech dostępnych metod: z menu Plik wybrać polecenie Zamknij, dwukrotnie kliknąć lewym przyciskiem ikonę pliku konfiguracji  w lewym górnym rogu okna pliku, posłużyć się kombinacją klawiszy Ctrl + F4, lub kliknąć przycisk Zamknij  znajdujący się w prawym górnym rogu okna pliku konfiguracji. Jeżeli przed zamknięciem pliku konfiguracji został on zmodyfikowany, a wprowadzone zmiany nie zostały uprzednio zapisane, na ekranie wyświetlone zostanie okno komunikatu ze stosownym zapytaniem.



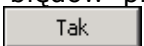
Rys. 4-26 Pytanie o zapisanie na dysk zmian pliku konfiguracji.

Kliknięcie przycisku  spowoduje zapisanie zmian na dysku, kliknięcie przycisku  - spowoduje utracenie wprowadzonych modyfikacji, natomiast kliknięcie  unieważnia polecenie zamknięcia pliku konfiguracji.

W czasie zapisu pliku konfiguracji program sprawdza poprawność jego budowy. Jeżeli zostaną wykryte nieprawidłowości wyświetlane jest okno przedstawione na Rys. 4-27, sygnalizujące błędy struktury zapisywanego pliku. Wybór znacznika *Nie wyświetlaj tej listy błędów* powoduje, że okno to zostanie ponownie wyświetlone tylko w sytuacji, gdy zostanie wygenerowany nowy błąd – aktualnie raportowane błędy będą ignorowane. Rozmiar okna można zmienić, poprzez ciągnięcie prawego-dolnego rogu okna. Rozmiar jest zapisany w rejestrze systemu Windows. Przy kolejnym wyświetleniu tego okna rozmiar jest odtwarzany.



Rys. 4-27 Okno błędów struktury pliku konfiguracji.

Pomimo stwierdzonych błędów plik konfiguracji może zostać zapisany. W tym celu, należy kliknąć przycisk . Należy jednak wziąć pod uwagę fakt, iż tego typu zapis może wiązać się z modyfikacjami pliku konfiguracji, np. mogą zostać usunięte powiązania pól z nieistniejącymi leksykonami czy maskami. Zmiana pliku konfiguracji sygnalizowana jest w komunikacie o błędach.

W większości przypadków występowania błędów struktury pliku konfiguracji można bez problemu zrezygnować z zapisu i poprawić definicję pliku konfiguracji.

FormAnalyzer Designer wykrywa następujące błędy struktury pliku konfiguracji:

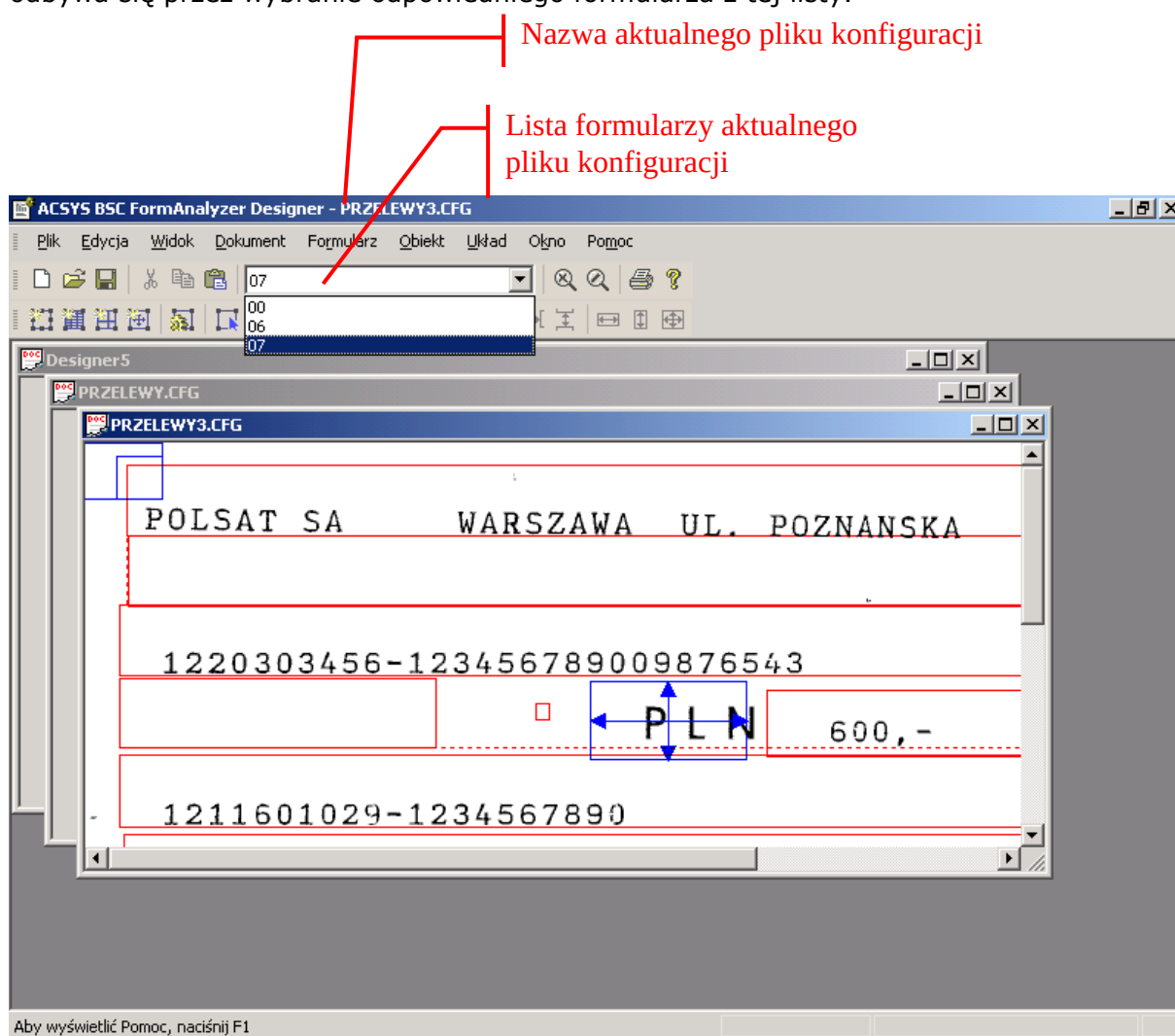
- ERR1 W kryterium poprawności "%s" w szablonie "%s" dla zmiennej podstawowej "%s" brak zdefiniowanego pola. Kryterium jest nieprawidłowe i zostanie usunięte."
- ERR2 W kryterium poprawności "%s" w szablonie "%s" dla zmiennej "%s" brak zdefiniowanego pola. Zmienna zostanie usunięta.
- ERR3 W kryterium poprawności "%s" w szablonie "%s" zmienna "%s" ma zły typ.
- ERR4 W formularzu "%s" w polu "%s" wybrany został skrypt, który nie istnieje (powiązanie zostanie usunięte).
- ERR5 W formularzu "%s" w polu "%s" w skrypcie "%s" zmienna "X" ma nieprawidłowy typ.
- ERR6 W formularzu "%s" w polu "%s" wybrana została maska "%s", która nie istnieje (powiązanie zostanie usunięte).
- ERR7 W formularzu "%s" w polu "%s" wybrany został leksykon, który nie istnieje (powiązanie zostanie usunięte).
- ERR8 W formularzu "%s" w polu "%s" data ma nieprawidłowy format.

- ERR9 W formularzu "%s" w polu "%s" data w eksporcie ma nieprawidłowy format.
- ERR10 W szablonie "%s" występuje pole weryfikacji, które powiązane jest z nieistniejącym polem (powiązanie zostanie usunięte).
- ERR11 W formularzu "%s" brak zdefiniowanych podstawowych znaczników pozycjonujących.
- ERR12 W formularzu "%s" brak znaczników pozycjonujących służących usuwaniu przekosu.
- ERR13 Szablon "%s" nie zawiera formularzy.
- ERR14 Typ formularza "%s", wykorzystywanego w szablonie "%s", nie został zdefiniowany.
- ERR15 Szablony "%s" i "%s" zawierają jednakową liczbę formularzy i dla obu szablonów pierwszym formularzem jest "%s".
- ERR16 Plik konfiguracyjny nie zawiera zdefiniowanych szablonów.
- ERR17 Pola "%s" i "%s" w formularzu "%s" zawierają ten sam identyfikator.
- ERR18 W formularzu "%s" w polu "%s" wystąpił konflikt leksykonu z trigramem.
- ERR19 W formularzu "%s" w polu "%s" maska jest pusta.
- ERR20 W strukturze przetwarzania po procesie 'Verifier2' powinien następować proces 'Arbiter'.
- ERR21 W strukturze przetwarzania występował szablon, który już nie istnieje (powiązanie zostanie usunięte).
- ERR22 W strukturze przetwarzania występowało pole, które już nie istnieje (powiązanie zostanie usunięte).
- ERR23 W formularzu "%s" w polu "%s" wybrany został leksykon przetwarzania końcowego, który nie istnieje (powiązanie zostanie usunięte).
- ERR24 W formularzu "%s" w polu "%s" minimalna liczba znaków jest większa od maksymalnej liczby znaków (wartości zostaną znormalizowane).
- ERR25 W kryterium poprawności "%s" w szablonie "%s" ustawiono typ uwierzytelnienia 'BASIC', a nie został zdefiniowany dostęp do zasobu '%s'.
- ERR26 W formularzu "%s" w polu "%s" ustawiono typ uwierzytelnienia 'BASIC', a nie został zdefiniowany dostęp do zasobu '%s'.
- ERR27 Dla szablonu "%s" ustawiono błędne opcje eksportu. Pliki z obrazami stron na wolumenie są tylko do odczytu.
- ERR28 Dla szablonu "%s" ustawiono błędne opcje eksportu. Wybrany format nie dopuszcza tworzenia plików wielostronicowych.
- ERR29 Dla szablonu "%s" ustawiono błędne opcje eksportu. Wybrany format nie dopuszcza tworzenia plików kolorowych.

ERR30 Dla szablonu "%s" ustawiono błędne opcje eksportu. Nie wybrano trybu eksportowanych stron.

FormAnalyzer Designer pozwala na równoczesną pracę z wieloma plikami konfiguracji.

Każdy z otwartych plików konfiguracji wyświetlany jest w osobnym oknie. Bez względu na ilość zdefiniowanych w pliku formularzy, w każdym oknie pliku konfiguracji wyświetlony jest zawsze jeden formularz. Pozostałe zdefiniowane formularze dostępne są w rozwijalnej liście znajdującej się w standardowym pasku narzędzi. Na Rys. 4-28 przedstawiono trzy otwarte pliki konfiguracji: *Designer5*, *PRZELEWY.CFG* i *PRZELEWY3.CFG*. Aktywnym plikiem konfiguracji jest *PRZELEWY3.CFG*. W standardowym pasku narzędzi rozwijalna lista formularzy zawiera wszystkie formularze zdefiniowane w tym pliku konfiguracji. Przelączenie wyświetlanego w oknie pliku konfiguracji formularza odbywa się przez wybranie odpowiedniego formularza z tej listy.



Rys. 4-28 Praca z wieloma plikami konfiguracji w FormAnalyzer Designer

4.7. Operacje na elementach formularza

Program FormAnalyzer Designer umożliwia wykonywanie szeregu operacji na elementach formularza.

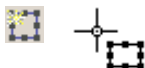
Zasady dotyczące tworzenia, modyfikacji (geometrycznej) i operowania na elementach formularza są niezależne od ich typu. W niniejszym punkcie przedstawiony został interfejs użytkownika związany z obsługą elementów formularza.

4.7.1. Umieszczenie nowego elementu.

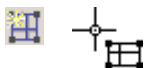
W celu wstawienia nowego elementu na formularz, należy wybrać żądany rodzaj elementu posługując się jedną z możliwych metod: kliknąć na pasku narzędzi stosowny przycisk odpowiadający konkretnemu elementowi, lub z menu Obiekt → Nowy wybrać odpowiedni rodzaj elementu.

Poniżej zostały zaprezentowane wszystkie dostępne rodzaje elementów, ich odpowiedniki na pasku narzędzi, oraz kształt, jaki przyjmie kursor po wybraniu odpowiedniego rodzaju elementu

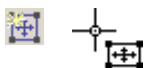
Pole rozpoznawane



Znacznik pozycjonowania



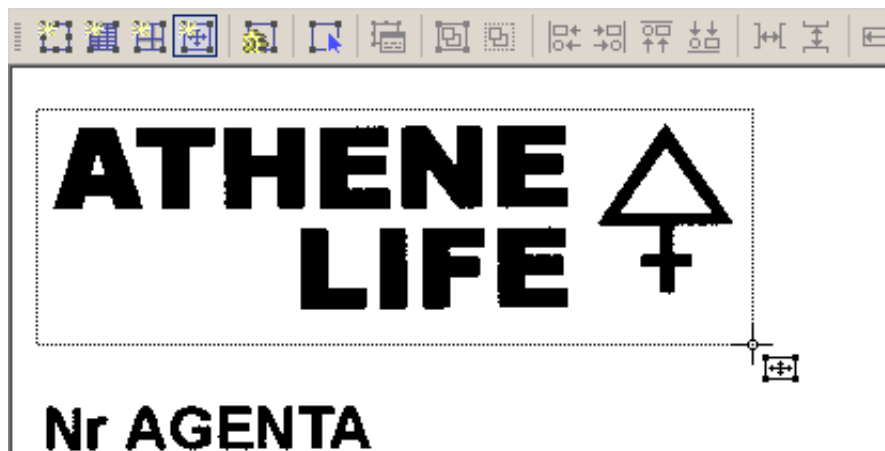
Obszar pozycjonowania

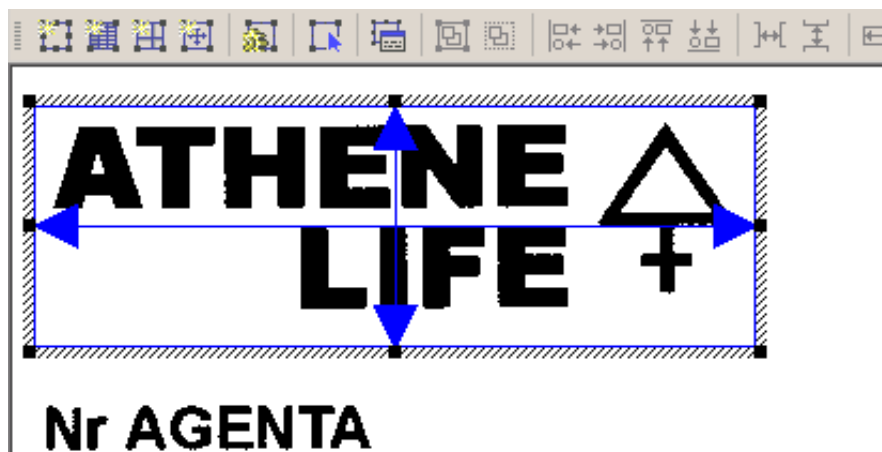


Każdy z wyżej wymienionych elementów ma kształt prostokątny. Aby umieścić nowy element na formularzu, należy nacisnąć i przytrzymać lewy przycisk myszki w miejscu, w którym będzie znajdował się jeden z wierzchołków prostokąta, a następnie przeciągnąć kursor do położenia, gdzie będzie znajdował się przeciwległy jego wierzchołek. Zwolnić przycisk myszki. Wstawiony element będzie automatycznie zaznaczony, wszelkie kolejne operacje będą dotyczyły ostatniego, zaznaczonego elementu.

W przypadku znacznika lub obszaru pozycjonowania puszczenie klawisza myszy kończy procedurę umieszczania elementu na formularzu. Dla pola rozpoznawanego trzeba jeszcze ustawić jego właściwości w wyświetlanym oknie dialogowym.

Na poniższych rysunkach przedstawiono widok formularza przy tworzeniu nowego obszaru pozycjonowania: górna część rysunku pokazuje stan tuż przed puszczeniem lewego przycisku myszki, dolna część – stan bezpośrednio po puszczeniu przycisku.





Rys. 4-29 Przykład tworzenia nowego elementu (obszaru pozycjonowania).

4.7.2. Powielanie elementów

W programie FormAnalyze Designer dostępny jest standardowy zestaw operacji edycji elementów z wykorzystaniem schowka. Służą do tego następujące polecenia znajdujące się w menu *Edycja: Kopiuj, Wytnij, Wklej*.

Wklejenie obiektów ze schowka, umieszczonych tam za pomocą poleceń: *Kopiuj*, lub *Wytnij*, może wymagać zmiany nazwy nowego obiektu (obiektów). Na formularzu muszą się znajdować obiekty o unikalnych nazwach. Jeżeli na formularzu znajduje się już obiekt, o takiej samej nazwie jak obiekt wklejany, to nazwa obiektu wklejanego jest rozszerzana o kolejny numer, dobierany tak, by uniknąć konfliktu nazw. Na przykład, jeżeli na formularzu istnieje obiekt *Pole (1)* to w wyniku wklejania zostanie utworzony obiekt o nazwie *Pole (2)* itd.)

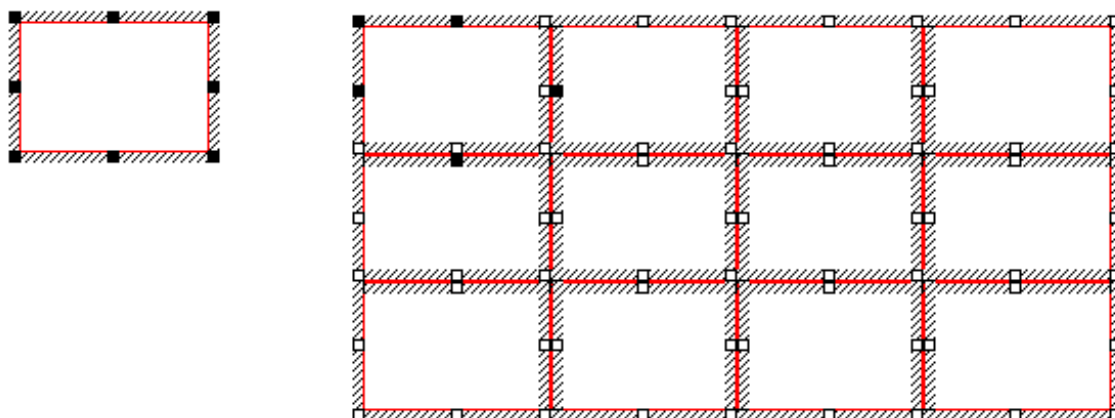
Pojedynczy element można wielokrotnie powielić wybierając polecenie *Powiel* z menu *Edycja*. Na ekranie zostanie wyświetlone okno dialogowe *Powiel pokazane*. To polecenie będzie aktywne tylko w przypadku, gdy na formularzu zaznaczony jest dokładnie jeden element.



Rys. 4-30 Okno powielania elementu formularza.

W polach *Wiersze* i *Kolumny*, należy podać żadaną liczbę rzędów i kolumn tabeli, której komórki będą utworzone z kopii powielanego elementu. Kopie są niezależnymi elementami takiego samego typu jak element źródłowy. Element źródłowy jest umieszczony w górnym lewym rogu i jest on zaznaczony.

Na poniższym rysunku przedstawiono zaznaczony do powielania element oraz wynik wykonania polecenia *Powiel* z uprzednio zdefiniowanymi parametrami w oknie dialogowym pokazanym na rysunku 4-30.



Rys. 4-31 Wielokrotne powielanie pojedynczego elementu formularza.

4.7.3. Zaznaczanie elementów.

W programie FormAnalyze Designer umożliwia zaznaczanie elementów na wiele sposobów.

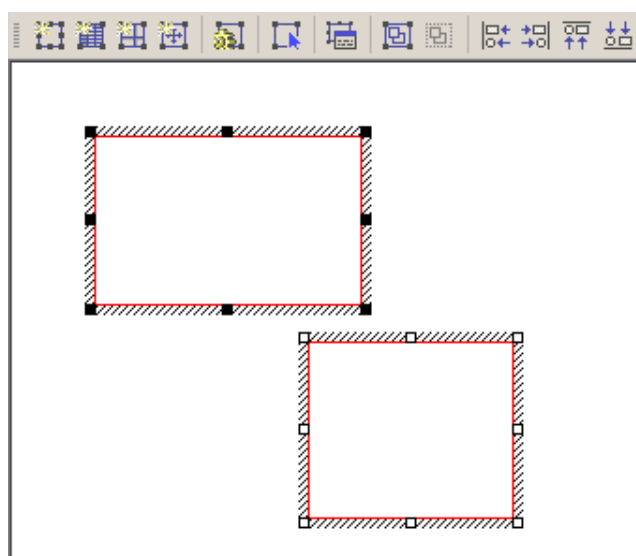
Najprostszą metodą zaznaczenia pojedynczego elementu jest kliknięcie lewym przyciskiem myszki w jego wnętrzu. Jeżeli wcześniej były zaznaczone jakieś elementy, to z chwilą zaznaczenia nowego elementu, on jako jedyny pozostaje zaznaczony na formularzu.

Istnieje możliwość zaznaczenia kilku elementów. W tym celu, należy przytrzymując wciśnięty klawisz Shift, kliknąć każdy z żądanych elementów. Jeżeli wykonując powyższą operację klikniemy na element już uprzednio zaznaczony, to jego zaznaczenie zostanie usunięte.

Aby zaznaczyć jednocześnie kilka elementów leżących na danym obszarze, możemy również otoczyć je prostokątem wyboru. Prostokąt taki wyznacza się po przez przeciągnięcie zaznaczenia, przytrzymując wciśnięty lewy przycisk myszki, tak, aby wszystkie elementy, które mają zostać zaznaczone leżały wewnątrz prostokąta. Zwolnienie przycisku myszki spowoduje zaznaczenie elementów.

W celu zaznaczenia wszystkich elementów znajdujących się na formularzu, należy wybrać polecenie Zaznacz wszystko z menu Edycja.

Gdy zaznaczonych jest wiele elementów, jeden z nich będziemy nazywać elementem odniesienia. Różni się on od pozostałych zaznaczonych elementów kolorem uchwytów na zaznaczeniu. Są one jednolicie czarne, gdy uchwyty pozostałych zaznaczonych elementów mają białe wnętrza. Na Rys. 4-32 Przedstawiono dwa zaznaczone elementy - widać, że element większy, z jednolicie zaznaczonymi uchwytami, jest elementem odniesienia.



Rys. 4-32 Przykład zaznaczenia wielu elementów.

Elementem odniesienia na formularzu będzie element zaznaczony jako ostatni.

Kiedy zaznaczamy elementy na formularzu za pomocą metody z przytrzymaniem wciśniętego klawisza *Shift*, elementem odniesienia staje się ten, który zaznaczyliśmy jako ostatni. W przypadku zaznaczania elementów za pomocą prostokąta wyboru lub polecenia Zaznacz Wszystkie element odniesienia zostanie wskazany automatycznie.

Aby wśród zaznaczonych elementów zmienić element odniesienia, należy przytrzymując wciśnięty klawisz *Shift* odznaczyć, a następnie zaznaczyć żądany element. Zostanie on wtedy zaznaczony, jako ostatni i stanie się jednocześnie elementem odniesienia.

Zaznaczenie można anulować klikając poza obszarem elementów formularza.

4.7.4. Przesuwanie elementu.

Funkcja przesuwania elementów staje się dostępna, gdy kursor myszki  umieścimy we wnętrzu żadanego elementu. Kształt kursora zmieni się na . Klikając i przytrzymując wciśnięty lewy przycisk myszki przesuwamy element do docelowej pozycji, a następnie zwalniamy przycisk myszki.

Operacja przesunięcia może być wykonana także na wielu elementach jednocześnie. Przyciskamy lewy przycisk myszki we wnętrzu któregośkolwiek zaznaczonego elementu i całość przesuwamy do pozycji docelowej.

Przesuwanie (jednego lub wielu elementów) można też wykonać za pomocą klawiatury. Klawisze kursora $\rightarrow$ $\leftarrow$ $\uparrow$ $\downarrow$ przesuwają zaznaczone elementy o jeden piksel we współrzędnych strony. Przy skalach wyświetlania obrazu innych niż 1:1, takie przesunięcie może nie być widoczne na ekranie - skala wyświetlania nie pozwala na odwzorowanie każdego jednopikselowego przesunięcia. Użycie klawiszy kursora z wciśniętym klawiszem *Shift* zwiększa skok przesuwanego elementu.

4.7.5. Zmiana rozmiaru elementu.

Każdy zaznaczony element posiada osiem uchwytów. Służą one do zmiany jego rozmiaru. W zależności od zaznaczonego uchwytu kursor przyjmie odpowiedni kształt i umożliwi to edycję elementu zgodnie z poniższym opisem:

	Uchwyty na środku pionowych krawędzi prostokąta umożliwiają zmianę tylko szerokość elementu.
	Uchwyty na środku poziomych krawędzi prostokąta umożliwiają zmianę tylko wysokość elementu.
	Uchwyty w wierzchołkach prostokąta umożliwiają zmianę obydwu wymiarów elementu.

Zmiana rozmiaru prostokąta polega na chwyceniu odpowiedniego uchwytu, a następnie przytrzymując wciśnięty lewy przycisk przeciągnięciu do momentu uzyskaniażądanego rozmiaru. Zmiana rozmiaru może być wykonana tylko na pojedynczym elemencie.

Można także zmienić rozmiar elementu posługując się klawiaturą. W tym celu, należy posłużyć następującymi kombinacjami klawiszy: *Shift* + *←* i *Shift* + *→* odpowiednio zmniejsza i zwiększa rozmiar poziomy (szerokość) zaznaczonych obiektów. *Shift* + *↑* i *Shift* + *↓* służą do zmniejszania i zwiększania wysokości zaznaczonych obiektów. Wykorzystując powyższe kombinacje klawiszy rozmiar elementów zmieni się o jeden piksel (na stronie).

Aby zwiększyć rozmiar elementów o większą liczbę pikseli, należy skorzystać z kombinacji klawiszy *Ctrl* + *Shift* + *klawisz kursora*.

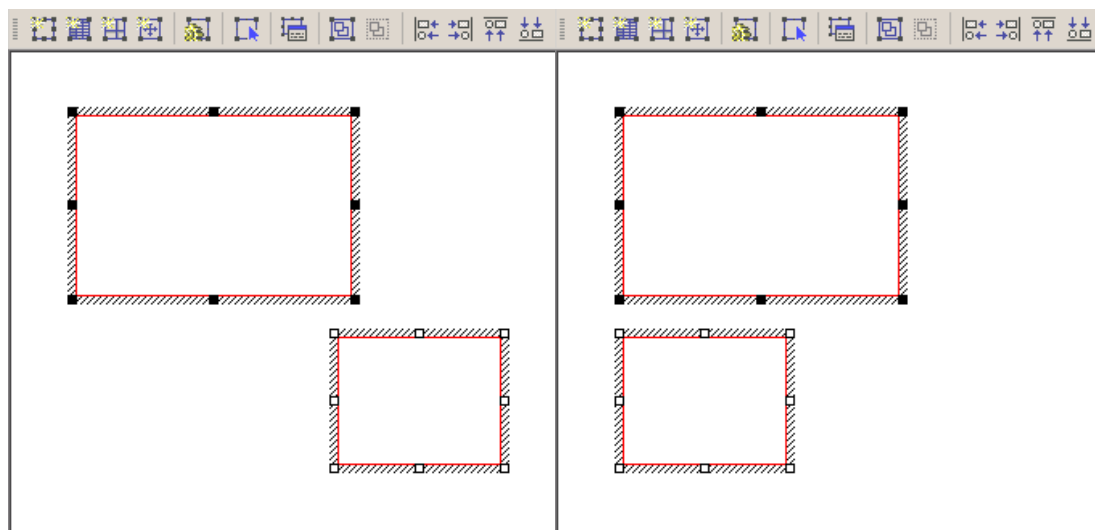
4.8. Operacje na wielu elementach

FormAnalyze Designer umożliwia zmianę położenia obiektów względem elementu odniesienia. Poszczególne polecenia dotyczące zmian położenia elementów zostały opisane poniżej.

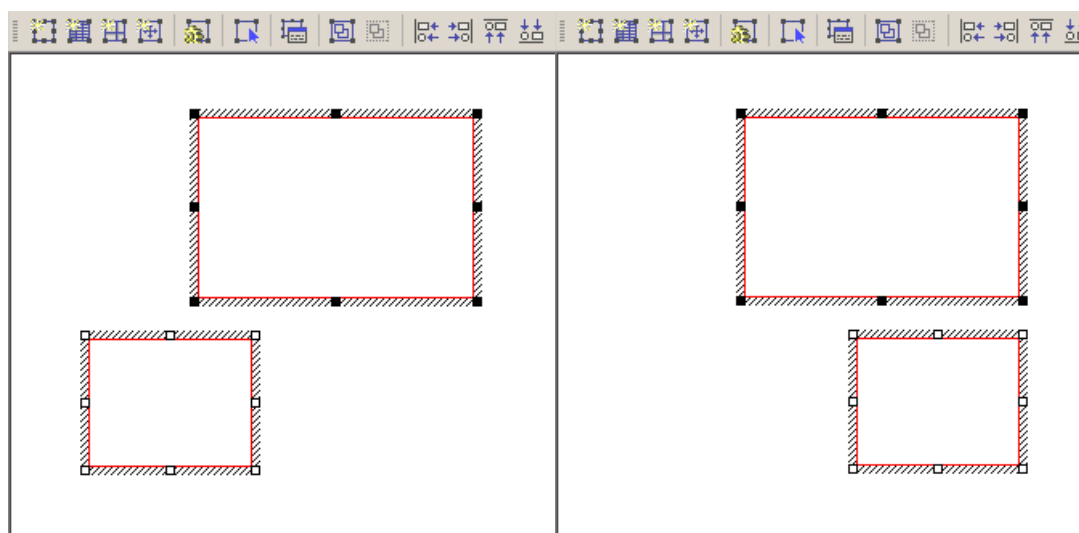
W menu Układ → Wyrównaj dostępne są cztery polecenia umożliwiające ustawianie względem siebie w określony sposób krawędzi elementów:

Polecenie Do lewej  umożliwia wyrównanie elementów do lewej krawędzi obiektu odniesienia; Do Prawej  – do prawej krawędzi obiektu odniesienia; Do góry  – wyrównuje elementy w stosunku do górnej krawędzi elementu odniesienia; Do dołu  – do dowolnej krawędzi obiektu odniesienia.

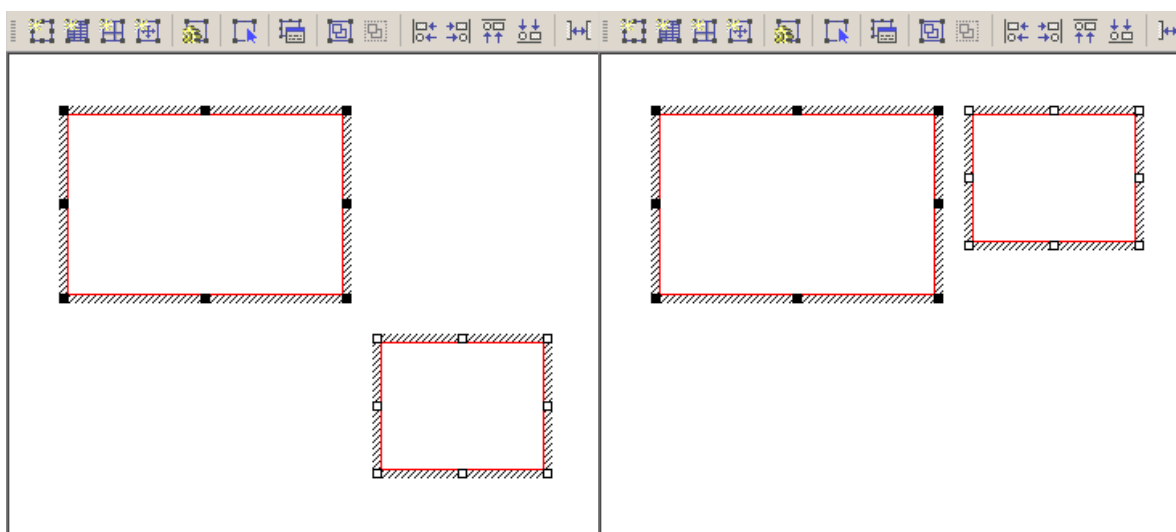
Na rysunkach: 4-33, 4-34, 4-35, 4-36 pokazano przykłady wykonania powyższych operacji dla dwóch zaznaczonych elementów. W każdym przypadku obiekt większy jest obiektem odniesienia.



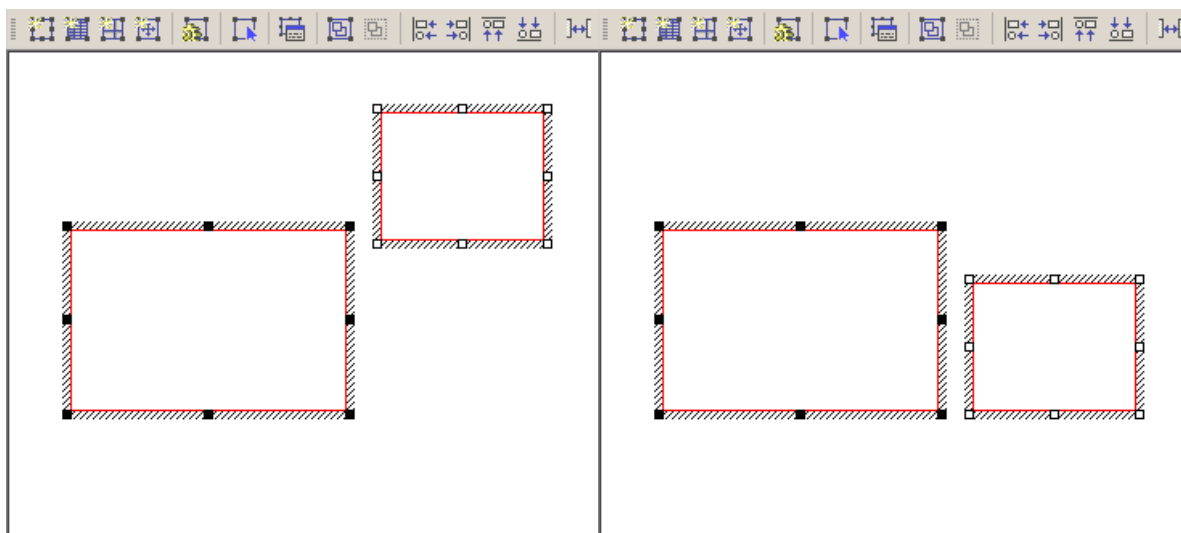
Rys. 4-33 Efekt zastosowania polecenia *Wyrównaj*→*Do lewej* lub przycisku na pasku narzędzi:



Rys. 4-34 Efekt zastosowania polecenia *Wyrównaj*→*Do prawej* lub przycisku na pasku narzędzi:



Rys. 4-35 Efekt zastosowania polecenia *Wyrównaj*→*Do góry* lub przycisku na pasku narzędzi:

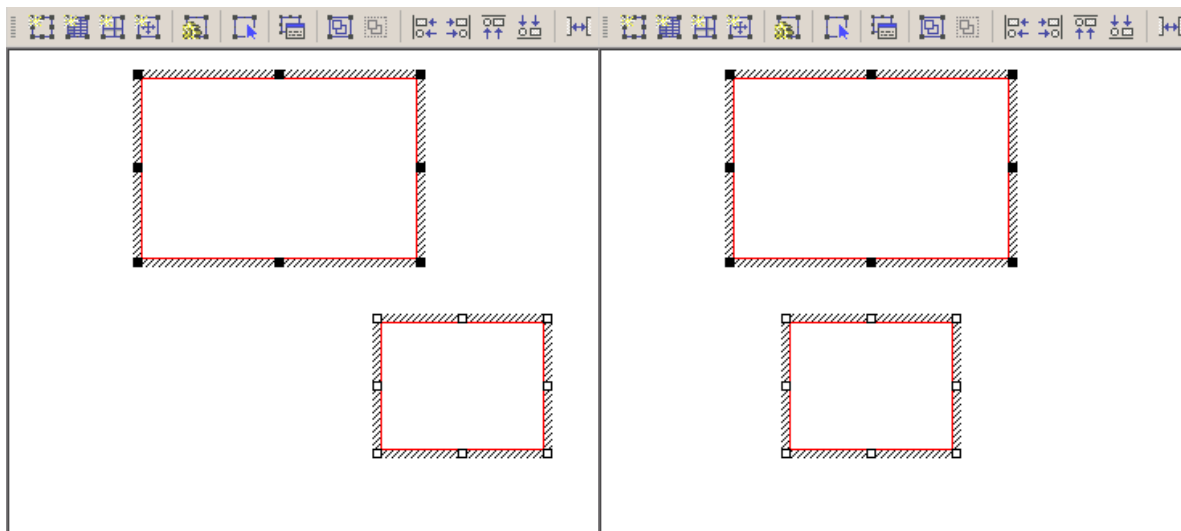


Rys. 4-36 Efekt zastosowania polecenia *Wyrównaj* → *Do dołu* lub przycisku na pasku narzędzi: ,

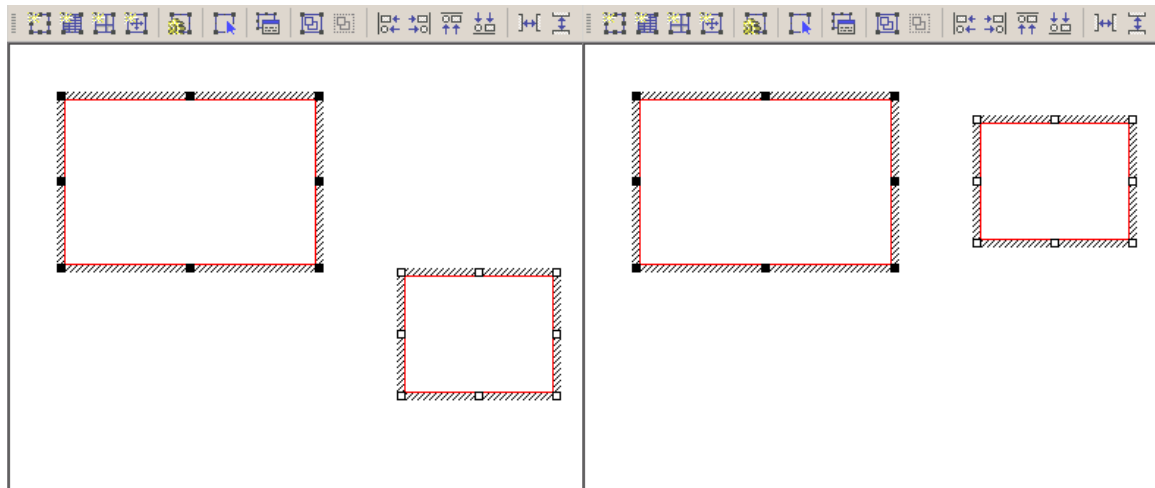
W menu Układ → Wycentruj znajdują się dwa polecenia umożliwiające ustawianie względem siebie środków elementów:

Polecenie W poziomie  ustawia środki zaznaczonych elementów tej samej linii poziomej, co środek obiektu odniesienia; W pionie  – ustawia w tej samej linii pionowej, co środek obiektu odniesienia.

Na rysunkach 4-37 i 4-38 pokazano przykłady wykonania powyższych operacji dla dwóch zaznaczonych elementów.



Rys. 4-37 Efekt zastosowania polecenia *Wycentruj w poziomie* lub przycisku na pasku narzędzi: .

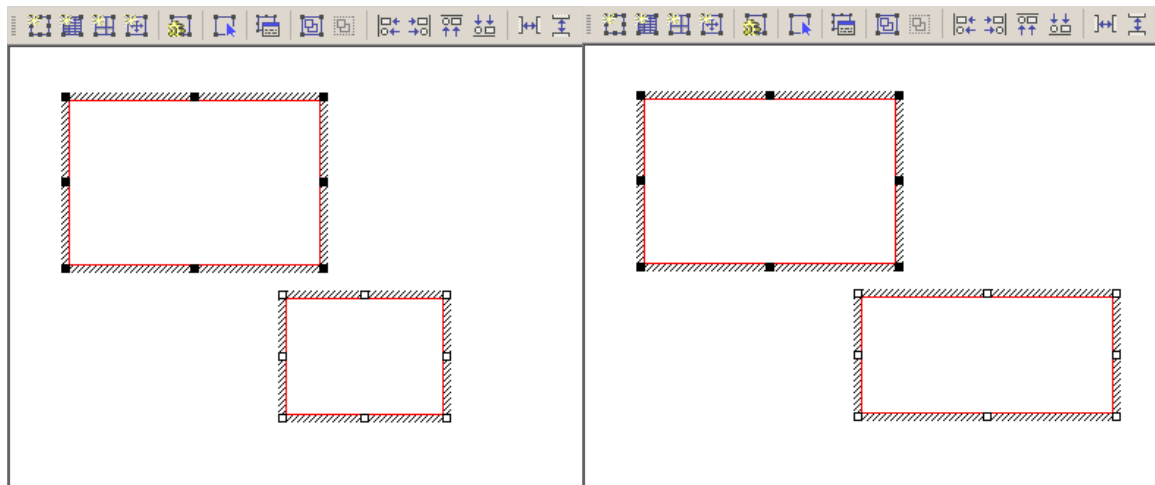


Rys. 4-38 Efekt zastosowania polecenia *Wycentrum pionie* lub przycisku na pasku narzędzi: 

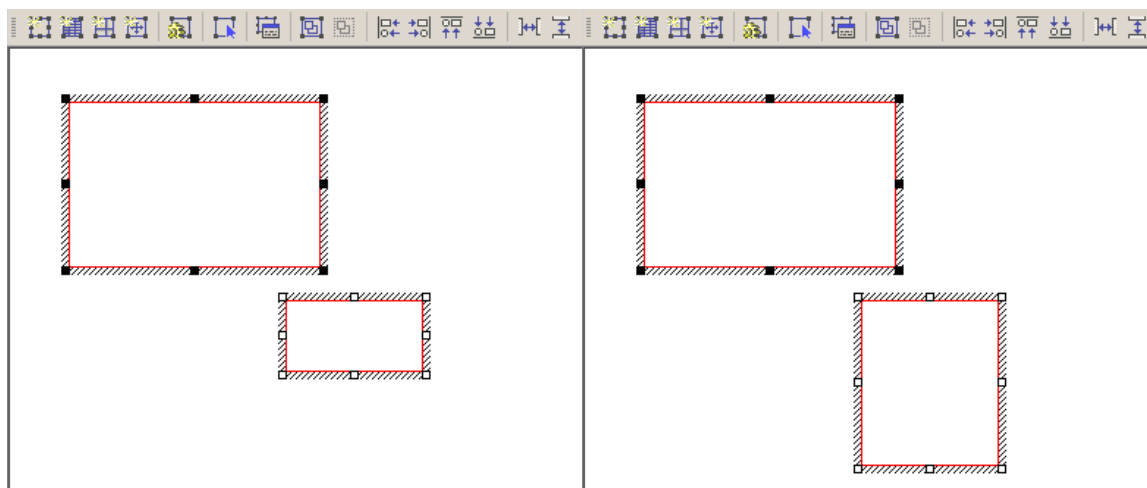
W menu Układ → dopasuj rozmiar dostępne są trzy polecenia:

Szerokość  umożliwia dopasowanie szerokości zaznaczonych elementów do szerokości obiektu odniesienia; Wysokość  – dopasowanie wysokości do obiektu odniesienia; Oba  – umożliwia zmianę obydwu rozmiarów stosunku do obiektu odniesienia.

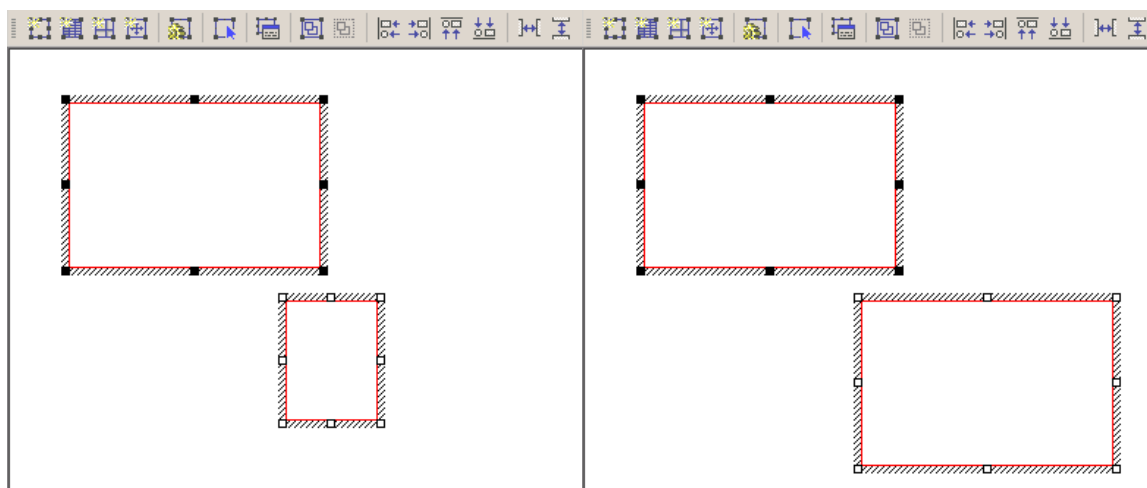
Na rysunkach 4-39, 4-40, 4-41 pokazano przykłady wykonania powyższych operacji dla dwóch zaznaczonych elementów.



Rys. 4-39 Efekt zastosowania polecenia *Dopasuj rozmiar → Szerokość* lub przycisku na pasku narzędzi: .



Rys. 4-40 Efekt zastosowania polecenia *Dopasuj rozmiar*→*Wysokość* lub przycisku na pasku narzędzi: .

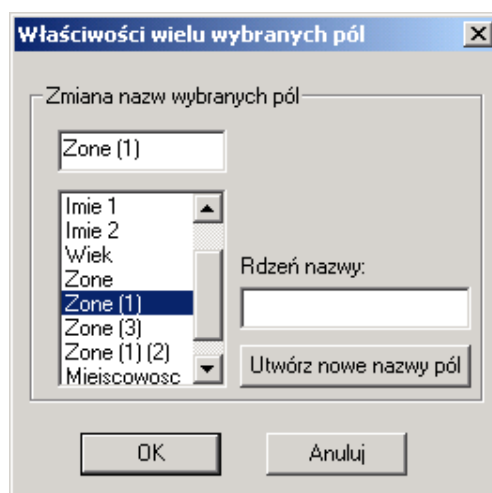


Rys. 4-41 Efekt zastosowania polecenia *Dopasuj rozmiar*→*Oba* lub przycisku na pasku narzędzi: .

4.9. Właściwości wielu wybranych obiektów.

Okno właściwości obiektu jest wykorzystywane z reguły do ustawiania parametrów pojedynczych pól. Parametry te w głównej mierze opisują sposób przetwarzania i rozpoznawania, dlatego też ich dokładny opis umieszczony został przy okazji prezentacji samych pól. Okno właściwości obiektu można wyświetlić wybierając polecenie Właściwości z menu Obiekt.

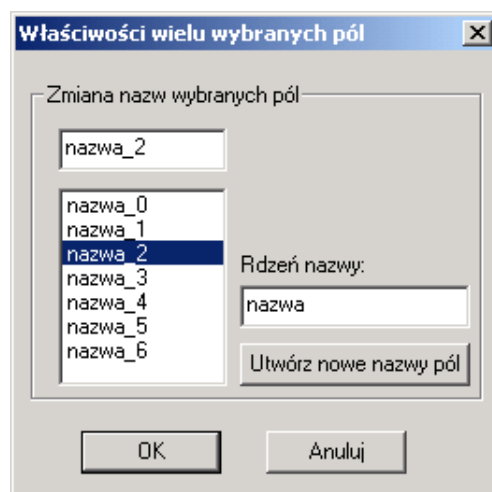
Okno właściwości można także wykorzystywać dla wielu zaznaczonych obiektów, Okno to umożliwia dokonanie szybkiej zmiany nazw pól na formularzu. Przykładowe okno dialogowe Właściwości wielu wybranych pól pokazane zostało na poniższym rysunku.



Rys. 4-42 Okno właściwości wielu wybranych obiektów.

Pierwsza metoda zmiany nazwy dotyczy pojedynczego pola. Aby zmienić nazwę pola należy zaznaczyć je, klikając na jego nazwie w dostępnej liście pól lewym przyciskiem myszki. W polu znajdującym się powyżej dostępnej listy wyświetlona zostanie wskazana nazwa, tam należy dokonać jej edycji. Powyższą operację można użyć w celu dokonania edycji pozostałych pól wyświetlonych na liście.

Druga metoda służy do zmiany nazw wszystkich zaznaczonych pól. Nowe nazwy pól składają się z dwóch części: rdzenia nazwy, wspólnego dla wszystkich pól, oraz dodawanego do rdzenia kolejnego identyfikatora. Wspólną część nazwy wszystkich zaznaczonych pól należy wpisać w polu *Rdzeń nazwy*. Kliknięcie przycisku **Utwórz nowe nazwy pól** spowoduje zmianę nazw wszystkich obiektów z listy. Do nowego rdzenia nazwy dodany zostanie kolejny numer obiektu w liście.

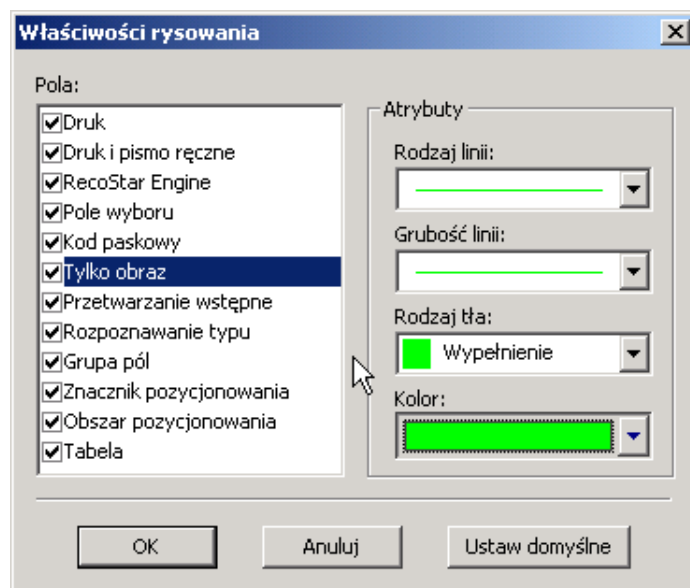


Rys. 4-43 Zmiana nazwy wszystkich obiektów w liście.

4.10. Zmiana atrybutów wyświetlania

Elementy umieszczane na formularzu mogą być wyświetlane w różny sposób (bądź też mogą być niewyświetlane wcale). Atrybuty wyświetlania poszczególnych elementów można zdefiniować w oknie dialogowym Właściwości rysowania. W celu jego wyświetlenia,

należy z menu Widok wybrać polecenie Właściwości rysowania. Przykładowe okno dialogowe zostało przedstawione poniżej.



Rys. 4-44 Okno atrybutów wyświetlania elementów formularza.

W liście *Pola* znajdują się elementy, dla których użytkownik może zdefiniować sposób wyświetlania. Włączone wyświetlanie elementu danego typu wskazuje zaznaczone pole wyboru ☒ znajdujące się z lewej strony nazwy elementu. Ponowne kliknięcie pola wyboru wyłącza wyświetlanie elementów danego typu.

Dla każdego typu elementu można ustawić atrybuty wyświetlania: rysowania krawędzi – linii elementu oraz jego wypełnienie. Z rozwijalnej listy *Rodzaj linii*, należy wybrać sposób jej rysowania (linia ciągła, przerywana). Z rozwijalnej listy *Grubość linii* należy wybrać jedną z grubości rysowania krawędzi. W rozwijalnej liście *Rodzaj tła* należy zdefiniować sposób wypełniania wnętrza elementu. Wybranie wartości *Brak* oznacza, że wnętrze elementu nie zostanie wypełniane.

Kliknięcie spowoduje zapamiętanie zdefiniowanych właściwości wyświetlania elementów; kliknięcie spowoduje powrót do poprzednich ustawień właściwości wyświetlania. Przycisk przywraca ustawienia domyślne programu FormAnalyzer Designer.

4.11. Panel właściwości

Panel właściwości jest narzędziem pozwalającym na szybki podgląd i modyfikację ustawień elementów składowych formularza. Panel ten jest dopełnieniem - z pewnymi ograniczeniami - funkcjonalność okna właściwości elementu. Dostępny jest równocześnie z obrazem formularza, co znacznie ułatwia korzystanie z niego.

Wyświetlaniem panelu właściwości steruje polecenie *Właściwości* dostępne w menu *Widok*.

Na poniższym rysunku przedstawiono widok panelu właściwości w czasie pracy z formularzem *Athene Life*. Z lewej strony widać panel właściwości, gdy na formularzu nie zaznaczono żadnych obiektów - są w nim wyświetlone zdefiniowane ustawienia samego

formularza. Z prawej strony pokazano panel właściwości dla zaznaczonego pola *Nr Agenta*.

Nazwa	Wartość
Typ	Formularz
Nazwa	Athene Life
Obrót	0 stopni
Tryb rozpoznawania	Standardowy
Tryb usuwania przekosu	Na podstawie obrazu
Dynamiczne znajdowanie pól wyboru	
Tło	C:\FormAnal\Athene Life\al.tif
Opis	
Szerokość	1648
Wysokość	2338
Rozdzielczość	200
Liczba pól	27

Nazwa	Wartość
Typ	Druk i pismo ręczne
Nazwa	Nr agenta
Leksykon	Cyfry
Klasyfikator	Pismo ręczne
Znaki nie stykają się	
Kratki dla znaków	☒
Tekst bez spacji	☒
Pole wieloliniowe	
Trigramy	
Usuwanie szumu	☒
Usuwanie tła	
Typ przetwarzania końcowego	brak
Wartość domyślna	
Słowniki	ADMIN.Lkplmiona
Nazwa pola	imie
Limit czasu	0
Wykorzystaj w rozpoznawaniu	☒
Słownik kompletny	
Nazwa skryptu	Wiek
Typ argumentu	tekst
Treść skryptu	dim Wiek1\nWiek1 = Year(Date())\n
Uaktualnij wartość pola	
Kryterium poprawności DLL	

Rys. 4-45 Panel właściwości dla formularza Athene Life (z lewej) i pola Nr Agenta (z prawej).

Każda linia na panelu właściwości rozgranicza kolejne ustawienia elementu. Dla każdej kolejnej właściwości elementu podana jest nazwa i wartość. Jeżeli wartość wyświetlona jest w kolorze czarnym oznacza to, że może być ona poddana edycji. Kolor szary wartości uniemożliwia użytkownikowi jej edycję.

Typ poszczególnych ustawień elementu określa, jaki rodzaj wartości należy wprowadzić w odpowiednie pola. Na przykład: dla wartości binarnych (włącz/wyłącz) włączenie opcji jest sygnalizowane ikoną ☒, (gdy opcja jest wyłączona pole wartości jest puste). W powyższym przykładzie aktywne są opcje: Kratki dla znaków, Tekst bez spacji, Usuwanie szumu, Wykorzystaj w rozpoznaniu.

W przypadku, gdy w polu znajduje się symbol, oznacza to, że konkretną wartość należy wskazać z rozwijanej listy. Dla pola *Nr Agenta* są to np. *Typ* (pola), *Leksykon* i *Klasyfikator* używane w czasie rozpoznawania jego treści.

Ustawienia zdefiniowane w linii *Typ* będą miały decydujące znaczenia, jakie właściwości będą dostępne w pozostałej części panelu. Edycja innych ustawień może także spowodować rozszerzenie lub skrócenie dostępnej listy ustawień. W panelu właściwości pola *Nr Agenta*, po wybraniu z listy słowników *LkpAgenci* spowodował uzupełnieniu listy ustawień poleceniami związanymi ze słownikami: *Nazwa pola* (słownika), *Limit czasu*, *Wykorzystaj w rozpoznawaniu* i *Słownik kompletny*.

Panel właściwości jest wiernym odwzorowaniem (pod względem treści, jak i możliwości ustawień) okien właściwości poszczególnych elementów formularzy. Poszczególne okna dialogowe związane z konkretnymi elementami został omówione w kolejnych rozdziałach. W rozdziale 5 - Formularze omówione zostały właściwości formularza i pól rozpoznawania

typu formularza, w rozdziale 6 Pozycjonowanie pól na formularzu – właściwości elementów pozycjonujących. W rozdziale 7 Pola rozpoznawane opisane zostały podane właściwości pól rozpoznawanych na formularzu. W rozdziałach 8 - Grupowanie pól

Celem grupowanie pól jest utworzenie zależności pomiędzy polami składowymi. Może ona ułatwiać proces rozpoznawania, dzieje się to np. przy grupowaniu pól tekstowych z włączoną opcją *Wybierz najlepszy wynik*. Relacja ta może umożliwić sprawdzanie poprawności danych, dzieje się to np. przy grupowaniu pól wyboru z włączoną opcją *Łącz pola składowe*. Grupowanie może także służyć jedynie do stworzenia formatki weryfikacyjnej, dzieje się to przy włączeniu opcji *Wiersz danych* lub *Sekcja danych*.

W każdym przypadku efektem grupowania jest nowe pole nadrzędne o unikalnej nazwie. Przy włączonej opcji *Łącz pola składowe* i *Wybierz najlepszy wynik*, pola wchodzące w skład grupy nie są dostępne w procesie przetwarzania. Przy włączonej opcji *Wiersz danych* i *Sekcja danych*, pola wchodzące w skład grupy dostępne są w procesie walidacji i eksportu. W tym przypadku pole nadrzędne nie podlega eksportowi.

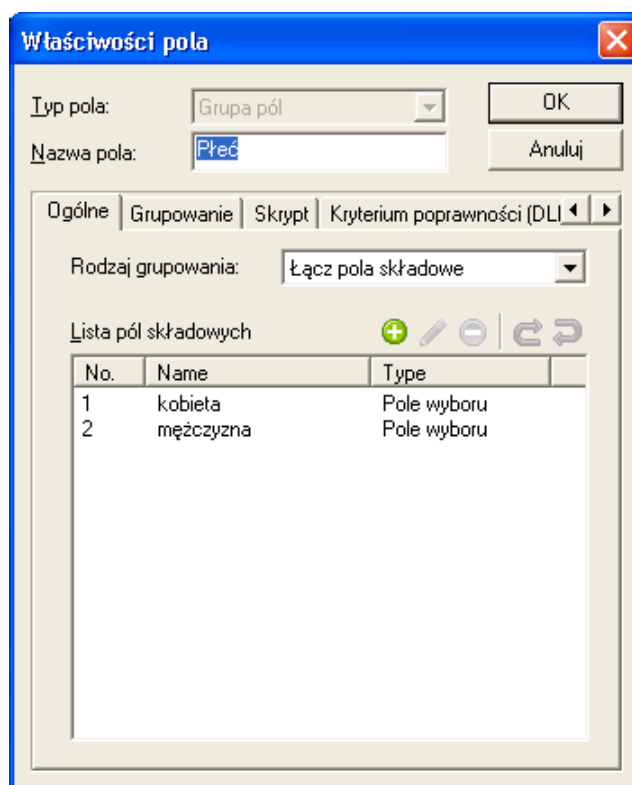
Pola zgrupowane można grupować zgodnie z następującymi zasadami:

- Opcja *Wybierz najlepszy wynik* dostępna jest tylko dla pól tekstowych
- Opcja *Łącz pola składowe* dostępna jest tylko dla pól, które generują ten sam typ danych: tekst lub pola wyboru
- Opcja *Wiersz danych* nie jest dostępna dla pól zgrupowanych metodą *Wiersz danych* i *Sekcja danych*
- Opcja *Sekcja danych* nie jest dostępna dla pól zgrupowanych metodą *Sekcja danych*

Niezależnie od sposobu grupowania, który zastosujemy, tworzenie grupy pól przebiega tak samo. Najpierw należy zdefiniować na formularzu pola składowe, pamiętając o wybraniu właściwych parametrów rozpoznawania, weryfikacji i eksportu. Jest to bardzo istotne, ponieważ po utworzeniu grupy pól, dostęp do poszczególnych składowych jest utrudniony. Żeby zmienić parametry pola składowego, trzeba zgrupowane pola rozgrupować, zmienić parametry pola i ponownie utworzyć grupę. Inną metodą zmiany właściwości pól zgrupowanych jest ustawienie odpowiednich cech w dialogu *Pola* w sekcji *Pola wewnętrzne*. Dokładny opis znajduje się w rozdziale 9 - Właściwości pól.

Pola, które mają utworzyć grupę, należy wybrać. Kolejność wybierania pól jest istotna przy łączeniu treści pól. Ich wartości złożą się na wartość grupy w kolejności wybierania.

W celu zgrupowania pól należy wyświetlić okno dialogowe *Właściwości pola* pokazane na poniższym rysunku korzystając z jednej z dostępnych metod: po uprzednim zaznaczeniu żądanych pól klikając na pasku narzędzi przycisk , lub wybierając z menu *Obiekt* polecenie *Grupuj*. Na ekranie zostanie wyświetlone okno dialogowe *Właściwości pola*.



Rys. 8-1 Przykładowe okno właściwości grupy pól.

Na pierwszej zakładce należy podać nazwę nowego pola grupowego, zgodnie z zasadami nazywania pól opisanymi w punkcie 7.2 - Typy i nazwy pól. W zależności od typów pól składowych w rodzaju grupowania dostępne mogą być następujące elementy do wyboru: *łącz pola składowe*, *Wybierz najlepszy wynik*, *Wiersz danych* i *Sekcja danych*. W *Liście pól składowych* podane są kolejno pola wchodzące w skład grupy. Kolejność pól składowych można zmienić używając przycisków (przeniesienie pola o jedną pozycję do góry) i (przeniesienie pola o jedną pozycję na dół).

Na drugiej zakładce użytkownik może zdefiniować szczegółowe cechy związane z grupowaniem i walidacją pola.

Grupy pól wyboru 9 - Grupy pól tekstowych omówiono właściwości grup pól rozpoznawanych. Informacje o właściwościach poszczególnych elementów mają zastosowanie także w panelu właściwości.

Pomimo wielu podobieństw pomiędzy panelem, a oknem właściwości istnieją także pewne różnice. W panelu właściwości pewne ustawienia nie są dostępne, np. w przypadku parametrów związanych z przetwarzaniem wstępnym. W panelu istnieje możliwość, włączenia dowolnego kroku przetwarzania wstępnego, nie można jednak ustawić jego parametrów. W celu zdefiniowania lub edycji tych parametrów, należy skorzystać z okna dialogowego właściwości odpowiedniego pola.

Drugą istotną różnicą między panelem, a oknem właściwości jest moment sprawdzania poprawności ustawień. W przypadku okna właściwości sprawdzenie następuje dopiero przy zamykaniu okna, w momencie kliknięcia przycisku . W panelu właściwości sprawdzenie poprawności dokonanych zmian następuje po zakończeniu edycji dowolnej pozycji na liście właściwości.

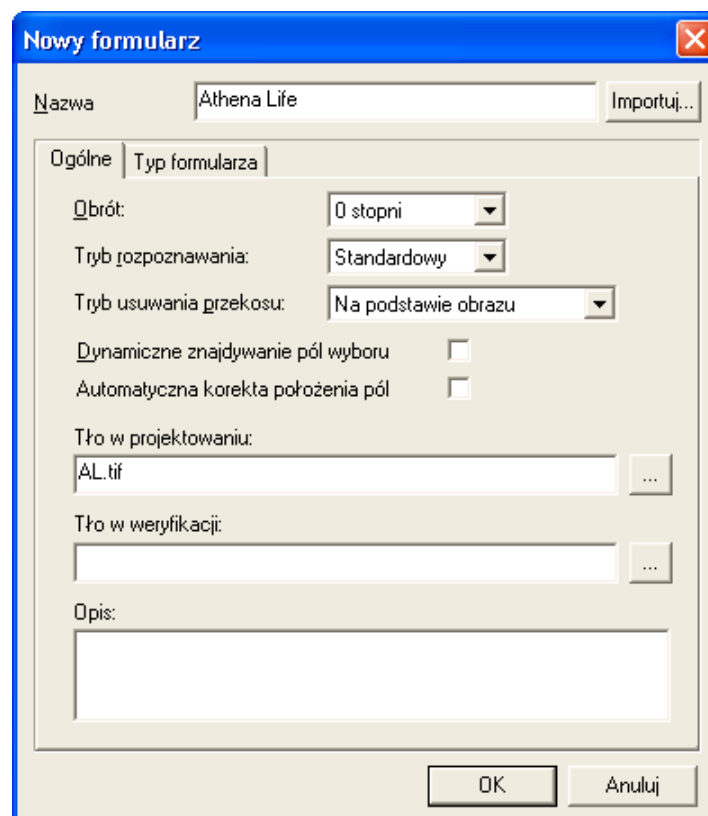
5. Formularze

Pierwszym krokiem wykonywanym przez FormAnalyze Engine w procesie rozpoznawania strony jest rozpoznanie typu formularza. W module FormAnalyze Designer należy zdefiniować informacje umożliwiające pewne rozróżnienie typów formularzy istniejących w jednym pliku konfiguracji. Do identyfikacji typu formularza można wykorzystać różne metody, w zależności od rodzaju przetwarzanych formularzy.

5.1. Dodanie nowego formularza

W celu utworzenia nowego formularza, należy z menu *Formularz* wybrać polecenie *Nowy...*

Na ekranie zostanie wyświetlone okno dialogowe *Nowy formularz*.



Rys. 5-1 Okno dialogowe definiowania nowego formularza – pierwsza zakładka

W pierwszej zakładce *Ogólne*, należy wprowadzić podstawowe dane dotyczące formularza oraz określić sposób jego przetwarzania wstępnego.

Dane ogólne obejmują:

1. Nazwę formularza.
2. Obraz wzorca formularza (*Tło w projektowaniu*).

Przed rozpoczęciem jakichkolwiek działań z nowym formularzem, należy wybrać plik zawierający wzorcowy obraz formularza. Na tym wzorcowym formularzu zostaną rozmieszczone wszelkie elementy związane z jego przetwarzaniem i rozpoznawaniem. Plik obrazu wzorcowego, należy wskazać w standardowym oknie otwierania pliku systemu Windows, które wyświetlone zostanie po uprzednim kliknięciu przycisku . Plik obrazu wzorcowego o rozszerzeniu TIFF musi być czarno-biały, zapisany bez kompresji albo z kompresją CCITT G3-1D lub G4. Ze względu na dokładność rozmieszczania elementów formularz wzorcowy musi mieć usunięty przekos (przekrzywienie obrazu formularza powstające w czasie skanowania).

3. Dodatkowy podgląd w procesie weryfikacji (*Tło w weryfikacji*).

Obrazek ten będzie wykorzystany w procesie weryfikacji. W podglądzie całej strony będzie wyświetlany obraz powstały poprzez nałożenie oryginalnej, zeskanowanej strony na podgląd zdefiniowany w procesie projektowania. Jest to szczególnie wygodne podczas przetwarzania dokumentów z usuniętym tłem (color dropout). W sytuacji tej użytkownik może mieć problem z weryfikacją danych, ponieważ będzie mu brakowało kontekstu pól. Plik podglądu, należy wskazać w standardowym oknie otwierania pliku systemu Windows, które wyświetlone zostanie po uprzednim kliknięciu przycisku . Plik o rozszerzeniu TIFF powinien być wersją kolorową lub grayscale obrazu wzorca formularza.

4. Automatyczna korekta położenia pól. Opcja dotyczy pozycjonowania pól formularza i ma zastosowanie do formularzy, w przypadku których dopuszczamy drobne zmiany w rozmiarze pomiędzy poszczególnymi egzemplarzami formularza (zmiany rzędu od -5% do +5%). Dotyczy to np. formularzy które są udostępniane klientom w formie elektronicznej (np. PDF), a następnie drukowane przez klienta, podpisywane i odsyłane w formie papierowej. Wydruk formularza z wykorzystaniem różnych drukarek, powoduje, że poszczególne egzemplarze będą różniły się między sobą rozmiarem - mogą być delikatnie ściśnięte lub rozciągnięte w stosunku do formularza wzorcowego w oparciu o który przygotowano szablon. Zaznaczenie opcji powoduje zmianę sposobu pozycjonowania pól formularza w procesie jego rozpoznawania przez moduł Engine. Standardowo po wykryciu znaczników pozycjonowania przeprowadzana jest korekta położenia pozostałych pól na formularzu. Po zaznaczeniu powyższej opcji, na podstawie odległości pomiędzy wykrytymi znacznikami pozycjonowania obliczane są współczynniki o jakie zmienił się rozmiar formularza w stosunku do wzorca (osobno w pionie i w poziomie). Następnie w oparciu o obliczony współczynnik wprowadzana jest korekta tak w położeniu jak i rozmiarze wszystkich pól formularza (dotyczy to również pól pozycjonowania). Po wprowadzeniu korekty ponownie wykrywane są wszystkie znaczniki pozycjonowania a następnie, w oparciu o nie, przeprowadzana jest korekta położenia pozostałych pól formularza, po czym moduł Engine przechodzi do dalszego przetwarzania formularza.

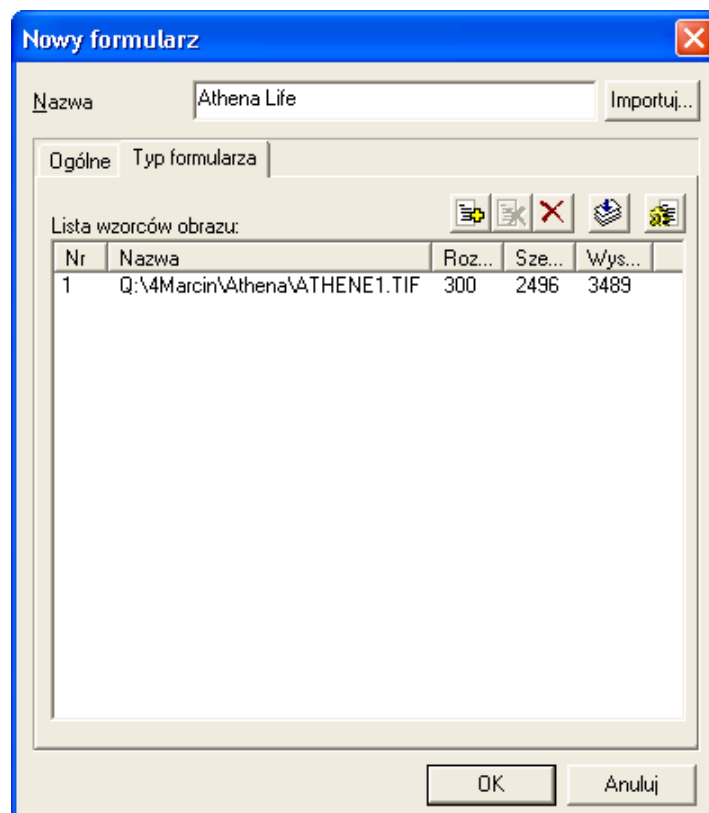
UWAGA! Na formularzach co do których przewidujemy możliwość delikatnych zmian w ich rozmiarze nie należy stosować pól pozycjonowania typu „Obszar pozycjonowania”. Na tego typu formularzach pola te nie będą rozpoznawane poprawnie, a ich zastosowanie może prowadzić do pogorszenia jakości procesu pozycjonowania pól.

Więcej informacji na temat pozycjonowania pól formularza znajdzie czytelnik w rozdziale „Pozycjonowanie pól na formularzu”.

5. Opis formularza.

6. Tryb rozpoznawania formularza. W wersji obecnej programu są stosowane dwa tryby rozpoznawania: standardowy i faks. Jak sugeruje nazwa, tryb standardowy jest używany w większości przypadków, w których obrazy rozpoznawanych dokumentów są uzyskiwane przez skanowanie dokumentów papierowych. W takich przypadkach, w zależności od typu dokumentów stosuje się skanowanie z rozdzielczością od 200 do 400 punktów na cal (dpi). Jakość obrazu uzyskanego ze skanera jest dobra. Odmienne przedstawia się sprawa z dokumentami, których obrazy uzyskano z faksu. Faksy przesyłają obrazy z rozdzielczością 150 dpi, a jakość obrazu silnie zależy od zakłóceń na łączach. W trybie faks są stosowane algorytmy uwzględniające małą rozdzielczość obrazu i duże, specyficzne dla faksowania zakłócenia obrazu.

W drugiej zakładce *Typ formularza* znajduje się lista obrazów wzorcowych, czyli plików w formacie TIFF, zawierających obrazy zeskanowanych formularzy.



Rys. 5-2 Okno dialogowe definiowania nowego formularza –druga zakładka

Przetwarzanie wstępne całego formularza to dwie operacje, których zadaniem jest przekształcenie formularza rozpoznawanego do postaci formularza wzorcowego:

- Obrót formularza. Jak już wspomniano wcześniej, formularz może być z różnych względów skanowany w orientacji innej niż obraz wzorcowy. Wybranie jednej z czterech wartości kąta obrotu formularza ma doprowadzić rozpoznawany formularz do takiej orientacji jak formularz wzorcowy.
- Usunięcie przekosu formularza. Wskutek nierównomiernego przesuwu papieru w skanerze obrazy dokumentów są najczęściej pochylone o pewien niewielki kąt. Stanowi to poważną trudność w precyzyjnym pozycjonowaniu pól rozpoznawanych. Wybranie opcji usuwania przekosu spowoduje, że dla każdego formularza będzie obliczany kąt przekosu, a następnie formularz zostanie obrócony o obliczoną

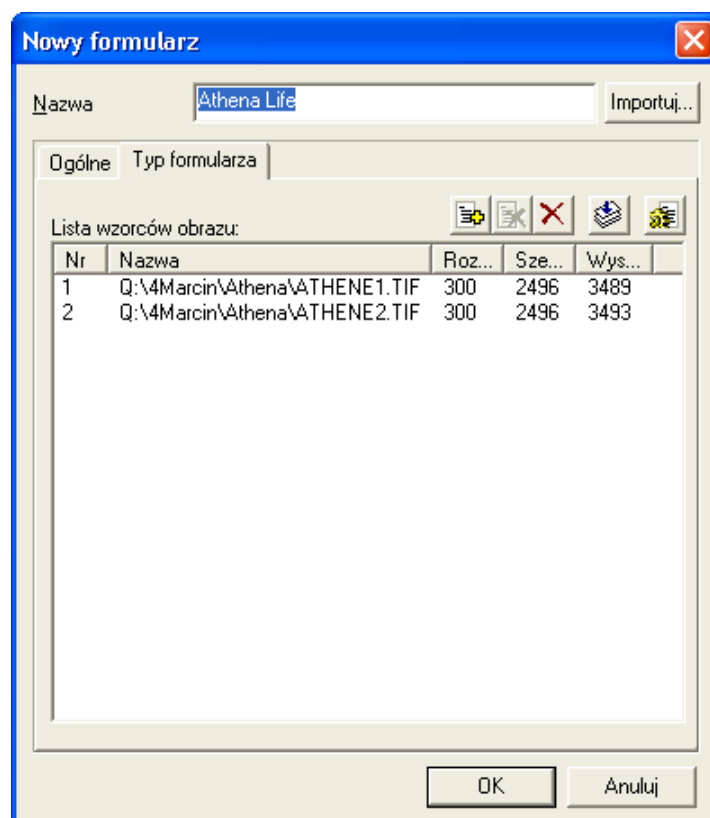
wartość. Przy korzystaniu ze skanera automatycznie kontrolującego przekos dokumentu w momencie skanowania, można zrezygnować z wyżej wymienionej opcji. Przekos możemy usuwać na podstawie obrazu strony - tę opcję stosujemy, gdy stałe fragmenty formularza nie są usuwane w czasie skanowania. Usuwanie przekosu na podstawie znaczników należy stosować, gdy formularz jest wydrukowany w kolorze usuwanym w czasie skanowania.

Do wyznaczenia przekosu zostaną użyte wydrukowane na czarno znaczniki. Informacje podane w zakładce *Ogólne* nie zawierają danych pozwalających na identyfikację formularza. Dane, wymagane do automatycznego rozpoznania typu formularza, można podać w zakładce *Typ formularza*.

5.2. Automatyczne rozpoznanie typu formularza.

Automatyczne rozpoznawanie typu formularza jest definiowane w zakładce *Typ formularza* w oknie dialogowym *Właściwości formularza*. W zakładce tej, należy wprowadzić dane pozwalające obliczyć parametry automatycznego rozpoznawania wyglądu formularza. Dane te stanowi lista obrazów wzorcowych. Są to pliki w formacie TIFF zawierające obrazy zeskanowanych formularzy.

Orientacja formularzy powinna być taka sama, jak formularzy skanowanych w trakcie eksploatacji systemu.



Rys. 5-3 Okno dialogowe definiowania nowego formularza - parametry rozpoznawania wyglądu.

W celu dodania nowego pliku zawierającego obraz formularza, należy w oknie dialogowym *Właściwości formularza* kliknąć przycisk *Dodaj* , a następnie w

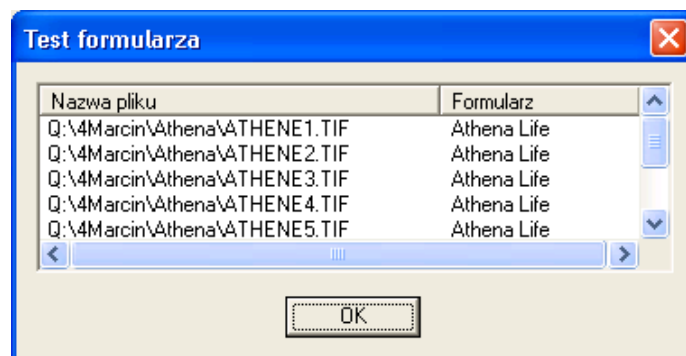
standardowym oknie dialogowym Otwórz systemu Windows wskazać żądany plik. Kolejność plików na liście jest bez znaczenia.

W celu usunięcia pliku z Listy wzorców obrazu, należy zaznaczyć żądany plik, a następnie kliknąć przycisk Usuń . Kliknięcie przycisku  spowoduje usunięcie wszystkich plików z listy wzorców.

Po dodaniu obrazów formularzy, należy obliczyć parametry opisujące wygląd formularza danego typu. W tym celu, należy kliknąć przycisk Przelicz .

Polecenie to wykonywane jest dla całego pliku konfiguracji. Istnieje możliwość dodania obrazów wzorcowych dla wielu formularzy, a następnie użycia polecenia przelicz dla wszystkich formularzy.

Automatyczne rozpoznawanie wyglądu formularza można przetestować. W tym celu, należy kliknąć przycisk Test . Operacja ta polega na identyfikacji typu formularza poprzez automatyczne rozpoznanie wyglądu. Podczas testu brane pod uwagę są wszystkie typy formularzy zdefiniowanych w pliku konfiguracji. Kliknięcie przycisku Test spowoduje wyświetlenie standardowego okna otwierania plików, w którym należy wskazać pliki obrazów formularzy przeznaczone do testu. Wynikiem testu jest nazwa formularza rozpoznana dla każdego pliku obrazu lub informacja, że formularz nie został zidentyfikowany. Przykładowy wynik testu został przedstawiony na rysunku poniżej.



Rys. 5-4 Okno wyników testu automatycznego rozpoznawania formularza.

Automatyczne rozpoznanie wyglądu formularza nie zawsze jest wystarczające. W przypadku, gdy wyniki testu automatycznego rozpoznawania wyglądu formularza zawierają błędy lub w pliku konfiguracyjnym występują formularze bardzo podobne, należy użyć innych metod identyfikacji formularza: pola detekcji typu lub pola nazwy formularza.

5.3. Rozpoznanie typu formularza - pole rozpoznawania typu

Pole rozpoznawania typu służy do rozróżniania formularzy, dla których automatyczne rozpoznanie wyglądu nie daje jednoznacznych wyników lub, które przy większej ich liczbie mogą zostać rozpoznane błędnie. Użytkownik decyduje, które formularze należy traktować, jako podobne.

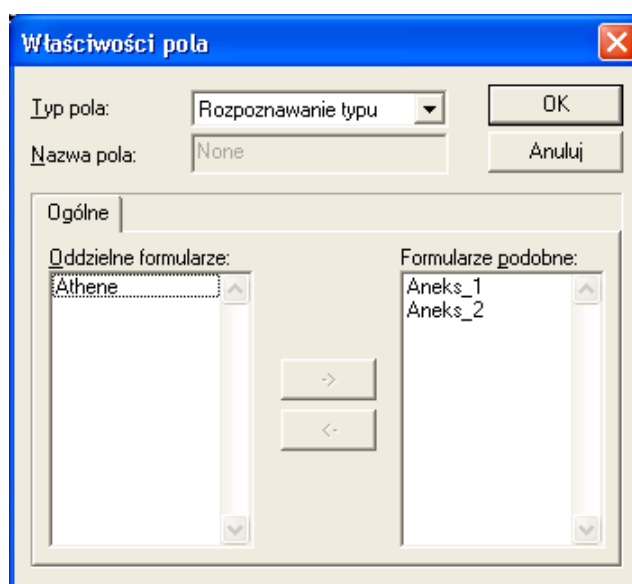
Na każdym z formularzy zaliczonych do grupy podobnych, należy umieścić przynajmniej jedno pole rozpoznawania typu. Pole to umieszcza się na elemencie graficznym, który

jest nieobecny na innych formularzach. Kiedy mamy do czynienia kilkoma rodzajami podobnych formularzy, na każdym z nich pole rozpoznawania typu może zostać umieszczone w innym miejscu.

Przed rozpoczęciem operacji rozpoznawania wykonywane jest pozycjonowanie pola rozpoznawania typu, czyli obliczenie jego pozycji na konkretnym formularzu. Dodatkowo do rozpoznawania typu formularza będą potrzebne pola pozycjonujące. Zasady rozmieszczania i ustawiania parametrów pól pozycjonujących omówiono w rozdziale 6 - Pozycjonowanie pól na formularzu. Kolejność wykonywania operacji umieszczania pól na formularzu nie jest istotna. Można najpierw dodać do formularza pole rozpoznawania typu, a następnie elementy pozycjonujące.

Tworzenie pola rozpoznawania typu przebiega analogicznie, jak w przypadku każdego innego pola rozpoznawanego. Szczegółowe informacje na ten temat znajdują się w rozdziale 7). Aby utworzyć pole rozpoznawania typu, należy z menu Obiekt → Nowy

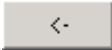
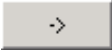
wybrać polecenie Pole. Kursor przyjmie postać: . Następnie, należy zaznaczyć prostokątny obszar pola rozpoznawanego, analogicznie do tworzenia prostokątnych elementów na formularzu. Pierwszy wierzchołek jest zaznaczany w punkcie, w którym przyciśniemy lewy klawisz myszy, wierzchołek przeciwległy w punkcie, w którym puścimy lewy klawisz myszy. Bezpośrednio po zaznaczeniu prostokąta zostanie otwarte okno dialogowe Właściwości pola. Z listy Typy pola, należy wybrać pozycję Rozpatrywanie typu.

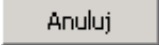


Rys. 5-5 Okno właściwości pola detekcji typu.

Prostokąt *Nazwa pola* pozostaje nieaktywny. Wartość umieszczona w tym polu służy jedynie do identyfikacji typu i nie jest później używana.

W zakładce *Ogólne* okna dialogowego *Właściwości pola* znajdują się dwa obszary: *Oddzielne formularze* i *Formularze podobne*. Znajdują się w nich wszystkie formularze zdefiniowane w pliku konfiguracji, z wyjątkiem formularza bieżącego, na którym umieszczamy pole rozpoznawania typu. Obydwa wymienione obszary służą do zdefiniowania relacji podobieństwa między formularzami. W obszarze *Oddzielne formularze* umieszczone są formularze, które można jednoznacznie odróżnić od bieżącego w procesie automatycznego rozpoznawania wyglądu. Do zadań użytkownika, należy umieszczenie odpowiednich formularzy w obszarze *Formularze podobne*.

Do przenoszenia formularzy między obszarami służą przyciski:  i .

Przenoszenie z obszaru *Podobnych* do *Oddzielnych* odbywa się bez żadnych ograniczeń. Aby formularz można było przenieść do obszaru *Podobnych*, musi on mieć ustawiony taki sam kąt obrotu w przetwarzaniu wstępnym jak formularz bieżący. Jeżeli ten warunek nie jest spełniony, zostanie wyświetlone okno komunikatu o niezgodności kątów obrotu, a formularz nie zostanie przeniesiony do listy *Podobnych*. Kliknięcie  spowoduje dodanie do definicji formularza pola rozpoznawania typu, natomiast kliknięcie  - anulowanie dodawania nowego pola.

5.4. Rozpoznanie typu formularza - pole nazwy formularza.

Wstawianie pola nazwy formularza jest metodą dającą największą dokładność rozpoznawania typu formularza. Pole nazwy formularza jest polem rozpoznawanym, którego tekstowa wartość jest nazwą formularza. Na formularzu może znajdować się tylko jedno pole nazwy.

Jeżeli w pliku konfiguracji istnieje, chociaż jeden formularz zawierający pole nazwy formularza, to w pierwszej kolejności rozpoznawana jest wartość tego pola. W przypadku, gdy pole nazwy umieszczone zostało na więcej niż jednym formularzu, a położenie lub parametry rozpoznawania Formularze pól są różne, kolejno są rozpoznawane obszary, w których może się znajdować nazwa formularza. Jeżeli wynik rozpoznania odpowiada nazwie formularza, to rozpoznawanie typu jest zakończone. Jeżeli żaden z wyników rozpoznawania nie jest nazwą formularza, FormAnalyze Engine przechodzi do automatycznego rozpoznawania wyglądu formularza omówionego wcześniej.

Wartość pola musi być tekstem, dlatego też dopuszczalne są tylko pola kodu paskowego, tekstu drukowanego i tekstu drukowanego lub pisanego ręcznie. Pole nazwy formularza jest identyfikowane przez nazwę: **__FORMNAME**. Na formularzu pola muszą mieć unikalne nazwy, nie można umieścić więcej niż jednego pola nazwy. Dodatkowo, pole nazwy formularza musi znajdować się, jako pierwsze na liście pól.

Pole nazwy formularza jest polem specjalnym. Jako jedyne pole będące polem rozpoznawanym nie jest pozycjonowane na formularzu. W przetwarzaniu pola jest uwzględniany jedynie obrót formularza.

Pole nazwy formularza nie jest pozycjonowane, dlatego też, należy zwrócić uwagę na odpowiednio duży margines pozostawiony wokół treści nazwy formularza. Konsekwencją zbyt małego marginesu może być trafienie fragmentów obrazów innych pól. W przypadku pól drukowanych czy drukowanych i pisanych może to prowadzić do błędów rozpoznawania. Z praktyki wynika, że najskuteczniejsze wyniki rozpoznawania uzyskuje się, gdy w polu nazwy formularza znajduje się kod paskowy.

Pole nazwy definiuje się umieszczając na formularzu pole rozpoznawane. W tym celu, należy z menu Obiekt → Nowy wybrać polecenie Pole. Po zaznaczeniu obszaru pola otwiera się okno dialogowe Właściwości pola. Po wybraniu odpowiedniego typu pola, należy wprowadzić jego nazwę: **__FORMNAME**. Istotne są zarówno dwa podkreślenia na początku, jak i duże litery, jakimi nazwa jest napisana.

Po wykonaniu obydwu czynności, należy przystąpić do definiowania właściwości pola nazwy, co zostało szczegółowo opisane w rozdziale 7 - Pola rozpoznawane.

Pole nazwy formularza jest pełnoprawnym polem rozpoznawanym. Dlatego też można do niego stosować skrypty i funkcje z zewnętrznych bibliotek DLL modyfikujące rozpoznaną wartość pola.

Po zakończeniu definiowania formularza trzeba sprawdzić, czy pole `__FORMNAME` jest pierwsze w liście pól rozpoznawanych. Jeżeli nie, to należy zmienić kolejność tak, by pole nazwy formularza było pierwsze.

6. Pozycjonowanie pól na formularzu

Położenia pól na rozpoznawanym formularzu różnią się od położenia zdefiniowanych na formularzu wzorcowym. Może być to spowodowane podczas wydruku, chociaż najczęściej problemy tego typu pojawiają się w procesie skanowania dokumentu. Aby skompensować przesunięcia pól w stosunku do formularza wzorcowego, stosuje się pozycjonowanie. Ten proces polega na odszukaniu na formularzu rozpoznawanym położenia elementów pozycjonujących zdefiniowanych we wzorcu. Obszar poszukiwania jest ograniczony - ma to na celu zmniejszenie prawdopodobieństwa umieszczenia elementu pozycjonującego na wyglądającym podobnie detalu formularza. Ograniczenie pola poszukiwania może w niektórych przypadkach doprowadzić do sytuacji, że element pozycjonowania nie zostanie odnaleziony. W takim przypadku, jego funkcję przejmują sąsiednie elementy pozycjonujące. W przypadku, gdy na formularzu nie odnaleziono żadnego elementu pozycjonującego, jest zgłaszany błąd pozycjonowania, a formularz nie jest dalej rozpoznawany.

Na złożonych formularzach wymagających precyzyjnego pozycjonowania można zadać dwupoziomą strukturę elementów pozycjonujących. Elementy, charakterystyczne dla danego formularza, należy zaznaczyć jako podstawowe elementy pozycjonujące. Będą one wyszukiwane na formularzu w pierwszej kolejności. Dopiero, gdy zostanie ustalona ich pozycja, zostaną obliczone obszary, w których mogą znajdować się pozostałe elementy pozycjonujące. W tych obszarach będą wyszukiwane elementy, które uznano za nie będące podstawowymi.

Wykorzystanie elementów pozycjonujących jest następujące:

Dla pola rozpoznawanego są ustalane najbliższe elementy pozycjonujące odszukane na formularzu. FormAnalyze Engine zna położenie pola rozpoznawanego względem tych elementów pozycjonujących na wzorcu. Znając współrzędne elementów pozycjonujących na konkretnym dokumencie obliczane są współrzędne pola rozpoznawanego (dla tego konkretnie dokumentu). Dopiero teraz pole jest rozpoznawane.

FormAnalyze Designer pozwala na użycie dwóch typów elementów pozycjonujących:

- Znacznika pozycjonowania
- Obszaru pozycjonowania

Sposób definiowania i właściwości tych elementów omówione zostały w kolejnych punktach.

6.1. Znacznik pozycjonowania

Znacznik pozycjonowania to wzorec geometryczny utworzony przez przecinające się lub stykające się linie pionowe i poziome. Tego typu wzorce można spotkać niemal na każdym formularzu (np. narożniki tabel) i z dużą pewnością rozpoznać ustalając przy tym dokładnie współrzędne elementu.

Aby wstawić na formularz nowy znacznik pozycjonowania, należy z menu Obiekt → Nowy wybrać polecenie Znacznik pozycjonowania lub na pasku narzędzi kliknąć przycisk Nowy

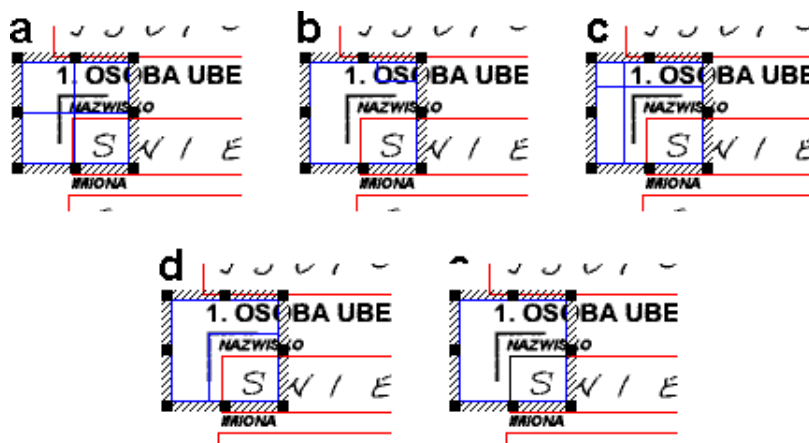
znacznik pozycjonowania . Gdy kursor przyjmie następujący kształt  , należy zaznaczyć prostokątny obszar znacznika na formularzu.

Rozmiar tego obszaru jest bardzo istotny. Tylko w nim będzie poszukiwany znacznik. Im większy zaznaczony obszar, tym większe przesunięcia formularza pozycjonowanego, przy których znacznik może być wykryty. Z drugiej strony, w większym obszarze mogą znaleźć się inne fragmenty formularza przypominające wzór linii znacznika. Może to spowodować nieprawidłowe rozpoznanie położenia znacznika pozycjonowania na dokumencie.

Po zaznaczeniu obszaru pola, umieszczone zostają w nim przecinające się linie oznaczające dokładne współrzędne znacznika. Zazwyczaj, bezpośrednio po ustawieniu znacznika na formularzu, linie nie pokrywają się dokładnie z punktem przecięcia na obrazie dokumentu. Precyzyjne ustawienie współrzędnych znacznika pozycjonowania, należy zdefiniować wybierając polecenie Format wewnętrzny z menu Obiekt → Nowy. Gdy

kursor zostanie umieszczony wewnątrz znacznika, przyjmie on następującą postać: . Aby zmienić położenie przecięcia linii, należy kliknąć i przytrzymać lewy przycisk myszki

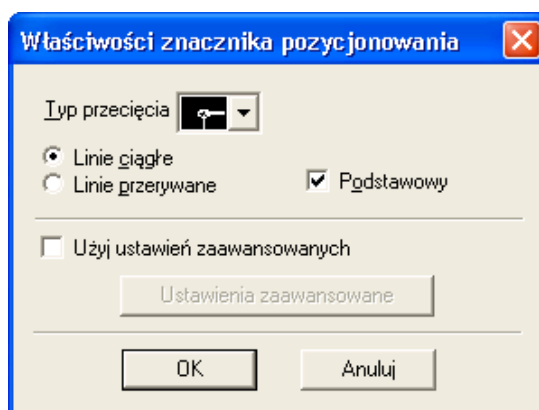
w dowolnym miejscu wewnątrz znacznika (kursor ponownie zmieni postać: ) i przeciągnąć linie w żądane miejsce. W zależności od położenia, może zmieniać się kształt linii przecięcia rysowanej przez FormAnalyze Designer. Zmiana kształtu jest spowodowana wykryciem w sąsiedztwie punktu innego wzoru przecinających się linii. Zmieniając położenie punktu przecięcia, warto zwrócić uwagę, czy w obszarze znacznika nie ma innych miejsc, w których uzyskujemy taki sam układ linii. Są to miejsca, które mogą zostać pomyłone ze znacznikiem.



Rys. 6-1 Modyfikacja rozkładu pola znacznika pozycjonowania.

Na rysunku powyżej przedstawiono kolejne fazy modyfikacji rozkładu pola znacznika. Rysunek (a) przedstawia położenie bezpośrednio po narysowaniu prostokąta znacznika na formularzu. Rysunki (b), (d) i (e) obrazują różne wzorce linii wykrywane przez FormAnalyze Designer w polu znacznika. Rysunek (b) pokazuje, że w niektórych przypadkach nie tylko linie mogą być interpretowane, jako znacznik. Na rysunku (e) widać, że nie zawsze prawidłowe linie, mogą zostać wzięte za znacznik. Prawidłową interpretację linii znacznika przedstawia rysunek (d), na którym występuje ten sam układ linii. Jeżeli program nie jest w stanie automatycznie rozpoznać wzoru przecięć pozostawia jego domyślne ustawienie (obrazek (c)).

W przypadku, gdy w obszarze znacznika znajdują się elementy, które mogą zostać zinterpretowane jako znacznik, należy zmniejszyć jego rozmiar. Obszar zawierający fragment obrazu o tym samym układzie linii, co znacznik powinien znaleźć się poza polem znacznika.



Rys. 6-2 Okno właściwości znacznika pozycjonowania.

Właściwości znacznika pozycjonowania można edytować także w oknie Właściwości znacznika pozycjonowania. Okno właściwości dla wybranego znacznika, można wyświetlić, po uprzednim zaznaczeniu żądanego znacznika pozycjonowania, jedną z dostępnych metod: kliknąć na pasku narzędzi przycisk Podaj właściwości obiektu , wybrać z menu Obiekt polecenie Właściwości lub kliknąć na obszarze zaznaczonego znacznika pozycjonowania prawym przyciskiem myszki i z menu kontekstowego wybrać polecenie Właściwości.

W oknie Właściwości znacznika pozycjonowania można ustawić rodzaj przecinających się linii, wybierając odpowiednią opcję: ciągłe lub przerywane. Z listy Typ przecięcia można wybrać typ przecięcia linii. Wszystkie dziewięć dostępnych typów przecięcia linii zaprezentowano poniżej.



Rys. 6-3 Typy przecięć linii w znaczniku pozycjonowania.

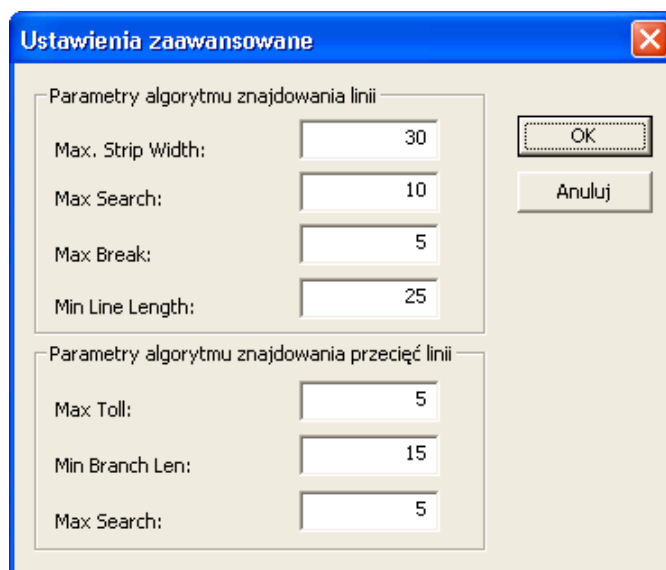
W prawej części okna znajduje się opcja Podstawowy. Jest ona domyślnie zaznaczona i umożliwia wyszukiwane na formularzu w pierwszej kolejności wszystkich znaczników pozycjonowania, które posiadają tą opcję włączoną. Gdy zostanie ona wyłączona znacznik wyszukiwany będzie w drugiej kolejności - po odszukaniu podstawowych elementów pozycjonujących.

Algorytm wyszukiwania znaczników pozycjonowania składa się z dwóch podstawowych kroków:

1. Znajdowanie linii.
2. Wyszukiwanie przecięć linii, oraz wybór elementu odpowiadającego wzorcowi znacznika.

W obszarze poszukiwania znacznika pozycjonowania mogą znaleźć się różne „zanieczyszczenia” (fragmenty linii, tabel, litery, szumy). Jak napisano wcześniej należy starać się unikać takich sytuacji, jednak nie zawsze jest to możliwe i tak np. litera T może zostać uznana za znacznik pozycjonowania. W związku z powyższym algorytm wyszukiwania znaczników musi odrzucać różne zanieczyszczenia. Jeżeli znacznik jest bardzo mały może to powodować, że nie będzie on poprawnie rozpoznawany. Rozwiązaniem problemu jest oczywiście umieszczenie na formularzu odpowiednio dużych i odseparowanych od treści formularza znaczników pozycjonowania. Niestety często mamy do czynienia z gotowymi, wydrukowanymi formularzami. Jeżeli znaczniki pozycjonowania są niewielkich rozmiarów a wokół nich istnieje sporo wolnej przestrzeni

możemy zmienić wartość współczynników wykorzystywanych przez algorytm wyszukiwania znaczników. W celu zmiany tych współczynników w oknie „Właściwości znacznika pozycjonowania” należy zaznaczyć opcję „Użyj ustawień zaawansowanych” a następnie kliknąć przycisk „Ustawienia zaawansowane”. Zmiany wartości współczynników dokonujemy w oknie „Ustawienia zaawansowane”.



Rys. 6-4 Okno Ustawienia zaawansowane.

Standardowe wartości współczynników wykorzystywanych przez algorytm przedstawiono na rysunku powyżej. W przypadku „małych” znaczników pozycjonowania (a za takie uważane są te dla których tworzące je linie mają długość mniejszą niż 0,5 cm) proponujemy zmianę wartości współczynników algorytmu na podane poniżej (w nawiasach podano standardowe wartości parametrów):

- Max. Strip Width: 15 (30)
- Max. Search: 5 (10)
- Max Break: 5 (5)
- Min Line Length: 10 (25)
- Max Toll: 5 (5)
- Min Branch Len: 5 (15)
- Max Search: 5 (5)

Użytkownik może spróbować doświadczalnie dobrać inne wartości parametrów, jednak nie zaleca się stosowania wartości mniejszych niż zaproponowane powyżej.

UWAGA! Na formularzu musi znajdować się przynajmniej jeden podstawowy element pozycjonujący (znacznik albo obszar pozycjonowania).

6.2. Obszar pozycjonowania

Obszar pozycjonowania jest wyróżnionym elementem graficznym, względem którego pola rozpoznawane mają stałą pozycję. Podstawową różnicą między obszarem pozycjonowania a znacznikiem pozycjonowania jest metoda wyszukiwania elementu na obrazie formularza. W przypadku znacznika pozycjonowania jest to detekcja przecinających się (stykających się) linii. Dla obszaru pozycjonowania jest porównywana

charakterystyka obrazu w obszarze. Proces „dopasowania” charakterystyk ze wzorca i formularza rozpoznawanego umożliwia sprawdzenie:

- Czy w obszarze pozycjonowania na wzorcu i formularzu obrazy są na tyle podobne, że chodzi o ten sam element pozycjonujący?

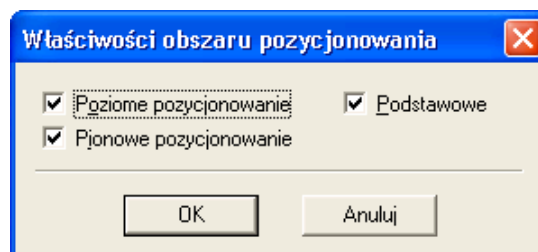
Jakie jest przesunięcie elementu na formularzu w stosunku do elementu na wzorcu?

W celu wstawienia na formularz nowego obszaru pozycjonowania, należy z menu Obiekt → Nowy wybrać polecenie Obszar pozycjonowania lub na pasku narzędzi kliknąć przycisk

Nowy obszar pozycjonowania . Gdy kursor przyjmie następującą postać: , należy zaznaczyć prostokąt obszaru pozycjonowania, analogicznie do pozostałych elementów formularza.

Stosunek rozmiarów prostokąta obszaru i rozmiaru elementów graficznych wchodzących do obszaru pozycjonowania ma podobne znaczenie jak w przypadku znaczników. Im większy prostokąt tym większe przesunięcia formularza da się skompensować. Jednocześnie istnieje zagrożenie, trafienia do wnętrza nowego obszaru niepożądanych fragmentów obrazów mogących spowodować błąd pozycjonowania (niewłaściwe współrzędne) lub całkowicie uniemożliwić pozycjonowanie. Błędy pozycjonowania pojawiają się jednak dopiero wtedy, gdy charakterystyka obrazu w obszarze zostanie mocno zakłócona.

Obszar pozycjonowania może służyć do ustalenia przesunięcia formularza w stosunku do wzorca w jednej lub obu współrzędnych. Ustawienia te można zdefiniować w oknie dialogowym Właściwości obszaru pozycjonowania, które należy wyświetlić po uprzednim zaznaczeniu żądanego obszaru pozycjonowania, używając jednej z dostępnych metod: kliknąć na pasku narzędzi przycisk Podaj właściwości obiektu , wybrać z menu Obiekt polecenie Właściwości lub kliknąć na zaznaczonym obszarze pozycjonowania prawym przyciskiem myszki i z menu kontekstowego wybrać polecenie Właściwości.



Rys. 6-5 Okno właściwości obszaru pozycjonowania.

Domyślnie zaznaczone są obydwie opcje pozycjonowania. Oznacza to, że w trakcie przetwarzania, będą dopasowywane do siebie charakterystyki obrazu w obydwóch współrzędnych. Wyłączenie dopasowania poziomego, powoduje, że w przetwarzaniu jest obliczane tylko przesunięcie formularza w stosunku do wzorca w pionie. Wyłączenie dopasowania pionowego, powoduje, że w przetwarzaniu jest obliczane tylko przesunięcie formularza w stosunku do wzorca w poziomie.

Odpowiednio zmienia się graficzna reprezentacja obszaru pozycjonowania. Na rysunku poniżej przedstawiono obszar pozycjonowania w obydwu kierunkach (a), obszar pozycjonowania w poziomie (b) i obszar pozycjonowania w pionie (c).



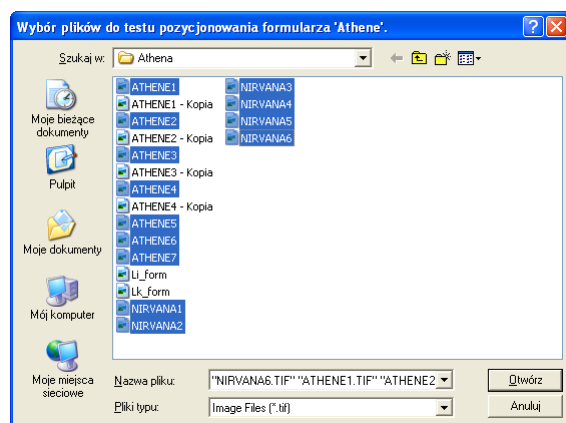
Rys. 6-6 Obszary pozycjonujące w różnych kierunkach.

Podobnie, jak w przypadku znaczników pozycjonowania, opcja *Podstawowy* jest domyślnie ustawiona.

6.3. Testowanie pozycjonowania formularza

Po zdefiniowaniu znaczników pozycjonowania na stronie formularza należy przystąpić do testowania definicji na rzeczywistych obrazach dokumentów. Pracę tą można wykonać w module FormAnalyzer Engine lub FormAnalyzer Designer. Testowanie pozycjonowania w module FormAnalyzer Engine wymaga wykonywania dodatkowych czynności w kilku modułach systemu FormAnalyzer pracujących w środowisku rozproszonym. Oznacza to, że ten sposób testowania definicji pozycjonowania jest czasochłonny i niewygodny. Testowanie pozycjonowania formularzy w module FormAnalyzer Designer można przeprowadzić wybierając z menu *Formularz* polecenie *Testowanie pozycjonowania...*

Na ekranie zostanie wyświetlone okno dialogowe *Wybór plików do testu pozycjonowania formularza*

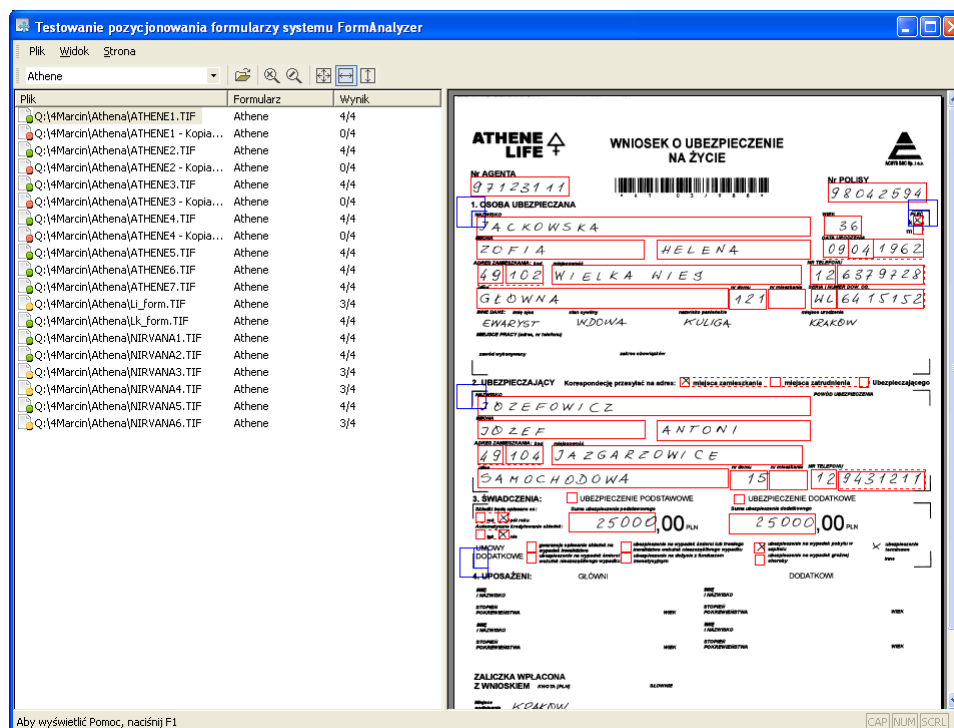


Rys. 6-7 Okno dialogowe wybór plików do testu pozycjonowania formularza

Użytkownik wskazuje wcześniej zeskanowane pliki obrazów, które będą służyły do testowania pozycjonowania aktualnie wybranego formularza. Powinny to być obrazy rzeczywistych dokumentów, zeskanowane w taki sam sposób jak w fazie produkcji.

Po zaznaczeniu plików i naciśnięciu przycisku *Otwórz* rozpocznie się proces obliczania pozycji pól aktualnie wybranego formularza dla wszystkich zaznaczonych obrazów. Proces obliczania pozycji pól poprzedzony jest fazą wstępnego przetwarzania obrazu zgodnie z definicją formularza. Etap wstępnego przetwarzania obrazu składa się z obrotu o zdefiniowany kąt oraz z usunięcia przekosu obrazu.

Na ekranie zostanie wyświetlone okno dialogowe *Testowanie pozycjonowania formularzy systemu FormAnalyzer*



Rys. 6-8 Okno dialogowe testowanie pozycjonowania

Podczas procesu przetwarzania obrazów i obliczania pozycji pól w pasku stanu wyświetlany jest postęp całego procesu.

Okno dialogowe podzielone jest na dwie części. Po lewej stronie okna wyświetlana jest lista testowanych obrazów oraz skrótowe informacje o wynikach testowania. W kolumnie *Plik* prezentowana jest nazwa testowanego obrazu oraz symboliczna informacja o wyniku testu. Ikona oznacza, że wszystkie znaczniki pozycjonowania prawidłowo się rozpoznały. Ikona oznacza, że wszystkie znaczniki pozycjonowania błędnie się rozpoznały. Jeżeli część znaczników pozycjonowania rozpoznała się prawidłowo, a część błędnie wyświetlana jest ikona . W kolumnie *Formularz* wyświetlana jest informacja o nazwie testowanego formularza. W kolumnie *Wynik* wyświetlana jest informacja o liczbie prawidłowo wypozycjonowanych znaczników w stosunku do liczby wszystkich znaczników na formularzu. Po prawej stronie okna wyświetlany jest podgląd aktualnie zaznaczonej strony. Na obrazie strony wyświetlane są znaczniki pozycjonowania oraz pozostałe pola formularza. Użytkownik może sprawdzić jakość pozycjonowania poprzez wzrokowe sprawdzenie pozycji pól formularza. W przypadku systematycznych błędów pozycjonowania należy zmienić definicję formularza. Zmiana definicji może polegać na zmianie rozmiaru i atrybutów znaczników pozycjonowania. Może zdarzyć się, że system FormAnalyzer nie będzie w stanie rozpoznać konkretnych znaczników pozycjonowania. Należy wtedy zastąpić błędny znacznik innym. Jeżeli wnioskiem z testowania pozycjonowania jest spostrzeżenie, że konkretny znacznik nie rozpoznaje się, to należy

zmienić definicję formularza, ponieważ w module FormAnalyze Engine problem ten będzie także występował.

W celu przeprowadzenia testu pozycjonowania formularza dla dodatkowych obrazów należy z wybrać z menu *Plik* wybrać polecenie *Otwórz...*

W celu przeprowadzenia testu pozycjonowania dla innego formularza należy zmienić formularz w pasku narzędzi a następnie wybrać z menu *Plik* wybrać polecenie *Przetwarzaj*.

Aby zamknąć okno dialogowe testowania pojonowania należy z menu *Plik* wybrać polecenie *Zamknij*.

7. Pola rozpoznawane

Pola rozpoznawane stanowią ostatni, najbardziej interesujący użytkownika, element projektu formularza, ponieważ to one składają się na treść (dane) dokumentu. Nie wszystkie z pól omówionych w tym rozdziale są rzeczywiście rozpoznawane, tzn. dają w wyniku treść dokumentu. Takimi wyjątkami są pola przetwarzania wstępnego i pola obrazu. Omówione wcześniej pole rozpoznawania typu formularza również nie daje treści dokumentu, pozwala jednak na identyfikację formularza. Pola przetwarzania wstępnego służą jedynie do wykonania pewnych operacji pomocniczych przy rozpoznawaniu. Pole obrazu jest przeznaczone do obliczenia współrzędnych fragmentu dokumentu, który będzie wyświetlany w trakcie weryfikacji.

Przetwarzanie pola rozpoczyna się od ustalenia jego współrzędnych na aktualnie rozpoznawanym formularzu. Wykorzystuje się tutaj obliczone uprzednio współrzędne elementów pozycjonujących. Do ustalenia dokładnego położenia pola rozpoznawanego są używane elementy pozycjonujące leżące najbliżej danego pola. Mając zestaw współrzędnych dla elementów pozycjonujących na formularzu wzorcowym i rozpoznawanej stronie dokumentu, oraz położenie pola na wzorcu, można obliczyć współrzędne pola dla konkretnej strony dokumentu.

Po ustaleniu współrzędnych pola jest ono przetwarzane. Przetwarzanie (i rozpoznawanie) jest operacją lokalną, tzn. dotyczy tylko tego fragmentu obrazu, który został zaznaczony w FormAnalyze Designer jako pole rozpoznawane. Przetwarzanie i rozpoznawanie pola nie modyfikuje danego obrazu. Tak, więc jeżeli kilka pól rozpoznawanych korzysta ze wspólnego fragmentu obrazu, to przy rozpoznawaniu każdego z nich fragment obrazu będzie taki sam.

Odstępstwem od powyższej reguły jest pole przetwarzania wstępnego. Operacje przetwarzania zadane we właściwościach pola modyfikują obraz strony w obszarze tego pola. Dlatego też, wszystkie pola, które będą rozpoznawane po polu przetwarzania wstępnego będą korzystały z obrazu zmodyfikowanego.

7.1. Kolejność przetwarzania pól

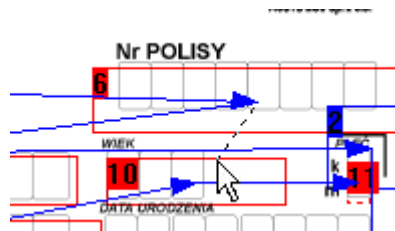
Pola rozpoznawane są przetwarzane w kolejności ustalonej w FormAnalyze Designer. Początkowo pola są ustawione w kolejności ich tworzenia. Kolejność pól można zmienić, w tym celu należy z menu Formularz wybrać polecenie Kolejność pól. W oknie formularza są zaznaczone wszystkie przetwarzane elementy formularza połączone w kolejności przetwarzania. Elementy pozycjonujące i pola rozpoznawania typu formularza, będą przetwarzane wcześniej. Kolejność pól na formularzu ma znaczenie jedynie dla pól rozpoznawanych.

Zmianę kolejności elementów na formularzu obrazuje poniższy przykład. Poniższy rysunek przedstawia sytuację przed dokonaniem zmian.

The diagram shows a life insurance application form titled "ATHENE LIFE" and "WNIOSEK O UBEZPIECZENIE NA ŻYCIE". The form includes fields for "Nr AGENTA", "Nr POLISY", "1. OSOBA UBEZPIECZANA" (with sub-fields: NAZWISKO, WIEK, DATA URODZENIA, ADRES ZAMIESZKANIA, NR TELEFONU, SERIA / NUMER DOW. OS., IMIĘ OJCA, stan cywilny, nazwisko panieńskie, miejsce urodzenia, MIEJSCE PRACY, zespół wykonywany, zakres obowiązków), and "2. UBEZPIECZAJĄCY". Blue arrows indicate the sequence of processing, starting from field 1 and moving through various sub-fields and sections. A barcode is present between the agent and policy numbers.

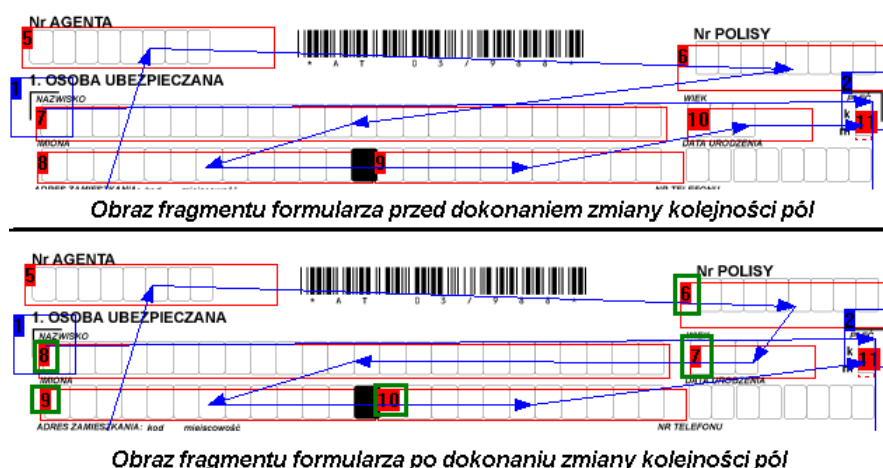
Rys. 7-1 Kolejność przetwarzania pól.

Aby zmienić kolejność elementów na formularzu, należy kliknąć na żądanym elemencie lewym przyciskiem myszki, a następnie przytrzymując go przeciągnąć przerywaną linię w odpowiednie pole. Rysunek 7 -1a przedstawia powyższą operację.



Rys. 7-1a Zmiana kolejności elementów

W zaprezentowanym przykładzie zmiana kolejności pól będzie polegała na umieszczeniu pola numer 10 za polem nr 6. Gdy po zakończonej operacji przeciągania zostanie zwolniony lewy przycisk myszki, dotychczasowe pole nr 10 zostanie opatrzone numerem 7. Rysunek nr 7 - 1b prezentuje nową kolejność rozpoznawania pól na formularzu, a także ukazuje zmiany kolejności pól, jakie zaszły w stosunku do sytuacji wyjściowej. Numery pól (zaznaczone na rysunku zielonymi ramkami), zmienione zostały na skutek wykonania operacji przeciągania.



Rys. 7-1b Efekt finalny zmiany kolejności pól

Zmiany dotyczą jedynie pól pierwotnie znajdujących się przed elementem nr 10. Tak, więc pole nr 6 pozostało bez zmian, pole nr 10 zostało zmienione na nr 7. Dotychczasowe pola nr 7, 8, 9 zostały opatrzone numerami 8, 9, 10. Kolejność pozostałych pól na formularzu nie zmieniła się.

Pod czas ustawiania kolejności pól do przetwarzania, należy zwrócić uwagę na pola, które częściowo lub całkowicie się pokrywają. W obydwu przypadkach, należy uzyskać całkowity dostęp do pola, np. przez przesunięcie jednego z pól w celu odsłonięcia jego obszaru. Tylko wtedy zmiana kolejności takich pól zostanie przeprowadzona prawidłowo.

Aby usunąć z formularza linie wskazujące kolejność pól, należy z menu Formularz wybrać polecenie Kolejność pól.

7.2. Typy i nazwy pól

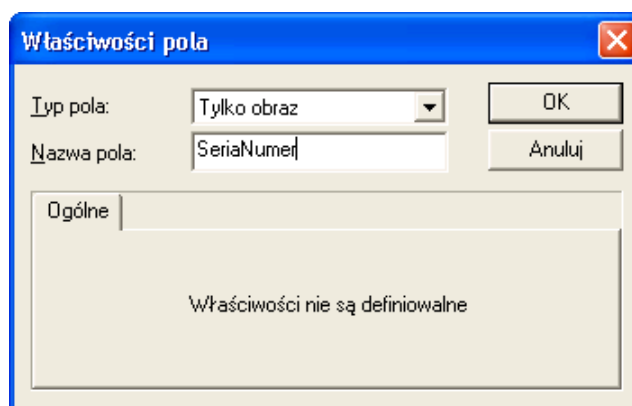
Elementy znajdujące się na formularzu muszą posiadać unikalne nazwy. Wyjątkiem pól nieposiadających nazw są elementy pozycjonujące oraz pola detekcji typu formularza i przetwarzania wstępnego. Wyżej pola nie przenoszą jakichkolwiek danych.

Nazwę pola należy wprowadzić w oknie dialogowym Właściwości pola. Domyślna nazwa jest kombinacją słowa Zone i kolejnego numeru pola rozpoznawanego.

Typ pola również należy ustawić w oknie dialogowym Właściwości pola, wybierając go z listy rozwijalnej Typ pola. W zależności od wybranego typu zmieniają się właściwości, które można dla danego pola zdefiniować. W momencie wyboru typu pola z rozwijalnej listy poszczególne obszary okna dialogowego Właściwości pola wypełniane są wartościami domyślnymi.

7.3. Pole Tylko obraz

Pole Tylko obraz jest wykorzystywane do obliczenia współrzędnych fragmentu obrazu na aktualnie przetwarzanym formularzu. Współrzędne te są wyliczane i wykorzystywane przez moduł FormAnalyzer Verifier. Pole Tylko obraz może zawierać dowolny fragment formularza, który następnie będzie wykorzystywany jedynie do przeglądania, bez możliwości jego edycji.



Rys. 7-2 Właściwości pola tylko obraz.

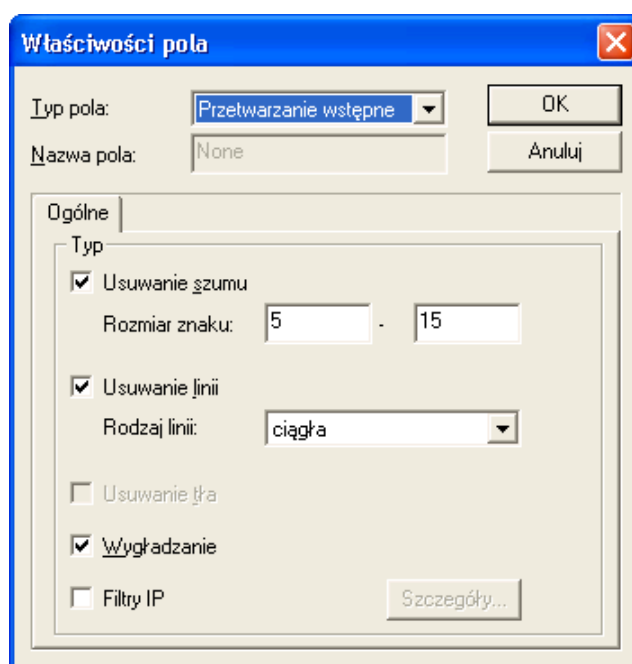
Po wybraniu typu pola Tylko obraz, należy w oknie dialogowym Właściwości pola zdefiniować jego nazwę. Pozostałe właściwości dla tego pola są niedostępne.

7.4. Pole Przetwarzanie wstępne

Pole przetwarzania wstępnego jest polem nie dającym żadnych wyników rozpoznania. Służy jedynie do modyfikowania obrazu strony dokumentu, wykonując wybrane operacje przetwarzania obrazów. Zmiana obrazu odbywa się jedynie w FormAnalyze Engine - obrazy stron zapisane w bazie danych nie ulegają jakimkolwiek zmianom. Pole to jest stosowane do poprawy jakości obrazu, a w konsekwencji do poprawienia jakości rozpoznawania.

Metody przetwarzania zadane w tym polu modyfikują obraz strony dokumentu. Pola rozpoznawane znajdujące się kolejno za polem przetwarzania wstępnego, będą korzystały z obrazu zmodyfikowanego.

Na rysunku poniżej przedstawiono okno dialogowe Właściwości pola przetwarzania wstępnego.



Rys. 7-3 Właściwości pola przetwarzania wstępnego.

W przypadku pola przetwarzania wstępnego, nie nadaje mu się nazwy. Przetwarzanie wstępne jest procesem składającym się z trzech operacji: Usuwanie szumu, Usuwanie linii, Wygładzanie. Projektując formularz można wybrać jedną lub więcej operacji, które następnie mają być wykonywane w ramach przetwarzania wstępnego. Kolejność wykonywania operacji jest analogiczna do kolejności zaprezentowanej w oknie właściwości pola, tzn. usuwanie szumu, usuwanie linii, wygładzanie i zaawansowane filtry IP.

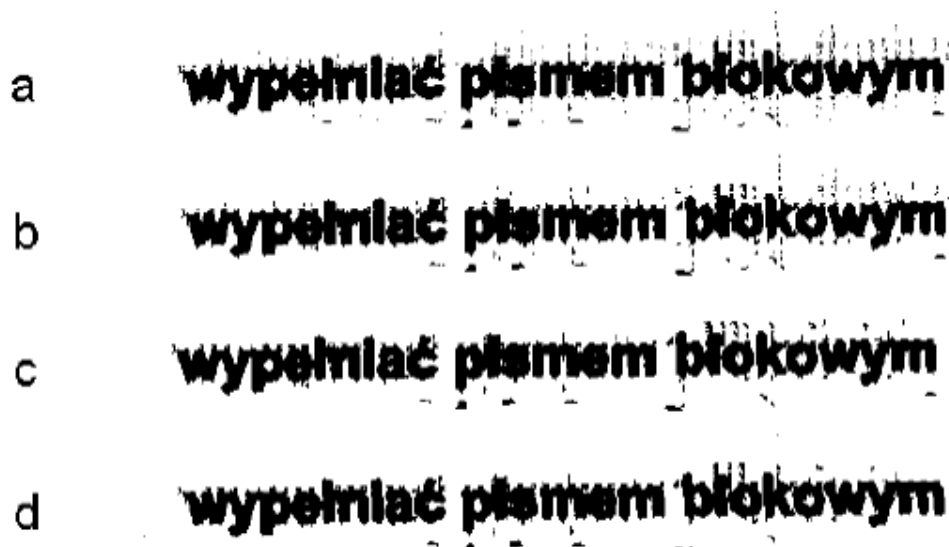
W kolejnych punktach omówiono poszczególne operacje przetwarzania wstępnego.

7.5. Usuwanie szumu

Ten rodzaj przetwarzania jest używany do usunięcia drobnych elementów obrazu (poniżej pewnego rozmiaru progowego), które mogą pojawić się na obrazach dokumentów. Należy zwrócić szczególną uwagę na właściwe dobranie rozmiaru progowego. Zmniejsza to niebezpieczeństwo usunięcia razem z szumem fragmentów znaków, które mają być rozpoznawane. Jego wartość podaje się pośrednio poprzez określenie rozmiaru znaków rozpoznawanych w polu Rozmiar w obszarze okna Parametry obrazu. W obydwie pola oddzielone znakiem myślnika należy wprowadzić minimalny i maksymalny rozmiar znaku podany w pikselach dla rozdzielczości 200 dpi. W przypadku operacji usuwania szumu istotniejszą wartością, jest minimalna wartość wprowadzona w pole Rozmiar. Elementy o rozmiarze liniowym mniejszym niż podana wartość minimalna, o ile spełniają dodatkowe, złożone ograniczenia dotyczące powierzchni i kształtu, są z obrazu usuwane. Wartością domyślną rozmiaru minimalnego jest 5 (pikseli przy rozdzielczości 200 dpi). Odpowiada to ok. 0,6 milimetra. Przy takiej wartości parametru z obrazu nie zostaną usunięte elementy, takie jak, np. znaki interpunkcyjne.

Jeżeli rozpoznawany formularz jest w rozdzielczości innej niż 200 dpi, to wartości parametru Rozmiar są przeliczane do odpowiedniej rozdzielczości.

Na rysunku poniżej pokazano przykłady usuwania szumu dla różnych ustawień minimalnej wartości Rozmiaru znaku. Rys. 7-5(a) przedstawia obraz oryginalny, (b) z usuwaniem szumu przy minimalnej wartości 1, (c) przy wartości 3 i (d) przy wartości 5.



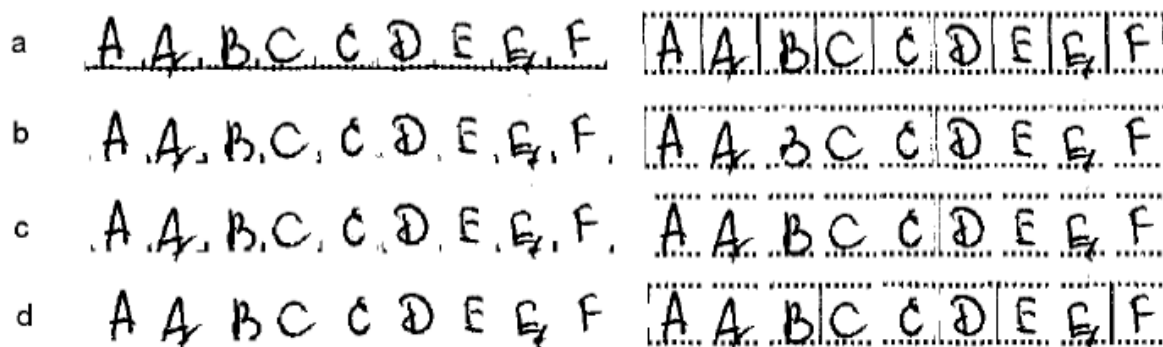
Rys. 7-4 Usuwanie szumu w zależności od wartości minimalnej Rozmiar znaku.

7.5.1.Usuwanie linii

Procedurę usuwania linii stosuje się w przypadku rozpoznawania danych wpisywanych w wydrukowany raster, tabelki itp. Operacja usuwania linii polega na usunięciu z obrazu linii, które mogłyby zostać potraktowane, jako fragmenty elementów przeznaczonych do rozpoznawania. Jeżeli, w przypadku, gdy linia ma styczność ze znakami, usunięcie jej nie może jednocześnie zmieniać obrazu znaków. Dla funkcji usuwania linii wybiera się z listy typ linii, która pojawia się na formularzu:

- *ciągła* - linia ciągła;
- *przerywana* - linia przerywana;
- *siatka* - linia ciągła z separatorami pisma blokowego.

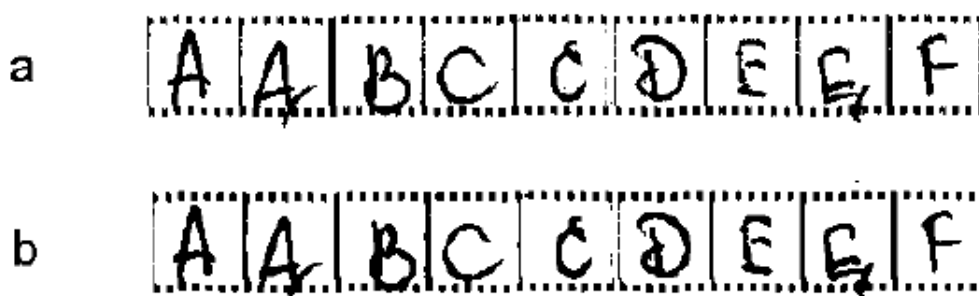
Na poniższym rysunku przedstawiono działanie funkcji usuwania linii z różnym parametrem typu linii. Podpunkt (a) na rysunku to obraz oryginalny, (b), (c) i (d) to obrazy usuwania linii odpowiednio z parametrem ciągła, przerywana i siatka.



Rys. 7-5 Usuwanie linii w zależności od ustawionego rodzaju linii.

7.5.2. Wygładzanie

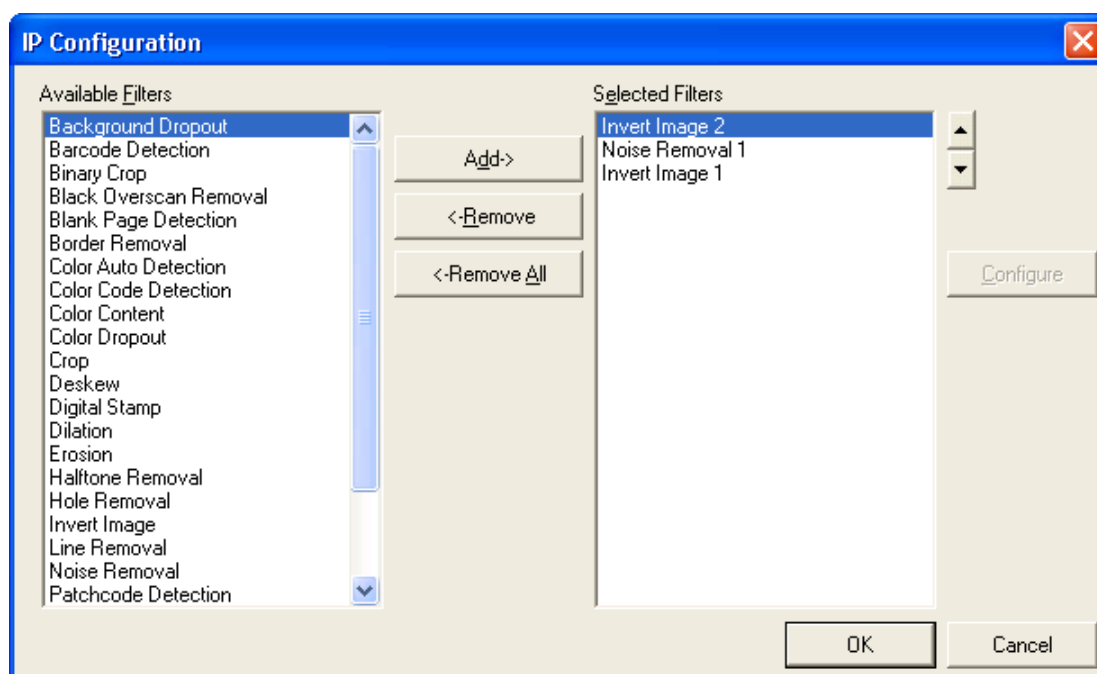
Operacja wygładzania polega na zastosowaniu do obszaru przetwarzania filtra o odpowiednio dobranej charakterystyce. Jego zadaniem jest zmniejszenie zakłóceń powstałych wskutek próbkowania i digitalizacji obrazu w skanerze. Wygładzanie daje szczególnie dobre efekty, gdy np. obraz znaków jest poszarpany czy nawet poprzerrywany. Ponieważ wygładzanie powoduje także pewne „rozmycie” nie należy go stosować, gdy znaki są bardzo blisko siebie. Może to dawać bardzo niekorzystny efekt zlepiania znaków. Wygładzanie najlepiej stosować dla pisma ręcznego o stosunkowo dużych literach. Na Rys. 7-6 pokazano przykład zastosowania wygładzania. Rys. 7-6(a) to oryginalny fragment obrazu, natomiast (b) - po wygładzaniu.



Rys. 7-6 Przykład wygładzania obrazu.

7.5.3. Filtry IP

Funkcja ta umożliwia przetworzenie wycinka obrazu przez zdefiniowaną listę filtrów. Aby skonfigurować ten proces należy zaznaczyć *Filtry IP*, a następnie wybrać polecenie *Szczegóły...*. Zostanie wyświetlone okno dialogowe przedstawione na Rys. 7-7.



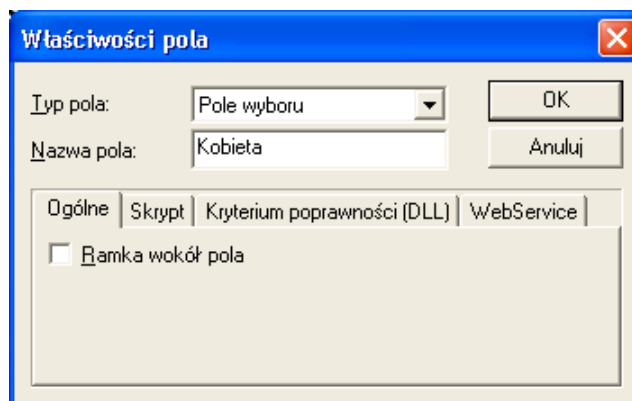
Rys. 7-7 Okno definiowania ustawień filtrów w FormAnalyzer Designer.

W prawej części okna w obszarze *Available Filters* znajdują się wszystkie filtry, których można użyć w procesie rozpoznawania dokumentów. Szczegółowy opis działania tych filtrów znajduje się w podręczniku FormAnalyzer Scan & Administrator.

W tym procesie należy wykorzystać filtry modyfikujące obraz operujące na obrazie czarno-białym, tj.: Background Dropout, Binary Crop, Black Overscan Removal, Border Removal, Crop, Deskew, Dilation, Erosion, Halftone Removal, Hole Removal, Invert Image, Line Removal, Noise Removal, Rotation, Scaling, Skeleton, Smoothing, Threshold.

7.6. Pole wyboru

Pole wyboru jest rzadko używane samodzielnie. Wartością tego pola jest zaznaczenie lub jego brak.



Rys. 7-8 Właściwości pola wyboru.

Dla pola wyboru można zdefiniować tylko jedną właściwość – Ramka wokół pola. Ramka otaczająca pole będzie uwzględniana w procesie. Opcja ta jest domyślnie zaznaczona.

Wyłączenie opcji Ramka wokół pola oznacza, że na zeskanowanym dokumencie pole wyboru nie jest otoczone ramką. Najczęściej taka sytuacja występuje w przypadku, gdy ramka jest wydrukowana w kolorze usuwanym przez skaner (drop-out color).

W przypadku pól wyboru z ramką, najlepszą jakość rozpoznawania uzyskuje się zaznaczając pole wyboru wewnątrz ramki.



Rys. 7-9 Zaznaczenie pola wyboru w ramce.

W zakładkach Skrypt i Kryterium poprawności (DLL) i Webservice można zdefiniować sposób sprawdzania poprawności pola. Dla pojedynczych pól wyboru te właściwości są rzadko wykorzystywane. Więcej informacji na temat skryptów weryfikujących i sprawdzania poprawności przy użyciu zewnętrznych bibliotek DLL podano w punktach 7.8.5 - Skrypty, 7.8.6 - Funkcje weryfikacji z zewnętrznych bibliotek DLL oraz 7.8.7 - Funkcje weryfikacji poprawności danych pola z wykorzystaniem Webservice.

7.7. Pole kod paskowy

Pole Kod paskowy jest przeznaczone do rozpoznawania kodów paskowych. Na rysunku poniżej pokazano okno dialogowe właściwości pola kodu paskowego. Pierwszą cechą, którą należy wybrać, jest klasyfikator. Dostępne są trzy klasyfikatory służące do rozpoznawania kodów paskowych:

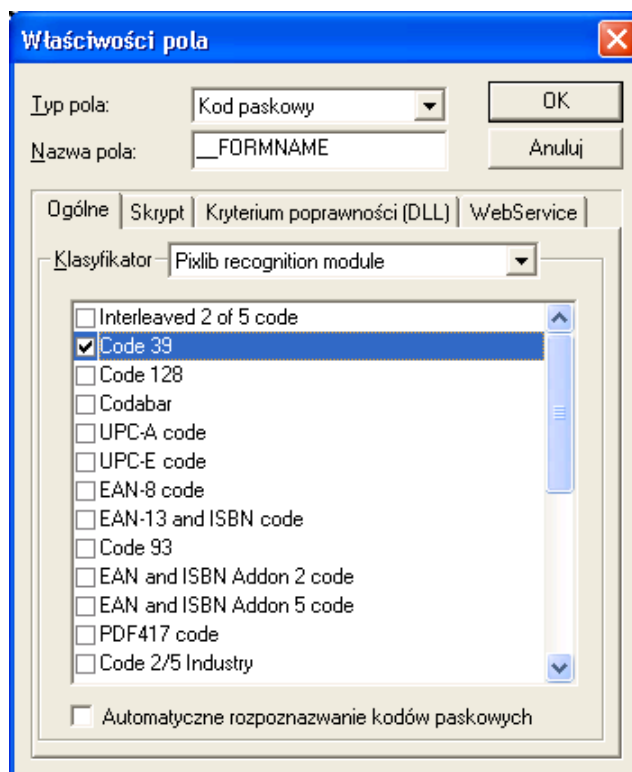
- *Pixlib recognition module* – wbudowany moduł do rozpoznawania kodów paskowych jedno i dwuwymiarowych (kody jedno wymiarowe nie wymagają dodatkowej licencji)
- *BAR recognition module* – zewnętrzny moduł do rozpoznawania kodów paskowych jednowymiarowych wymagający dodatkowej licencji (Recognita lub Nuance)
- *Amplib recognition module* – zewnętrzny moduł do rozpoznawania kodów paskowych dwuwymiarowych wymagający dodatkowej licencji (Nuance)

W jednym szablonie dokumentu można wykorzystywać różne klasyfikatory służące rozpoznawaniu kodów paskowych. Należy przestrzegać zasad zawartych w Tabeli 1.

Tabela 1 - Dostępność rozpoznawania kodów paskowych w systemie FormAnalyze.

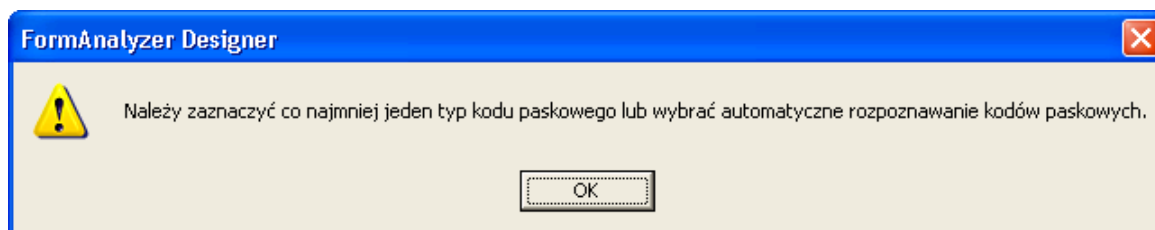
Typ kodu paskowego	Pixlib	BAR (Recognita)	BAR (Nuance)	Amplib
Interleaved 2 of 5 code	Tak	Tak	Tak	Tak (nie łączyć z Code2/5)
Code 39	Tak	Tak	Tak	Tak
Code 128	Tak	Tak	Tak (nie łączyć z Code128/EAN128)	Tak (nie łączyć z Code128/EAN128)
Codabar	Tak	Tak	Tak	Tak
UPC-A code	Tak	Tak	Tak (nie łączyć z EAN)	Tak (nie łączyć z EAN)
UPC-E code	Tak	Tak	Tak	Tak
EAN-8 code	Tak	Tak	Tak (nie łączyć z UPC-A)	Tak (nie łączyć z UPC-A)
EAN-13 and ISBN code	Tak	Tak	Tak (nie łączyć z UPC-A)	Tak (nie łączyć z UPC-A)
Code 93	Tak			Tak
EAN and ISBN Addon 2 code	Tak			
EAN and ISBN Addon 5 code	Tak			
PDF417 code	Tak		Tak (samodzielnie)	Tak (samodzielnie)
Code 2/5 Industry	Tak			Tak (nie łączyć z ITF)
Code 2/5 IATA	Tak			Tak (nie łączyć z ITF)
Code 2/5 Datalogic	Tak			Tak (nie łączyć z ITF)
Code 2/5 Inverted	Tak			Tak (nie łączyć z ITF)
Code BCD Matrix	Tak			
Code 2/5 Matrix	Tak			Tak (nie łączyć z ITF)
Code 32	Tak			Tak
Code 128 / EAN 128	Tak		Tak (nie łączyć z Code128)	Tak (nie łączyć z Code128)
Postnet code	Tak	Tak	Tak (samodzielnie)	Tak (samodzielnie)
Data Matrix	Tak			Tak (samodzielnie)
QR Code	Tak			
Royal Post	Tak			
Australian Post	Tak			
Intelligent Mail	Tak			

Dla pola tego typu można podać rodzaj kodu paskowego używanego na formularzu lub też pozostawić identyfikację rodzaju kodu procedurze rozpoznającej zaznaczając pole wyboru Automatyczne rozpoznawanie kodów paskowych.



Rys. 7-10 Właściwości pola Barcode.

W przypadku, nie zaznaczenia opcji Automatyczne rozpoznawanie kodów paskowych lub, gdy typ kodu nie zostanie wybrany z dostępnej w oknie dialogowym listy, przejście do następnej zakładki albo zapamiętanie wprowadzonych zmian spowoduje wyświetlenie okna komunikatu.



Rys. 7-11 Komunikat o konieczności ustawienia typu kodu paskowego.

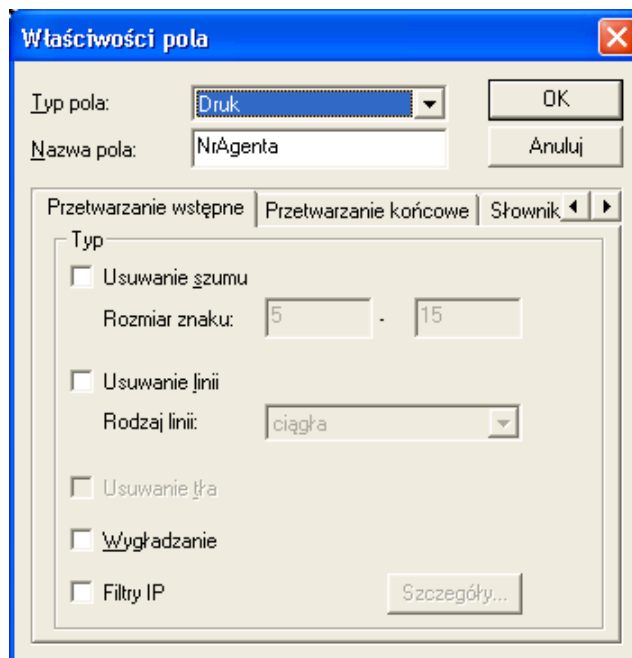
W systemie FormAnalyze dostępne są rodzaje kodów paskowych pozwalające na zapisanie cyfr (i w niektórych przypadkach znaków specjalnych) oraz informacji alfanumerycznych.

7.8. Pole Druk

Pole Druk jest polem zawierającym tekst drukowany. Proces przetwarzania, rozpoznawania oraz weryfikacji pola drukowanego został podzielony na cztery etapy. Ten podział jest odzwierciedlony w oknie właściwości pola, w którym właściwości zostały umieszczone na czterech zakładkach.

7.8.1. Przetwarzanie wstępne

W pierwszej kolejności wykonywane zostaje przetwarzanie wstępne obrazu pola. Na poniższym rysunku przedstawiono zakładkę Przetwarzanie wstępne. Parametry przetwarzania wstępnego są takie same, jak w przypadku pola Przetwarzania wstępnego, szczegółowo omówionego w punkcie 7.4.



Rys. 7-12 Zakładka Przetwarzanie wstępne pola Druk.

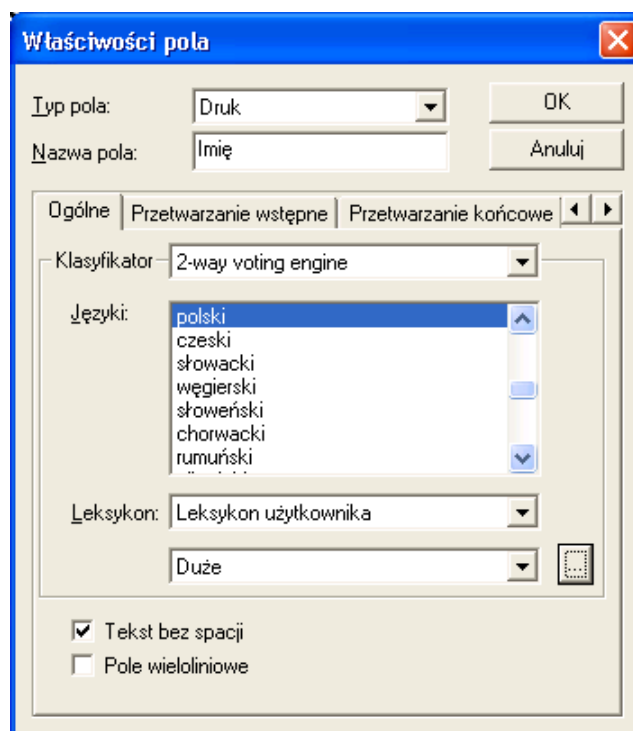
Przetwarzanie wstępne pola drukowanego modyfikuje kopię obrazu pola, która później będzie rozpoznawana. W tym przypadku obraz dokumentu nie jest modyfikowany.

7.8.2. Rozpoznawanie treści pola

Kolejnym krokiem jest rozpoznawanie treści pola. W zakładce Ogólne, okna dialogowego Właściwości pola, można zdefiniować szereg parametrów rozpoznawania. Właściwy dobór tych parametrów ma duży wpływ na jakość rozpoznawania pola. Pierwszą cechą, którą należy wybrać, jest klasyfikator. Dostępne są następujące klasyfikatory służące do rozpoznawania pisma drukowanego i ręcznego:

- *M/TEXT omnifont recognition module* – zewnętrzny moduł do rozpoznawania pisma drukowanego w formie pełno tekstowej. Moduł wspiera jedynie podzbiór dostępnych w liście języków (angielski, francuski, niemiecki). Wymaga dodatkowej licencji Nuance.
- *Multi-lingual omnifont recognition module* – zewnętrzny moduł do rozpoznawania pisma drukowanego w formie pełno tekstowej. Moduł wspiera wszystkie dostępne w liście języki. Wymaga dodatkowej licencji Recognita lub Nuance.
- *2-way voting engine* – zewnętrzny moduł do rozpoznawania pisma drukowanego w formie pełno tekstowej. Wynik rozpoznawania jest efektem przetwarzania z wykorzystaniem silników: *M/TEXT omnifont recognition module* i *Multi-lingual omnifont recognition module*. Moduł wspiera wszystkie dostępne w liście języki. Wymaga dodatkowej licencji Nuance.

- *FRX engine* – zewnętrzny moduł do rozpoznawania pisma drukowanego w formie pełno tekstowej. Moduł wspiera wszystkie dostępnych w liście języki. Wymaga dodatkowej licencji Nuance.
- *3-way voting engine* – zewnętrzny moduł do rozpoznawania pisma drukowanego w formie pełno tekstowej. Wynik rozpoznawania jest efektem przetwarzania z wykorzystaniem silników: *M/TEXT omnifont recognition module*, *Multi-lingual omnifont recognition module* oraz *FRX engine*. Moduł wspiera wszystkie dostępnych w liście języki. Wymaga dodatkowej licencji Nuance.
- *9-pin draft dot-matrix recognition module* – zewnętrzny moduł do rozpoznawania pisma drukowanego stworzonego za pomocą drukarki igłowej w formie pełno tekstowej. Moduł wspiera wszystkie dostępnych w liście języki. Wymaga dodatkowej licencji Recognita lub Nuance.
- *Handprinted numeral recognition module* – zewnętrzny moduł do rozpoznawania pisma ręcznego zawierającego jedynie cyfry. Wymaga dodatkowej licencji Nuance.
- *reRecognition Handprint recognition module* – zewnętrzny moduł do rozpoznawania pisma ręcznego zawierającego dowolne znaki alfanumeryczne. Moduł wspiera wszystkie dostępnych w liście języki. Wymaga dodatkowej licencji Nuance.



Rys. 7-13 Zakładka Ogólne okna właściwości pola Druk.

Pierwsza dostępna lista Języki w obszarze Klasyfikator zakładki Ogólne służy do zdefiniowania języka, którym będą wypełniane pola. Na liście tej można jednocześnie wskazać kilka języków, klikając na każdym kolejnym lewym przyciskiem myszy. Kliknięcie na pozycji wcześniej już zaznaczonej spowoduje anulowanie jej zaznaczenia. W przypadku formularzy spotyka się najczęściej pola korzystające z jednego języka.

W następnym obszarze zakładki Leksykony należy zdefiniować, jakich znaków można spodziewać się w polu.

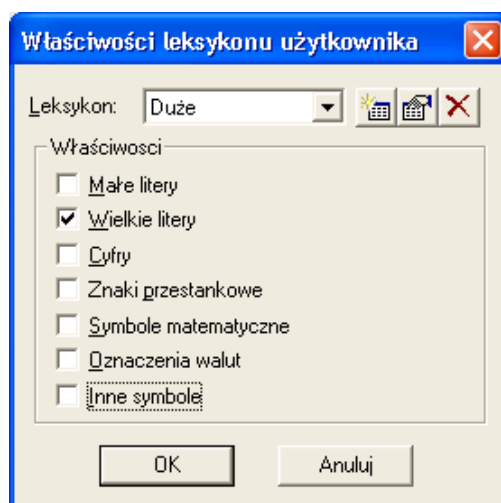
Ograniczenie zbioru znaków, które pojawiają się w polu, zwiększa jakość rozpoznawania, np. cyfra .1, gdy nie ma ograniczeń jest bardzo podobna do znaków „I”, „l”, „/”, „|” „i”, i „7”. W przypadku korzystania z leksykonu Cyfry dostępny będzie zbiór znaków podobnych zawierający „1” i „7”. Predefiniowane zbiory znaków to:

- *Litery* - w tym zbiorze znajdują się małe i duże litery
- *Cyfry*
- *Data* - cyfry i znaki pozwalające na zapisanie daty
- *Godzina* - cyfry i znaki pozwalające na zapisanie czasu
- *Kwota* - cyfry i znaki pozwalające na zapisanie kwoty
- *Numer telefonu* - cyfry i znaki pozwalające na zapisanie numeru telefonicznego.

Jeżeli w polu może wystąpić kombinacja znaków, której nie odpowiada żaden z leksykonów predefiniowanych, możemy zdefiniować leksykon użytkownika.

W tym celu należy wybrać z rozwijalnej listy Leksykony pozycję Leksykon użytkownika. Jeżeli w bieżącym pliku konfiguracji zostały uprzednio zdefiniowane leksykony użytkownika, to można je wybrać z listy.

Gdy leksykonów nie ma lub będzie definiowany nowy leksykon, należy kliknąć przycisk  Spowoduje to wyświetlenie okna dialogowego Właściwości leksykonu użytkownika

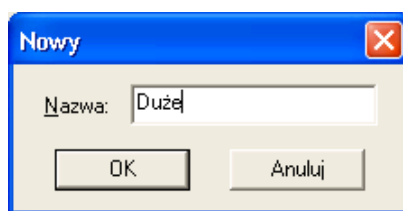


Rys. 7-14 Okno właściwości leksykonu użytkownika.

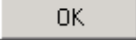
Korzystając z listy Właściwości, należy zaznaczyć odpowiedni leksykon. Właściwości leksykonu mogą być dowolnie modyfikowane. W celu zapisania wprowadzonych zmian, należy po zakończonym procesie definiowania właściwości kliknąć przycisk .

Leksykony są identyfikowane po nazwie, w której rozróżniane są małe i duże litery. Definiowanie właściwości nowego leksykonu, należy rozpocząć od nadania nowej nazwy.

W tym celu w oknie Właściwości leksykonu użytkownika, należy kliknąć przycisk Nowy . Jeżeli w pliku konfiguracji nie zdefiniowane zostały dotychczas żadne leksykony użytkownika funkcja Nowy będzie jedyną aktywną opcją. W oknie dialogowym Nowy należy wprowadzić nazwę leksykonu.



Rys. 7-15 Okno nazwy nowego leksykonu.

Kliknięcie przycisku  spowoduje utworzenie nowego leksykonu zawierającego wszystkie grupy znaków. W rozwijalnej liście znajdującej się po prawej stronie przycisków funkcyjnych okna, należy wskazać odpowiednie zestawy znaków, zaznaczając jedną z dostępnych pozycji na liście.

Kliknięcie przycisku Zmień nazwę  umożliwia zmianę nazwy leksykonu użytkownika, natomiast kliknięcie przycisku Usuń  spowoduje usunięcie leksykonu.

W zakładce Ogóle okna dialogowego Właściwości leksykonu użytkownika dostępne są jeszcze następujące opcje:

- *Tekst bez spacji* - zaznaczenie tej opcji powoduje usunięcie z rozpoznanego tekstu spacji. Jest to bardzo przydatne w przypadku pól, które mogą zostać wypełnione drukiem rozstrzelonym, a zawierają tylko jedno słowo.
- *Pole wieloliniowe* - zaznaczenie tej opcji oznacza, że w polu należy spodziewać się więcej niż jednej linii. Dla takiego pola najpierw wykonywany jest podział na linie tekstu, po czym linie są kolejno rozpoznawane.

7.8.3. Przetwarzanie końcowe

Kolejnym krokiem, jest przetwarzanie końcowe. W procesie tym sprawdza się czy wynik rozpoznania jest zgodny z wymaganą postacią (formatem).

Na Rys. 7-16 przedstawiono zakładkę *Przetwarzanie końcowe* okna dialogowego Właściwości pola dla pól drukowanych.

Rys. 7-16 Zakładka Przetwarzanie końcowe okna właściwości pola Druk.

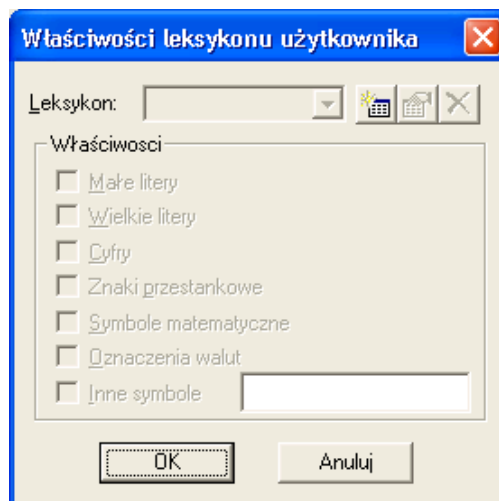
W obszarze Format, należy określić format danych, któremu powinna odpowiadać treść rozpoznana na dokumencie. Z wyjątkiem opcji Liczba, która ma znaczenie dla skryptów i funkcji DLL modyfikujących wartość pola, ustawienia parametrów przetwarzania końcowego nie wpływają na działanie FormAnalyze Engine. Są one brane pod uwagę dopiero w czasie weryfikacji (w module FormAnalyze Verifier) do sprawdzenia poprawności pola. W systemie FormAnalyze jest dostępnych pięć postaci danych wyjściowych. Dla niektórych z nich należy dodatkowo sprecyzować ich format.

Tekst

Umożliwia utworzenie leksykonu grupy znaków, które później będą dostępne w procesie Weryfikacji dla użytkownika o uprawnieniach Verifier.

W celu zdefiniowania grupy znaków dostępnych w procesie weryfikacji, należy kliknąć przycisk przeglądania [...] znajdujący się po prawej stronie rozwijalnej listy obok formatu Tekst.

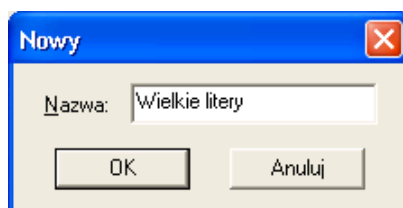
Wyświetlone zostanie okno Właściwości leksykonu użytkownika.



Rys. 7-17 Niewypełnione okno Właściwości leksykonu użytkownika.

W oknie tym, posługując się przyciskami znajdującymi się po prawej stronie pola *Leksykon*, można:

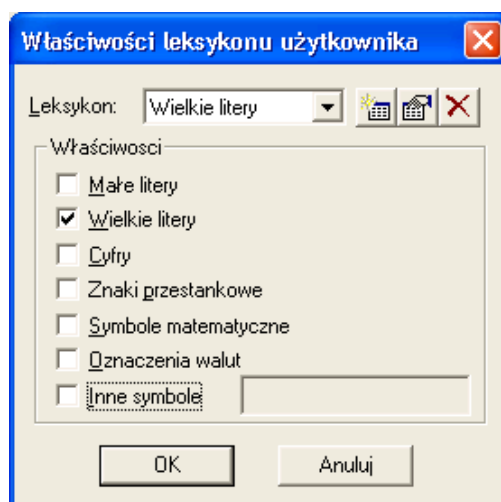
- klikając przycisk Nowy , utworzyć nowy leksykon użytkownika. Kliknięcie tego przycisku spowoduje wyświetlenie okna Nowy:



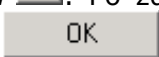
Rys. 7-18 Okno Nowy

W oknie tym w pole *Nazwa* należy wprowadzić nazwę nowotworzonego leksykonu, a następnie kliknąć przycisk , umożliwi to zdefiniowanie grup znaków dla nowego leksykonu użytkownika. Kliknięcie przycisku  spowoduje zakończenie procedury tworzenia leksykonu użytkownika i powrót do okna dialogowego *Właściwości leksykonu użytkownika*.

Nazwa nowoutworzonego leksykonu użytkownika zostanie wyświetlona w rozwijalnej liście *Leksykon* w oknie dialogowym *Właściwości leksykonu użytkownika*. Jednocześnie uaktywnione zostaną wszystkie pola wyboru w obszarze *Właściwości*. Użytkownik może zaznaczając lub odznaczając odpowiednie pole wyboru zdefiniować grupy znaków, które będą dostępne podczas weryfikacji. Użytkownik ma do wyboru następujące grupy znaków: *Małe litery*, *Wielkie litery*, *Cyfry*, *Znaki przestankowe*, *Symbole matematyczne*, *Oznaczenia walut* oraz *Inne symbole* (zaznaczenie tego pola wyboru umożliwi wpisanie dowolnych znaków, które następnie będą dostępne podczas weryfikacji). Poniżej zaprezentowane zostało przykładowe okno *Właściwości leksykonu użytkownika* ze zdefiniowanym przykładowym leksykonem *Wielkie litery*:



Rys. 7-19 Wypełnione okno Właściwości leksykonu użytkownika.

- W rozwijalnej liście *Leksykon* znajdować się będą wszystkie leksykony zdefiniowane w całym pliku konfiguracyjnym. Wybierając dany leksykon z dostępnej listy może edytować grupy uprzednio zdefiniowanych znaków, zmienić nazwę takiego leksykonu, klikając przycisk *Zmień nazwę*  lub usunąć leksykon klikając przycisk *Usuń* . Po zdefiniowaniu wszystkich parametrów, należy kliknąć przycisk  w oknie *Właściwości leksykonów użytkownika*. Spowoduje to powrót do okna *Właściwości pola*. W rozwijalnej liście obok wartości *Tekst* zostanie wyświetlona nazwa zdefiniowanego leksykonu.

Data

Jest stosowana dla pola zawierającego datę. Należy tu dodatkowo określić wejściowy format zapisu daty, w którym powinna być ona zapisana na dokumencie. Format daty należy wybrać z rozwijalnej listy Format wprowadzania daty, dostępnej tylko dla tej opcji.

Kwota

Pole rozpoznawane zawiera kwotę, czyli (opcjonalny) symbol waluty i samą kwotę.

Liczba

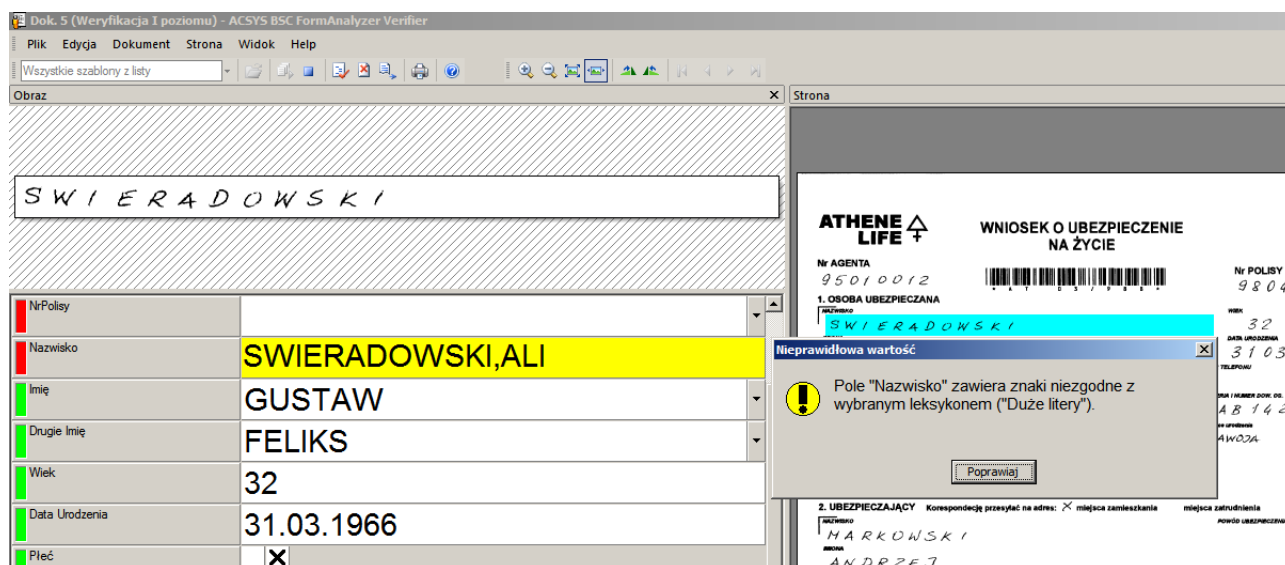
Rozpoznawane pole zawiera liczbę.

Warunkiem korzystania ze skryptów i funkcji DLL, których argumentem jest liczba całkowita, to pole, którego wartość będzie tym argumentem musi mieć zaznaczoną powyższą opcję.

PESEL

W rozpoznawanym polu jest wpisany ciąg cyfr stanowiący numer Pesel.

W aplikacji FormAnalyzer Veryfier, w przypadku próby zaakceptowania pola, z przypisanym leksykonem, gdy wprowadzono w nie znaki nie określone w leksykonie, system wyświetli odpowiedni komunikat, zaprezentowany na rysunku poniżej:



Rys. 7-20 Okno komunikatu po wprowadzeniu znaku niezgodnego z leksykonem.

W obszarze *Maska pola* można ściśle zdefiniować format pola, na którym będzie pracował Weryfikator.

Maska umożliwia wskazanie, jakie konkretne znaki będą mogły występować w danym polu. Definiuje także dokładną ilość znaków w danym polu (długość takiego pola, nie będzie mogła przekroczyć ilości znaków zdefiniowanych we wzorcu) oraz możliwość występowania odpowiednich separatorów (np.: w formatach mogą występować znaki: - / . \)

Zaznaczenie pola wyboru *Zawsze dostosuj wartość pola do maski* oznacza, że w przypadku, gdy w trakcie ustawiania wartości do weryfikacji odnalezione zostaną inne znaki niż zdefiniowane we wzorcu, to zostaną one pominięte. Jeżeli pole wyboru pozostanie niezaznaczone, wszystkie wartości zostaną przesłane do weryfikacji.

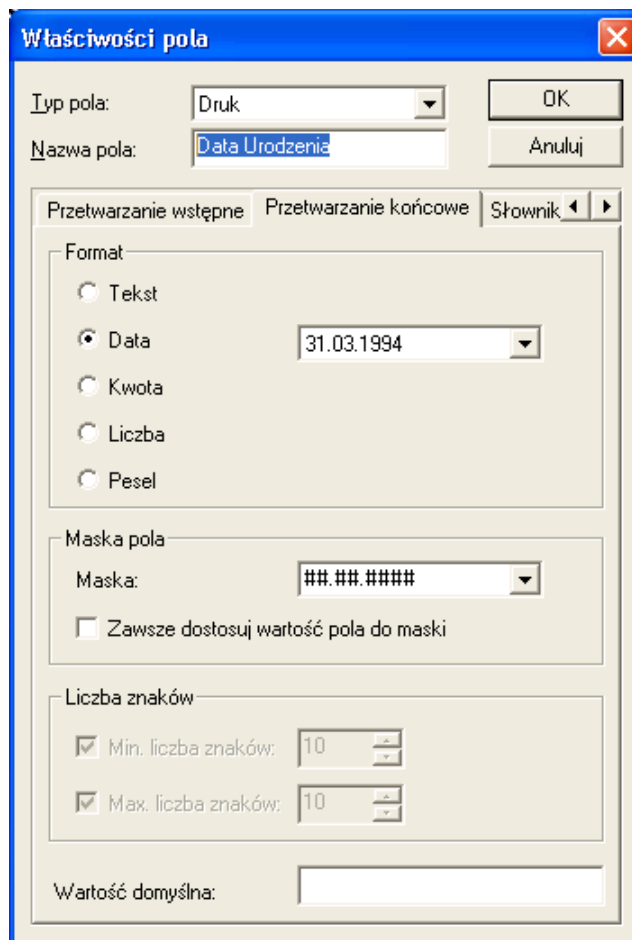
Do sporządzania masek można wykorzystywać następujące kombinacje znaków:

- # - cyfra (np. format daty z odpowiednim separatorem - ##.##.#### [przykładowa akceptowalna wartość: 09.09.2009] lub format numeru NIP odpowiednim separatorem - ###-###-##-## [przykładowa akceptowalna wartość: 521-009-90-98], użycie # uniemożliwi wprowadzanie żadnego innego znaku poza cyfrą);
- A – litery i cyfry (użycie znaku „A” uniemożliwi wprowadzenie żadnego innego znaku poza literą i/lub cyfrą);
- ? – tylko litery (użycie ? uniemożliwi wprowadzenie innych znaków poza dużymi, bądź małymi literami);
- > - duże litery (przy użyciu > i # można np. ustawić maskę dla numeru dowodu osobistego: >>> ##### -[przykładowa akceptowalna wartość: ACC 324567], użycie > uniemożliwi wprowadzenie żadnego innego znaku poza dużymi literami);
- < - małe litery (użycie < uniemożliwi wprowadzenie żadnego innego znaku poza małymi literami);
- & - znak ANSI;
- \ - znak, który należy użyć poprzedzając wystąpienie jednego ze znaków specjalnych wymienionych powyżej (np. jeżeli w polu dowód osobisty

występować będą dokumentów, których serie zaczynać się będą na literę A, taką maskę należałoby zapisać w sposób następujący \A>> ##### [przykładowa akceptowalna wartość: ADF 234567].

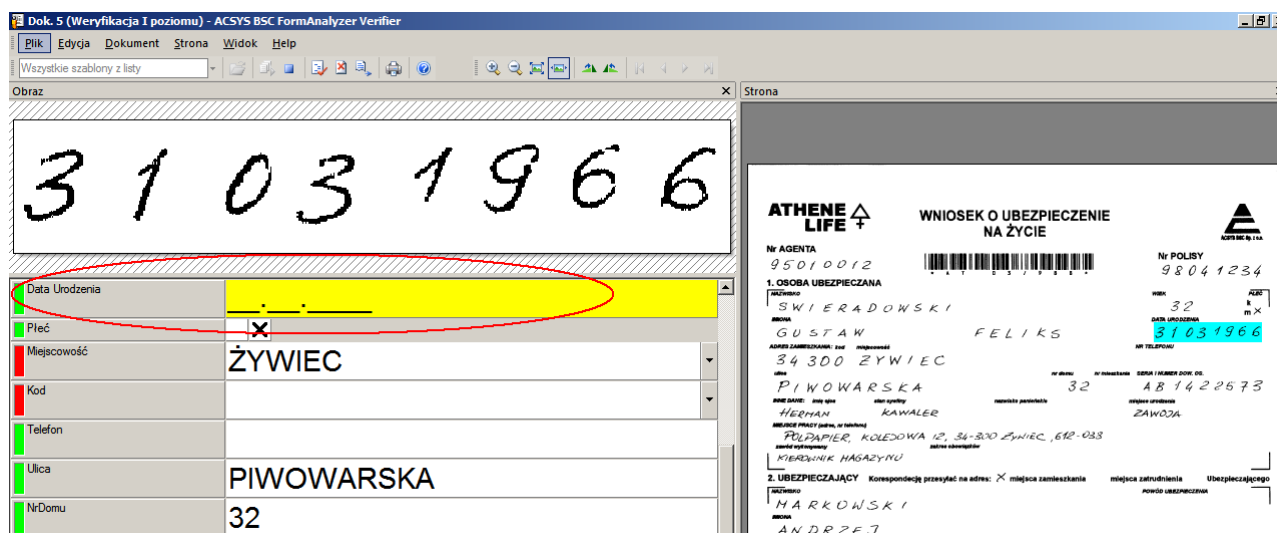
Zastosowanie znaków specjalnych użytych do definicji maski dotyczącej daty zostało zaprezentowane na poniższym przykładzie:

Pierwszy obrazek prezentuje ustawienia maski zdefiniowane w oknie dialogowym *Właściwości pola*, w zakładce *Przetwarzanie końcowe* dla pola Data urodzenia:



Rys. 7-21 Przykładowa maska pola Data urodzenia.

Obrazek drugi prezentuje efekt zastosowania maski dla pola Data urodzenia w aplikacji FormAnalyzer Verifier:



Rys. 7-22 Efekt zastosowania maski w aplikacji FA Verifier.

Program uniemożliwi wprowadzenie innych znaków niż cyfry, dodatkowo pole jest ograniczone do konkretnej liczby znaków oddzielonych separatorami zdefiniowanymi w masce.

W obszarze *Liczba znaków* można zdefiniować liczbę znaków, które będą mogły występować w danym polu. W celu zdefiniowania odpowiednich wartości można zaznaczyć jedno i/lub drugie pole wyboru. Zaznaczenie pola *Min. liczba znaków* spowoduje, że wartości w polu nie będą mogły mieć mniej niż wskazana liczba znaków. Zaznaczenie pola *Max. liczba znaków* spowoduje, że wartości w polu nie będą mogły mieć więcej niż wskazana liczba znaków. Zaznaczenie obydwu pól wyboru spowoduje nałożenie warunku na ilość maksymalną i minimalną.

W poniższym przykładzie zaprezentowano fragment okna dialogowego, w którym zdefiniowano ustawienia dla przykładowego pola Nazwisko: minimalna liczba znaków 3, maksymalna – 32. Na drugim obrazku zaprezentowano okno komunikatu w przypadku próby zaakceptowania pola, w które wprowadzono mniejszą ilość znaków niż uprzednio zostało określone w grupie *Liczba znaków* w oknie dialogowym *Właściwości pola*, w zakładce *Przetwarzanie końcowe*.

Jeżeli w trakcie weryfikacji dokumentu, na etapie akceptacji pola zostanie wykryta nieprawidłowa liczba znaków, nie zgadzająca się ze zdefiniowanymi wartościami, zostanie wyświetlone odpowiednie okno komunikatu.

W polu Wartość domyślna jest wpisywana wartość, która zostanie przypisana polu, w przypadku, gdy pole nie zostało wypełnione.

7.8.4. Słowniki wartości

W przypadku rozpoznawania treści formularzy dla wielu pól zestaw wartości może być ograniczony do pewnego zbioru. Zbiór dopuszczalnych wartości pola jest w systemie FormAnalyzer definiowany w postaci słownika.

Słowniki wartości pól są zapamiętane w bazie danych. Ich definiowanie i modyfikacja jest możliwa przy użyciu narzędzi Sybase SQL Anywhere.

W module FormAnalyzer Designer pole rozpoznawane wiązane jest z odpowiednim słownikiem. Trzeba pamiętać, że w trakcie weryfikacji i/lub rozpoznawania będzie

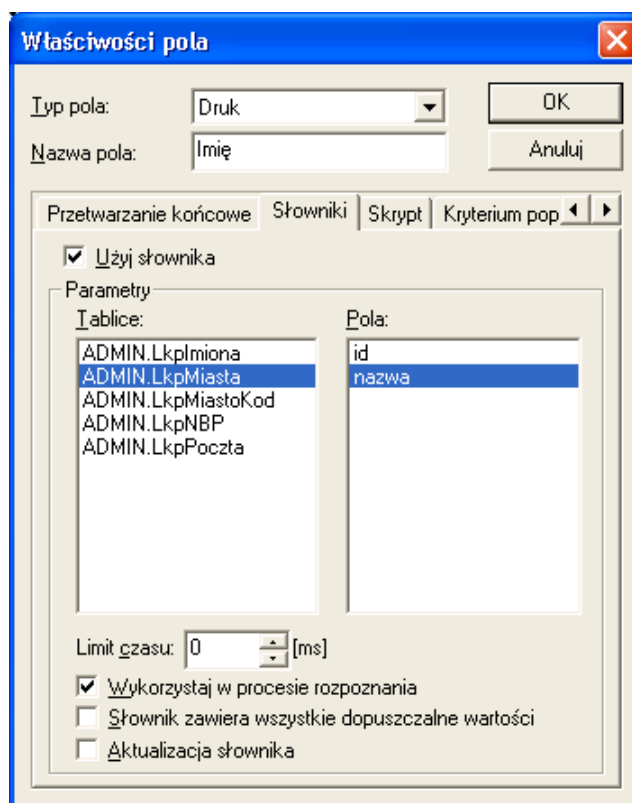
wykorzystywany aktualny zbiór wartości słownika w bazie danych. Bardzo istotna jest także inna konsekwencja umieszczenia słowników w bazie danych: przy weryfikacji i rozpoznawaniu dokumentu można korzystać ze słowników znajdujących się w tej samej bazie danych, co dokument. Zdefiniowanie dla jednego dokumentu słowników z różnych baz danych spowoduje błąd rozpoznawania i/albo weryfikacji.

Słowniki mogą być wykorzystane w czasie weryfikacji a także rozpoznawania dokumentu. Gdy pole rozpoznawane jest powiązane ze słownikiem będzie on użyty przy weryfikacji dokumentu. Aby stosować słownik w czasie rozpoznawania należy w zakładce Słowniki okna dialogowego Właściwości pola zaznaczyć opcję Wykorzystaj w procesie rozpoznawania.

Po rozpoznaniu wartości pola jest wykonywane sprawdzenie zgodności wyniku rozpoznania ze słownikiem. Słownik wartości jest wykorzystywany następująco:

- Sprawdza się, czy rozpoznana wartość pola jest w słowniku wartości. Jeżeli wartość znaleziono w słowniku przetwarzanie jest kończone.
- Gdy wartości pola nie ma w słowniku, wybierana jest z niego wartość najbliższa (najbardziej podobna) do wartości pola.
- Jeżeli odległość wartości pola od najbliższej wartości ze słownika jest mniejsza od pewnego określonego progu, wartość pola jest zmieniana na wartość ze słownika. W przypadku przeciwnym, wartość pola nie jest zmieniana, (czyli pole ma wartość, której nie ma w słowniku).

Wybór słownika związanego z polem umożliwia zaznaczenie opcji Użyj słownika. Po zaznaczeniu tej opcji staje się aktywny obszar zakładki Parametry, w którym na listach Tablice i Pola znajdują się nazwy pól, oraz wszystkie dostępne słowniki. Korzystając z obydwu list można przypisać dany słownik do pola, zaznaczając w obydwu listach po jednym z elementów. Należy pamiętać, że do jednego pola można przypisać tylko jeden słownik.



Rys. 7-23 Okno słowników bazy danych FormAnalyze.

W polu *Limit czasu* można podać maksymalny czas wyszukiwania wartości w słowniku. Jest to szczególnie istotne w przypadku bardzo dużych słowników, w których wyszukiwanie może trwać bardzo długo.

Zaznaczenie opcji *Słownik zawiera wszystkie dopuszczalne wartości*, spowoduje, że wartość pola jest poprawna tylko wtedy, gdy jest zdefiniowana w słowniku wartości. Ta opcja jest istotna jedynie przy weryfikacji dokumentu: pole zawierające wartość spoza słownika jest polem niepoprawnym. Opcja *Słownik zawiera wszystkie dopuszczalne wartości* nie zmienia sposobu wykorzystania słowników w FormAnalyze Engine. Jeżeli różnica pomiędzy wartością rozpoznaną a najbliższą wartością zdefiniowaną w słowniku jest zbyt duża, wartość pola nie jest zmieniana. W polu pozostaje wartość spoza słownika wartości.

Zaznaczenie opcji *Aktualizacja słownika* spowoduje, że w procesie weryfikacji akceptacja pola, którego wartość nie znajduje się w słowniku, automatycznie uaktualni słownik o nową wartość. Szczegółowy opis automatycznej aktualizacji słowników znajduje się w podręczniku FormAnalyze Verifier.

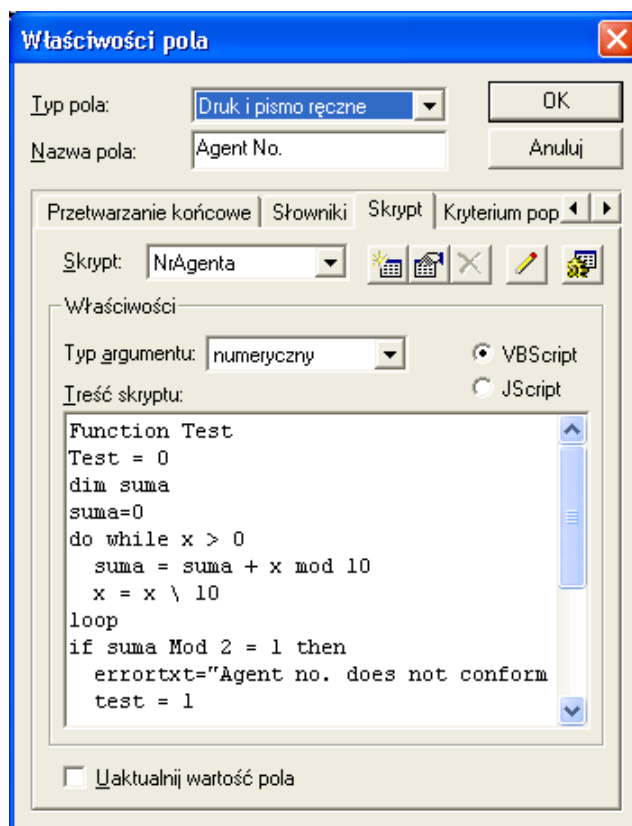
7.8.5. Skrypty

Skrypty definiowane dla pola mają dwojakie zastosowanie. Ich podstawowym zastosowaniem jest weryfikacja wartości pola. W tym przypadku wykonanie skryptu nie modyfikuje wartości pola – sprawdza, czy wartość pola spełnia kryteria poprawności zadane w skrypcie.

Po drugie, skrypty można wykorzystać do modyfikacji wartości pola. To rozwiązanie jest wykorzystywane do automatycznego poprawiania błędów rozpoznawania, które mogą zostać wykryte i poprawione automatycznie.

Niezależnie od tego, czy wartość pola jest modyfikowana, czy też nie, podstawową informacją wynikającą z wykonania skryptu jest to czy pole spełnia kryteria weryfikacji.

Silnik skryptów wykorzystywany w programie FormAnalyze obsługuje języki VBScript oraz JScript. Właściwości skryptu dostępne są w zakładce *Skrypt*.



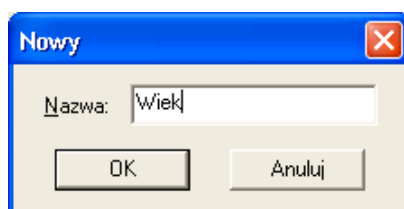
Rys. 7-24 Zakładka Skrypt pola Druk.

Skrypt wykonywany dla danego pola można wybrać z listy *Skrypt*. Na liście są umieszczone wszystkie skrypty zdefiniowane w aktualnym pliku konfiguracji. Dodatkowo jest na niej umieszczona pozycja pusta, pozwalająca na nie związanie z polem żadnego skryptu.

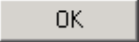
Po wybraniu skryptu z listy są wypełniane pola Właściwości skryptu: Typ argumentu i Treść skryptu. Jeżeli skrypt nie wymaga modyfikacji, na tym kończy się definiowanie skryptu dla pola.

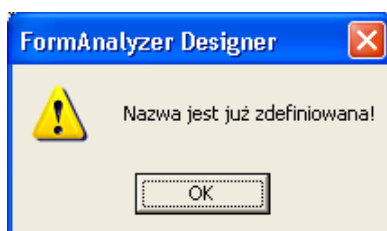
W przypadku modyfikowania skryptu (zmiany jego treści lub typu argumentu), należy pamiętać, że skrypt może być już wykorzystywany w innych polach. Wprowadzone zmiany mogą mieć wpływ na zmianę weryfikacji (i modyfikacji) innych pól na różnych formularzach. Gdy nie jest do końca jasne, z jakimi polami jest związany skrypt, który chcemy zmodyfikować, należy zdefiniować nowy skrypt.

W celu rozpoczęcia definiowania nowego skryptu, należy kliknąć przycisk Nowy , znajdujący się w zakładce Skrypt okna dialogowego Właściwości pola tuż obok listy rozwijalnej Skrypt. Na ekranie wyświetlone zostanie okno dialogowe Nowy, w którym w pole Nazwa, należy wprowadzić nową nazwę skryptu.



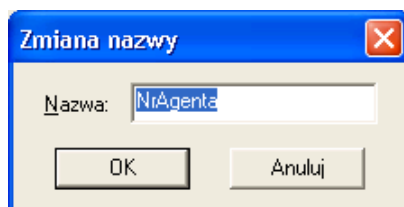
Rys. 7-25 Okno nazwy nowego skryptu.

Kliknięcie przycisku  spowoduje utworzenie nowego skryptu o typie argumentu *tekst* i pustej treści. (Jeżeli skrypt o podanej nazwie już istnieje, program wyświetli stosowne okno komunikatu [zaprezentowane na rysunku poniżej], a skrypt nie zostanie utworzony.) Skrypt o pustej treści zawsze, niezależnie od konkretnej wartości pola, dla której jest wykonywany, stwierdza poprawność tej wartości.



Rys. 7-26 Komunikat o istnieniu wybranej nazwy skryptu.

Przycisk *Zmień nazwę*  umożliwia zmianę nazwy skryptu. Wyświetlone zostanie okno dialogowe *Zmiana nazwy*, w którym umieszczona jest bieżąca nazwa skryptu. Można teraz dokonać jej edycji. Podobnie, jak przy tworzeniu nowego skryptu program sprawdza, istnienia nowo podanej nazwy. W przypadku występowania w bazie danych skryptu o wskazanej nazwie wyświetlone zostanie stosowne okno komunikatu, a użytkownik będzie musiał ponownie dokonać edycji nazwy skryptu w oknie dialogowym *Zmień nazwę*.



Rys. 7-27 Okno zmiany nazwy skryptu.

W celu zmiany treści skryptu możemy posłużyć się oknem *Treść skryptu* na zakładce *Skrypt* lub kliknąć przycisk . Kliknięcie przycisku  spowoduje wyświetlenie okna *Edytor Skryptu* z poziomu którego możemy dokonać zmiany treści skryptu, oraz przetestować poprawność jego działania.

7.8.6. Funkcje weryfikacji z zewnętrznych bibliotek DLL

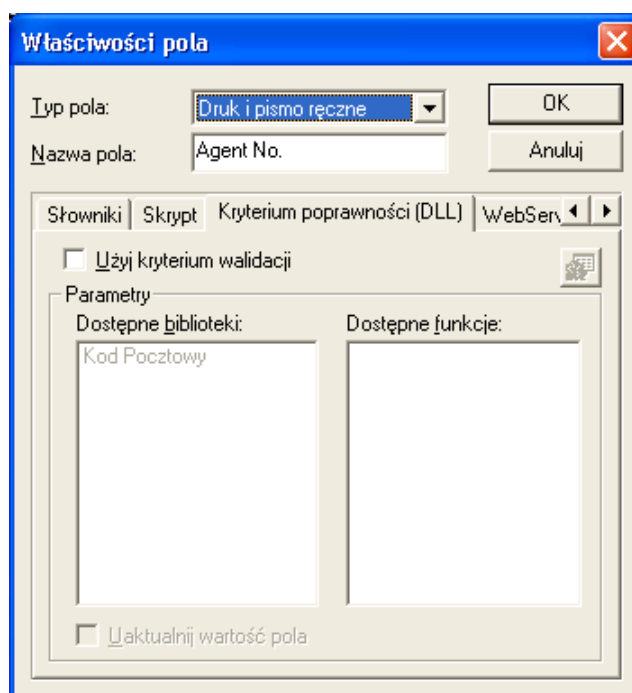
Inną metodą weryfikacji wartości pola jest skorzystanie z funkcji weryfikacji zdefiniowanej w zewnętrznej, dynamicznie ładowanej bibliotece (DLL). Podobnie jak w przypadku skryptów, funkcje weryfikacji mogą być wykorzystywane jedynie do sprawdzenia poprawności jak i równocześnie do modyfikacji wartości pola.

O ile definiowanie skryptów odbywa się w całości w FormAnalyze Designer, to funkcje weryfikacji są definiowane w czasie przygotowywania biblioteki DLL.

Tak jak w przypadku skryptów weryfikacji, podstawowym zadaniem funkcji weryfikacji jest ustalenie czy wartość pola spełnia kryteria poprawności. Dodatkowo, funkcje weryfikacji można wykorzystać do modyfikacji wartości pola. To rozwiązanie jest wykorzystywane do automatycznego poprawiania takich błędów rozpoznawania, które mogą zostać wykryte i poprawione automatycznie.

Ponieważ FormAnalyzer Designer korzysta z gotowych już funkcji weryfikacji, niektórych opcji nie można zdefiniować. Definiowanie tych opcji odbyło się na poziomie tworzenia biblioteki DLL. Dotyczy to w szczególności opcji *Uaktualnij wartość pola*. O tym czy funkcja będzie modyfikowała wartość pola decyduje odpowiednie jej zdefiniowanie. Podobnie jak w przypadku modyfikacji wartości pola przez skrypty, wartość pola zostanie zmieniona jedynie wtedy, gdy nowa wartość będzie spełniać kryteria poprawności.

Na rysunku poniżej przedstawiono zakładkę *Kryterium poprawności (DLL)* okna dialogowego *Właściwości pola* dla pól drukowanych.



Rys. 7-28 Zakładka Kryterium poprawności DLL pola Druk.

W obszarze Parametry na liście *Dostępne biblioteki* są wyświetlone nazwy wszystkich bibliotek DLL zawierających funkcje weryfikacji. FormAnalyzer Designer poszukuje tych bibliotek jedynie w folderze, w którym został uruchomiony. W celu skorzystania z funkcji znajdujących się w określonej bibliotece DLL, należy skopiować tę bibliotekę do foldera, w którym znajduje się program *Designer.exe*.

Aby skorzystać z funkcji weryfikacji należy najpierw zaznaczyć opcję *Użyj kryterium walidacji*. Zaznaczenie tej opcji spowoduje, że nazwy bibliotek w liście *Dostępne biblioteki* staną się aktywne. Na liście *Dostępne funkcje* są wyświetlane funkcje weryfikacji zdefiniowane dla wskazanej na poprzedniej liście biblioteki. Z funkcji tych można skorzystać podczas procesu weryfikacji pojedynczego pola. (Funkcje służące do weryfikacji pojedynczego pola mają tylko jeden parametr - pole, którego wartość ma być weryfikowana).

Funkcja, którą zostanie zaznaczona na liście będzie używana do weryfikacji wartości pola. W celu zrezygnowania ze stosowania funkcji weryfikacji, należy wyłączyć opcję *Użyj*

kryterium walidacji - program zapamiętuje wybraną bibliotekę i funkcję, jednak nie będą one dalej wykorzystywane.

7.8.7. Funkcje weryfikacji poprawności danych pola z wykorzystaniem WebService.

W systemie FormAnalyze istnieje możliwość sprawdzania poprawności danych rozpoznanych i wprowadzonych z wykorzystaniem mechanizmu WebService.

WebService jest komponentem programowym niezależnym od platformy i implementacji, dostarczającym określonej funkcjonalności. Na bazie WebService użytkownik może konstruować rozproszone aplikacje i systemy.

Wykorzystywanie WebService'ów pozwala współdziałać ze sobą komponentom programowym poprzez Internet, niezależnie od swojej lokalizacji i szczegółów implementacji.

Interfejsy usług WebService mogą być opisane na tyle szczegółowo, aby użytkownik mógł zbudować aplikację kliencką, komunikującą się z tymi usługami.

Usługi WebService są rejestrowane, co umożliwia użytkownikom w łatwy sposób odnaleźć potrzebne im komponenty.

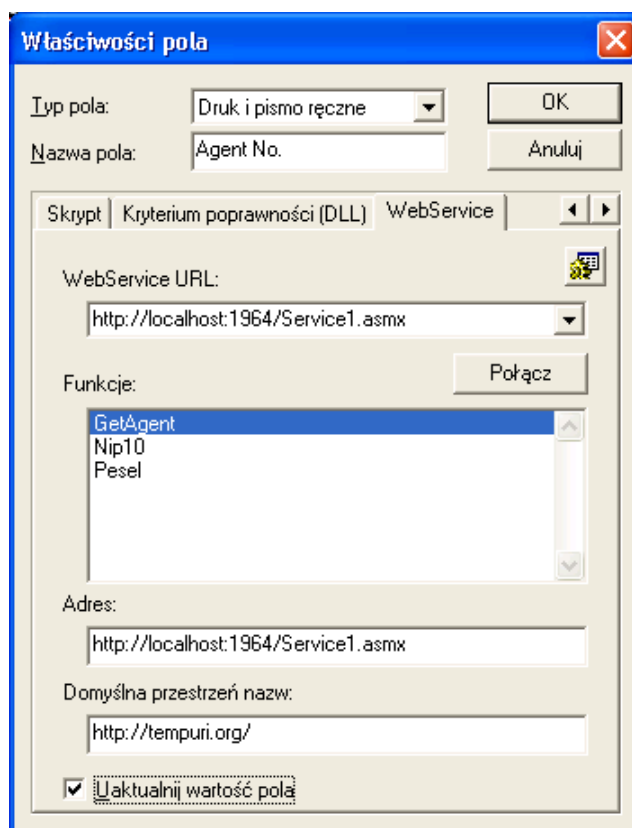
W przypadku systemu FormAnalyze korzystanie z WebService'ów umożliwia użytkownikowi utworzenie zestawu funkcji umożliwiających sprawdzanie poprawności lub modyfikację wartości pól dokumentu.

W celu skorzystania z istniejącego WebService'u, w oknie dialogowym Właściwości pola, należy przejść do zakładki WebService.

The screenshot shows a Windows-style dialog box titled "Właściwości pola" (Field Properties) with a red close button in the top right corner. The dialog is divided into several sections. At the top, there are two labels: "Typ pola:" (Field type) with a dropdown menu showing "Druk i pismo ręczne" (Print and handwriting), and "Nazwa pola:" (Field name) with a text box containing "Agent No.". To the right of these are "OK" and "Anuluj" (Cancel) buttons. Below this is a tabbed interface with three tabs: "Skrypt" (Script), "Kryterium poprawności (DLL)" (Validity criterion (DLL)), and "WebService" (which is selected). The "WebService" tab contains a "WebService URL:" label followed by a text box and a small icon. Below this is a "Funkcje:" (Functions) label next to a large, empty list box with a vertical scrollbar. To the right of the list box is a "Połącz" (Connect) button. Further down are two more text boxes labeled "Adres:" (Address) and "Domyślna przestrzeń nazw:" (Default namespace). At the bottom left, there is a checkbox labeled "Uaktualnij wartość pola" (Update field value).

Rys. 7-29 Zakładka WebService pola Druk i pismo ręczne

W pole *WebService URL* należy wprowadzić adres posiadanego WebService. Pole *WebService URL* zawiera listę wszystkich adresów wykorzystywanych w pliku konfiguracyjnym. Kliknięcie przycisku **Połącz** spowoduje połączenie z WebService w celu pobrania zdefiniowanych w nim funkcji.



Rys. 7-30 Funkcje zdefiniowane w Webservice, wyświetlone w zakładce WebService pola Druk i pismo ręczne

Dzięki usługom WebService użytkownik systemu FormAnalyzer:

- może w łatwy sposób uzyskać połączenie z zewnętrznymi klienckimi bazami danych (np.: jeżeli w polu dokumentu znajdują się dane dotyczące numeru agenta, poprzez usługę WebService użytkownik może połączyć się z zewnętrzną bazą danych i sprawdzić, czy taki numer zgadza się z imieniem i nazwiskiem agenta);
- posiada dostęp do funkcji zdefiniowanych w WebService, co zapewnia większe bezpieczeństwo dla danych, do których nie wszyscy użytkownicy powinni mieć dostęp, a są przechowywane w pliku konfiguracyjnym.
- tworzyć WebServisy w jednym z dostępnych języków: C# lub Java.

Funkcje WebService dla pojedynczego pola powinny być zdefiniowane w następujący sposób:

- powinny posiadać tylko jeden argument;
- powinny zwracać wartość typu ciąg znaków;
- funkcja, która jest wykorzystywana w FormAnalyzer Engine zwraca nową wartość wołanego pola;
- funkcja, która jest wykorzystywana w FormAnalyzer Verifier zwraca pusty ciąg, w przypadku gdy sprawdzanie zakończyło się prawidłowo) lub komunikat o błędzie;
- wartości typu chbox przekazywane są przez zmienną typu ciąg znaków (sekwencja 0 i 1).

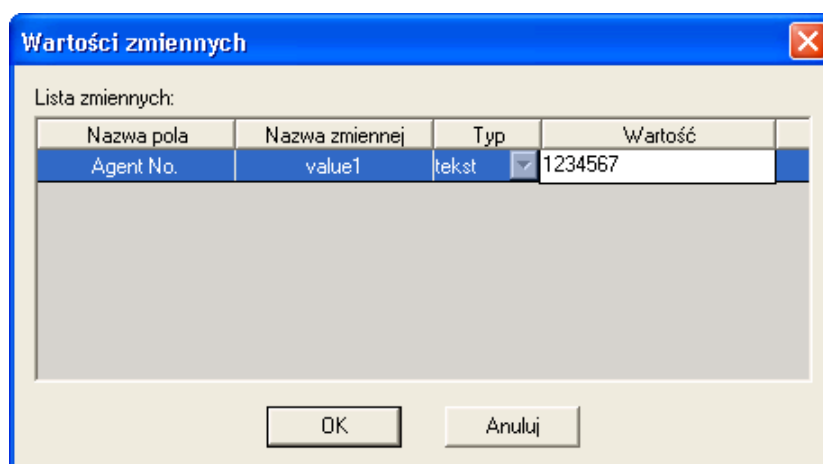
7.8.8. Uaktualnianie wartości pól i sprawdzanie poprawności funkcji weryfikacji.

W systemie FormAnalyze istnieje możliwość automatycznego uaktualniania pól przy pomocy utworzonych uprzednio kryteriów poprawności. Użytkownik może skorzystać z opcji *Uaktualnij wartość pola* w przypadku wszystkich trzech kryteriów poprawności opisanych kolejno w punktach 7.7.5, 7.7.6. i 7.7.7.

W momencie zaznaczenia opcji *Uaktualnij wartość pola* wartość pola może być zmodyfikowana przez jedno z kryteriów poprawności. Jest to możliwe tylko wtedy, gdy nowa wartość spełnia kryteria weryfikacji. Jeżeli nowa wartość pola nie spełnia tych kryteriów, w polu jest zachowywana poprzednia wartość.

Bardzo istotną czynnością jest sprawdzenie działania kryteriów poprawności. Ponieważ w liście dostępnych funkcji podane są jedynie nazwy funkcji, nie ma pewności, czy wybrana funkcja weryfikacji pojedynczego pola jest odpowiedniego typu. Żeby uniknąć błędów w czasie rozpoznawania należy koniecznie sprawdzić w procesie projektowania działanie funkcji weryfikacji.

W tym celu, należy kliknąć przycisk Test w zakładce Kryteria poprawności . Warunkiem przeprowadzenia testu są zaznaczone w obszarze zakładki Parametry jednej z dostępnych bibliotek i jednej z dostępnych dla nich funkcji. Na ekranie wyświetlone zostanie okno dialogowe Wartości zmiennych pokazane na poniższym rysunku.

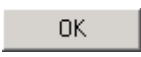


Nazwa pola	Nazwa zmiennej	Typ	Wartość
Agent No.	value1	tekst	1234567

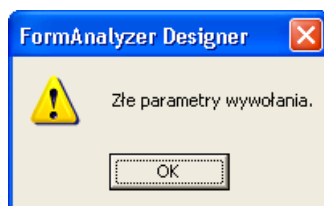
Rys. 7-31 Okno wartości parametrów wywołania funkcji weryfikacji.

W pierwszych trzech kolumnach są podane dane identyfikujące argumenty funkcji. Są to kolejno: nazwa pola (w dokumencie), nazwa zmiennej (używana dla identyfikacji wartości pola w funkcji) oraz typ zmiennej. Te wartości nie mogą być zmieniane. W kolumnie Wartość podajemy wartości pól (argumentów), dla których zostanie wykonana funkcja weryfikacji.

W przedstawionym na Rys. 7-31 Okno wartości parametrów wywołania funkcji weryfikacji, przykładzie wprowadzono wartość zmiennej. Po wpisaniu wartości należy zaznaczyć inny argument lub inną kolumnę; spowoduje to zakończenie edycji wartości.

Kliknięcie przycisku  spowoduje wywołanie funkcji wskazanej funkcji. Najpierw jest sprawdzana poprawność parametrów wywołania funkcji z jej definicją. W przypadku,

gdy do weryfikacji pola użyta została funkcja wymagająca parametru innego typu zostanie wyświetlone okno zawierające stosowny komunikat.

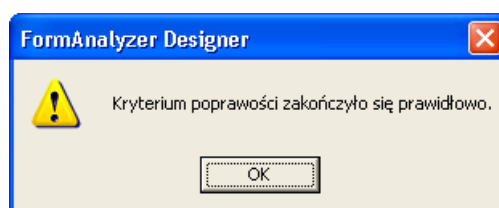


Rys. 7-32 Komunikat o niezgodności typu parametru funkcji z typem pola.

Wystąpienie tego błędu oznacza, że wybrana została niewłaściwa funkcja. W taki wypadku, należy wybrać właściwą funkcję, po czym powtórzyć sprawdzenie działania funkcji.

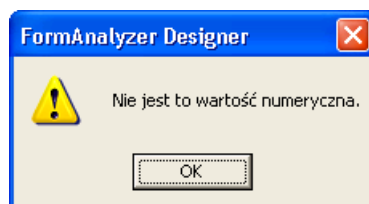
Powyższy błąd może wynikać też z niepoprawnej definicji funkcji. W takim przypadku należy poprawić kod źródłowy biblioteki i ponownie ją skompilować.

Gdy parametr jest odpowiedniego typu następuje wykonanie funkcji weryfikacji. Jeżeli wartość pola, (wprowadzona ręcznie w oknie parametrów) spełnia kryteria weryfikacji wyświetlone zostanie okno komunikatu pokazane na poniższym Rys. 7-33 Komunikat o spełnieniu przez zadaną wartość kryterium poprawności.



Rys. 7-33 Komunikat o spełnieniu przez zadaną wartość kryterium poprawności.

Powyższy komunikat jest skutkiem wykonania funkcji *Numeric* z biblioteki *Przykładowy DLL* z argumentem 1234567. Funkcja *Numeric* sprawdza, czy ciąg znaków, który jest parametrem wywołania jest wartością numeryczną. Wykonanie tej funkcji dla parametru 123456?, lub innego nie będącego liczbą, spowoduje wyświetlenie komunikatu pokazanego na poniższym rysunku Rys. 7-34 Komunikat błędu weryfikacji funkcji numerycznej.



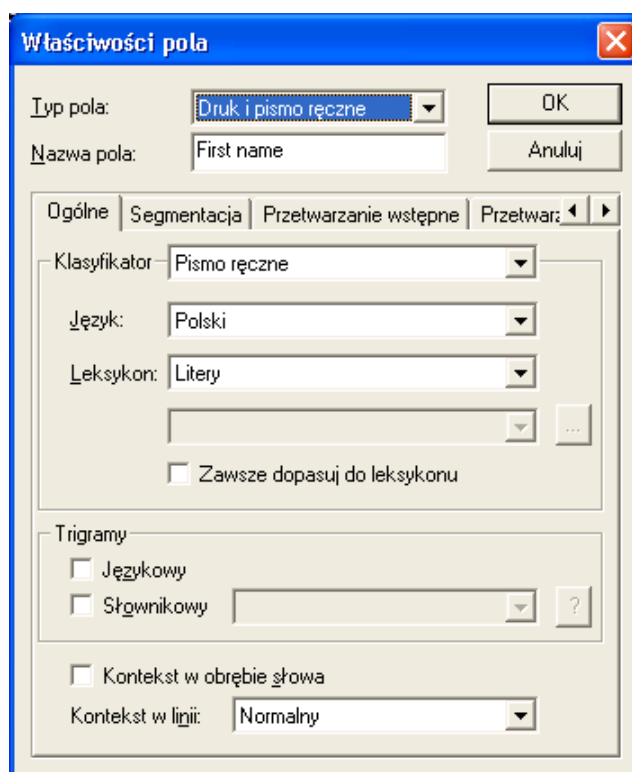
Rys. 7-34 Komunikat błędu weryfikacji funkcji numerycznej.

7.9. Pole Druk i pismo ręczne

Pole *Druk i pismo ręczne* jest polem zawierającym tekst pisany ręcznie lub drukowany. Podobnie jak w przypadku pola *Druk*, proces rozpoznawania i przetwarzania wartości pola jest wieloetapowy. Na wstępie jest wykonywane przetwarzanie początkowe, po którym następuje rozpoznanie wartości pola. Dla rozpoznanej wartości mogą być wykonane

kolejno: przetwarzanie końcowe, sprawdzenie wartości pola w słowniku, skrypt weryfikacji wartości oraz funkcja weryfikacji z zewnętrznej biblioteki DLL.

W zakładkach *Przetwarzanie początkowe*, *Przetwarzanie końcowe*, *Słowniki*, *Skrypt i Kryterium poprawności (DLL)* ustawia się właściwości przetwarzania wstępnego, parametry przetwarzania końcowego, sposób korzystania ze słowników wartości, skrypt weryfikacji oraz funkcji weryfikacji. Sposób definiowania parametrów w tych zakładkach, oraz sposób ich wykorzystania jest analogiczny, jak w przypadku pól drukowanych. Dokładny opis poszczególnych czynności został umieszczony w poprzednim punkcie: 7.8 - Pole Druk.



Rys. 7-35 Zakładka Ogólne okna właściwości pola Druk i pismo ręczne.

Na powyższym przedstawiono zakładkę *Ogólne* okna dialogowego *Właściwości pola Druk i pismo ręczne*. Została ona podzielona na trzy obszary. W pierwszym obszarze *Klasyfikator* należy zdefiniować sposób rozpoznawania pola. Można wybrać jeden z dziewięciu klasyfikatorów. Wyboru klasyfikatora dokonuje się wybierając odpowiednią pozycję z listy.:

Druk - przeznaczony do rozpoznawania znaków drukowanych.

Pismo ręczne - rozpoznaje pismo ręczne.

Druk i pismo ręczne - rozpoznaje w polu zarówno tekst drukowany, jak i napisany ręcznie.

Klasyfikator cyfr - klasyfikator rozpoznający cyfry, znaki przestankowe i symbole matematyczne zarówno pisane ręcznie, jak i drukowane.

Karta kredytowa (F7B) - przeznaczony do rozpoznawania numerów kart kredytowych.

OCR-A - klasyfikator rozpoznający cyfry drukowane czcionką OCR-A.

OCR-B - klasyfikator rozpoznający cyfry drukowane czcionką OCR-B.

Cmc7 - klasyfikator rozpoznający cyfry drukowane czcionką Cmc7.

E13b - klasyfikator rozpoznający cyfry drukowane czcionką E13b.

W przypadku klasyfikatorów Karta kredytowa, Cmc7 i E13b w grupie Leksykony nie ma możliwości zmiany ustawień - klasyfikatory rozpoznają tylko cyfry. W pozostałych przypadkach można skorzystać z leksykonów predefiniowanych, lub zdefiniować leksykon użytkownika. Przy definiowaniu leksykonu użytkownika, będą aktywne tylko te grupy znaków, które aktualnie wybrany klasyfikator rozpoznaje. Szczegółowy opis tworzenia leksykonu użytkownika został podany w poprzednim punkcie: 7.8 - Pole Druk.

W liście rozwijalnej Język można zdefiniować język, którym będą wypełniane pola. Na liście tej można jednocześnie wskazać kilka języków, klikając na każdym kolejnym lewym przyciskiem myszy. Kliknięcie na pozycji wcześniej już zaznaczonej spowoduje anulowanie jej zaznaczenia. W przypadku formularzy spotyka się najczęściej pola korzystające z jednego języka.

Zaznaczenie opcji Zawsze dopasuj do leksykonu ma wpływ na wyniki rozpoznawania. Włączenie tej opcji powoduje, że system przyjmuje, że w danym polu nie występują inne znaki, oprócz tych, które są określone w leksykonie. Dzięki temu system rzadziej generuje znaki nierozpoznane.

W tabeli poniżej znajdują się przykładowe rozpoznane znaki oraz interpretacja ich przez system przy wyłączonej i włączonej opcji Zawsze dopasuj do leksykonu. W tym przykładzie leksykon musiałby być zdefiniowany tylko dla cyfr. Znak ~ oznacza, że przy wyłączonej opcji Zawsze dostosuj do leksykonu znak nie zostałby prawidłowo zinterpretowany. Zawężenie możliwych wartości, które mogą pojawić się w polu przez utworzenie leksykonu ułatwiają systemowi dopasowanie odpowiedniego znaku.

Obraz oryginalny	Wyłączona opcja Zawsze dopasuj do leksykonu.	Włączona opcja Zawsze dopasuj do leksykonu.
	~8~16	88816
	723~7	72307

Kolejny obszar w zakładce Ogólne dla pola *Druk i pismo ręczne* służy do ustawiania parametrów trigramów używanych przy rozpoznawaniu wartości pola. Trigramy używane w FormAnalyze są informacjami o częstotliwości występowania trzyliterowych sekwencji znaków. Np. dla polskich nazwisk sekwencja liter 'SKI' jest bardziej prawdopodobna niż sekwencja 'CYR', natomiast sekwencja 'BBE' w ogóle w polskich nazwiskach nie występuje. Wykorzystanie tych informacji pozwala na poprawę jakości rozpoznawania.

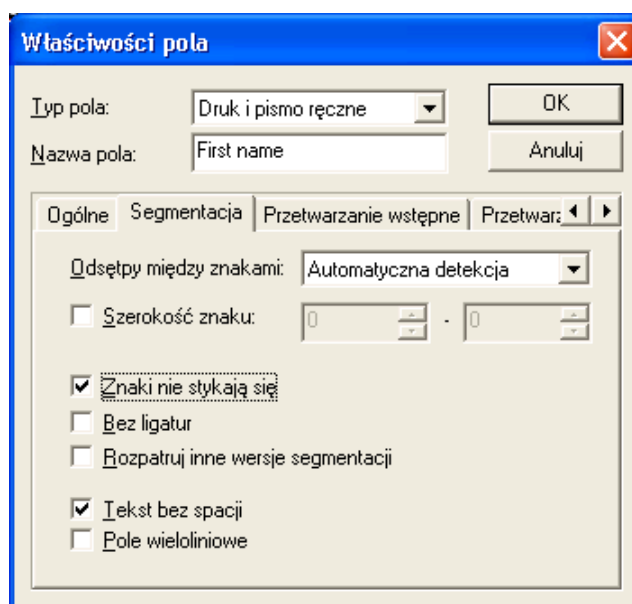
Istnieją dwie opcje wyszukiwania trigramów:

- językowy - występowanie trigramów obliczane jest w obrębie wszystkich występujących słów.

- Słownikowy – występowanie trigramów obliczane jest dla pewnego zestawu wartości, które może przyjmować pole (np. dla słownika wartości).

W przypadku wybrania trigramów słownikowych, kliknięcie przycisku , znajdującego się po prawej stronie obok rozwijalnej listy, umożliwi wyświetlenie dodatkowych informacji w formie okna komunikatu o wskazanym słowniku.

Opcja Kontekst w obrębie słowa umożliwia rozpoznawanie znaków w obrębie słowa, które mogą posiadać odpowiedniki w postaci symboli cyfr np.: 2 Z; 8 B; 5 S; 9 g. W momencie zaznaczenia tej opcji rozpatrywane jest położenie konkretnych znaków na formularzu. Wynikiem rozpoznawania będzie znak, który ostatecznie najbardziej odpowiada konkretnemu usytuowaniu na formularzu. Z rozwijalnej listy Kontekst w linii można wybrać jedną z dostępnych czterech opcji precyzjności rozpoznawania podobnych znaków,



Rys. 7-36 Zakładka okna właściwości pola Druk i pismo ręczne.

W drugiej zakładce Segmentacja można zdefiniować następujące ustawienia:

- Odstępy między znakami
Odstępy między znakami uzależnione są od kroju czcionki drukowanej lub sposobu zapisania tekstu za pomocą pisma ręcznego. Rozpoznawanie odstępów między znakami w programie FormAnalyze Designer można przeprowadzić na trzy sposoby definiując znaki równo odległe, znaki proporcjonalne lub zlecając silnikowi automatyczną detekcję.
- Szerokość znaku
Opcja ta wykorzystywana jest w przypadku, gdy na formularzu znajdują się znaki ze sobą połączone. Ustawienie minimalnej i maksymalnej szerokości znaku wskazuje, jaką szerokość maksymalną lub minimalną będzie posiadał dzielony znak. Wyłączenie tej opcji spowoduje rozpoznawanie znaków połączonych, jako całości, bez uwzględnienia podziału. Wielkości podawane są w pikselach, przeliczone zgodnie z rozdzielnością zeskanowanego formularza.
- Znaki nie stykają się

Opcja ta wykorzystywana jest w momencie, gdy użytkownik ma pewność, że w danym polu nie występują żadne znaki stykające się ze sobą.

- Bez ligatur

Zaznaczenie tej opcji oznacza, że w polu używana jest czcionka, w której nie występują ligatury, czyli połączenia pewnych liter np.: ft, fi, w kroju czcionki Times Roman.

- Rozpatruj inne wersje segmentacji

Zaznaczenie tej opcji spowoduje zadanie rozpoznawania wszystkich alternatywnych możliwości występowania określonych znaków na formularzu w zależności od poziomu zaufania. Niezaznaczenie opcji oznacza rozpoznawanie danego tekstu najprostszą i najbardziej prawdopodobną dla programu metodą.

- Tekst bez spacji

W procesie rozpoznawania pomijane będą spacje występujące w tekście.

- Pole wieloliniowe

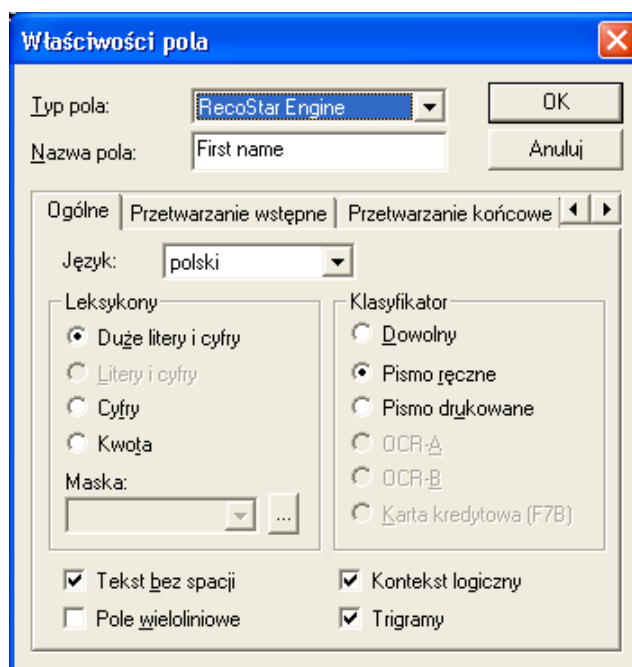
W przypadku zaznaczenia tej opcji wynikiem rozpoznawania będzie tekst znajdujący się w więcej niż jednej linii. Nie zaznaczenie tej opcji spowoduje rozpoznanie tekstu w jednej linii będącej najbardziej dominującą ze wszystkich znajdujących się w danym polu.

7.10. Pole RecoStar Engine

Pole *RecoStar Engine* jest przeznaczone do rozpoznawania pól zawierających tekst pisany ręcznie lub drukowany. W stosunku do pola *Druk i pismo ręczne* różni się ono zastosowanym pakietem rozpoznawania. Podobnie jak w przypadku już omówionych pól rozpoznawania pisma, proces rozpoznawania i przetwarzania wartości pola jest wieloetapowy. Na wstępie jest wykonywane przetwarzanie początkowe, po którym następuje rozpoznanie wartości pola. Dla rozpoznanej wartości mogą być wykonane kolejno: przetwarzanie końcowe, sprawdzenie wartości pola w słowniku, skrypt weryfikacji wartości oraz funkcja weryfikacji z zewnętrznej biblioteki DLL.

W zakładkach *Przetwarzanie początkowe*, *Przetwarzanie końcowe*, *Słowniki*, *Skrypt i Kryterium poprawności (DLL)* ustawia się właściwości przetwarzania wstępnego, parametry przetwarzania końcowego, sposób korzystania ze słowników wartości, skrypt weryfikacji wartości oraz funkcję weryfikacji wartości. Parametry ustawiane na tych zakładkach, oraz sposób ich wykorzystania, jest dokładnie taki sam jak w przypadku pola *Druk*. Szczegółowy opis znajduje się w punkcie 7.8 - Pole Druk.

Parametry dotyczące samego sposobu rozpoznawania treści pola - specyficzne dla pola typu *RecoStar Engine* - są ustawiane w zakładce *Ogólne* okna dialogowego Właściwości tego pola.



Rys. 7-37 Zakładka Ogólne okna właściwości pola RecoStar Engine.

Podstawowe znaczenie dla sposobu rozpoznawania treści pola ma wybór klasyfikatora używanego przy rozpoznawaniu. Dostępne są następujące klasyfikatory:

- *Pismo ręczne* - przeznaczony do rozpoznawania pisma ręcznego.
- *Pismo drukowane* - rozpoznaje pismo drukowane.
- *Dowolny* - rozpoznaje zarówno pismo ręczne jak i pismo drukowane.
- (Na podstawie analizy rozpoznawanego pola jest podejmowana decyzja o tym, czy stosować klasyfikator pisma ręcznego czy drukowanego.)
- *OCR-A* - przeznaczony do rozpoznawania znaków drukowanych czcionką OCR-A.
- *OCR-B* - przeznaczony do rozpoznawania znaków drukowanych czcionką OCR-B.
- *Karta kredytowa (F7B)* - przeznaczony do rozpoznawania numerów kart kredytowych.

Dostępność poszczególnych klasyfikatorów jest związana z wybraną pozycją z rozwijalnej listy *Język*, z której należy wskazać język, w którym będzie wypełnione pole.

Klasyfikatory *OCR-A*, *OCR-B* i *Karta kredytowa (F7B)* są dostępne jedynie przy ustawieniu w polu *Język* wartości *międzynarodowy*. Klasyfikatory pisma drukowanego i ręcznego (oraz dowolny) są dostępne, przy wybraniu każdej innej opcji niż *międzynarodowy*.

Zestaw dostępnych klasyfikatorów, a także leksykonów i opcji pozwalających na poprawę jakości rozpoznania, które zostały omówione w dalszej części niniejszego punktu, jest różny dla różnych języków. Np. ustawienia austriacki jest dostępny jedynie klasyfikator pisma ręcznego z leksykonem cyfr i znakami umożliwiającymi zapis kwot. W języku niemieckim dostępny jest pełen zestaw klasyfikatorów i leksykonów.

Wybranie klasyfikatora pozwala pakietowi rozpoznawania znaków dostosować metodę rozpoznawania do sposobu wypełnienia pola lub, w przypadku klasyfikatora *Dowolny*, wybrać tę metodę na podstawie analizy pola. Leksykony służą do ograniczenia zbioru rozpoznawanych znaków. Określenie, jakie znaki mogą wystąpić w polu pozwala, w wielu przypadkach, na znaczną poprawę jakości rozpoznawania. Np. cyfra „0” może być łatwo pomyłona z literami „O” i „D”. Jeżeli w danym polu mogą pojawić się jedynie cyfry, należy ustawić leksykon Cyfry - w takiej sytuacji „0” nie będzie mylone z „O” i „D”.

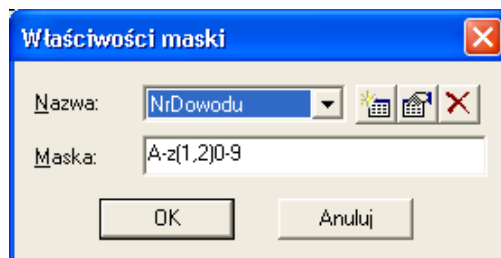
Dostępne dla konkretnego pola leksykony zależą od omówionych wcześniej ustawień języka i klasyfikatora. Jeżeli leksykon jest aktywny jego znaczenie jest zawsze takie samo - ogranicza zbiór znaków w polu do pewnego podzbioru. Znaczenie leksykonów jest następujące:

- *Duże litery i cyfry* - rozpoznawane znaki to duże litery i cyfry.
- *Litery i cyfry* - rozpoznawane znaki to litery duże i małe oraz cyfry.
- *Cyfry* - rozpoznawane znaki to cyfry.
- *Kwota* - rozpoznawane znaki to cyfry i znaki umożliwiające zapis kwoty.

Właściwy dobór klasyfikatora i leksykonu pozwala na zwiększenie jakości rozpoznawania. W niektórych przypadkach można podać, bardziej precyzyjnie niż pozwalają na to leksykony, znaki, których należy spodziewać się w polu. Skorzystanie z takich dodatkowych informacji dotyczących struktury treści pola umożliwiają maski.

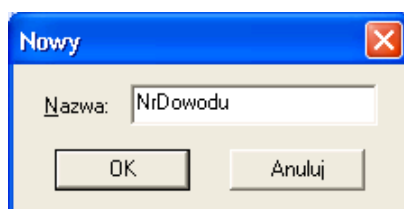
Korzystając z rozwijanej listy *Maska* można wybrać jedną z masek zdefiniowanych już w pliku konfiguracji. Na tej liście, oprócz nazw masek, znajduje się także pusta pozycja. Służy ona do zaniechania stosowania maski przy rozpoznawaniu pola.

W celu zdefiniowania nowej maski lub zmodyfikowania już istniejącej, należy kliknąć przycisk  znajdujący się z prawej strony pola. Na ekranie wyświetlone zostanie okno dialogowe Właściwości maski.



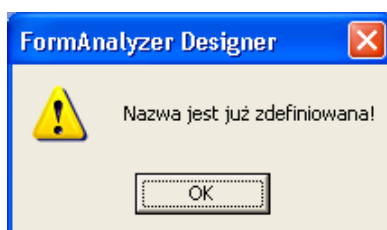
Rys. 7-38 Okno właściwości maski.

Na poniższym rysunku pokazane zostało przykładowe okno Właściwości maski dla pola *NrDowodu*, zdefiniowanej w celu rozpoznawania serii i numeru dowodu osobistego. Aby zdefiniować nową maskę należy kliknąć przycisk Nowy  znajdujący się po prawej stronie pola Nazwa. Na ekranie wyświetlone zostanie okno dialogowe Nowy umożliwiające zdefiniowanie nazwy dla nowotworzonej maski.



Rys. 7-39 Okno nazwy nowej maski dla pola typu RecoStar Engine.

Wprowadzona nazwa musi być unikalna w całym pliku konfiguracji. Jeżeli utworzona zostanie maska o nazwie już używanej w pliku konfiguracji przez inną maskę, to przy próbie zaakceptowania nowej nazwy wyświetlone zostanie okno dialogowe ze stosownym komunikatem. Spowoduje to także ponowną próbę zdefiniowania nazwy dla nowotworzonej maski. W przypadku tworzenia nowej maski pole Nazwa nie może pozostać puste, bo wtedy maska nie zostanie utworzona.



Rys. 7-40 Komunikat o powtórzeniu nazwy maski.

Po poprawnym wprowadzeniu nazwy należy zdefiniować samą maskę, wprowadzając odpowiednie symbole znaków w polu *Maska* w oknie dialogowym *Właściwości maski*. W najprostszym przypadku maska definiuje zbiór dopuszczalnych w polu znaków. Przy wprowadzaniu znaków można korzystać z dwóch metod:

- Przez jawne wyliczenie dopuszczalnych znaków.

Maska pozwalająca na wystąpienie w polu dużych liter A, B, C, D i cyfr 1, 2, 3, 4 zostałaby zapisana w tej postaci jako: ABCD1234.

- Przez podanie zakresu znaków, w postaci znak początkowy - znak końcowy.

Korzystając z tej metody zapisu, maska z poprzedniego punktu będzie następująca: A-D1-4.

Obydwa sposoby zadawania zbioru znaków mogą być stosowane równocześnie. Maskę dla dużych liter A, B, C, D i cyfr 1, 2, 3, 4 można zapisać też jako np.: A-D1234, ABCD1-4, A-CD12-4.

Ponieważ znak „-” jest używany do oddzielania początku i końca zakresu znaków, włączenie go do zbioru znaków rozpoznawanych wymaga użycia kombinacji znaków: znaku specjalnego „\” oraz samego znaku „-”.

Maskę dopuszczającą użycie w polu cyfr od 1 do 4 oraz znaku „-” można zapisać tak:

1-4\-

Dodatkową, bardzo cenną właściwością maski, jest możliwość wskazania, które znaki w polu mają pochodzić ze zdefiniowanego zbioru. Aby zastosować podany zbiór dla pojedynczego znaku podajemy - za definicją zbioru - jego pozycję w rozpoznanym napisie: {1} oznacza pierwszy znak rozpoznany w polu, {2} drugi, itd. Można także

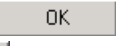
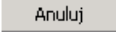
wskazać ciąg kolejnych znaków: {1,2} oznacza stosowanie zbioru do pierwszych dwóch znaków pola, {3,5} do znaków od trzeciego do piątego, itd.

W przykładzie maski *NrDowodu* jako pierwsze dwa znaki dopuszcza się duże litery. Definicja zbioru to A-Z, za nią jest podany zakres pozycji znaków w polu, do których stosuje się ten zbiór {1,2}. Za A-Z{1,2} pojawia się kolejna definicja zbioru znaków, obejmującego cyfry: 0-9. Ponieważ w tym przypadku nie podano, dla których pozycji stosować zbiór, będzie on stosowany aż do końca linii (pola).

Jeśli długość pola, a dokładniej liczba znaków, które muszą zostać w danym polu wpisane, jest ściśle określona możemy zdefiniować maskę dla takiej liczby znaków. Np. seria dowodu osobistego składa się z dwóch znaków, a jego numer z 7 cyfr. Maskę *NrDowodu* można byłoby zapisać jako: A-Z{1,2}0-9{3,9}.

Takie rozwiązanie pociąga za sobą pewną niedogodność: maska jest stosowana jedynie w przypadku, gdy w polu rozpoznano **dokładnie** tyle znaków ile jest wyspecyfikowanych w masce.

W masce mogą być używane tylko takie znaki, które są dopuszczane przez używany dla pola klasyfikator i leksykon. Użycie w masce znaku spoza tego zakresu prowadzi do błędów rozpoznawania.

Po zadaniu odpowiedniej postaci maski, należy kliknąć przycisk  - nowa maska zostanie dodana do pliku konfiguracji. Naciśnięcie  oznacza zaniechania definiowania nowej maski.

W oknie dialogowym Właściwości maski korzystając z przycisków funkcyjnych znajdujących się po prawej stronie pola Nazwa, można zmienić nazwę lub usunąć maskę.

Aby zmienić nazwę maski należy, po uprzednim zaznaczeniu żądanej maski kliknąć przycisk Zmień nazwę . Na ekranie wyświetlone zostanie okno dialogowe Zmiana nazwy. W oknie tym należy dokonać edycji nazwy maski. Nazwa musi być unikalna. Jeżeli zostanie wprowadzona nazwa, zapisana już uprzednio w pliku konfiguracji, wyświetlony zostanie stosowny komunikat, a użytkownik zmuszony zostanie do ponownej edycji nazwy maski.

Kiedy nowa nazwa jest unikalna, maska zostanie ponownie zapisana. Nazwa zmieni się nie tylko w polu, w którego właściwościach dokonać edycji nazwy, lecz we wszystkich polach korzystających z tej maski.

W celu usunięcia żądanej maski należy po uprzednim jej zaznaczeniu kliknąć przycisk Usuń . Operacja ta, podobnie jak zmiana nazwy, może dotyczyć wielu pól rozpoznawanych, w różnych formularzach i szablonach dokumentów.

W oknie właściwości maski można także zmienić postać samej maski. Przy takich zmianach trzeba mieć na uwadze, że maska może być wykorzystywana także w innych polach. Jeśli nie ma pewności, czy modyfikacja maski będzie właściwa dla wszystkich pól korzystających z niej, należy zdefiniować nową maskę.

8. Grupowanie pól

Celem grupowanie pól jest utworzenie zależności pomiędzy polami składowymi. Może ona ułatwiać proces rozpoznawania, dzieje się to np. przy grupowaniu pól tekstowych z włączoną opcją *Wybierz najlepszy wynik*. Relacja ta może umożliwić sprawdzanie poprawności danych, dzieje się to np. przy grupowaniu pól wyboru z włączoną opcją *Łącz pola składowe*. Grupowanie może także służyć jedynie do stworzenia formatki weryfikacyjnej, dzieje się to przy włączeniu opcji *Wiersz danych* lub *Sekcja danych*.

W każdym przypadku efektem grupowania jest nowe pole nadrzędne o unikalnej nazwie. Przy włączonej opcji *Łącz pola składowe* i *Wybierz najlepszy wynik*, pola wchodzące w skład grupy nie są dostępne w procesie przetwarzania. Przy włączonej opcji *Wiersz danych* i *Sekcja danych*, pola wchodzące w skład grupy dostępne są w procesie walidacji i eksportu. W tym przypadku pole nadrzędne nie podlega eksportowi.

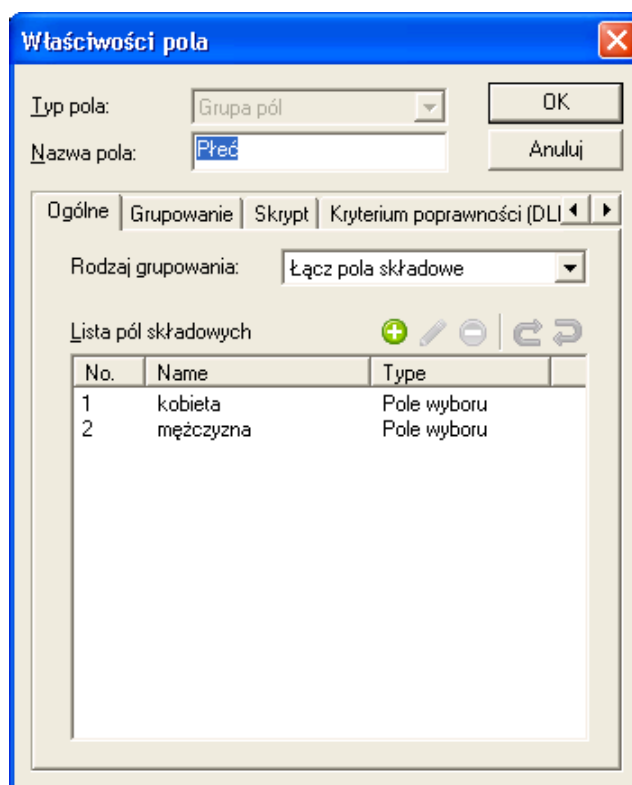
Pola zgrupowane można grupować zgodnie z następującymi zasadami:

- Opcja *Wybierz najlepszy wynik* dostępna jest tylko dla pól tekstowych
- Opcja *Łącz pola składowe* dostępna jest tylko dla pól, które generują ten sam typ danych: tekst lub pola wyboru
- Opcja *Wiersz danych* nie jest dostępna dla pól zgrupowanych metodą *Wiersz danych* i *Sekcja danych*
- Opcja *Sekcja danych* nie jest dostępna dla pól zgrupowanych metodą *Sekcja danych*

Niezależnie od sposobu grupowania, który zastosujemy, tworzenie grupy pól przebiega tak samo. Najpierw należy zdefiniować na formularzu pola składowe, pamiętając o wybraniu właściwych parametrów rozpoznawania, weryfikacji i eksportu. Jest to bardzo istotne, ponieważ po utworzeniu grupy pól, dostęp do poszczególnych składowych jest utrudniony. Żeby zmienić parametry pola składowego, trzeba zgrupowane pola rozgrupować, zmienić parametry pola i ponownie utworzyć grupę. Inną metodą zmiany właściwości pól zgrupowanych jest ustawienie odpowiednich cech w dialogu *Pola* w sekcji *Pola wewnętrzne*. Dokładny opis znajduje się w rozdziale 9 - Właściwości pól.

Pola, które mają utworzyć grupę, należy wybrać. Kolejność wybierania pól jest istotna przy łączeniu treści pól. Ich wartości złożą się na wartość grupy w kolejności wybierania.

W celu zgrupowania pól należy wyświetlić okno dialogowe *Właściwości pola* pokazane na poniższym rysunku korzystając z jednej z dostępnych metod: po uprzednim zaznaczeniu żądanych pól klikając na pasku narzędzi przycisk , lub wybierając z menu *Obiekt* polecenie *Grupuj*. Na ekranie zostanie wyświetlone okno dialogowe *Właściwości pola*.



Rys. 8-1 Przykładowe okno właściwości grupy pól.

Na pierwszej zakładce należy podać nazwę nowego pola grupowego, zgodnie z zasadami nazywania pól opisanymi w punkcie 7.2 - Typy i nazwy pól. W zależności od typów pól składowych w rodzaju grupowania dostępne mogą być następujące elementy do wyboru: *łącz pola składowe*, *Wybierz najlepszy wynik*, *Wiersz danych* i *Sekcja danych*. W *Liście pól składowych* podane są kolejno pola wchodzące w skład grupy. Kolejność pól składowych można zmienić używając przycisków (przeniesienie pola o jedną pozycję do góry) i (przeniesienie pola o jedną pozycję na dół).

Na drugiej zakładce użytkownik może zdefiniować szczegółowe cechy związane z grupowaniem i walidacją pola.

8.1. Grupy pól wyboru

Ze względu na kryteria poprawności bardzo często umieszcza się pola wyboru w grupach. Na przykład w celu ustalenia płci respondenta umieszcza się na formularzu dwa pola wyboru odpowiadające odpowiednio płci męskiej i żeńskiej. Płeć zostanie prawidłowo rozpoznana, gdy zostanie zaznaczone dokładnie jedno z pól wyboru.

Aby utworzyć grupę pól można zgrupować zaznaczone wcześniej pola. W tym celu należy wybrać wszystkie pola, które mają należeć do grupy, a następnie z menu Obiekt wybrać polecenie Grupuj. Na ekranie zostanie wyświetlone okno dialogowe Właściwości pola.

Właściwości pola

Typ pola: Grupa pól OK

Nazwa pola: Płeć Anuluj

Ogólne Grupowanie Skrypt Kryterium poprawności (DLI) < >

Rodzaj grupowania: Łącz pola składowe

Lista pól składowych

No.	Name	Type
1	kobieta	Pole wyboru
2	mężczyzna	Pole wyboru

Rys. 8-2 Okno właściwości grupy pól wyboru (zakładka Ogólne).

Zgrupowane pola wyboru będą traktowane jako jedno pole jeżeli w liście *Rodzaj grupowania* zostanie wybrana pozycja *Łącz pola składowe*. Dlatego trzeba podać nazwę nowego pola grupowego, zgodnie z zasadami nazywania pól opisanymi w punkcie 7.2 - Typy i nazwy pól.

Na drugiej zakładce (Rys. 8-3) w grupie pól *Warunki walidacji znaczników wyboru*, należy wprowadzić dwa parametry pozwalające na weryfikację poprawności wypełnienia pola. Są to minimalna i maksymalna liczba zaznaczonych pól. Wartościami domyślnymi są odpowiednio 0 i liczba pól wyboru w grupie. Oznacza to, że jako poprawne jest traktowane dowolne zaznaczenie (lub jego brak).

W oknie właściwości, parametry *Min. liczba wybranych pozycji* i *Max. liczba wybranych pozycji* mają tę samą wartość. Pole grupowe *Płeć* będzie spełniało warunki poprawności, kiedy zostanie zaznaczone dokładnie jedno z tworzących go pól wyboru (kobieta albo mężczyzna).

Właściwości pola

Typ pola: Grupa pól OK

Nazwa pola: Polec Anuluj

Ogólne Grupowanie Skrypt Kryterium poprawności (DLI)

Warunki walidacji znaczników wyboru

Min. liczba wybranych pozycji: 1

Max. liczba wybranych pozycji: 1

Separator pól składowych:

Format wiersza/sekcji

☐ Kolor tła: Default

☐ Kolor tekstu: Default

Rys. 8-3 Warunki poprawności grupy pól wyboru (zakładka Grupowanie).

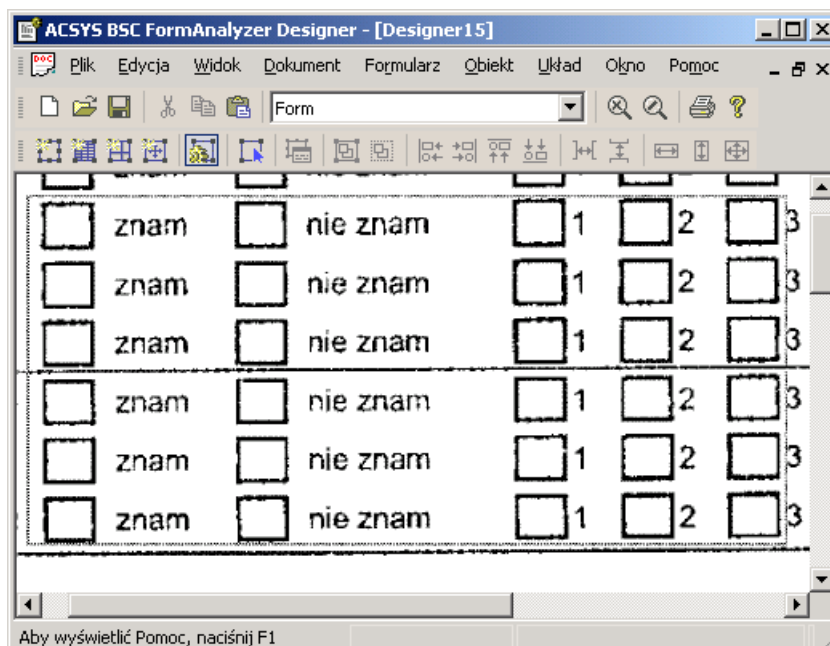
Kolejne zakładki służą do zdefiniowania bardziej zaawansowanych warunków poprawności. Zasady tworzenia skryptów, bibliotek DLL i funkcji WebService jest bardzo podobna jak dla pól tekstowych. Różnica istnieje jedynie w typie przekazywanych wartości z pola do funkcji walidujących.

Opisaną powyżej metodę tworzenia grup pól wyboru można stosować w przypadku niewielkiej liczby zaznaczonych pól. Jeśli liczba pól wyboru jest duża, należy skorzystać z funkcji automatycznego tworzenia grup pól wyboru. Ta funkcja w zaznaczonym obszarze sama wyszukuje elementy, które są polami wyboru, i grupuje je według ustalonego kryterium.

Automatyczne tworzenie grupy pól wyboru umożliwia polecenie Automat dla pól wyboru z menu Obiekt → Nowy lub klikając na pasku narzędzi przycisk Wyszukaj pola wyboru . Następnie należy zaznaczyć prostokątny obszar, w którym będą wyszukiwane pola wyboru. Bezpośrednio po zaznaczeniu obszaru wyszukiwania jest wyświetlany kreator parametrów wyszukiwania pól wyboru. Daje on możliwość precyzyjnego ustawienia parametrów wyszukiwania. Pozwala też na zadanie właściwości grup pól wyboru, które mogą być automatycznie tworzone przy okazji wyszukiwania.

Na poniżej pokazano fragment formularza, dla którego zastosowane będzie automatyczne wyszukiwanie pól wyboru. Rzędy pól wyboru są do siebie podobne. W każdym rzędzie można wyróżnić dwie grupy pól: dwuelementową (opisaną znam, nie znam) i trzelementową (opisaną cyframi od 1 do 3). W grupie dwuelementowej, nazwijmy ją *Znajomość*, musi zostać zaznaczone dokładnie jedno pole wyboru. W grupie trzelementowej, o nazwie *Stopień*, może być zaznaczone, co najwyżej jedno pole wyboru.

Omawiając kreator wyszukiwania pól wyboru parametry wyszukiwania muszą być zadane tak, by dla zaznaczonych sześciu rzędów powyższe grupy zostały utworzone automatycznie.



Rys. 8-4 Fragment dokumentu, którym będą automatycznie wyszukiwane pola wyboru.

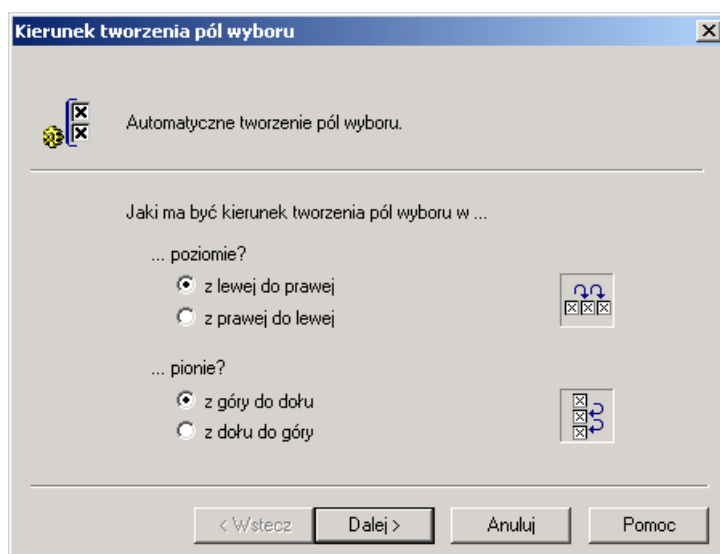
W pierwszej kolejności należy zaznaczyć żądane elementy. W naszym przykładzie zaznaczonych zostało wszystkich sześć widocznych elementów. Bezpośrednio po zakończeniu rysowania prostokąta obszaru jest wyświetlany kreator wyszukiwania pól wyboru. Kliknięcie przycisku **Dalej >** powoduje przejście do kolejnej strony kreatora; przycisk **< Wstecz** służy do przejścia na poprzednią stronę.

Pierwsze okno dialogowe kreatora Kierunek tworzenia pól wyboru, przedstawione na poniżej, służy do wybrania kierunku tworzenia pól wyboru. Kolejność tworzenia pól wyboru na formularzu ma największe znaczenie dla kolejności przetwarzania pól. Jeżeli nie użytkownik nie zdefiniuje kolejności, to pola będą przetwarzane w zgodzie z kolejnością tworzenia.

Kolejność tworzenia jest także istotna przy automatycznym nadawaniu nazw polom czy grupom pól.

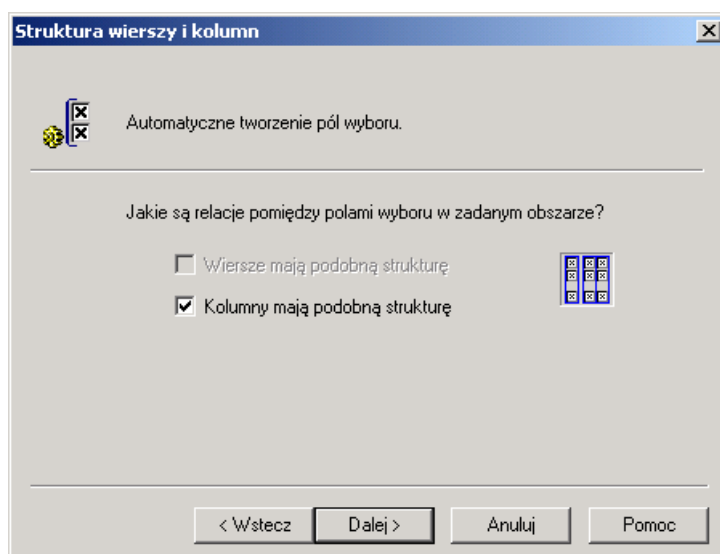
Domyślnie ustawione jest tworzenie pól od lewej do prawej i z góry na dół.

Dla niniejszego przykładu automatycznego wyszukiwania pozostawione zostały wartości domyślne.



Rys. 8-5 Kreator wyszukiwania pól wyboru - kierunek tworzenia pól wyboru.

Kolejne okno dialogowe Struktura wierszy i kolumn pozwala na określenie struktury wyszukiwanych pól wyboru. Definiowana jest tutaj relacja występująca pomiędzy polami wyboru w wierszach lub kolumnach. Domyślnie ustawiona jest opcja Wiersze mają podobną strukturę. Oznacza to, że w trakcie wyszukiwania FormAnalyzer Designer będzie starał się dopasować pola wyboru w wierszu do struktury zadanej na ostatniej stronie kreatora.



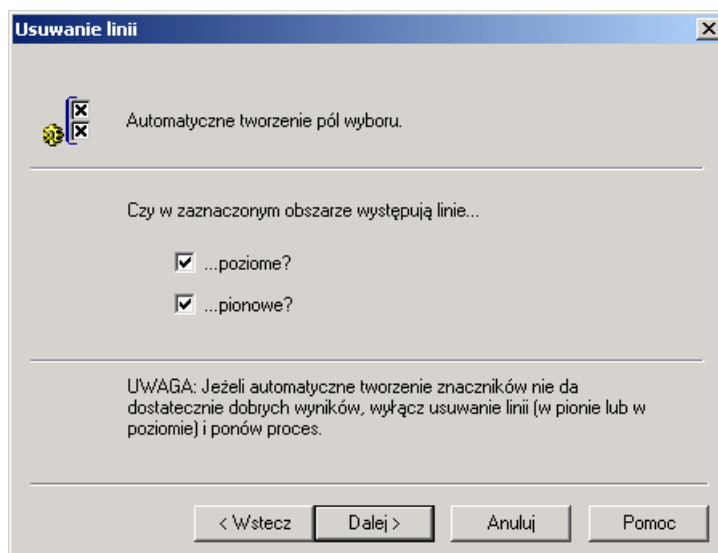
Rys. 8-6 Kreator wyszukiwania pól wyboru - struktura wierszy i kolumn.

Zaznaczenie podobnej struktury kolumn jest teraz nieaktywne. Nie można jednocześnie ustawić podobnej struktury wierszy i kolumn. W celu zaznaczenia opcji Kolumny mają podobną strukturę, należy najpierw odznaczyć pole wyboru przy opcji Wiersze mają podobną strukturę. Dla podobnej struktury kolumn FormAnalyzer Designer będzie starał się dopasować pola wyboru w kolumnie do struktury zadanej na ostatniej stronie kreatora.

W przypadku, gdy obydwie opcje pozostaną niezaznaczone, pola wyboru będą wyszukiwane bez dodatkowych ograniczeń. W takiej sytuacji automatyczne tworzenie pól wyboru nie powiedzie się.

W omawianym przykładzie, pola wyboru mają podobną strukturę wierszy. Zatem także na tej stronie, pozostawione zostaną ustawienia domyślne.

W trzecim oknie dialogowym kreatora Usuwanie linii), należy podać informacje dotyczące linii znajdujących się w obszarze wyszukiwania: poziome lub pionowe. Te informacje w istotny sposób usprawniają wyszukiwanie pól wyboru. Jest to szczególnie istotne w przypadku stykania się linii z ramkami pól wyboru.



Rys. 8-7 Kreator wyszukiwania pól wyboru - usuwanie linii.

W omawianym przykładzie, w obszarze wyszukiwania znajdują się jedynie linie poziome.

W celu zdefiniowania jedynie linii poziomych, należy kliknąć lewym przyciskiem myszki na polu wyboru ☐ znajdującym się przy opcji ... pionowe?. Zaznaczenie tego pola zostanie anulowane.

W ostatnim oknie dialogowym kreatora Nazwa i struktura pól wyboru, przedstawionym na Rys. 8-8, należy zdefiniować precyzyjnie strukturę wyniku wyszukiwania pól wyboru. Należy tu rozpatrzyć dwa przypadki.

Pierwszy z nich ma miejsce, jeśli w oknie dialogowym kreatora Struktura wierszy i kolumn, nie zdefiniowano podobieństwa struktury wierszy ani kolumn. Wynik wyszukiwania pól wyboru jest wtedy zbiorem pojedynczych pól. W takiej sytuacji można jedynie zdefiniować nazwę pola – rdzeń nazwy, do którego dla każdego pola jest dodawany modyfikator wynikający z kolejności jego tworzenia. Kliknięcie przycisku **Zakończ** rozpocznie proces poszukiwania i tworzenia pól wyboru w zadanym obszarze.

Drugi przypadek ma miejsce, gdy w oknie dialogowym Struktura wierszy i kolumn zdefiniowano podobieństwo wierszy albo kolumn. Upřednie zdefiniowanie podobieństwa wierszy lub kolumn spowoduje, że w oknie dialogowym Nazwa i struktura pól wyboru, trzeba będzie określić liczbę pól wyboru oraz ewentualnie sposób łączenia ich w grupy. Analogicznie do pierwszego przypadku można zdefiniować nazwy pól i/lub grup pól wyboru..

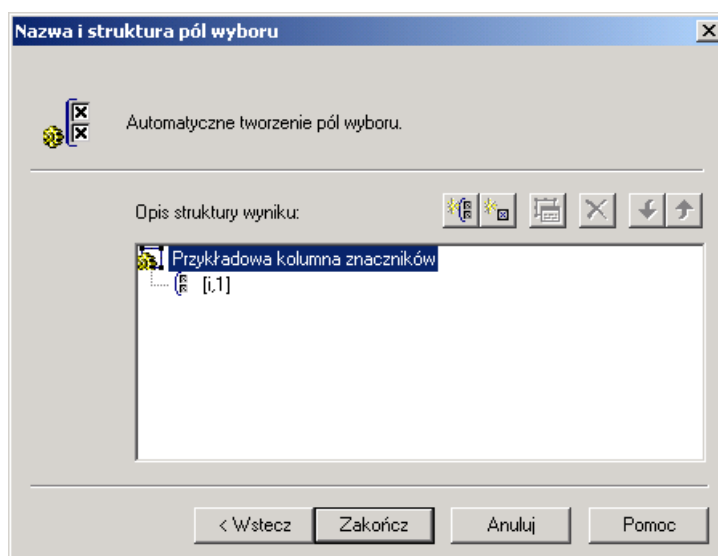
W oknie *Opis struktury wyniku* jest wyświetlana aktualna struktura kolumny lub wiersza (dla zwięzłości będziemy mówić rząd, mając na myśli kolumnę lub wiersz pól wyboru).

Najwyższy poziom (korzeń) oznacza cały rząd. Grupy pól są zaznaczone za pomocą , natomiast pola wyboru za pomocą symbolu . W strukturze jest wyróżniony element aktualny – jego nazwa jest wyświetlona na niebieskim pasku.

Kliknięcie przycisku  lub  spowoduje dodanie do aktualnego elementu odpowiednio nowej grupy pól lub nowego pola. Bezpośrednio po dodaniu nazwa nowego elementu jest ustawiana w tryb edycji tak, by można ją było od razu odpowiednio zmodyfikować.

Przyciśnięcie  powoduje usunięcie aktualnego elementu.

Przyciski  i  służą do zamiany kolejności elementu aktualnego i jego następnika lub poprzednika.



Rys. 8-8 Kreator wyszukiwania pól wyboru - nazwy i struktura pól wyboru.

Każda nazwa tworzonej grupy pól wyboru składa się z dwóch części: stałego rdzenia nazwy oraz modyfikatora. Rdzeń nazwy jest dowolnym tekstem. Zadaniem modyfikatora jest wygenerowanie unikalnego dla pola (lub grupy) identyfikatora. W zależności od wybranej struktury nowy identyfikator jest generowany odpowiednio dla:

- Kolejnego pola wyboru, gdy nie zostały zdefiniowane podobieństwa struktury wierszy i kolumn.
- Kolejnego wiersza pól wyboru, gdy zdefiniowane zostało podobieństwo struktury wierszy.
- Kolejnej kolumny pól wyboru, zdefiniowane zostało, podobieństwo struktury kolumn.

Kolejne identyfikatory oblicza się zwiększając lub zmniejszając o 1 ich aktualną wartość. Zarówno kierunek zliczania, jak i wartość początkowa, są zadawane w modyfikatorze. Dla zwiększających się identyfikatorów od wartości początkowej modyfikator będzie zwiększał się o 1 [**i**, wartość_początkowa]; natomiast dla identyfikatorów malejących od wartości początkowej modyfikator będzie zmniejszał się o 1 [**d**, wartość_początkowa].

Na powyższym rysunku Rdzeń nazwy jest pusty, natomiast modyfikator jest zadany w postaci [i,1]. Oznacza to, że dla kolejnych wierszy pól wyboru zostaną wygenerowane nazwy 1, 2, .. itd.

Modyfikator może wystąpić, tak jak w przykładzie, za rdzeniem nazwy, ale można go także umieścić przed lub nawet w środku rdzenia.

W celu edycji wzorca nazwy, należy zaznaczyć żądany wzorzec nazwy, a następnie kliknąć na jego obszarze, aby udostępnić go do edycji. Po dokonaniu edycji tekstu bądź modyfikatora, należy zatwierdzić wprowadzone zmiany naciskając klawisz Enter lub klikając na dowolnym obszarze poza edytowanym wzorcem nazwy.

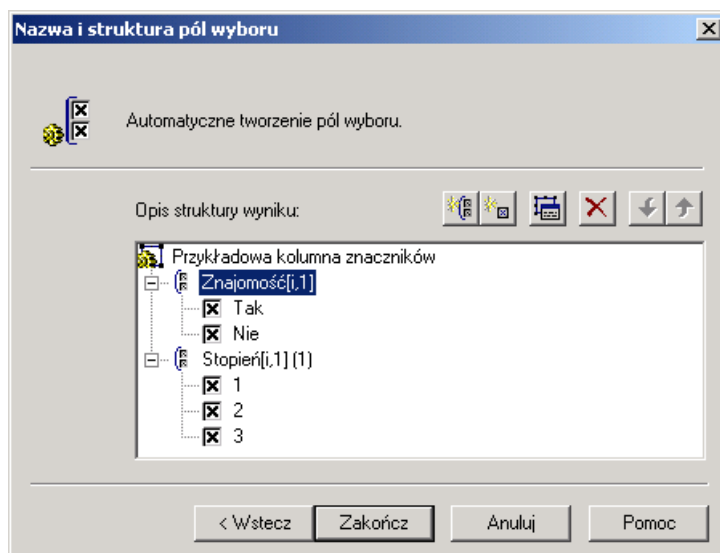
Aby dokonać zmiany nazwy można również posłużyć się inną metodą, polegającą na otwarciu okna dialogowego Właściwości pola, za pomocą kliknięcia na pasku narzędzi przycisku Podaj właściwości obiektu , kliknięcia na obszarze pól wyboru prawym przyciskiem myszy i wybraniem z menu kontekstowego polecenia Właściwości, lub po uprzednim zaznaczeniu pól wyboru z menu Obiekt wybrać polecenie Właściwości. W przypadku grupy pól, okno właściwości jest otwierane tylko wtedy, gdy grupa zawiera, co najmniej dwa elementy.

W wyświetlonym oknie dialogowym, należy w obszarze Nazwa pola dokonać edycji nazwy lub modyfikatora.

Można tu także zaznaczyć minimalną i maksymalną liczbę zaznaczonych pól wyboru i zmienić nazwę grupy.

Na poniższym rysunku wiersz składa się z dwóch grup pól wyboru. Grupa *Znajomość* składa się z pól wyboru *Tak* i *Nie*. We wzorcach nazw grupy i zawierających się w niej pól występuje modyfikator zwiększający z wartością początkową 1. Modyfikator w przypadku nazw pól nie musi się pojawić - nazwy pól *Tak* i *Nie* są w każdym wierszu przykrywane nazwą grupy.

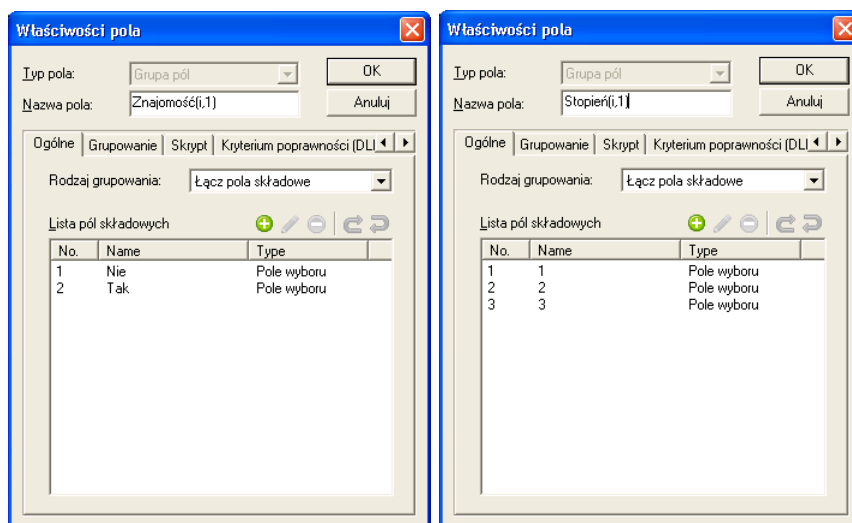
Grupa *Stopień* składa się z trzech pól wyboru o nazwach 1, 2, 3.



Rys. 8-9 Nazwy i struktury pól wyboru dla przykładu wyszukiwania automatycznego.

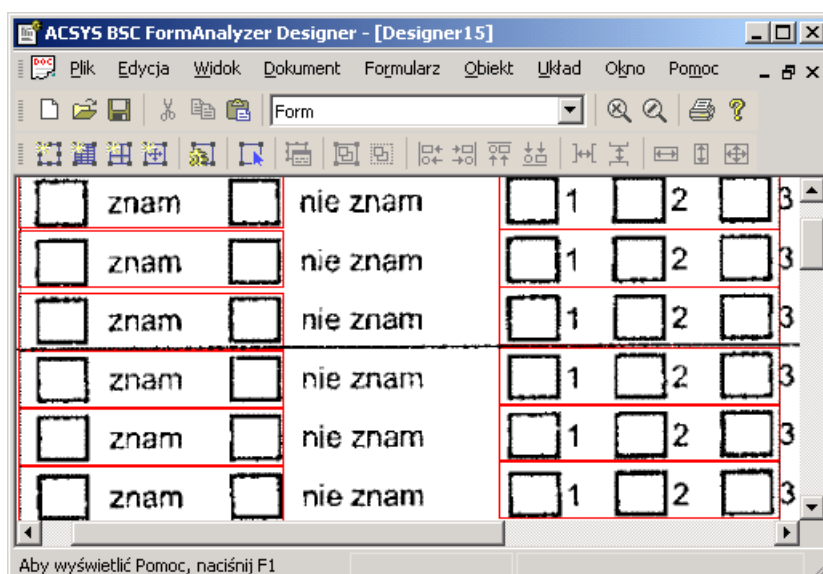
W omawianym przykładzie dla każdej grupy pól można było określić kryterium poprawności. W grupie pól *Znajomość* powinno być zaznaczone dokładnie jedno pole wyboru. W grupie pól *Stopień* może być zaznaczone najwyżej jedno pole wyboru. Na Rys. 8-10 przedstawiono okna właściwości obu grup.

Minimalna i maksymalna liczba zaznaczonych pól wyboru ma w obu przypadkach wartość 1.

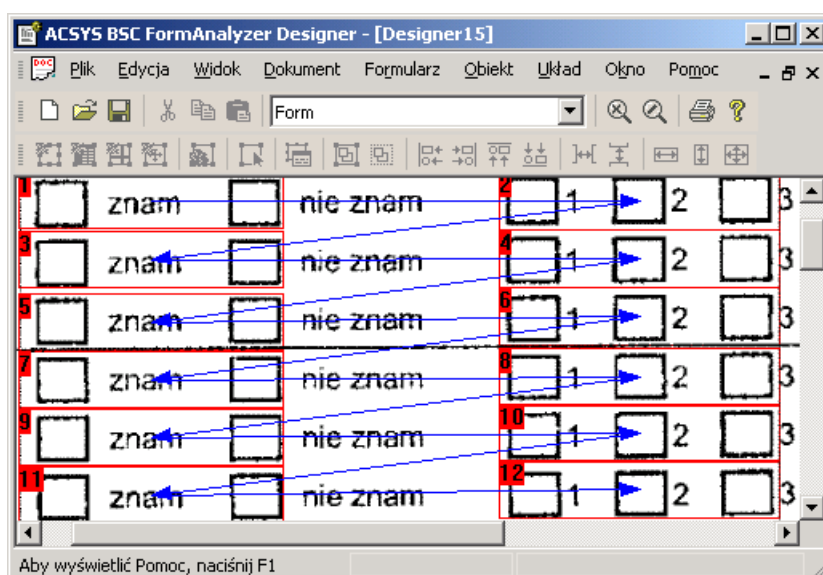


Rys. 8-10 Okna właściwości grup *Znajomość* i *Stopień*.

Efekt automatycznego wyszukiwania pól przedstawiono na rysunkach 8-10 i 8-11. Na Rys. 8-11 8-10 widać odszukane pola wyboru oraz utworzone automatycznie grupy pól. Na rysunku 8-11 po prawej stronie jest zaznaczona kolejność pól.



Rys. 8-11 Wynik automatycznego wyszukiwania pól wyboru (utworzone grupy).



Rys. 8-12 Wynik automatycznego wyszukiwania pól wyboru (kolejność ich rozpoznawania).

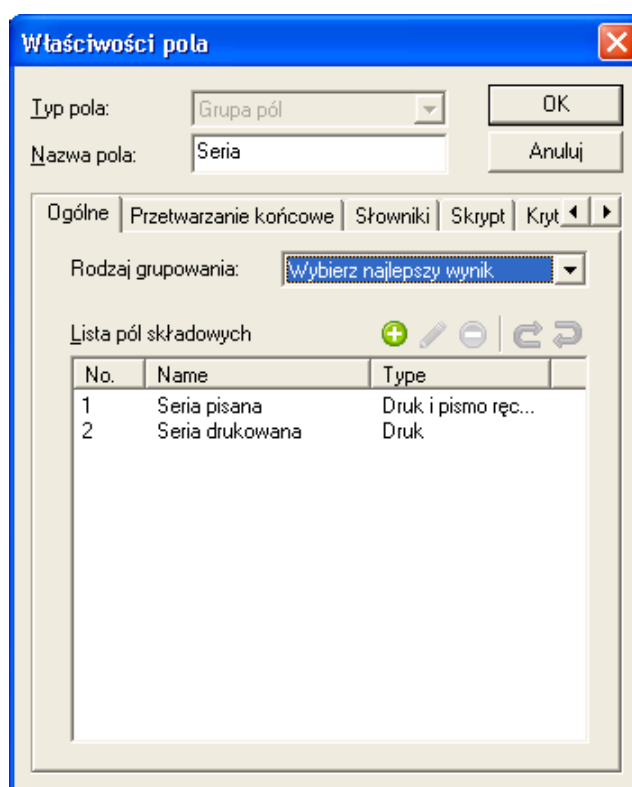
8.2. Grupy pól tekstowych

Grupy pól tekstowych są używane przede wszystkim do poprawienia jakości rozpoznawania treści dokumentu. W aktualnej wersji systemu FormAnalyzer cel ten można zrealizować korzystając z dwóch sposobów wykorzystania informacji rozpoznanych w polach składających się na grupę.

Pierwszy z tych sposobów to połączenie wartości pól składowych w jedną wartość. Taka operacja samoistnie nie zapewnia żadnej poprawy jakości rozpoznawania - jest to zwykła konkatenacja napisów (czasami ten tryb grupowania wykorzystuje się tylko w celu scalenia wyników rozpoznawania kilku pól). Poprawę jakości rozpoznawania w tym przypadku uzyskuje się dzięki możliwości wybrania różnych parametrów rozpoznawania poszczególnych pól składowych. Najczęściej spotykaną sytuacją jest wykorzystanie różnych leksykonów dla poszczególnych pól składowych grupy. Na przykład: cechy dowodu tożsamości są najczęściej wykorzystywane jako jedna wartość tekstowa. Jej rozpoznawanie jest najlepiej przeprowadzić rozpoznając oddzielnie serię (z leksykonem zawierającym jedynie duże litery) i numer (z leksykonem zawierającym cyfry).

W drugim trybie grupowania pól tekstowych jako wartość grupy jest wybierana wartość najlepiej rozpoznanego pola. Bardzo istotną sprawą są tu kryteria jakości rozpoznania pola. Z każdym polem rozpoznawanym jest związany poziom ufności. W dużym przybliżeniu jest to obliczone przez moduły rozpoznające, prawdopodobieństwo tego, że wynik rozpoznania jest wynikiem poprawnym. Jeżeli nie ma dodatkowych kryteriów poprawności dla wartości pola, to polem najlepiej rozpoznawanym jest pole o najwyższym poziomie ufności. Trzeba pamiętać, że poziom ufności jest wartością statystyczną. Zatem pole, którego poziom ufności jest wysoki, może być rozpoznane niepoprawnie. I przeciwnie, pole o niskim poziomie zaufania, może być rozpoznane poprawnie.

Poprawę jakości rozpoznania uzyskiwana jest poprzez zastosowanie różnych parametrów rozpoznawania w każdym z pól i wybranie tego pola, które daje (statystycznie) najlepsze wyniki. Najczęstsze wykorzystanie tej metody to rozpoznawanie pól, w których mogą pojawić się napisy drukowane lub pisane. W takim przypadku, na tym samym fragmencie dokumentu są definiowane dwa pola, z których jedno rozpoznaje tekst drukowany, a drugie tekst pisany.

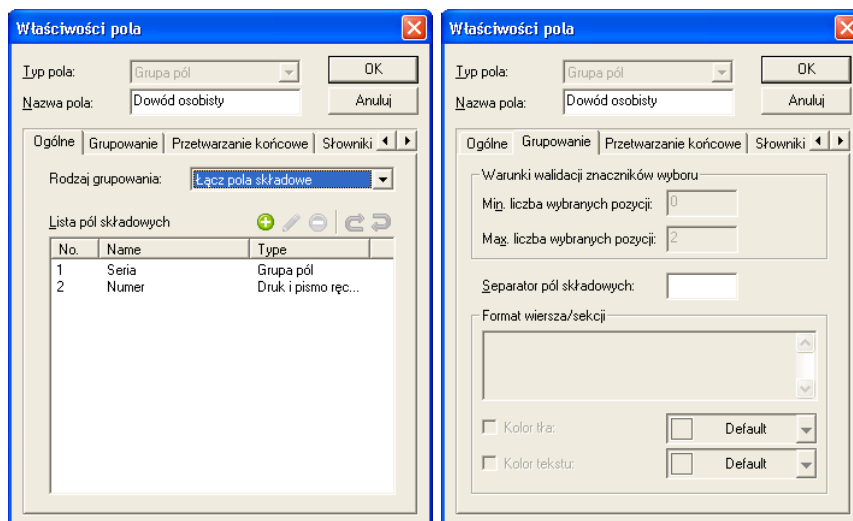


Rys. 8-13 Właściwości grupy pól tekstowych - wybór najlepiej rozpoznanego.

Podobnie jak w przypadku grupy pól wyboru, typ pola jest tu ustawiony na stałe. Nazwę pola grupowego podaje się zgodnie z omówionymi wcześniej zasadami nazewnictwa pól. W obszarze *Liście pól składowych* są wyświetlone pola składowe grupy. Dla każdego pola jest podany jego numer w grupie (gdy wybrano opcję łączenia, decydujący o kolejności łączenia pól), nazwa pola oraz jego typ.

W przykładzie została zaznaczona opcja *Wybierz najlepszy wynik* z dwóch pól: *Seria pisana* i *Seria drukowana*. Są to pola rozpoznające serię dowodu osobistego, odpowiednio przy pomocy klasyfikatora pisma ręcznego i pakietu rozpoznawania pisma drukowanego. Jako wartość grupy *Seria* zostanie wybrana wartość pola o większym poziomie zaufania. Pozwoli to na poprawne rozpoznanie serii dowodu osobistego, niezależnie od tego, czy zostanie ona napisana ręcznie czy wydrukowana.

Na poniższym rysunku pokazano przykładowe okno dialogowe Właściwości pola dla grupy pól tekstowych z łączeniem wartości pól. W oknie jest zaznaczona opcja *Łącz pola składowe*. Dodatkowo, przy łączeniu wartości, można podać separator, czyli tekst, który będzie wstawiany pomiędzy łączone wartości pól.



Rys. 8-14 Właściwości grupy pól tekstowych (łączenie pól).

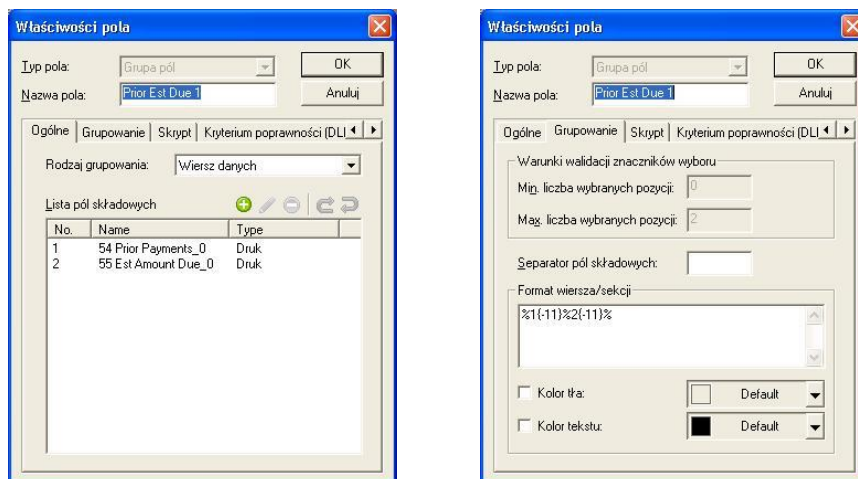
W przedstawionym przykładzie skorzystano z możliwości grupowania wielopoziomowego, tzn. możliwości użycia jako składnika grupy elementu, który sam może stanowić grupę. Na liście pól składowych znajduje się pole *Seria*, utworzona poprzednio grupa pól tekstowych z wyborem lepszego wyniku. Pole *Numer* jest rozpoznawane przy użyciu klasyfikatora cyfr pisanych i drukowanych, z wybranym leksykonem cyfr. Wartość grupy *Dowód osobisty* będzie się składała z liter (wartość grupy *Seria*) i następujących od razu za nimi, (ponieważ separator pozostał pusty) cyfr (wartość pola *Numer*).

Niezależnie od trybu grupowania (Łącz pola składowe lub Wybierz najlepszy wynik) określa się sposób przetwarzania grupy po rozpoznaniu. Okno Właściwości pola dla grupy pól tekstowych oprócz omówionej wyżej zakładki Ogólne posiada także zakładkę Przetwarzanie końcowe. Pozostałe zakładki: Słowniki, Skrypt i Kryterium poprawności (DLL) umożliwiają definiowanie sposobu korzystania ze słowników wartości, skryptów weryfikacji i parametrów przetwarzania końcowego. Ich działanie, jak i parametry, są takie same jak dla pola Druk i Druk i pismo ręczne. Zostały one opisane dokładnie w punkcie 7.8 - Pole Druk.

8.3. Wiersz i sekcja danych

Te metody grupowania mają na celu stworzenie formatki do procesu weryfikacji dokumentu. Dodatkowo każdy element nadrzędny może mieć zdefiniowane dodatkowe kryteria poprawności. Treść pola nadrzędnego nie jest eksportowana.

Włączona opcja *Wiersz danych* tworzy definicję jednej linii weryfikacji w module FormAnalyzer Verifier. Nazwa tej linii odpowiada nazwie zgrupowanego pola. Liczba edytowanych pól jest zgodna z definicją *Format wiersz/pola*. Użytkownik może dodatkowo zdefiniować kolor tła oraz kolor tekstu.

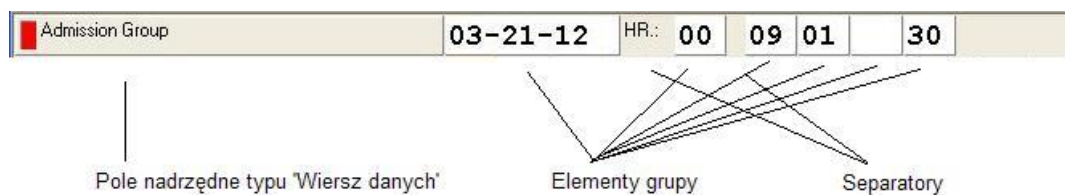
Rys. 8-15 Właściwości pola zgrupowanego metodą *Wiersz danych*

Grupowanie *Wiersz danych* charakteryzuje się następującymi cechami:

- wszystkie grupowane pola tworzą jeden edytowany wiersz w module FormAnalyzer Verifier, rozmiar pól grupowanych może być względny (proporcjonalny) lub bezwzględny wyrażony w liczbie znaków, pomiędzy edytowanymi polami mogą znajdować się separatory,
- każde grupowane pole jest edytowane niezależnie,
- wartość każdego grupowanego pola dostępna jest z poziomu kryteriów poprawności,
- każde grupowane pole eksportowane jest oddzielnie, pole zgrupowane nie jest dostępne w eksporcie danych,
- poprawność pola powstałego poprzez zgrupowanie *Wiersz danych* wynika ze zdefiniowanych kryteriów poprawności dla tego pola oraz ze stanu pól wchodzących w skład tego pola (np. aby pole było prawidłowe wszystkie pola wchodzące w jego skład muszą być prawidłowe),
- pole zgrupowane *Wiersz danych* jest filtrowane podczas weryfikacji, gdy wszystkie pola wchodzące w jego skład mogą być filtrowane (np. filtracja pól pustych następuje wtedy gdy wszystkie pola w grupie są puste),
- pola wchodzące w skład grupy nie podlegają sortowaniu przy wyświetlaniu w module FormAnalyzer Verifier,

Dla tej formy grupowania dostępne są następujące definiowalne parametry dostępne na zakładce *Grupowanie*:

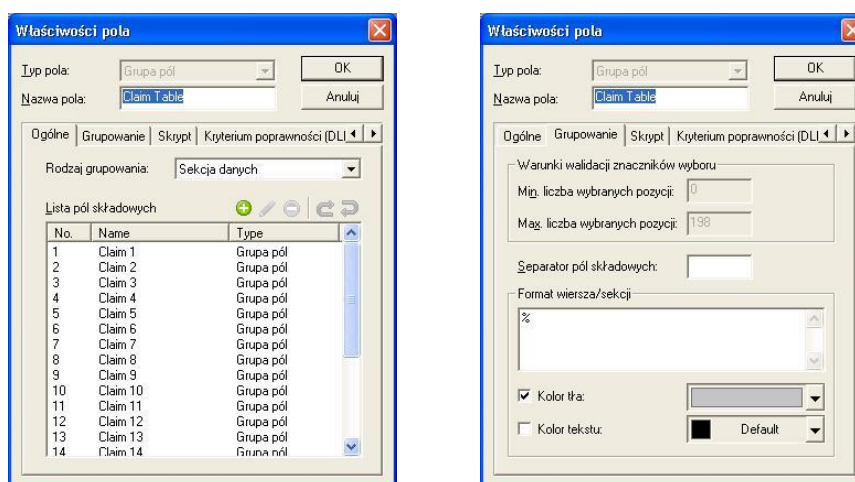
- *Separator pól składowych* – ciąg znaków wykorzystywany do tworzenia wartości pola zgrupowanego. Wartość dostępna jest jedynie z poziomu kryteriów poprawności.
- *Format wiersza/sekcji* – ciąg znaków definiujący formatkę w module FormAnalyzer Verifier np. %1{-10}% "HR.:{-3}%2{-3}% " "{-1}%3{-3}%4{-3}%5{-3}%6{-3}% gdzie:



Rys. 8-16 Linia edycyjna w procesie weryfikacji o formacie:
%1{-10}% "HR.:"{-3}%2{-3}% " " {-1}%3{-3}%4{-3}%5{-3}%6{-3}%

- % oznacza zarezerwowany obszar w linii edycyjnej
 - Cyfra n po znaku % oznacza, że jest to n-te edytowane pole w grupie (indeks odnosi się do numeru pola z pola *Lista pól składowych*)
 - dowolny ciąg znaków w podwójnych cudzysłowach "Abc" występujący po znaku % oznacza nieedytowany element – separator zawierający tekst
 - jeżeli po znaku % nie występuje cyfra ani ciąg znaków oznacza to nieedytowany element – separator nie zawierający tekstu
 - wartość liczbowa w nawiasach klamrowych oznacza szerokość elementu. Wartość mniejsza od zera oznacza procentową szerokość elementu. Wartość większa od zera oznacza szerokość elementu wyrażoną w liczbie znaków
 - jeżeli dla obszaru nie zdefiniowano szerokości, oznacza to, że szerokość będzie taka jaka szerokość wolnej przestrzeni w edytowanej linii
 - minimalna szerokość elementu edycyjnego wynosi jeden znak
 - jeżeli wyliczona szerokość pól jest zbyt duża, to szerokości elementów składowych jest jednakowa – proporcjonalna w stosunku do dostępnego obszaru w module FormAnalyzer Verifier
- *Kolor tła* – kolor tła elementów nieedytowanych – separatorów
 - *Kolor tekstu* – kolor tekstu elementów nieedytowanych – separatorów

Włączona opcja *Sekcja danych* tworzy definicję sekcji weryfikacji w module FormAnalyzer Verifier, w skład której wchodzi linia nagłówka sekcji oraz tyle linii edycyjnych ile jest pól składowych. Nazwa sekcji odpowiada nazwie zgrupowanego pola.



Rys. 8-17 Właściwości pola zgrupowanego metodą *Sekcja danych*

Grupowanie *Sekcja danych* charakteryzuje się następującymi cechami:

- wszystkie grupowane pola tworzą jedną sekcję danych – są zagłębione o jeden poziom i są poprzedzone stałym elementem oznaczającym początek sekcji. Wszystkie pola wchodzące w skład grupy są wyświetlane jako nowe wiersze w procesie weryfikacji
- każde grupowane pole jest edytowane niezależnie
- wartość każdego grupowanego pola dostępna jest z poziomu kryteriów poprawności
- każde grupowane pole eksportowane jest oddzielnie, pole zgrupowane nie jest dostępne w eksporcie danych
- poprawność pola powstałego poprzez zgrupowanie *Sekcja danych* wynika ze zdefiniowanych kryteriów poprawności dla tego pola oraz ze stanu pól wchodzących w skład tego pola (np. aby pole było prawidłowe wszystkie pola wchodzące w jego skład muszą być prawidłowe)
- pole zgrupowane *Sekcja danych* jest filtrowane podczas weryfikacji, gdy wszystkie pola wchodzące w jego skład mogą być filtrowane
- pola wchodzące w skład grupy nie podlegają sortowaniu w module FormAnalyze Verifier

Dla tej formy grupowania dostępne są następujące definiowalne parametry z zakładki *Grupowanie*:

- *Separator pól składowych* – ciąg znaków wykorzystywany do tworzenia wartości pola zgrupowanego. Wartość dostępna jest jedynie z poziomu kryteriów poprawności.
- *Format wiersza/sekcji* – ciąg znaków definiujący wygląd pierwszego wiersza sekcji w module FormAnalyze Verifier np. %, gdzie:

Value Codes		
39a Code Amount	80	0.11
39b Code Amount		
39c Code Amount		
39d Code Amount		
40a Code Amount		
40b Code Amount		
40c Code Amount		
40d Code Amount		
41a Code Amount		
41b Code Amount		

Początek sekcji danych

Elementy wchodzące w skład grupy

Zagłębienie sekcji danych

Rys. 8-18 Przykładowa sekcja danych w procesie weryfikacji

- % oznacza zarezerwowany obszar w wierszu,
 - dowolny ciąg znaków w podwójnych cudzysłowach "Abc" występujący po znaku % oznacza to nieedytowany element – separator zawierający tekst
 - jeżeli po znaku % nie występuje ciąg znaków oznacza to nieedytowany element – separator nie zawierający tekstu
 - wartość liczbową w nawiasach klamrowych oznacza szerokość elementu. Wartość większa od zera oznacza procentową szerokość elementu. Wartość mniejsza od zera oznacza szerokość elementu wyrażoną w liczbie znaków
- *Kolor tła* – kolor tła elementów nieedytowanych – separatorów
 - *Kolor tekstu* – kolor tekstu elementów nieedytowanych – separatorów

Przykład definicji sekcji i wiersza danych z Rys. 8-19:

Nazwa pola	Typ pola	Format wiersza/sekcji
Osoba Ubezpieczana	Pole zgrupowane – sekcja danych	%
Nazwisko	Pole tekstowe	
Imiona	Pole zgrupowane – wiersz danych	%1%{1}%2
Płeć	Pole zgrupowane – łącz pola składowe	
Adres – miejscowość	Pole zgrupowane – wiersz danych	%2%"Kod pocztowy"{6}%1{7}
Adres - ulica	Pole zgrupowane – wiersz danych	%1%"Nr.domu"{3}%2{3}% "Nr.mieszkania"{4}%3{3}

Osoba Ubezpieczana	
Nazwisko	JACKOWSKA
Imiona	ZOFIA HELENA
Płeć	X
Adres - miejscowość	WIELKA WIEŚ
Adres - ulica	GŁÓWNA
Kod pocztowy	49-102
Nr. domu	121
Nr. mieszkania	

Rys. 8-19 Przykładowa sekcja danych Osoba Ubezpieczana

Przykład definicji sekcji i wiersza danych z Rys. 8-20:

Nazwa pola	Typ pola	Format wiersza/sekcji
Zamówienie	Pole zgrupowane – sekcja danych	%%"Ilość"{3}%"Cena netto"{8}%"VAT"{3}%"Wartość netto"{9}%"Wartość VAT"{8}%"Wartość brutto"{9}
Wiersz1	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz2	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz3	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz4	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz5	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz6	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz7	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Wiersz8	Pole zgrupowane – wiersz danych	%1%2{3}%3{8}%4{3}%5{9}%6{8}%7{9}
Razem	Pole zgrupowane – wiersz danych	%%1{9}%2{8}%3{9}

Dok. 722671 (I poziom weryfikacji) - ACSYS BSC FormAnalyze Verifier

Edycja Dokument Strona Widok Pogoc

Faktury

Strona

Lp	Nazwa	PKWiU	j.m.	Ilość	Cena netto	Rabat [%]	VAT [%]	Wartość netto	VAT	Wartość brutto
1	EPSON TUSZ STYLUS PHOTO 950 LIGHT MAGENTA T0336		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26
2	EPSON TUSZ STYLUS PHOTO 950 YELLOW T0334		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26
3	EPSON TUSZ STYLUS PHOTO 950 MAGENTA T0333		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26
4	EPSON TUSZ STYLUS PHOTO 950 CYAN T0332		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26
5	EPSON TUSZ STYLUS PHOTO 950 BLACK T0331		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26
6	EPSON TUSZ STYLUS PHOTO LIGHT CYAN		szt.	3,00	61,00	0,00	22	183,00	40,26	223,26

według stawki VAT	wartość netto	kwota VAT	wartość brutto
Podstawowy podatek VAT 22%	1 098,00	241,56	1 339,56
Razem:	1 098,00	241,56	1 339,56

Zamówienie	Ilość	Cena netto	VAT	Wartość netto	Wartość VAT	Wartość brutto
Wiersz1 EPSON TUSZ STYLUS PHOTO 950 LIGHT MAGENTA T0336	3	61.00	22	183.00	40.26	223.26
Wiersz2 EPSON TUSZ STYLUS PHOTO 950 YELLOW T0334	3	61.00	22	183.00	40.26	223.26
Wiersz3 EPSON TUSZ STYLUS PHOTO 950 LIGHT MAGENTA T0333	3	61.00	22	183.00	40.26	223.26
Wiersz4 EPSON TUSZ STYLUS PHOTO 950 CYAN T0332	3	61.00	22	183.00	40.26	223.26
Wiersz5 EPSON TUSZ STYLUS PHOTO 950 BLACK T0331	3	61.00	22	183.00	40.26	223.26
Wiersz6 EPSON TUSZ STYLUS PHOTO LIGHT CYAN	3	61.00	22	183.00	40.26	223.26
Razem				1098.00	241.56	1339.56

Folder: \$(Dokumentacja) Szablon: Faktury Strona 1 z 1 Czas: 00:34

Rys. 8-20 Przykładowa sekcja danych Zamówienie

9. Właściwości pól

Dodatkowym oknem dialogowym służącym do definicji cech pól jest okno wybierane poleceniem *Pola* z menu *Dokument*. W oknie tym użytkownik może modyfikować właściwości pola opisane w rozdziale 7 - Pola rozpoznawane, usuwać pola oraz może definiować dodatkowe cechy wykorzystywane w procesie rozpoznawania, weryfikacji i eksportu.

W oknie dialogowym *Właściwości pól* istnieje możliwość tworzenia pól indeksowych (nierozpoznawanych).

Dodatkowo użytkownik może wyświetlić i zmodyfikować właściwości pól zgrupowanych.

Okno *Właściwości pól* uruchamiane jest poleceniem *Dokument*→*Pola*. W liście *Szablon dokumentu* są dostępne nazwy wszystkich szablonów zdefiniowanych w pliku konfiguracyjnym. Po otwarciu okna zaznaczony jest szablon, w którego skład wchodzi aktualnie edytowany formularz w głównym oknie programu. Dla wybranego szablonu w liście wyświetlane są pola związane z wybranym szablonem, których typ jest zgodny z wybranym filtrem w lewej części okna. Dostępne są następujące filtry:

- *Wszystkie pola* – wyświetlane są wszystkie pola bez względu na typ pola. Pola nadrzędne i zgrupowane w trybie *Wiersz danych/Sekcja danych* wyświetlane są w zależności od rodzaju opcji np. w opcjach eksportu są dostępne pola zgrupowane, a w opcjach weryfikacji pola nadrzędne.
- *Pola wewnętrzne* – wyświetlane są pola zgrupowane, co umożliwia modyfikację właściwości pól bez konieczności rozgrupowania.
- *Pola indeksowe* – wyświetlane są pola niepowiązane ze stronami dokumentu, można wykorzystać w procesie klasyfikacji i weryfikacji.
- *Pismo ręczne* – wyświetlane są pola o typie *RecoStar Engine*, *Druk* i *pismo ręczne* oraz pola zgrupowane składające się z pól o typach wymienionych wcześniej.
- *Pismo drukowane* – wyświetlane są pola o typie *Druk* oraz pola zgrupowane składające się z pól o typie *Druk*.
- *Kod paskowy* – wyświetlane są pola o typie *Kod paskowy*.
- *Pola wyboru* – wyświetlane są pola wyboru oraz zgrupowane pola wyboru.

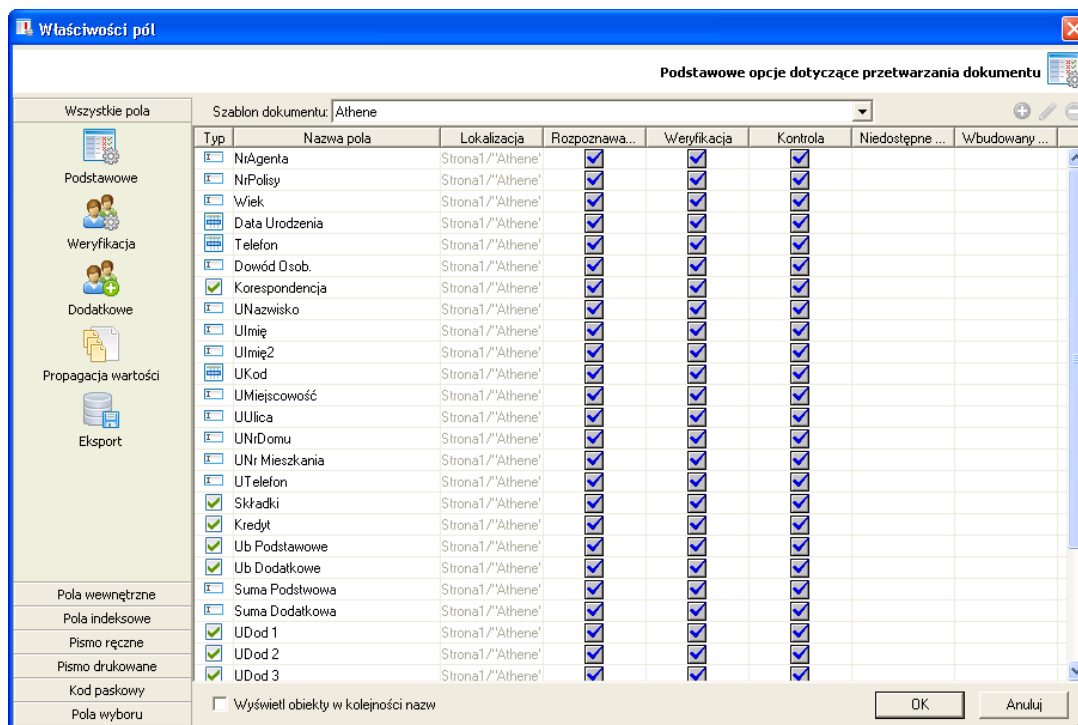
Przy włączonej opcji *Wyświetl obiekty w kolejności nazw* wyświetlane pola posortowane są nazwach. W przeciwnym przypadku wykorzystywana jest kolejność weryfikacji. Wyjątkiem jest wyświetlenie opcji eksportu podczas której wykorzystywana jest kolejność eksportu.

Każdemu typowi pola przyporządkowana jest ikona symbolizująca ten typ:

-  - pola tekstowe
-  - pola wyboru
-  - pola kod paskowy

-  - pola obrazu
-  - pola zgrupowane *Wiersz danych*
-  - pola zgrupowane *Sekcja danych*

Rozmiar okna można zmienić, poprzez ciągnięcie prawego-dolnego rogu okna. Rozmiar jest zapisany w rejestrze systemu Windows. Przy kolejnym wyświetleniu tego okna rozmiar jest odtwarzany.



Rys. 9-1 Okno Właściwości pól.

Przycisk  umożliwia dodanie nowego pola indeksowego. Funkcja ta jest dostępna tylko w przypadku gdy wybrany jest filtr *Pola indeksowe*. Funkcja ta została opisana w rozdziale 9.6 - Pola indeksowe.

Modyfikację właściwości pól umożliwia przycisk . Te zmiany są oknie dialogowym opisanym w rozdziale 7 - Pola rozpoznawane.

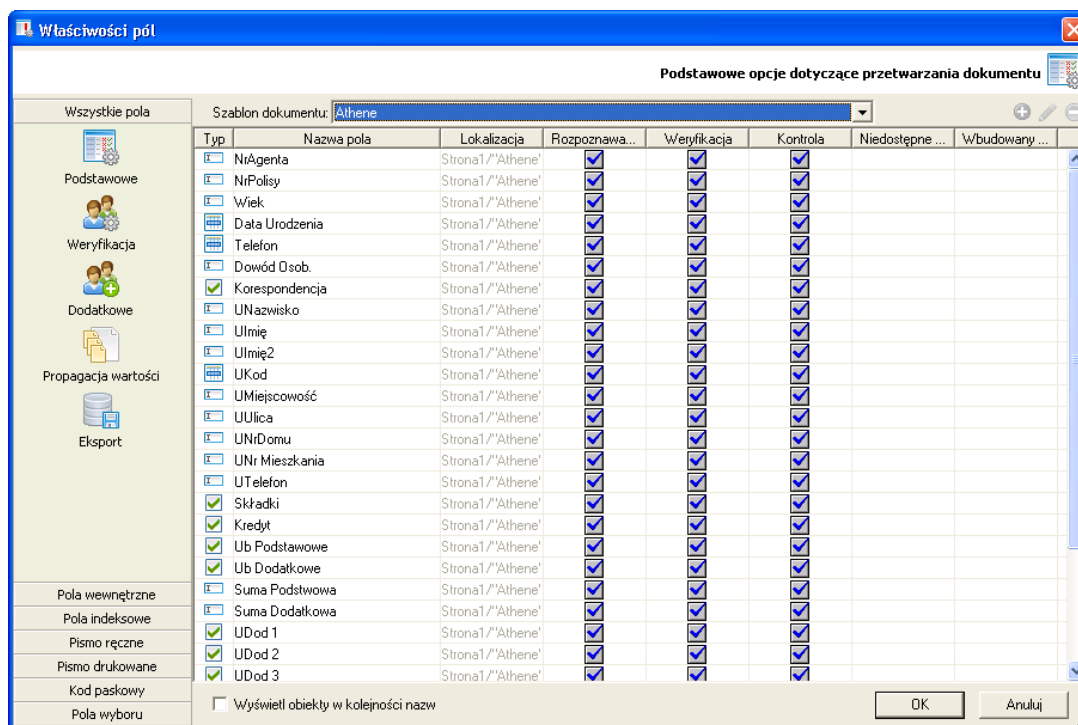
Przycisk  służy do usunięcia zaznaczonego pola. Funkcja ta nie jest dostępna w przypadku gdy wybrany jest filtr *Pola wewnętrzne*. Pole jest usuwane z listy natychmiast, jednak nie jest od razu usuwane z pliku konfiguracji. Dopiero naciśnięcie  i zamknięcie okna właściwości pola, powoduje nieodwracalne usunięcie pola. Zamknięcie okna pól przyciskiem  powoduje, że jest przywracany stan listy pól sprzed otwarcia okna.

9.1. Opcje podstawowe

Najczęściej wszystkie zdefiniowane w szablonie dokumentu pola są przetwarzane w taki sam sposób: na początku jest rozpoznawana ich treść, następnie jest weryfikacja, a na koniec jest opcjonalna kontrola. Można wyobrazić sobie sytuację, że część pól została zdefiniowana tylko do rozpoznawania, a inne pola tylko do weryfikacji. Aby można było

stworzyć tego rodzaju konfigurację, zostały wprowadzone opcje zgrupowane w grupie *Podstawowe*.

Na rysunku poniżej przedstawiono okno opcji podstawowych dla szablonu Athene.



Rys. 9-2 Okno opcji podstawowych z wyświetloną listą pól szablonu Athene.

Dla każdego pola z listy zostaną podane w kolejnych kolumnach następujące parametry:

- Typ pola w postaci piktoqramu
- Nazwa pola
- Strona i typ formularza, na którym pole się znajduje.
- Rozpoznawanie – ta opcja jest opcją wykorzystywaną przez FormAnalyzer Engine. Jeśli nie jest ustawiona, FormAnalyzer Engine pominie rozpoznawanie pola, jednak skrypty lub funkcje z bibliotek DLL będą mogły ustawić jego wartość. Ta opcja jest wyłączana dla pól, których wartość jest wyliczana automatycznie przez kryteria poprawności lub nie da się rozpoznawać automatycznie (np. podpis ubezpieczonego na polisie).
- Weryfikacja – wyłączenie tej opcji powoduje, że pole nie jest wyświetlane w liście pól do weryfikacji. Stosuje się ją najczęściej dla pól, których wartość jest wyznaczana automatycznie i nie wymaga interwencji operatora. Użytkownik poprawiający treść dokumentu może jednak zobaczyć i zmodyfikować wartości pól, dla których ta opcja jest wyłączona. W tym celu musi włączyć pokazywanie wszystkich pól w module FormAnalyzer Verifier.
- Kontrola – wyłączenie tej opcji powoduje, że pole nie jest wyświetlane w liście pól do kontroli.

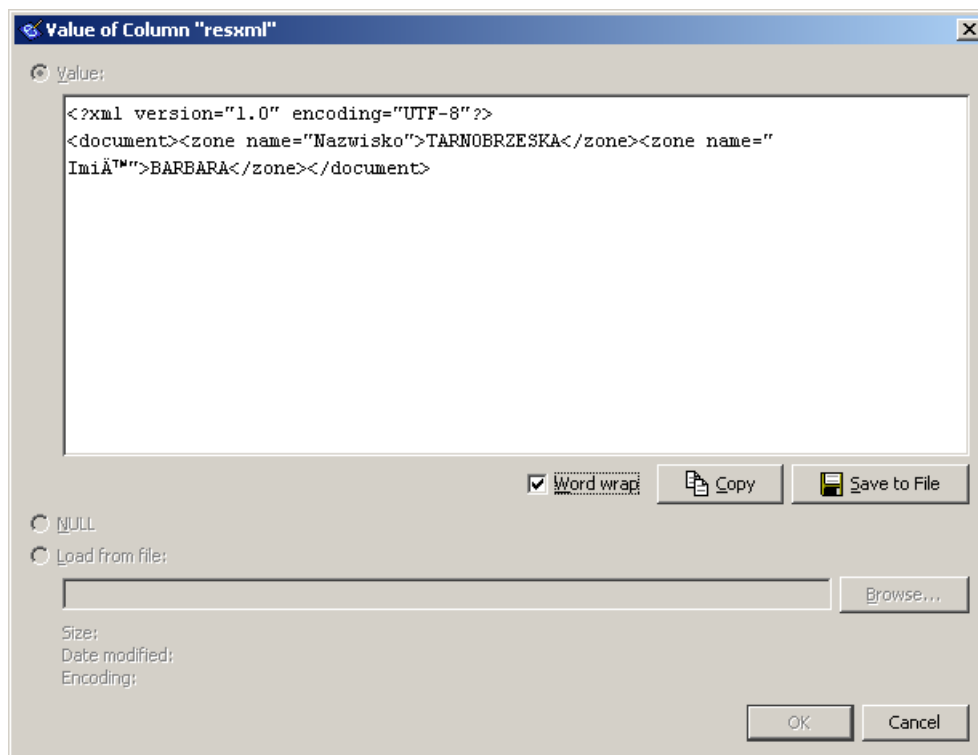
- Niedostępne dla operatora – włączenie tej opcji powoduje, że pole nie jest dostępne w procesie poprawiania wartości nawet po włączeniu pokazywania wszystkich pól w module FormAnalyzer Verifier.
- Wbudowany XML – włączenie tej opcji powoduje, że wartość tego pola będzie zapamiętywana w formie pliku XML, w tabeli *Results*, w kolumnie resXML. Pozwala to na uzyskanie dostępu do określonych informacji przechowywanych w bazie danych FormAnalyzer przy pomocy aplikacji zewnętrznej i zarządzanie nimi zgodnie ze zdefiniowanymi ustawieniami.

Zawartość kolumny resXML zawierającej wartości w formacie XML, pól wskazanych w oknie dialogowym *Właściwości pól* prezentuje poniższy obrazek. Do celów prezentacji wykorzystano program ISQL firmy Sybase.

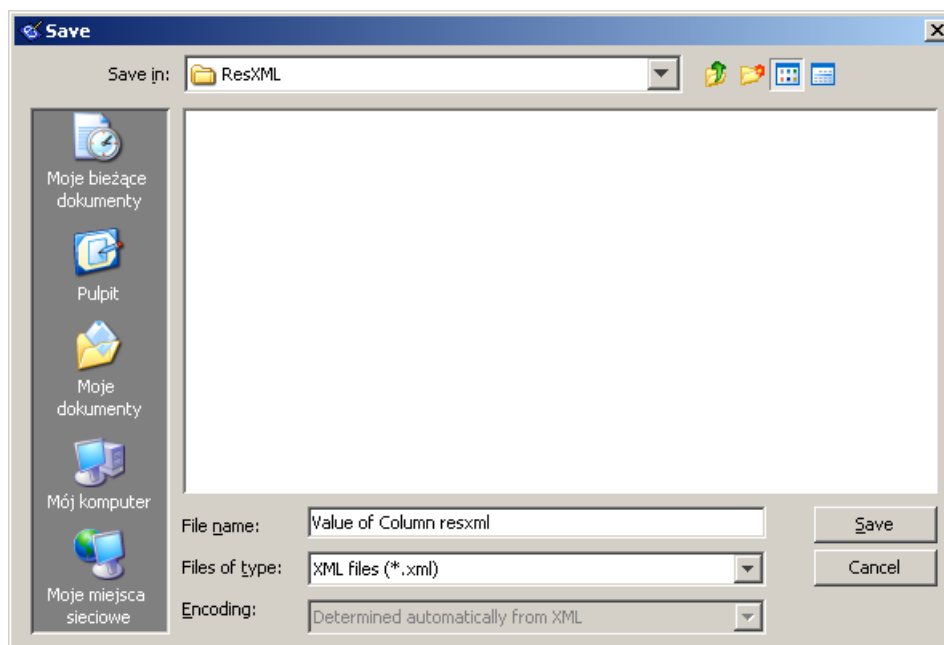
obid	res	resxml
1	25,086 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">JAMIL</zone><zone name="Middle Name"></zone><zone name="Last Name">SHMAINQU...
2	25,104 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">JAMIL</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
3	25,105 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
4	25,106 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
5	25,107 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
6	25,108 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
7	25,109 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
8	25,110 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
9	25,111 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
10	25,112 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
11	25,113 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
12	25,114 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
13	25,115 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
14	25,116 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
15	25,117 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
16	25,118 0x5244ffffffffff0100010000...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="First Name">KATHY</zone><zone name="Middle Name"></zone><zone name="Last Name">DAUGHER...
17	25,187 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">KWIAKOWSKI</zone><zone name="ImiA">ARTUR</zone></document>
18	25,190 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">ABAKANOWICZ</zone><zone name="ImiA">JAROSŁAW</zone></document>
19	25,191 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">TARNOWSKA</zone><zone name="ImiA">BARBARA</zone></document>
20	25,189 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">ZACORSKI</zone><zone name="ImiA">KRZYSZTOF</zone></document>
21	25,192 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">NIVHAIICID</zone><zone name="ImiA">~</zone></document>
22	25,193 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">~</zone><zone name="ImiA">~</zone></document>
23	25,186 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">SWIERADOWSKI</zone><zone name="ImiA">GUSTAW</zone></document>
24	25,186 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">JACKOWSKA</zone><zone name="ImiA">ZOFIA</zone></document>
25	25,188 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">MIZEROWSKI</zone><zone name="ImiA">MARIAN</zone></document>
26	25,194 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">ŁWIERZY</zone><zone name="ImiA">HERMAN</zone></document>
27	25,195 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">LESZYŃSKA</zone><zone name="ImiA">JADWIGA</zone></document>
28	25,196 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">NOWAKOWSKI</zone><zone name="ImiA">KAZIMIERZ</zone></document>
29	25,197 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">KOWALSKI</zone><zone name="ImiA">JANUSZ</zone></document>
30	25,198 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">POLKOWSKI</zone><zone name="ImiA">GRZEGORZ</zone></document>
31	25,199 0x524403000000000100020...	<?xml version="1.0" encoding="UTF-8"?><document><zone name="Nazwisko">NOWACKA</zone><zone name="ImiA">AGNIESZKA</zone></document>

Rys. 9-3 Dane w formacie XML przechowywane w tabeli Results w kolumnie resxml

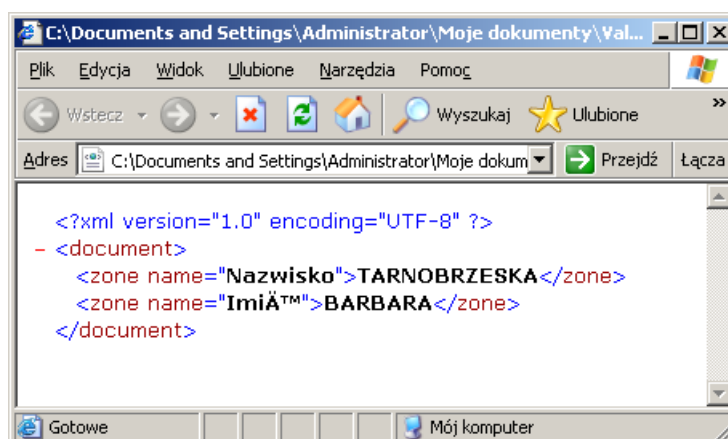
Użytkownik może wyświetlić zawartość każdego z wierszy, uprzednio klikając na przycisk znajdujący się na końcu takiego wiersza.

**Rys. 9-4 Okno prezentujące zawartość jednego z wierszy**

Przy użyciu poleceń  **Copy** użytkownik może skopiować zawartość wiersza znajdującą się w obszarze *Value*. Przycisk  **Save to File** służy do zapisywania zawartości wiersza do pliku XML. Kliknięcie tego przycisku spowoduje wyświetlenie standardowego okna zapisywania pliku systemu Windows:

**Rys. 9-5 Okno dialogowe umożliwiające zapisanie zawartości wiersza do pliku XML.**

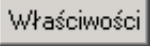
Poniższy obrazek prezentuje zapisany plik XML, otwarty przy użyciu aplikacji Windows Explorer.



Rys. 9-6 Struktura pliku zapisanego przy użyciu funkcji Save to File

Każdorazowy proces rozpoznawania lub weryfikacji automatycznie aktualizować będzie zawartość danych w tabeli *Results*, w kolumnie *resXML*.

Przechowywanie dużej informacji w formie XML może w znacznym stopniu zwiększać objętość bazy danych, co może wpływać negatywnie na wydajność pracy z taką bazą danych.

Poza wskazywaniem pól, których zawartość wyświetlana będzie w formie plików XML, w zakładce *Pola wbudowanego XML* w oknie dialogowym *Pola szablonu dokumentów* użytkownik ma możliwość wyświetlić okno właściwości zaznaczonego na liście pola, klikając przycisk .

Wykorzystując w/w funkcjonalność można stworzyć program prezentujący zawartość pól dokumentu lub eksportujący zawartość dokumentu do systemu zewnętrznego.

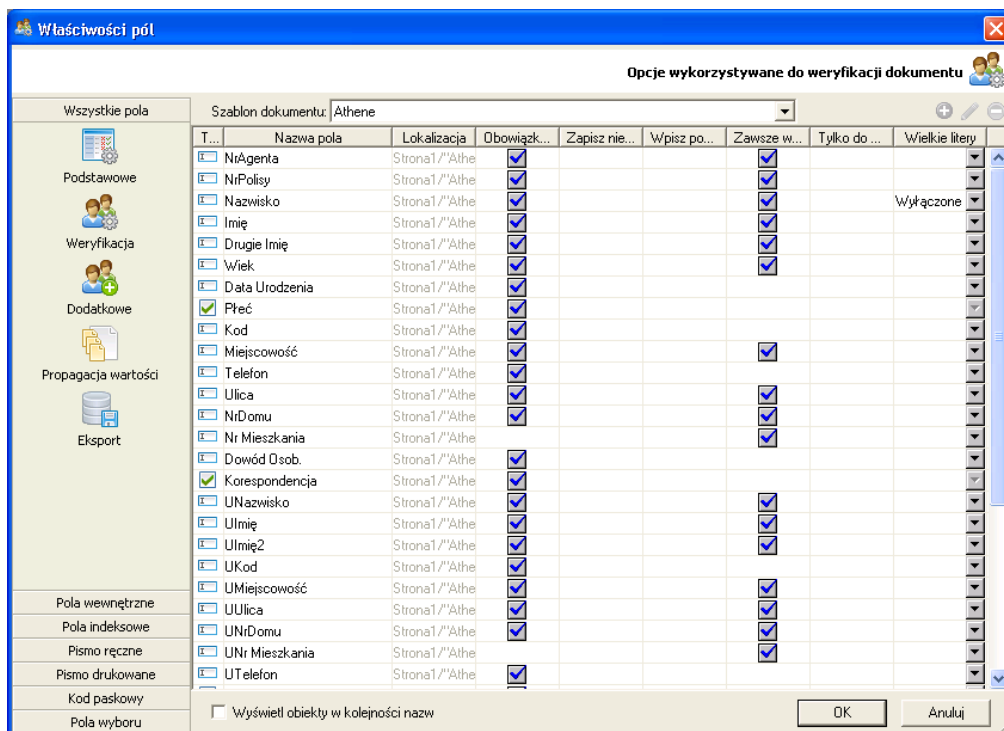
9.2. Opcje weryfikacji

Podstawowym zadaniem FormAnalyzer Designer jest zaprojektowanie sposobu rozpoznawania formularza. Oprócz przygotowania „instrukcji” dla FormAnalyzer Engine, który będzie rozpoznawał dokumenty, bardzo ważna jest pomoc, jakiej można udzielić operatorom weryfikacji, dokonującym sprawdzenia (i poprawienia) wyników automatycznego rozpoznania.

Chodzi o to by system automatycznie wykrywał sytuacje niepoprawne w dokumencie i sygnalizował je operatorom. W poprzednim rozdziale przedstawiono wiele przykładów kryteriów poprawności, które sygnalizują operatorom weryfikacji niezgodność danych z założeniami, które powinny spełniać w poprawnie wypełnionym dokumencie. Kryteria poprawności nie są jedynym, ani podstawowym sposobem ułatwienia pracy operatorom (oraz wymuszenia na nich odpowiednio dokładnej weryfikacji treści dokumentu).

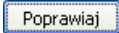
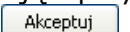
Ustawienia związane z weryfikacją dokumentów dostępne są w grupie opcji *Weryfikacja i Dodatkowe*.

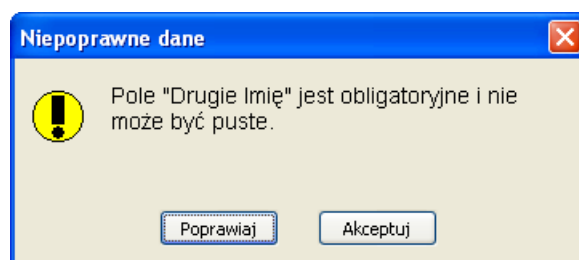
Na rysunku poniżej przedstawiono okno opcji weryfikacji dla szablonu Athene.



Rys. 9-7 Okno opcji weryfikacji z wyświetloną listą pól szablonu Athene.

Dla każdego pola z listy zostaną podane w kolejnych kolumnach następujące parametry:

- Typ pola w postaci piktogramu
- Nazwa pola
- Strona i typ formularza, na którym pole się znajduje.
- Obowiązkowe – ta opcja jest domyślnie ustawiona dla wszystkich pól. Oznacza ona, że w poprawnym dokumencie pole nie może być puste. W sytuacji, gdy bardzo rzadko zdarzają się takie dokumenty, w których wszystkie pola są wypełnione, przypisanie tej opcji do wszystkich pól dokumentu może spowodować uniemożliwienie zatwierdzenia poprawności dokumentów. Trzeba dokładnie przeanalizować treść dokumentu i wyłączyć opcję Obowiązkowe, dla tych pól, które muszą być wypełniane. Np. w szablonie *Athene* obowiązkowe są: *Nazwisko* i (pierwsze) *Imię*, a nieobowiązkowe *Drugie imię*.
- Zapisz niepoprawne – ta opcja pozwala na zapisanie pola (a zatem i zatwierdzenie dokumentu) nawet, jeśli nie spełnia ono kryteriów poprawności. W przypadku wykrycia takiego pola operatorowi weryfikacji jest wyświetlany komunikat, którego przykład pokazano na rysunku poniżej. Operator może powrócić do edycji pola klikając przycisk , lub zatwierdzić niepoprawną wartość klikając przycisk .



Rys. 9-8 Komunikat wyświetlany dla niewypełnionego pola obowiązkowego, dla którego zaznaczono opcję Zapisz niepoprawne.

- Wpisz ponownie – ustawienie tej opcji powoduje, że operator weryfikacji po poprawieniu i zatwierdzeniu treści pola rozpoznanej automatycznie musi ponownie wprowadzić, odczytując ją z fragmentu obrazu dokumentu. Po drugim zatwierdzeniu obydwie wersje treści są ze sobą porównywane. Jeśli są zgodne pole jest zatwierdzone. Jeżeli się nie zgadzają, procedura jest powtarzana. Ta opcja jest stosowana dla pól, których poprawność ma kluczowe znaczenie dla całego dokumentu. Trzeba pamiętać, że jest stosowanie znacznie spowalnia procedurę weryfikacji przez operatorów.
- Zawsze widoczne – włączenie tej opcji powoduje, że pole takie nie podlega opcji filtracji pól (ukrywaniu) w module FormAnalyze Verifier. Najczęściej opcję tą stosuje się do pól szczególnie istotnych, które zawsze muszą być sprawdzone przez użytkownika na etapie weryfikacji pomimo dobrych wyników rozpoznawania.
- Tylko do odczytu – włączenie tej opcji powoduje, że wartość tego pola nie jest możliwa do zmiany przez użytkownika weryfikacji. Wartość takich pól można zmodyfikować z poziomu skryptów modyfikujących wartości pola w module FormAnalyze Engine i FormAnalyze Verifier. Stosuje się ją najczęściej dla pól, których wartość jest obliczana lub pobierana automatycznie z bazy danych.

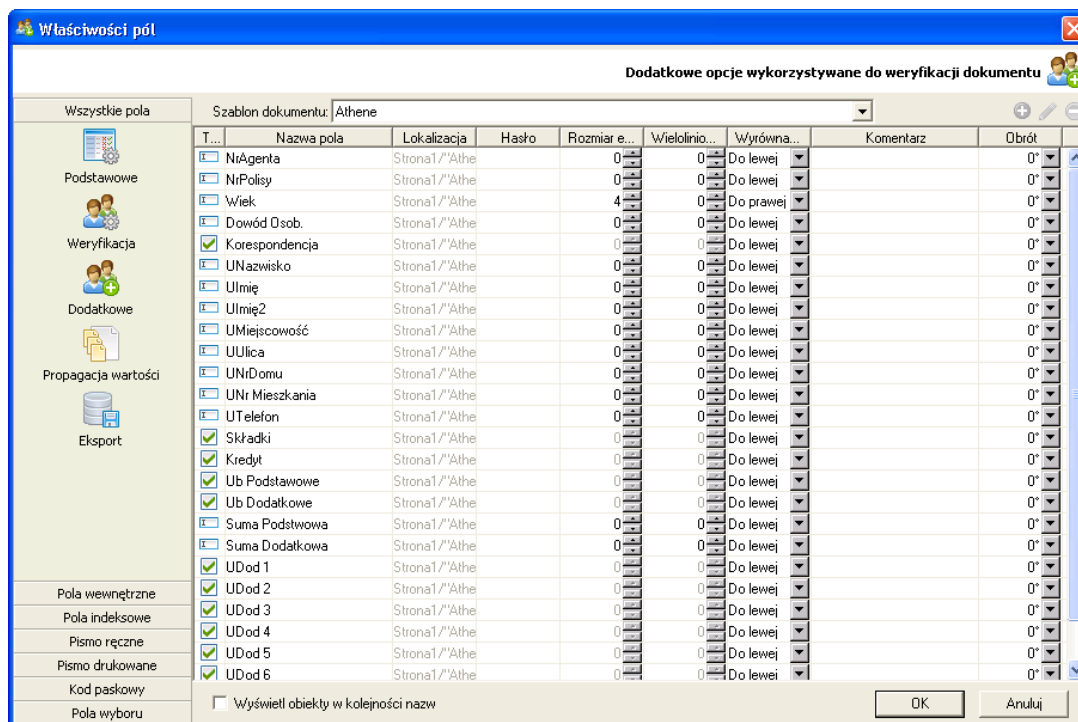
NrAgenta	97123111
NrPolisy	98042594
Nazwisko	JACKOWSKA
Imię	ZOFIA
Drugie Imię	HELENA
Wiek	36

Rys. 9-9 Fragment okna FormAnalyze Verifier z polami tylko do odczytu.

- Wielkie litery – opcja ta wymusza wpisywania małych lub wielkich znaków w polach tekstowych w module FormAnalyzer Verifier. Jeżeli wartość tej opcji nie została ustawiona (wartość pusta), wpisywanie małych/wielkich znaków będzie uzależniona od włączenia opcji 'Używaj tylko wielkich liter w polach weryfikacji' w module FormAnalyzer Verifier.

9.3. Dodatkowe opcje weryfikacji

Grupie dodatkowych opcji zdefiniowano opcje wykorzystywane na etapie weryfikacji, które powiązane są z wizualizacją danych. Mają one pomóc osobie sprawdzającej dane, poprawnie wpisać treść dokumentu.



Rys. 9-10 Okno opcji dodatkowych z wyświetloną listą pól szablonu Athene.

Dla każdego pola z listy zostaną podane w kolejnych kolumnach następujące parametry:

- Typ pola w postaci piktoqramu
- Nazwa pola
- Strona i typ formularza, na którym pole się znajduje.
- Hasło – włączenie tej opcji powoduje, że treść wprowadzana przez operatora jest wyświetlana w oknie weryfikacji za pomocą symboli gwiazdek (*) – podobnie, jak przy wpisywaniu hasła przy logowaniu się do systemu Windows).
- Rozmiar edytora – w tej kolumnie można wpisać liczbę, która określi rozmiar pola edycyjnego w module FormAnalyzer Verifier. Opcja ta jest dostępna tylko dla pól tekstowych. Nie ma ona wpływu na długość danych tekstowych, a jedynie na prezentację treści. Wartością domyślną jest 0, co oznacza że pole edycyjne zajmują maksymalną dopuszczalną przestrzeń w panelu edycyjnym. Wartość n większa od 0, oznacza że pole edycyjne ma rozmiar n standardowych znaków. Wartość n mniejsza od 0,

oznacza że pole edycyjne ma rozmiar n wyrażony w procentach w stosunku do rozmiaru przestrzeni edycyjnej. Na Rys. 9-11 przedstawiona są zmiany szerokości pól edycyjnych w zależności od rozmiaru panela edycyjnego. W przykładzie zdefiniowano następujące wartości rozmiaru edytora: NrAgenta – 50%, NrPolisy – 50%, Wiek – 4 znaki, Uwagi – 0.

The screenshots show the 'FormAnalyzer Verifier' interface. The top window shows the 'Uwagi' (Remarks) field with a list of missing fields: '1. DATA PODPISANIA' and '2. PODPIS UBEZPIECZ'. The bottom window shows the 'Uwagi' field with a list of missing fields: '1. DATA PODPISANIA' and '2. PODPIS UBEZPIECZONEGO'.

Form Data (Visible in both screenshots):

- ATHENE LIFE** logo
- WNIOSEK O UBEZPIECZENIE NA ŻYCIE**
- Nr AGENTA**: 97123111
- 1. OSOBA UBEZPIECZANA**
 - NALZWSKO**: JACKOWSKA
 - IMIONA**: ZOFIA HELENA
 - ADRES ZAMIESZKANIA: kod miejscowości**: 49102 WIELKA WIES
 - ulica**: GŁÓWNA
 - nr domu**: 12
 - IMIE DANE: imię ojca**: EWARYST
 - stan cywilny**: WIDOWA
 - nazwisko parafialne**: KULIGA
 - MIEJSCE PRACY (adres, nr telefonu)**: [empty]
- 2. UBEZPIECZAJĄCY**
 - NALZWSKO**: [empty]

Form Fields (Visible in both screenshots):

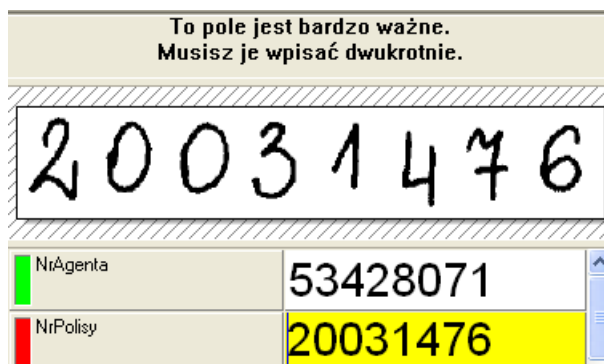
- NrAgenta**: 97123111
- NrPolisy**: 98042594
- Wiek**: 36
- Uwagi**: BRAKUJĄCE POLA:
1. DATA PODPISANIA
2. PODPIS UBEZPIECZ

Rys. 9-11 Prezentacja danych w module FormAnalyzer Verifier dla różnych ustawień dodatkowych opcji weryfikacji.

- Wieloliniowe – włączenie tej opcji umożliwia wprowadzenie wartości wielowierszowych w polu edycyjnym. Wartość większa od zera definiuje liczbę wierszy wyświetlonych w module FormAnalyzer Verifier. Na Rys. 9-11 przedstawiono przykład pola Uwagi dla którego zdefiniowano liczbę wierszy na 3.

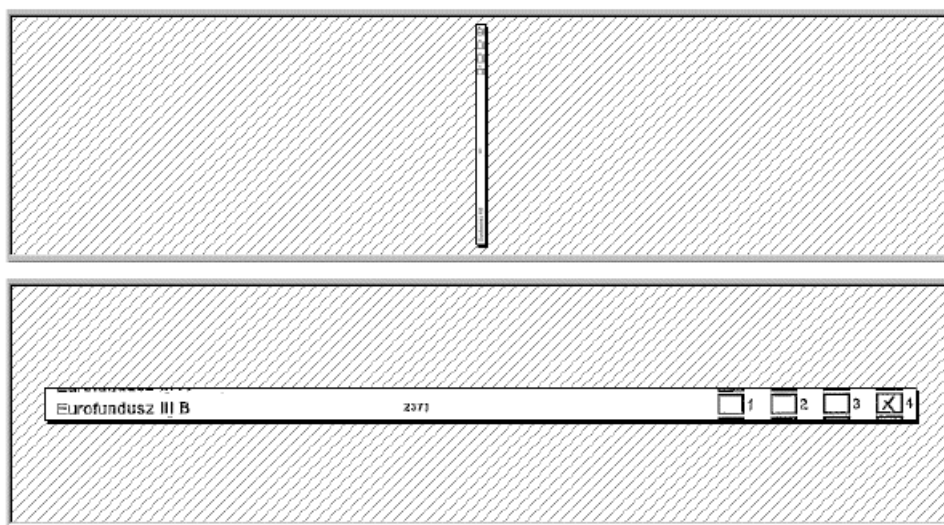
Wprowadzenia i eksport wartości wielowierszowych wymusza zastosowania formatu XML w procesie eksportu danych.

- Wyrównanie – opcja ta umożliwia wybór wyrównania danych w procesie weryfikacji. Dostępne są trzy wartości: *Do lewej*, *Do środka* i *Do prawej*. Domyślną wartością jest *Do lewej*. Na Rys. 9-11 przedstawiono przykład w którym pierwsze trzy pola numeryczne mają ustawioną opcję wyrównania do prawej, a pole Uwagi do lewej.
- Komentarz – w tej kolumnie możemy wpisać komentarz, który będzie wyświetlany w panelu komentarza programu FormAnalyze Verifier. Dobrze dobrany komentarz ułatwia zorientowanie się w pracy operatorom, którzy dopiero zapoznają się z weryfikacją określonego szablonu. Na poniższym rysunku przedstawiono przykładowy komentarz dla pola Nr polisy, które musi być wprowadzone dwa razy.



Rys. 9-12 Fragment okna FormAnalyze Verifier z włączonym podglądem komentarza.

- Obrót – czasami pola na stronach formularza są ustawione tak, że wyświetlenie ich w panelu obrazu pola FormAnalyze Verifier w takim ułożeniu jak na stronie formularza, spowoduje, że będą nieczytelne. Dotyczy to najczęściej pól wyboru (zwłaszcza, jeśli mają być wyświetlone razem z opisem na formularzu). W takim przypadku, należy „obrócić” pole w panelu obrazu pola o wielokrotność 90°. Poniżej pokazano przykład pola wyboru wyświetlonego bez obrotu, oraz z obrotem o 90°.



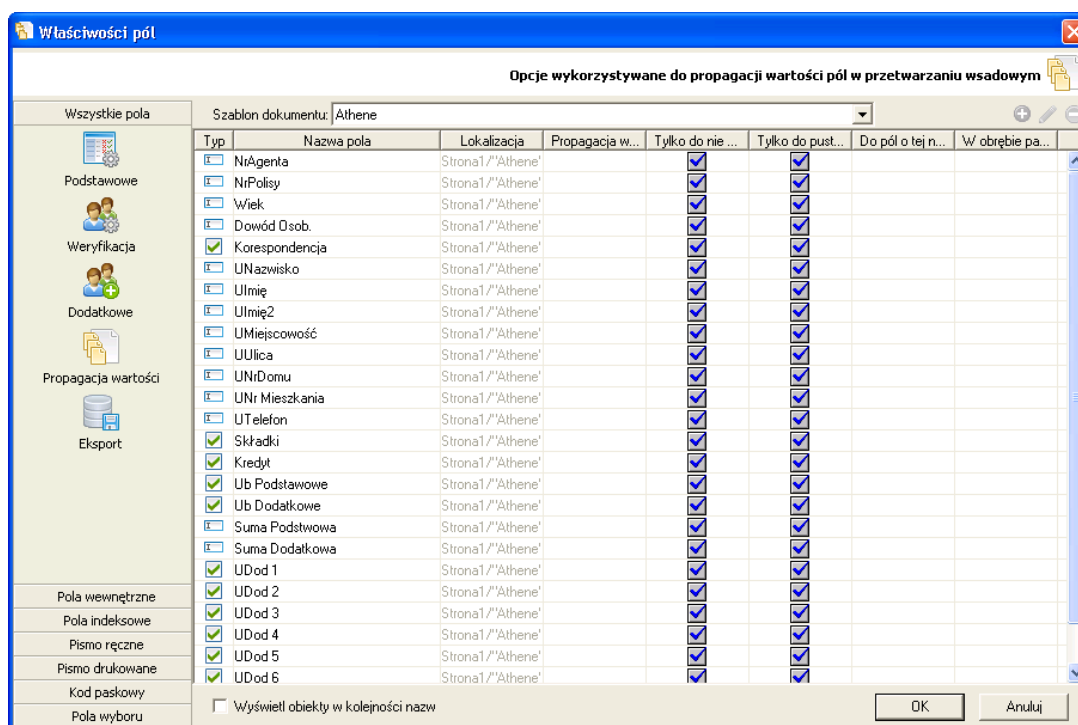
Rys. 9-13 Pole wyboru wyświetlane z domyślnym obrotem 0stopni (góra) i 90 stopni (dół).

9.4. Opcje propagacji wartości

W grupie opcji *Propagacja wartości*, okna dialogowego *Właściwości pól* istnieje możliwość zdefiniowania ustawień umożliwiających propagację wartości pomiędzy polami dokumentów logicznie ze sobą powiązanych. Funkcja ta ma największe zastosowanie w przypadku weryfikacji paczek dokumentów.

Przyjmijmy, że ustawienia propagacji pola zostały zdefiniowane dla pola o nazwie „Nazwisko”. W procesie weryfikacji dla pola „Nazwisko” zaakceptowana została wartość „Kowalski”. Wartość „Kowalski” trafi do pól pustych lub niepoprawionych (z dokumentów o tym samym szablonie, wszystkich dokumentów posiadających pola o konkretnej nazwie lub dokumentów należących do jednej paczki w zależności od bieżących ustawień) we wszystkich kolejnych, weryfikowanych dokumentach. Jeżeli w trakcie dalszego procesu weryfikacji, w którymś z dokumentów, w polu „Nazwisko” wartość „Kowalski” zostanie zmieniona na „Nowak”, wtedy do pozostałych dokumentów propagowaną wartością będzie „Nowak”.

W celu zdefiniowania ustawień dotyczących propagacji wartości, należy zaznaczyć pole wyboru *Propagacja wartości*.



Rys. 9-14 Okno opcji propagacji wartości z wyświetloną listą pól szablonu Athene.

Dla każdego pola z listy zostaną podane w kolejnych kolumnach następujące parametry:

- Typ pola w postaci piktogramu
- Nazwa pola
- Strona i typ formularza, na którym pole się znajduje.
- Propagacja wartości – włączenie propagacji pola.

- Tylko do niepoprawionych – zaznaczenie tej opcji spowoduje, że wartości będą propagowane tylko do pól niepoprawionych. W przypadku, gdy nastąpiła propagacja wartości do pól dokumentu, a dokument taki skierowany został do ponownej weryfikacji lub do drugiego etapu weryfikacji, wartości nie będą w dalszym ciągu propagowane.
- Tylko do pustych – zaznaczenie tej opcji spowoduje, że wartości będą propagowane tylko do pól pustych dokumentu.
- Do pól o tej nazwie – zaznaczenie tej opcji spowoduje, że system będzie propagował wartość do wszystkich pól o tej samej nazwie we wszystkich weryfikowanych dokumentach przypisanych do różnych szablonów.
- W obrębie paczki – zaznaczenie tej opcji spowoduje, że propagacja wartości zostanie przeprowadzona tylko w obrębie paczki, w której zaakceptowano wartości przeznaczone do dalszej propagacji przez system. Zakończenie weryfikacji takiej paczki zakończy propagację wartości. Propagowane wartości zostaną zainicjowane na wartości puste.

9.5. Opcje eksportu

W grupie opcji *Eksport* użytkownik ma do dyspozycji cechy pól, które wykorzystywane są do eksportu dokumentu z systemu FormAnalyzer. Cechy te stanowią dopełnienie do opcji które można zdefiniować w oknie Opcje eksportu, które zostały opisane w rozdziale 14 - Format danych wyjściowych.

T...	Nazwa pola	Lokalizacja	Eksportowa...	Typ danych	Format eksportu	Rozmiar
	NrAgenta	Strona1/"Athene	☒	Maska		8
	NrPolisy	Strona1/"Athene	☒	Ciąg znaków		
	Nazwisko	Strona1/"Athene	☒	Ciąg znaków		
	Imię	Strona1/"Athene	☒	Ciąg znaków		
	Drugie Imię	Strona1/"Athene	☒	Ciąg znaków		
	Wiek	Strona1/"Athene	☒	Ciąg znaków		
	Płeć	Strona1/"Athene	☒	Lista stanów	Lista indeksów	1
	Kod	Strona1/"Athene	☒	Ciąg znaków		
	Miejscowość	Strona1/"Athene	☒	Ciąg znaków		
	Ulica	Strona1/"Athene	☒	Ciąg znaków		
	NrDomu	Strona1/"Athene	☒	Ciąg znaków		
	Nr Mieszkania	Strona1/"Athene	☒	Ciąg znaków		
	Dowód Osob.	Strona1/"Athene	☒	Ciąg znaków		
	Korespondencja	Strona1/"Athene	☒	Lista stanów	Lista indeksów	1
	UNazwisko	Strona1/"Athene	☒	Ciąg znaków		
	UImię	Strona1/"Athene	☒	Ciąg znaków		
	UImię2	Strona1/"Athene	☒	Ciąg znaków		
	UMiejscowość	Strona1/"Athene	☒	Ciąg znaków		
	UUlica	Strona1/"Athene	☒	Ciąg znaków		
	UNrDomu	Strona1/"Athene	☒	Ciąg znaków		
	UNr Mieszkania	Strona1/"Athene	☒	Ciąg znaków		
	UTelefon	Strona1/"Athene	☒	Ciąg znaków		
	Składki	Strona1/"Athene	☒	Lista stanów	Lista indeksów	1
	Kredyt	Strona1/"Athene	☒	Lista stanów	Lista indeksów	1
	Ub Podstawowe	Strona1/"Athene	☒	Wartość logiczna		1

Rys. 9-15 Okno opcje eksportu z wyświetloną listą pól szablonu Athene.

Dla każdego pola z listy zostaną podane w kolejnych kolumnach następujące parametry:

- Typ pola w postaci piktogramu

- Nazwa pola
- Strona i typ formularza, na którym pole się znajduje.
- Eksportowane – można tu ustawić, czy konkretne pole ma być zapisywane w danych wyjściowych, czy nie. Pole jest zapisywane do danych wyjściowych, gdy w tej kolumnie jest znacznik ☒. Kliknięcie w odpowiedniej komórce kolumny Eksportowane powoduje zmianę stanu znacznika. Jeżeli był zaznaczony, to po kliknięciu znika - pole nie będzie eksportowane. Gdy znacznik nie był zaznaczony, po kliknięciu jest ustawiany – pole będzie eksportowane.
- Typ danych – zawiera informacje o sposobie zapisu danych wyjściowych – jej wartość zależy od typu pola.
- Format eksportu – zawiera informacje o formacie eksportowanego dokumentu – jej zawartość zależy od typu danych.
- Rozmiar – możemy tu ustawić długość (liczbę znaków) zapisywanych w pliku danych wyjściowych dla tego pola. To ustawienie ma znaczenie tylko w przypadku stosowania pól o stałym rozmiarze.

Ciąg znaków jest typem danych używanym dla pól tekstowych - oznaczonych ☐. W tym przypadku nie trzeba ustawiać formatu eksportu - wartość pola zostanie zapisana jako ciąg znaków. Długość (liczbę) tych znaków, należy zdefiniować w kolumnie *Rozmiar*. To ustawienie ma znaczenie tylko w przypadku stosowania pól o stałym rozmiarze.

Wartość logiczna jest typem danych używanym do zapisu wartości pojedynczego pola wyboru, oznaczonego ☐. Dla *Wartości logicznej* nie ustawia się dodatkowych parametrów (format, rozmiar) eksportu pola. Gdy pole wyboru jest zaznaczone w pliku danych wyjściowych jest zapisywane 1, kiedy pole nie jest zaznaczone jest zapisywane 0.

Lista stanów jest typem danych używanym do zapisu wartości grupy pól wyboru, oznaczonych jak pojedyncze pole wyboru - ☐. W kolumnie *Format eksportu* jest dla takich pól wyświetlana lista formatów, z której wybiera się odpowiedni format zapisu danych.

Tablica stanów jest formatem o stałej długości, równej liczbie pól w grupie. Dla każdego pola wyboru do pliku wyjściowego jest zapisywany jeden znak: 0 - gdy pole nie jest zaznaczone i 1 - gdy pole jest zaznaczone. Grupa siedmiu pól wyboru, z których są zaznaczone pola drugie i piąte zostanie zapisana w pliku danych wyjściowych jako:

0100100

W formacie *Lista indeksów* do pliku wyjściowego są zapisywane jedynie numery zaznaczonych pól wyboru. Ten format jest najczęściej stosowany w przypadku grup pól wyboru, w których ma być zaznaczone dokładnie jedno pole (grupa 1 z n). Wartością takiej grupy jest liczba oznaczająca numer zaznaczonej opcji.



Rys. 9-16 Lista stanów.

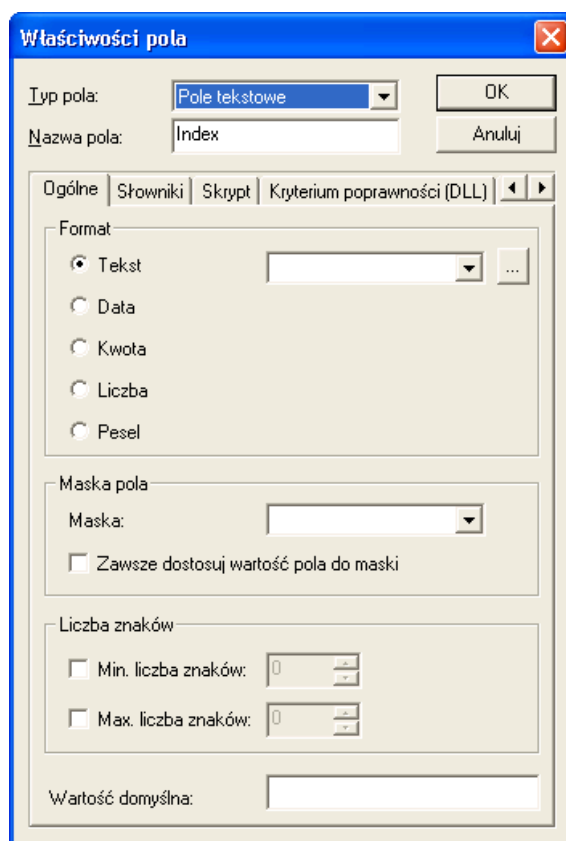
Po ustawieniu parametrów eksportu zalecane jest przeprowadzenie próby zapisu danych wyjściowych oraz pobrania ich do systemu docelowego. Pozwoli to na zweryfikowanie zadanego formatu danych. Jednocześnie zostaną sprawdzone prawa dostępu do foldera(ów), w których będą zapisywane dane wyjściowe.

9.6. Pola indeksowe

Pola indeksowe są polami, które nie są powiązane z obrazem, tym samym są nierozpoznawalne. Pola te pojawiają się w klasyfikacji i weryfikacji dokumentów. Wartości tych pól pojawiać się będą również obok pozostałych pól w procesie weryfikacji dokumentów. (Szczegółowy opis procesu klasyfikacji dokumentów znajduje się w podręczniku aplikacji FormAnalyze Verifier).

W aplikacji FormAnalyze Designer użytkownicy mogą tworzyć pola indeksowe w oknie *Właściwości pól* z włączonym filtrem *Pola indeksowe*.

W celu utworzenia nowego pola indeksowego, należy z rozwijalnej listy *Szablon dokumentu* wybrać odpowiedni szablon, a następnie wybrać filtr *Pola indeksowe*. Jeżeli dla tego szablonu zostały już wcześniej zdefiniowane pola indeksowe zostaną one wyświetlone na liście, jeżeli nie – lista będzie pusta. Następnie należy kliknąć przycisk  znajdujący się w pasku narzędzi. Spowoduje to wyświetlenie okna *Właściwości pola* dla pól indeksowych.



Rys. 9-17 Okno tworzenia nowego pola indeksowego – pole tekstowe.

Z rozwijalnej listy *Typ pola* należy wybrać jeden z dostępnych typów: *pole tekstowe* lub *pole wyboru*. Domyślnie wyświetlone zostanie okno dla tworzenia pola indeksowego typu tekstowego. W polu *Nazwa pola*, należy wprowadzić odpowiednią nazwę nowotworzonego pola. Domyślnie w tym polu będzie znajdować się wartość *Index*.

Okno tworzenia pola indeksowego typu pole tekstowe jest bardzo podobne do okna tworzenia pola rozpoznawalnego *Druk*. W przypadku pól indeksowanych nie ma zakładki związanych z przetwarzaniem, gdyż jak już zostało wspomniane wcześniej pola indeksowane są polami nierozpoznawalnymi. W pozostałych zakładkach można:

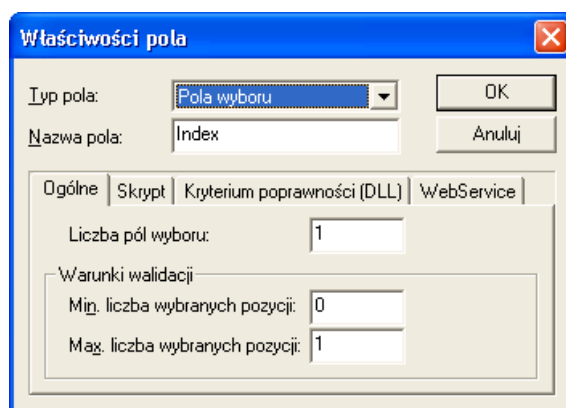
Ogólne – zdefiniować format pola, utworzyć maskę dla pola, ograniczyć liczbę znaków możliwych do wprowadzenia w pole lub nadać domyślną wartość dla pola.

Słowniki – przypisać do pola jeden z dostępnych słowników wartości.

Skrypt, *Kryterium poprawności (DLL)*, *WebService* – w jednej z tych zakładek można zdefiniować warunki poprawności danych.

Szczegółowy opis obsługi opcji i ustawień dostępnych w poszczególnych zakładkach znajduje się w rozdziale 7. Pola rozpoznawalne, w punkcie 7.7 Pole Druk.

W przypadku wybrania typu pola *Pole wyboru* z rozwijalnej listy *Typ pola* w oknie. Właściwości pola, okno dialogowe będzie wyglądało jak poniżej:



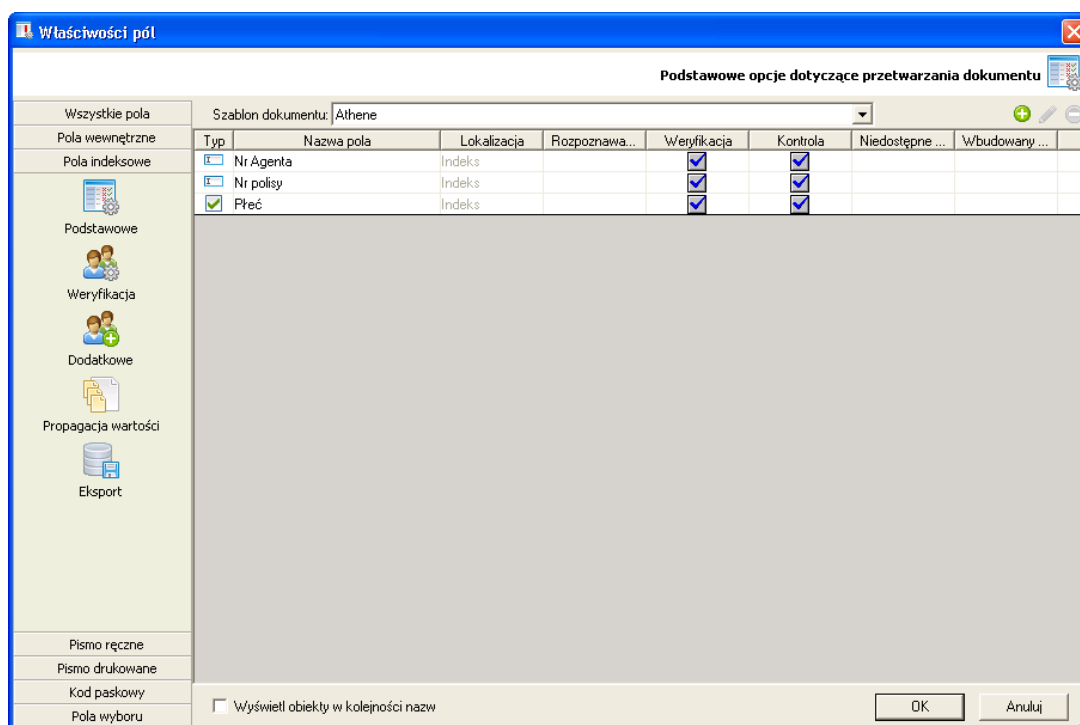
Rys. 9-18 Okno tworzenia nowego pola indeksowanego – pole wyboru.

Pole wyboru będzie mogło posiadać jedną z dwóch wartości: zaznaczone lub niezaznaczone.

W zakładce *Ogólne* należy zdefiniować ilość pól wyboru wprowadzając odpowiednią wartość w obszar *Liczba pól wyboru*. W obszarze *Warunki walidacji* można zdefiniować minimalną i/lub maksymalną ilość zaznaczonych pól wyboru w tworzonej polu indeksowym, np.; jeżeli w przypadku, gdy w polu *Liczba pól wyboru* wprowadzimy wartość 3 (co będzie oznaczać, że pole indeksowe składać będzie się z trzech pól wyboru) a w obszarze *Warunki walidacji* jako minimalną wartość wybranych pozycji wprowadzimy 0, a maksymalną – 1, to takie pole indeksowe, prawidłowo wypełnione, powinno mieć zaznaczone jedno z trzech pól wyboru lub nie mieć żadnego.

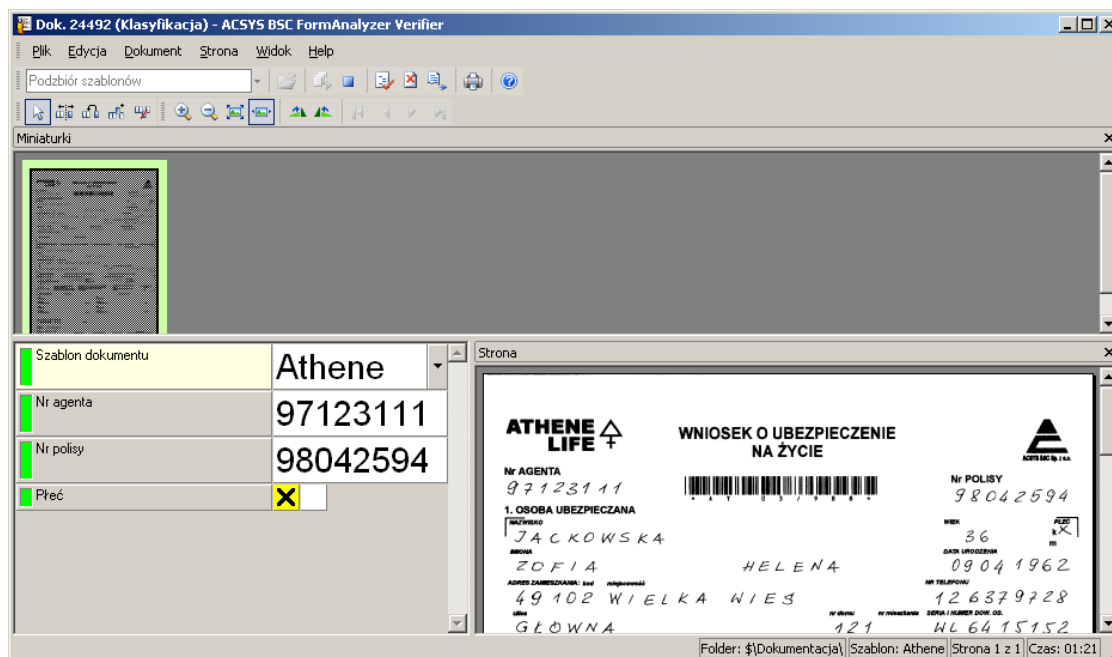
W pozostałych zakładkach można zdefiniować warunki poprawności danych (*Skrypt*, *Kryterium poprawności (DLL)*, *WebService*).

W przykładzie zaprezentowanym poniżej, w formularzu zostały zdefiniowane trzy pola indeksowe: typu tekst: Nr agenta i Nr polisy oraz typu pole wyboru: Płeć.



Rys. 9-19 Przykładowe pola indeksowe zdefiniowane dla szablonu Athene.

W aplikacji FormAnalyzer Verifier, w module Klasyfikacja dokumentów użytkownik będzie widział pola indeksowe w sposób następujący:



Rys. 9-20 Pola indeksowe w aplikacji FormAnalyzer Verifier.

10. Kolejność przetwarzania pól

Ustawienie kolejności przetwarzania powinna być sprawdzana i korygowana po dodaniu nowego pola do szablonu dokumentu. Nowo stworzone pole dodawana jest zawsze na koniec listy. Zmiana kolejności przetwarzania wykonywana jest w oknie dialogowym, które otwierane jest poleceniem *Dokument→Kolejność przetwarzania*. W liście *Szablon dokumentu* dostępne są nazwy wszystkich szablonów zdefiniowanych w pliku konfiguracji. Po otwarciu okna zaznaczony jest szablon, w którego skład wchodzi aktualnie edytowany formularz w głównym oknie programu. Dla wybranego szablonu w liście są wyświetlane pola dokumentu. W liście pól można zaznaczyć wiele obiektów, jednak tylko dla zaznaczonego pojedynczego obiektu dostępne są wszystkie funkcje opisane poniżej.

W pasku w lewej części okna dialogowego użytkownik może wskazać jakiego rodzaju kolejność chce wyświetlać i zmienić: rozpoznawania, weryfikacji lub eksportu.

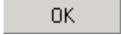
Zmianę kolejność pól można wykonać z wykorzystaniem metody drag&drop lub wykorzystując przyciski w pasku narzędzi. Żeby ustawić odpowiednią kolejność wystarczy zmienić ich kolejność w liście. Zmianę kolejności umożliwiają przyciski:

 zamienia pozycjami zaznaczone pole z następnym

 zamienia pozycjami zaznaczone pole z poprzednim

W oknie tym dostępne są także następujące funkcje:

- Zmiana właściwości pola. Po naciśnięciu przycisku  wyświetlane jest standardowe okno z właściwościami pola.
- Usunięcie pola. Po naciśnięciu przycisku  zaznaczone pola są usuwane z listy i z szablonu.
- Zaznaczenie wszystkich obiektów w liście pól - .
- Odznaczenie wszystkich obiektów w liście pól - .
- Wygenerowanie nazw dla zaznaczonych pól.

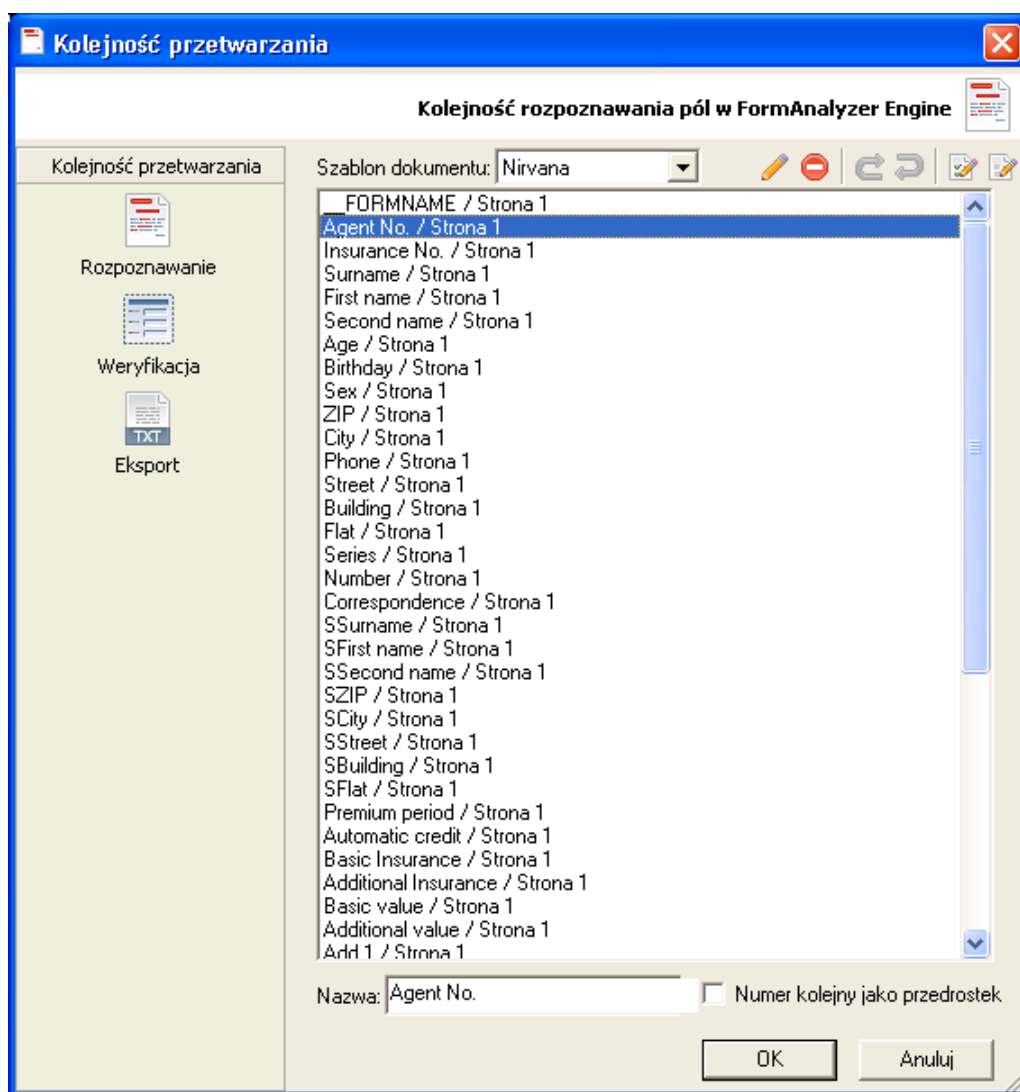
Ostateczna zmiany w szablonach dokonywane są dopiero po naciśnięciu  i zamknięcie okna kolejności przetwarzania. Zamknięcie okna przyciskiem  powoduje, że jest przywracany stan sprzed otwarcia okna.

Rozmiar okna można zmienić, poprzez ciągnięcie prawego-dolnego rogu okna. Rozmiar jest zapisany w rejestrze systemu Windows. Przy kolejnym wyświetleniu tego okna rozmiar jest odtwarzany.

10.1. Kolejność rozpoznawania pól

Po wybraniu opcji *Rozpoznawanie* użytkownik będzie miał możliwość zobaczyć listę wszystkich pól rozpoznawanych znajdujących się w zaznaczonym szablonie dokumentu. Należy zwrócić uwagę, że w liście wyświetlane są także pola wykorzystywane do rozpoznawania typu, pozycjonowania i przetwarzania wstępnego obrazu.

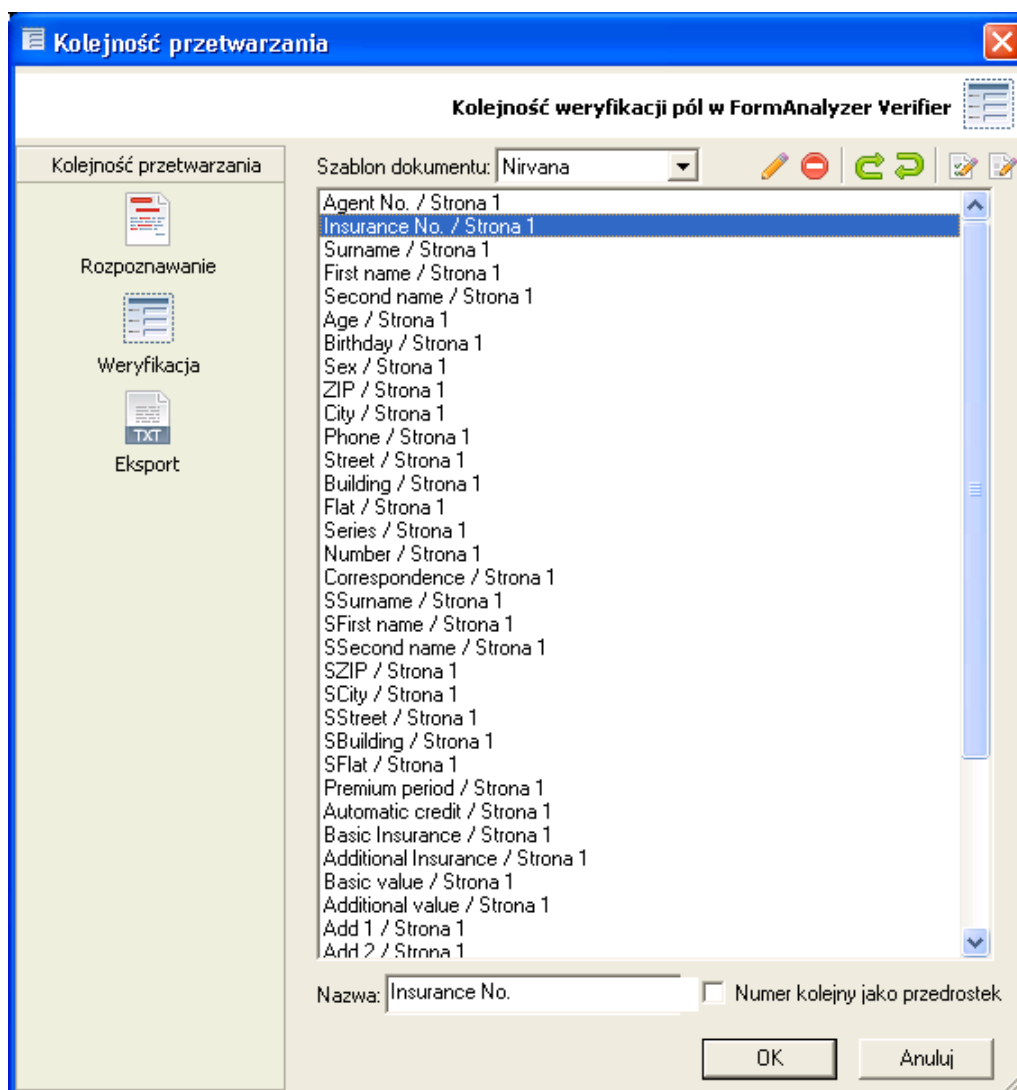
Lista rozpoznawanych pól wyświetlana jest jedynie w celach informacyjnych. Zmianę kolejności rozpoznawania można przeprowadzić dla konkretnego formularza w głównym oknie programu.



Rys. 10-1 Okno dialogowe z wyświetloną kolejnością rozpoznawania pól

10.2. Kolejność weryfikacji pól

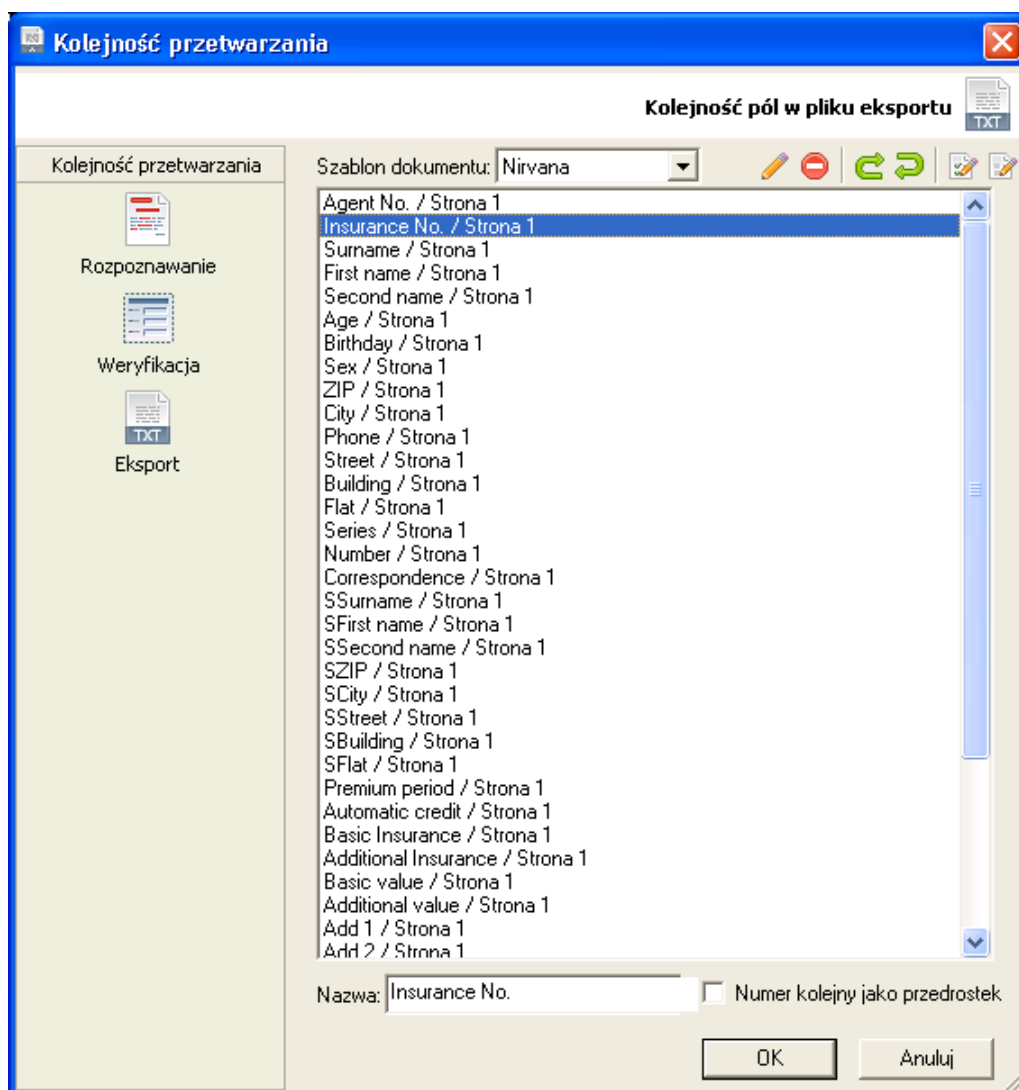
Po wybraniu opcji *Weryfikacja* użytkownik będzie miał możliwość edytować listę weryfikowanych pól znajdujących się w zaznaczonym szablonie dokumentu. Lista ta wykorzystywana jest w procesie klasyfikacji, weryfikacji i kontroli w programie FormAnalyzer Verifier pod warunkiem, że nie jest wybrana opcja *Wyświetlaj pola weryfikacji w kolejności nazw pól*. Należy zwrócić uwagę, że w liście tej nie są wyświetlane pola zgrupowane (podrzędne) w trybie *Wiersz danych* i *Sekcja danych*. Kolejność tych pól wynika ze sposobu grupowania.



Rys. 10-2 Okno dialogowe z wyświetloną kolejnością weryfikacji pól

10.3. Kolejność eksportu pól

Po wybraniu opcji *Eksport* użytkownik będzie miał możliwość edytować listę eksportowanych pól znajdujących się w zaznaczonym szablonie dokumentu. Lista ta wykorzystywana jest w procesie eksportu pod warunkiem, że nie jest wybrana opcja *Sortowanie po nazwie*, którą można ustawić w opcjach eksportu. Należy zwrócić uwagę, że w liście tej nie są wyświetlane pola zgrupowane (nadrzędne) w trybie *Wiersz danych* i *Sekcja danych*. Jest to spowodowane faktem, że pola te służą jedynie do zdefiniowania formy wizualnej wykorzystywanej w procesie weryfikacji.



Rys. 10-3 Okno dialogowe z wyświetloną kolejnością eksportu pól

11. Szablony dokumentów

FormAnalyze Engine rozpoznaje dokumenty zapisane w pliku konfiguracji jako szablony dokumentów. Dla każdego rozpoznawanego typu dokumentu musi być zdefiniowany odpowiedni szablon.

Szablon dokumentu jest wzorcem rozpoznawania danego rodzaju dokumentów i składa się z formularzy. Każdy formularz w szablonie opisuje sposób rozpoznawania jednej strony dokumentu. Szablon zawiera, więc tyle formularzy ile stron ma dany rodzaj dokumentu. Dla dokumentów jednostronicowych szablon składa się z jednego formularza.

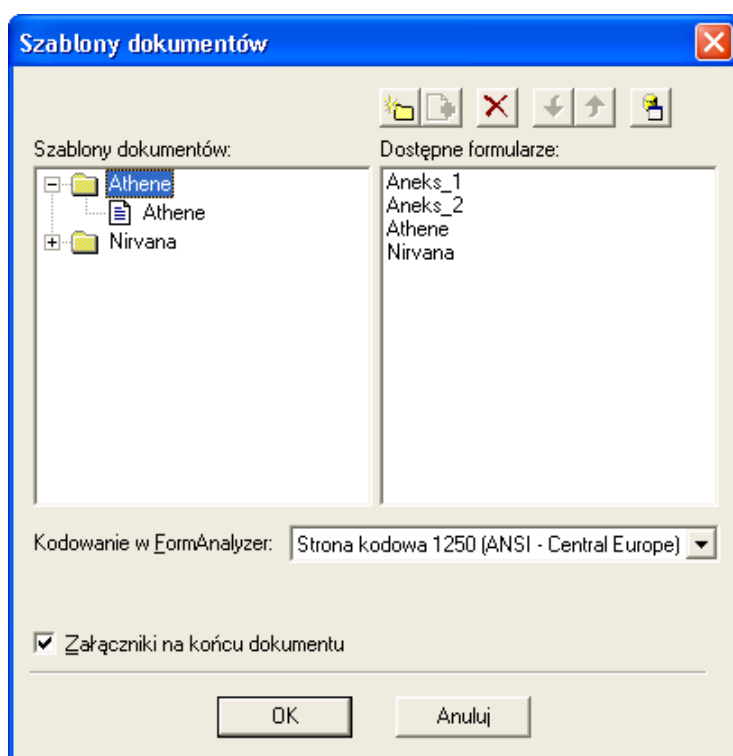
Kolejność występowania formularzy w szablonie musi dokładnie odpowiadać kolejności stron w dokumencie: typ formularza jest rozpoznawany jedynie w niezbędnych przypadkach dla pierwszej strony dokumentu. Kolejne strony są rozpoznawane zgodnie z kolejnymi formularzami znajdującymi się w szablonie.

W celu zdefiniowania nowego szablonu lub modyfikacji istniejącego, należy wybrać z menu Dokument polecenie Szablony... Na ekranie wyświetlone zostanie okno dialogowe Szablony dokumentów przedstawione na poniższym rysunku.

W oknie dialogowym Szablony dokumentów znajdują się dwa obszary: Szablony dokumentów, oraz Dostępne formularze. W pierwszym z nich wyświetlane są szablony dokumentów dostępne w danym pliku konfiguracji. Jeżeli lista ta jest pusta, oznacza to, że szablony dokumentów nie zostały jeszcze zdefiniowane. W drugim obszarze *Dostępne formularze* znajdują się nazwy wszystkich zdefiniowanych formularzy. Nazwy formularzy są posortowane alfabetycznie.

W dolnej części okna znajdują się dwie rozwijalne listy Kodowanie w FormAnalyze, oraz Kodowanie dla eksportu. Pierwsza z nich służy do wskazania strony kodowej, w której silnik będzie tworzył wyniki rozpoznawania. Dotyczy to głównie lokalnych ustawień takich, jak np.: polskie znaki. Moduły Engine, Export i Weryfikator będą korzystały ze strony kodowej wskazanej na tej liście. W sytuacji, gdy wybrana została odpowiednia strona kodowa dla w FormAnalyze, można dokonać jej konwersji w procesie eksportu. W tym celu należy zdefiniować odpowiednią stronę kodową dla eksportu wybierając żądaną pozycję z drugiej rozwijalnej listy Kodowanie dla eksportu.

Oprócz stałej struktury szablonów dokumentów oraz dostępnych formularzy można do każdego z szablonów dołączyć załączniki. Przydatne jest to w sytuacji, gdy np.: dla dokumentu zawierającego dane, jako integralna część dokumentu dołączone jest zdjęcie. Można wtedy umieścić taką część dokumentu jako załącznik, w tym celu, w oknie dialogowym Szablony dokumentów, należy zaznaczyć opcję Załączniki na końcu dokumentów.



Rys. 11-1 Okno szablonów dokumentów.

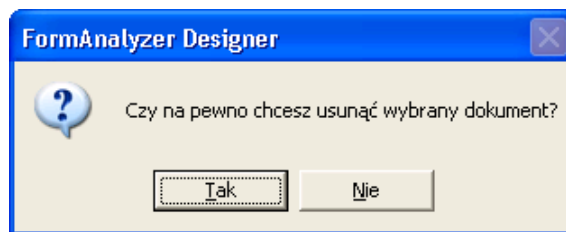
Nad obszarem Dostępne formularze znajdują się przyciski funkcyjne. Pierwszy z nich Nowy szablon dokumentu  służy do utworzenia nowego szablonu dokumentu. W obszarze Szablony dokumentów pojawi się nowy szablon, którego nazwa będzie znajdowała się w trybie edycji:  Dok. (1) , dzięki czemu można wprowadzić od razu właściwą nazwę szablonu. Po wprowadzeniu nowej nazwy, szablon dokumentu jest pusty, tzn. nie zawiera żadnych formularzy.

Żeby dodać nowy formularz (lub kilka formularzy), po uprzednim wskazaniu odpowiedniego szablonu, w obszarze Dostępne formularze, należy zaznaczyć odpowiednią ilość formularzy. Kliknięcie przycisku Nowe strony  spowoduje dodanie zaznaczonych formularzy do szablonu. Formularze zostaną dodane w kolejności, w jakiej znajdowały się w obszarze.

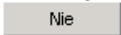
Do zmiany kolejności formularzy w szablonie służą przyciski Przesuń w dół  i Przesuń do góry  . Powodują one odpowiednio zamianę kolejności aktualnie wybranego formularza o jedną pozycję w dół lub w górę. Zmiana kolejności odbywa się jedynie w obrębie jednego szablonu.

Aby usunąć szablon lub formularz z obszaru Szablony dokumentów, należy zaznaczyć żądany szablon lub formularz, a następnie kliknąć przycisk Usuń  służy do usuwania formularzy i szablonów. Przed usunięciem należy paskiem wyboru w liście szablonów zaznaczyć element, który chcemy usunąć.

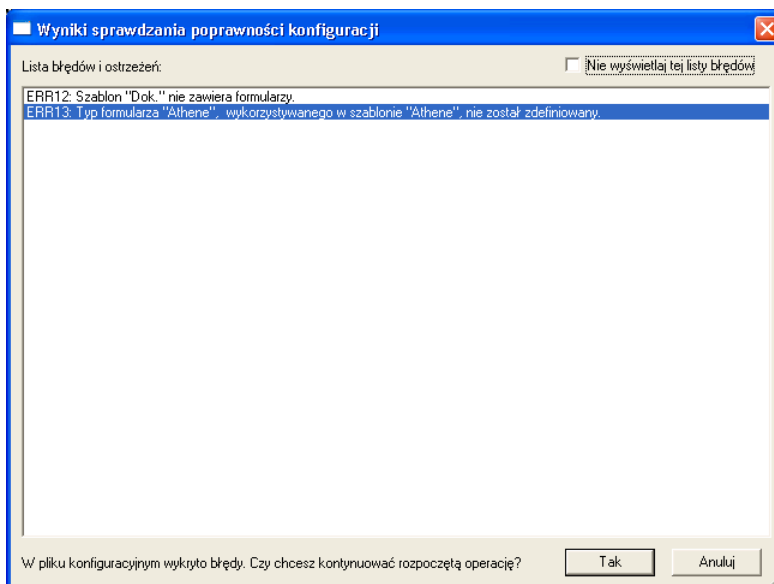
W momencie próby usunięcia szablonu, kliknięcie przycisku Usuń  spowoduje wyświetlenie okna komunikatu nakazującego potwierdzenie lub zaniechanie rozpoczętej operacji



Rys. 11-2 Okno potwierdzenia usuwania szablonu.

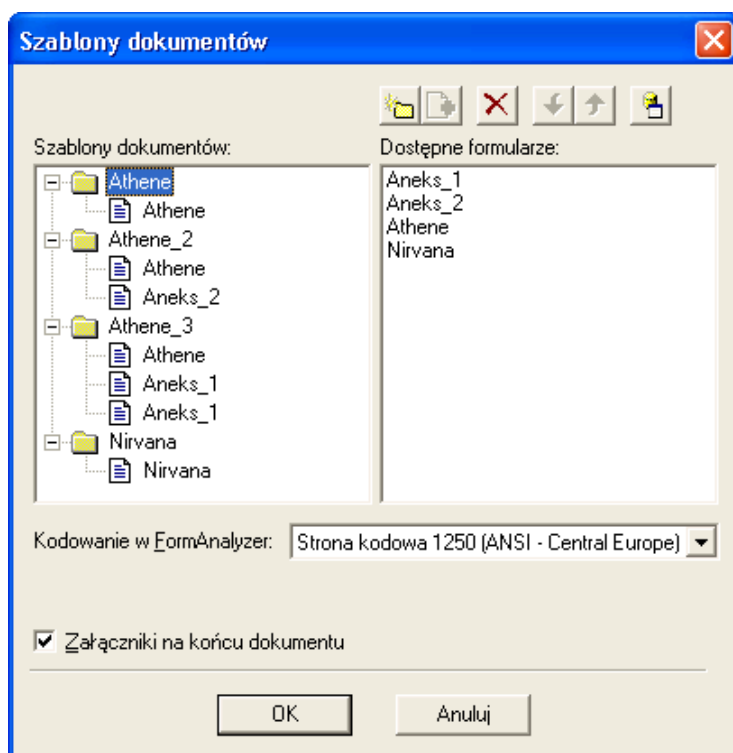
Kliknięcie przycisku  usuwa zaznaczony szablon z obszaru, kliknięcie przycisku  anuluje operację usuwania.

Przycisk Zapisz informację o szablonie dokumentu  służy do zapisywania informacji o szablonie dokumentu w pliku tekstowym. Na ekranie wyświetlone zostanie standardowe okno dialogowe *Zapisywanie jako* służące do wskazania lokalizacji i podania nazwy pliku tekstowego. Jeżeli zaznaczony szablon nie posiada formularzy, wyświetlany jest stosowny komunikat w oknie dialogowym Wyniki sprawdzania poprawności szablonu dokumentu pokazanym na poniższym rysunku



Rys. 11-3 Komunikat o braku formularzy w szablonie.

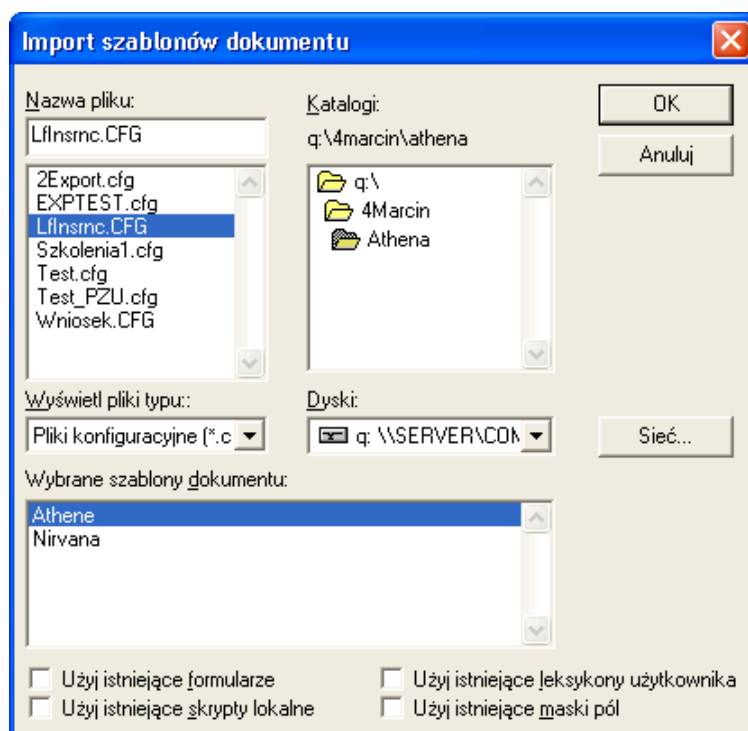
Poniżej przedstawiono przykładową strukturę szablonów pliku konfiguracji, przeznaczonego do rozpoznawania wniosków o ubezpieczenie firm Nirvana Insurance i Athene Life. Wniosek może mieć trzy warianty, różniące się liczbą załączników. Formularze załączników są takie same dla obu firm. Pierwsza strona wniosku ubezpieczeniowego firmy Athene Life jest opisana formularzem Athene Life. Formularz pierwszej strony wniosku Nirvana Insurance nazywa się Nirvana.



Rys. 11-4 Przykładowa struktura szablonów w pliku konfiguracji.

Dla każdej firmy zdefiniowano trzy szablony: dla dokumentu bez załączników oraz dokumentów z jednym załącznikiem. Szablony dokumentów jednej firmy mają różną liczbę stron, dlatego jest poprawne użycie dla tych szablonów takiego samego formularza pierwszej strony. W czasie rozpoznawania konkretnego dokumentu, w zależności od liczby jego stron, wybór będzie ograniczony do par szablonów *Athene - Nirvana*, *Athene 2 - Nirvana 2* lub *Athene 3 - Nirvana 3*. W każdej z powyższych par, pierwsza strona szablonu jest różna, zatem będzie można ustalić jednoznacznie szablon rozpoznając formularz pierwszej strony dokumentu.

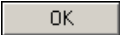
Bardzo istotna z punktu widzenia efektywności pracy (a w szczególności ponownego wykorzystania już zaprojektowanych szablonów lub ich składowych) jest możliwość importowania szablonów z innych plików konfiguracyjnych. Polecenie *Dokument→Import* otwiera okno importu szablonów dokumentów. Na poniższym rysunku przedstawiono okno importu szablonów z zaznaczonym szablonem *Athene* przy wyłączonych wszystkich opcjach importu. W takim przypadku do bieżącego pliku konfiguracji zostaną skopiowane wszystkie składowe tworzące zaznaczone szablony: formularze, lokalne kryteria poprawności VBScript, leksykony użytkownika oraz maski pól. Kopiowanie składowych może wiązać się z konfliktem nazw, gdy w bieżącym pliku konfiguracji istnieją już składowe (tzn. formularz, skrypt lokalny, leksykon lub maska) o takiej samej nazwie jak składowa importowana. W tej sytuacji do nazw składowych importowanych są dodawane przyrostki (*n*), gdzie *n* jest kolejnym numerem kopii.

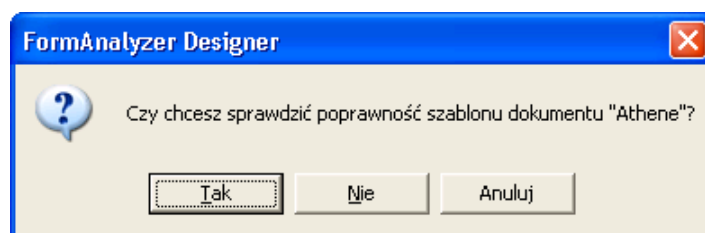


Rys. 11-5 Okno importu szablonów dokumentów.

W przypadku zaznaczenia opcji *Użyj istniejące formularze* przy imporcie szablonu są sprawdzane nazwy formularzy. Jeżeli w aktualnym pliku konfiguracji zostanie odnaleziony formularz o nazwie takiej samej jak formularz używany przez szablon importowany, w szablonie importowanym zostanie użyty formularz znajdujący się w pliku konfiguracji, do którego importujemy szablon. Porównywana jest tylko nazwa formularza. Ponieważ nazwy formularzy są unikalne tylko w pliku konfiguracji, mogą istnieć (w różnych plikach konfiguracji) różne formularze o tej samej nazwie. Przed włączeniem opcji *Użyj istniejące formularze* należy sprawdzić czy formularze posiadają tylko taką samą nazwę i czy różnią się zawartością.

W przypadku włączenia pozostałych opcji dotyczących używania składowych szablonu zdefiniowanych w bieżącym pliku konfiguracji, tzn. *Użyj istniejące skrypty lokalne*, *Użyj istniejące leksykony użytkownika* i *Użyj istniejące maski pól*, obowiązują inne zasady zastępowania składowych. Program sprawdza zarówno nazwę składowej, jak i wszystkie jej ustawienia - dla skryptu jest to jego treść, dla leksykonu zestaw znaków dla maski wzorzec znaków. Tylko w przypadku, gdy nazwa i ustawienia są takie same odpowiednia składowa jest zastępowana przez składową szablonu znajdującą się w bieżącym pliku konfiguracji.

Przed wykonaniem importu szablonów zalecane jest sprawdzenie ich poprawności. Po zaznaczeniu stosownych opcji importu szablonów i kliknięciu przycisku  wyświetlone zostanie okno komunikatu, pokazane na poniższym rysunku, z zapytaniem o sprawdzenie poprawności wskazanego szablonu dokumentu.



Rys. 11-6 Okno z pytaniem o sprawdzenie poprawności szablonu, który ma być importowany.

Powyższe okno komunikatu zostanie wyświetlone dla każdego z importowanych szablonów. Kliknięcie przycisku  spowoduje sprawdzenie poprawności szablonu. Kliknięcie przycisku  spowoduje nie sprawdzanie poprawności szablonu. Kliknięcie przycisku  spowoduje, że szablon nie zostanie zaimportowany. Poprawność jest sprawdzana przed wykonaniem importu. Jeżeli w trakcie importu zostanie zastąpiony formularz, szablon może posiadać błędy.

12. Definiowanie warunków poprawności danych

Dla szybkości działania systemu wprowadzania danych duże znaczenie ma weryfikacja danych. Jest to na ogół najbardziej czasochłonny etap przetwarzania dokumentów do postaci elektronicznej. Automatyczne wykonanie przynajmniej części weryfikacji pozwala na znaczne zwiększenie wydajności całego systemu.

W systemie FormAnalyze jest wiele metod pozwalających na automatyczne sprawdzanie warunków poprawności rozpoznania pola: format danych i słowniki dla pól tekstowych, minimalna i maksymalna wartość zaznaczonych pól wyboru w grupie. Te metody pozwalają na dość ograniczone ustalenie poprawności danego pola. Trzeba zwrócić uwagę, że takie sprawdzenie poprawności odbywa się jedynie na podstawie wartości sprawdzanego pola - brak tutaj możliwości skorzystania z wartości innych pól.

Omówioną wcześniej metodą weryfikacji jest też wykonanie skryptu weryfikacji pojedynczego pola lub wywołanie funkcji weryfikacji z zewnętrznej biblioteki DLL. Również w tym przypadku mamy do dyspozycji jedynie wartość weryfikowanego pola, jednak można tu zaimplementować bardzo różnorodne i dokładne kryteria weryfikacji.

Weryfikacja wartości pojedynczego pola (omówiona dla skryptów w punkcie 7.8.5 - Skrypty, a dla funkcji weryfikacji z zewnętrznych bibliotek DLL w punkcie 7.8.6 - Funkcje weryfikacji z zewnętrznych bibliotek DLL) jest nazywana weryfikacją lokalną. Znajduje ona zastosowanie do weryfikacji (lub modyfikacji) wartości pojedynczego pola bez uwzględnienia innych wartości rozpoznanych na formularzu. Takie sprawdzenie poprawności stosuje się przede wszystkim w przypadku pól, których wartość ma określoną, dającą się sprawdzić strukturę lub też jest określony zbiór (zakres) dopuszczalnych wartości pola.

W systemie FormAnalyze można skorzystać z drugiego rodzaju weryfikacji – weryfikacji sprawdzającej zależności pomiędzy wartościami wielu pól dokumentu (nazywanej też weryfikacją globalną). Weryfikacja globalna pozwala na sprawdzenie spójności danych rozpoznanych na dokumencie.

Niezależnie od typu weryfikacji (lokalna - globalna), sposobu jej realizacji (skrypt, funkcja, Webservice) oraz tego czy weryfikacja modyfikuje wartość pola, czy nie, proces weryfikacji odpowiada na pytanie: Czy pole (pola) spełnia kryteria poprawności badane w procesie weryfikacji? Jeżeli tak, proces weryfikacji przekazuje do systemu FormAnalyze wartość logiczną TRUE. Jeśli w procesie weryfikacji ustalono, że pole nie spełnia kryteriów poprawności do systemu jest przekazywana wartość FALSE.

W przypadku weryfikacji modyfikującej wartość pola rozpoznawanego kryterium poprawności jest sprawdzane dla nowej wartości (wyznaczonej w procesie weryfikacji). Jedynie wtedy, gdy ta nowa wartość spełnia kryteria poprawności jest modyfikowana wartość pola dokumentu. Jeżeli mimo modyfikacji wartości, pole w dalszym ciągu nie spełnia kryteriów poprawności, wartość pola dokumentu nie jest zmieniana. (Wartość, o której wiadomo, że jest niepoprawna zostaje do poprawienia ręcznego przez operatora.)

Bardzo istotną sprawą jest kolejność wykonywania poszczególnych procedur weryfikacji. Decyduje o tym kilka czynników:

1. W przypadku procedur uruchamianych w programie FA Engine: procedury lokalne są wykonywane po rozpoznaniu wartości pola (uruchomienie procedury poprzedzone jest ewentualnie przez zastosowanie przetwarzania końcowego i

słownika wartości). Procedury globalne są wykonywane dopiero po rozpoznaniu wszystkich pól dokumentu (czyli także po rozpoznaniu wszystkich formularzy). O kolejności uruchomienia procedur globalnych decyduje ich uszeregowanie w oknie definiowania kryteriów poprawności.

2. W przypadku procedur uruchamianych w programie FA Verifier:
 - a. Wszystkie procedury lokalne są uruchamiane podczas otwierania dokumentu (ma to na celu ustawienie statusu pól dokumentu).
 - b. Poszczególne procedury lokalne są uruchamiane podczas akceptacji pól z którymi zostały związane.
 - c. Procedury globalne są uruchamiane podczas akceptacji ostatniego z pól stanowiących argumenty procedury.
 - d. Podczas akceptacji dokumentu uruchamiane są wszystkie procedury lokalne i globalne, które wcześniej nie zostały zaakceptowane.
3. Jeżeli z danym polem związane zarówno skrypt, funkcję i Webservice to kolejność wykonywania procedur weryfikacyjnych jest następująca: skrypt, funkcja, Webservice.

12.1. Okno globalnych kryteriów poprawności

Sposób dostępu do właściwości kryteriów poprawności jest różny dla kryteriów lokalnych i globalnych. Lokalne procedury weryfikacji są dostępne w oknach właściwości pól, z którymi są związane. Nie wprowadza to żadnych ograniczeń, ponieważ weryfikacja lokalna korzysta jedynie z wartości pola, z którym jest związana. Sposób definiowania i modyfikacji właściwości procedur weryfikacji lokalnej został omówiony w punktach 7.8.5 - Skrypty, 7.8.6 - Funkcje weryfikacji z zewnętrznych bibliotek DLL oraz 7.8.7 - Funkcje weryfikacji poprawności danych pola z wykorzystaniem Webservice.

Globalne procedury weryfikacji są dostępne w oddzielnym oknie: *Kryteria poprawności dokumentów*.

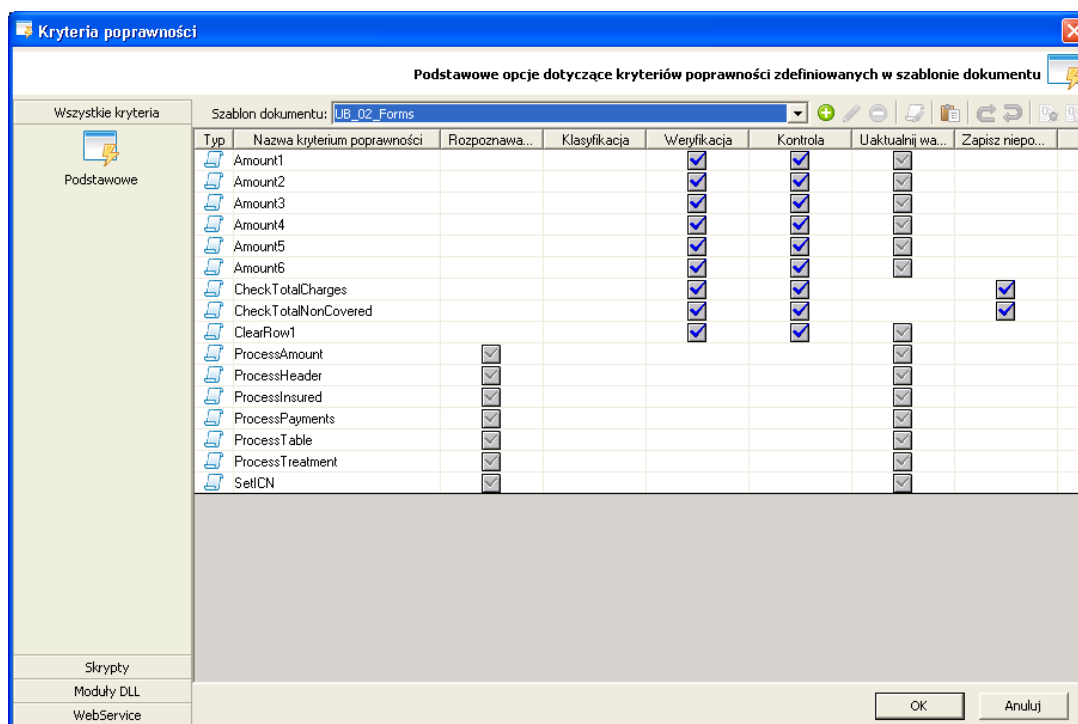
To okno jest otwierane poleceniem *Dokument*→*Kryteria poprawności*. W liście *Szablon dokumentu* są dostępne nazwy wszystkich szablonów zdefiniowanych w pliku konfiguracji. Po otwarciu okna zaznaczony jest szablon, w którego skład wchodzi aktualnie edytowany formularz w głównym oknie programu. Dla wybranego szablonu w liście są wyświetlane globalne procedury weryfikacji związane z tym szablonem. W liście zdefiniowanych kryteriów poprawności można wyróżnić jedno kryterium - do niego będą się odnosić operacje zadane przez użytkownika. W liście kryterium poprawności wyświetlane są następujące cechy:

- Typ:  (skrypty),  (biblioteki DLL),  (funkcje Webservice)
- Nazwa – unikalna w obrębie jednego szablonu
- Rozpoznawanie – kryterium uruchamiane jest na etapie rozpoznawania dokumentu w module FormAnalyzer Engine

- Klasyfikacja – kryterium uruchamiane jest na etapie klasyfikacji dokumentu w module FormAnalyzer Verifier
- Weryfikacja – kryterium uruchamiane jest na etapie weryfikacji dokumentu w module FormAnalyzer Verifier
- Kontrola – kryterium uruchamiane jest na etapie kontroli dokumentu w module FormAnalyzer Verifier
- Uaktualnij wartość – włączenie tego atrybutu umożliwia przekazanie wartości z kryterium do programu (FormAnalyzer Engine lub Verifier)
- Zapisz niepoprawne – włączenie tego atrybutu umożliwi akceptację dokumentu nawet wtedy, gdy kryterium poprawności nie jest spełnione

W pasku w lewej części okna dialogowego użytkownik może wskazać jakiego rodzaju obiekty chce wyświetlać: wszystkie kryteria, skrypty, funkcje DLL lub funkcje Webservice.

Rozmiar okna można zmienić, poprzez ciągnięcie prawego-dolnego rogu okna. Rozmiar jest zapisany w rejestrze systemu Windows. Przy kolejnym wyświetleniu tego okna rozmiar jest odtwarzany.



Rys. 12-1 Okno Kryteria poprawności dokumentów.

Kolejność procedur weryfikacji w tej liście odpowiada kolejności ich wykonywania. Żeby ustawić odpowiednią kolejność wykonywania weryfikacji wystarczy zmienić ich kolejność w liście. Zmianę kolejności umożliwiają przyciski:

 zamienia pozycjami w liście zaznaczoną procedurę z następną

 zamienia pozycjami w liście zaznaczoną procedurę z poprzednią

Przycisk  służy do usunięcia procedury weryfikacji. Procedura jest usuwana z listy natychmiast, jednak nie jest od razu usuwana z pliku konfiguracji. Dopiero naciśnięcie  i zamknięcie okna kryteriów poprawności, powoduje nieodwracalne usunięcie procedury. Zamknięcie okna skryptów przyciskiem  powoduje, że jest przywracany stan listy kryteriów sprzed otwarcia okna.

Ważna różnica występuje tutaj między skryptami weryfikacji a funkcjami z bibliotek DLL. Wynika ona z różnego położenia definicji procedury weryfikacji: skrypty są zdefiniowane w pliku konfiguracji, natomiast funkcje w bibliotece DLL. Usunięcie skryptu oznacza, zatem usunięcie jego definicji. (Przy omyłkowym usunięciu skryptu, jego odtworzenie wymaga ponownego napisania treści skryptu.) W przypadku funkcji z biblioteki DLL usunięcie polega na skasowaniu powiązania pomiędzy funkcją z biblioteki a polami dokumentu - definicja funkcji pozostaje niezmienną w bibliotece DLL.

Modyfikację właściwości procedury weryfikacji umożliwia przycisk . Te zmiany są wykonywane w kreatorze kryterium poprawności omówionym w kolejnym punkcie. Jeżeli mamy do czynienia ze skryptem dostępna jest także funkcja , która wyświetla okno z treścią skryptu. Opis edycji skryptów został przedstawiony w rozdziale 12.3. Dla innych bibliotek DLL i funkcji WebService funkcja ta nie jest dostępna.

Kryteria poprawności mogą także powstać poprzez zaimportowanie obiektów z innych szablonów dokumentu. Do tego służy funkcja , która opisana jest w rozdziale 12.4.

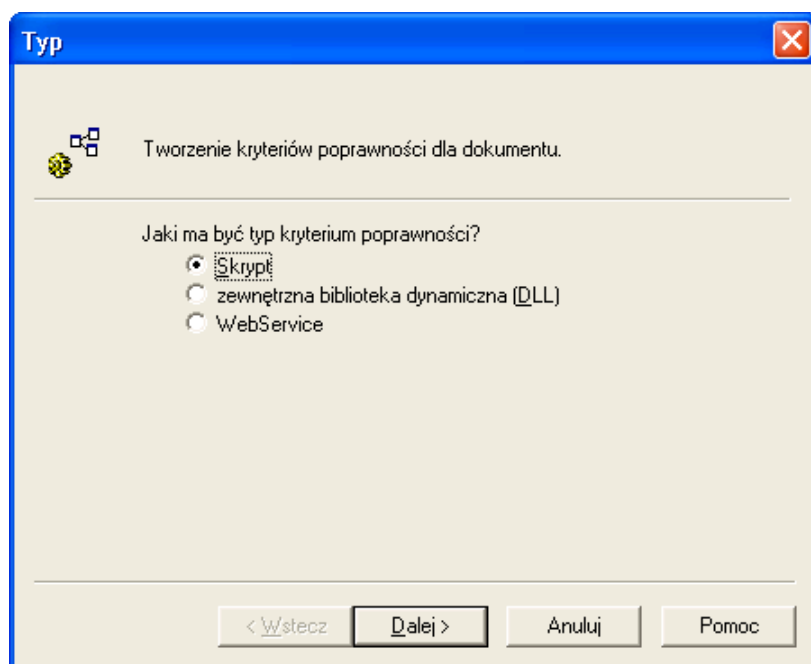
Przycisk  uruchamia proces sprawdzenia działania procedury weryfikacji. Program otwiera okno, w którym podaje się wartości parametrów wejściowych, po czym następuje wykonanie procedury weryfikacji.

Inną metodą przetestowania kryterium poprawności jest proces testowania z wykorzystaniem treści dokumentów zlokalizowanych w bazie danych FormAnalyze. Do tego celu służy przycisk . Dokładny opis tej procedury zostanie przedstawiony w rozdziale 12.5.

12.2. Definiowanie nowego kryterium poprawności

Przed rozpoczęciem definiowania nowego kryterium poprawności należy zwrócić uwagę na to, który szablon dokumentu jest wybrany w liście szablonów. Nowo tworzony kryterium zostanie dołączone do aktualnie wybranego szablonu. W FormAnalyze Designer nie ma możliwości przenoszenia kryteriów między szablonami dokumentów.

Definiowanie nowego kryterium rozpoczyna kliknięcie przycisku . Program otwiera pierwszą stronę kreatora pokazaną na poniższym rysunku. Na tej stronie wybiera się sposób realizacji kryterium poprawności: jako skrypt, jako funkcję z zewnętrznej biblioteki DLL lub jako Webservice.



Rys. 12-2 Kreator kryteriów poprawności – strona wyboru typu kryterium.

Dalszy przebieg tworzenia kryterium poprawności zależy od wybranego typu. Ponieważ część tego procesu jest taka sama dla wszystkich typów kryterium, dlatego przedstawimy je jednocześnie, zaznaczając różnice (w rozdziale 12.11.3 opisano procedurę tworzenia kryterium poprawności typu Web Service).

Jeśli kryterium ma być realizowane jako skrypt kolejnymi krokami będą:

1. Zdefiniowanie podstawowych parametrów kryterium (nadanie nazwy, wybranie miejsca uruchomienia skryptu, oraz wskazanie czy skrypt ma modyfikować wartość pól).
2. Wskazanie argumentów (zmiennych wejściowych) kryterium
3. Wskazanie zmiennych wyjściowych kryterium (tylko gdy skrypt modyfikuje wartość pól).
4. Napisanie treści skryptu

W przypadku, gdy kryterium będzie realizowane jako funkcja z biblioteki DLL, kolejnymi krokami jego tworzenia będą:

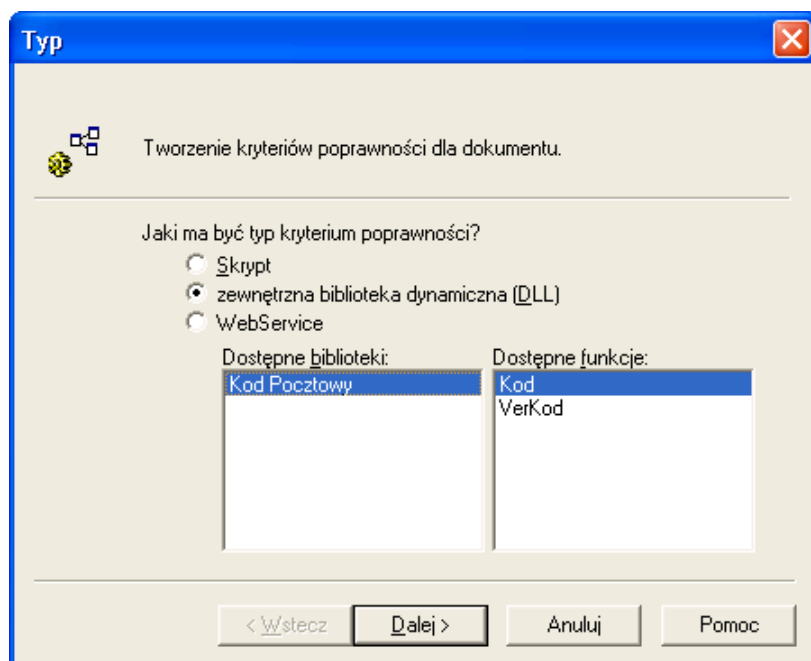
1. Wskazanie biblioteki DLL i funkcji weryfikacji w tej bibliotece
2. Określenie nazwy kryterium.
3. Wskazanie argumentów (zmiennych wejściowych) kryterium
4. Wskazanie zmiennych wyjściowych kryterium (tylko gdy skrypt modyfikuje wartość pól).

W przypadku, gdy kryterium będzie realizowane, jako funkcja WebService, kolejnymi krokami będą:

1. Wskazanie miejsca położenia definicji WebService (WSDL), wskazanie domyślnej przestrzeni nazw używanej w pliku definicji (WSDL) i wybranie odpowiedniej funkcji;
2. Wskazanie adresu URL wywołania funkcji WebService.
3. Zdefiniowanie podstawowych parametrów kryterium (nadanie nazwy, wybranie miejsca uruchomienia WebService, oraz wskazanie czy WebService ma modyfikować wartość pól).
4. Wskazanie argumentów (zmiennych wejściowych) kryterium
5. Wskazanie zmiennych wyjściowych kryterium (tylko gdy WebService modyfikuje wartość pól).

Jeżeli definiujemy skrypt weryfikacji, na pierwszej stronie kreatora zaznaczamy jedynie opcję *Skrypt*. Możemy teraz przejść na następną stronę kreatora, na której zdefiniujemy podstawowe właściwości kryterium (skryptu).

Jeśli do weryfikacji chcemy użyć funkcji z biblioteki DLL zaznaczamy opcję *zewnętrzna biblioteka dynamiczna (DLL)*. Na pierwszej stronie kreatora pojawią się listy wyboru biblioteki DLL i funkcji weryfikacji w tej bibliotece.



Rys. 12-3 Kreator kryteriów poprawności – wybór funkcji weryfikacji.

W liście Dostępne biblioteki są wyświetlone nazwy wszystkich bibliotek DLL zawierających funkcje weryfikacji. FormAnalyze Designer poszukuje tych bibliotek jedynie w folderze, w którym został uruchomiony. Żeby skorzystać z funkcji znajdujących się w określonej bibliotece DLL musimy skopiować tę bibliotekę do foldera, w którym znajduje się program designer.exe.

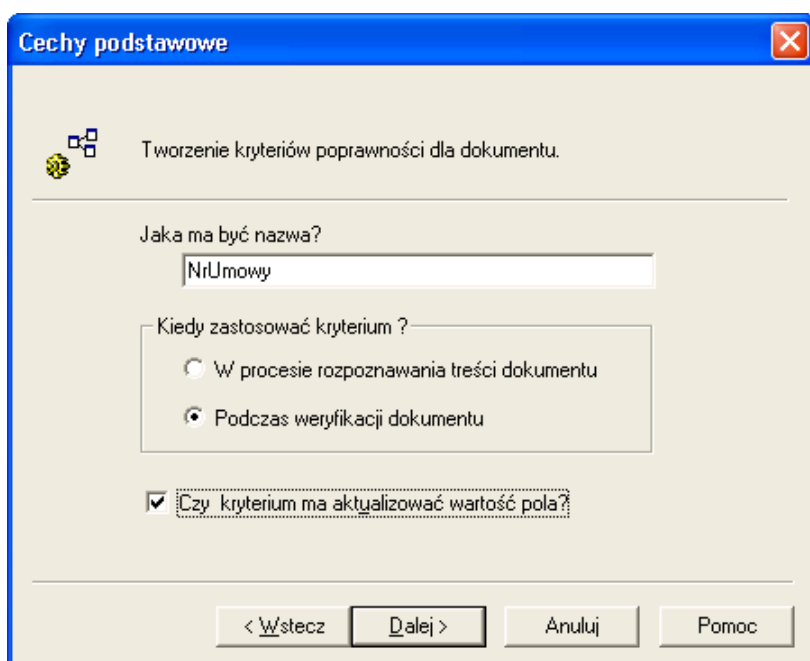
W liście Dostępne funkcje są wyświetlane funkcje weryfikacji zdefiniowane w zaznaczonej paskiem wyboru bibliotece. Ponieważ korzystamy z kreatora globalnych kryteriów poprawności w liście są wyświetlone tylko funkcje weryfikacji globalnej, czyli te, które mają więcej niż jeden parametr wejściowy. Funkcje weryfikacji lokalnej - z jednym parametrem wejściowym - są dostępne w zakładce Kryterium poprawności (DLL) okna

właściwości pól rozpoznawanych (patrz punkt 7.8.6 - Funkcje weryfikacji z zewnętrznych bibliotek DLL). Funkcja, którą wybierzemy ustawiając na niej pasek wyboru będzie używana jako kryterium weryfikacji globalnej.

We wszystkich trzech przypadkach istotnym fragmentem procesu dodawania kryterium powinno być sprawdzenie jego działania.

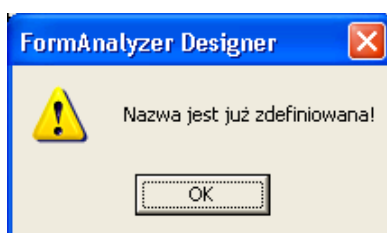
Po wybraniu funkcji weryfikacji możemy przejść na następną stronę kreatora, na której zdefiniujemy podstawowe właściwości kryterium poprawności.

Poniżej przedstawiono stronę podstawowych właściwości skryptu weryfikacji. Na tej stronie użytkownik podaje nazwę nowego skryptu, wybiera miejsce uruchomienia skryptu oraz wskazuje czy skrypt ma aktualizować wartość pól.



Rys. 12-4 Kreator kryteriów poprawności – strona parametrów podstawowych skryptu.

Nazwa skryptu nie może być pusta. Nie może też być taka sama jak nazwa skryptu zdefiniowanego już dla wybranego szablonu dokumentu. Obydwa te błędy są sygnalizowane komunikatem o błędzie przedstawionym na .



Rys. 12-5 Komunikat o powtórzeniu się nazwy tworzonego kryterium

Skrypty globalne korzystają najczęściej z wartości wielu rozpoznawanych pól dokumentu. Skrypt może służyć jedynie weryfikacji wartości pól (sprawdzeniu jej zgodności z wartościami innych rozpoznanych pól dokumentu). Wartością skryptu jest w tym przypadku jedynie informacja, czy zadane w treści skryptu kryteria poprawności są spełnione, czy też nie.

Skrypt może zostać też wykorzystany do modyfikacji wartości pola. Uzyskuje się to dzięki zaznaczeniu opcji *Czy kryterium ma aktualizować wartość pola?*. Uaktualnienie wartości pola (pól) jest wykonywane tylko w przypadku, gdy nowa wartość - wyznaczona w czasie wykonywania skryptu - spełnia zadane w nim kryteria poprawności. Inaczej mówiąc, wartość pól dokumentu może się zmienić tylko wtedy, gdy skrypt nie zakończył wykonania zgłoszeniem błędu.

Wartości pól dokumentu nie są również uaktualniane jeżeli wartość zmiennej przypisanej polu nie uległa zmianie w trakcie wykonania skryptu.

Po ustawieniu podstawowych parametrów skryptu możemy przejść na następną stronę kreatora, na której możemy wybrać pozostałe parametry skryptu.

W przypadku kryterium poprawności realizowanego jako funkcja z biblioteki DLL, strona właściwości podstawowych, przedstawiona na poniższym rysunku, jest bardzo podobna do omówionej już strony dla skryptu.

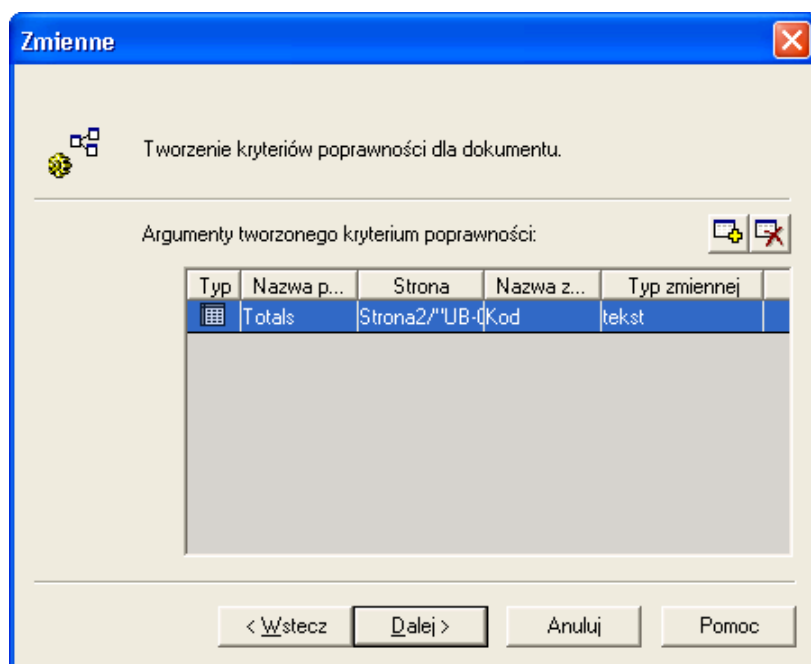
Rys. 12-6 Kreator kryteriów poprawności – właściwości podstawowe funkcji z biblioteki DLL.

Opcje *Kiedy zastosować kryterium?* oraz *Czy kryterium ma aktualizować wartość pola?* są nieaktywne. Stan tych opcji zależy od wybranej funkcji. To czy funkcja modyfikuje wartość pola, czy też nie, zależy od sposobu jej implementacji w bibliotece DLL.

Sposób wypełniania nazwy kryterium poprawności jest taki sam jak dla skryptu: nazwa musi być unikalna (nie może być także pusta).

Po ustawieniu podstawowych parametrów kryterium poprawności, korzystającego z biblioteki DLL, możemy przejść na następną stronę kreatora, na której możemy wybrać pozostałe parametry kryterium.

Kolejna strona kreatora kryteriów poprawności służy do wskazania zmiennych wejściowych (argumentów) kryterium. Sposób obsługi okna jest niezależny od typu kryterium.



Rys. 12-7 Kreator Kryteriów poprawności – strona pól dokumentu używanych przez procedurę weryfikacji.

W liście Zmienne dla tworzonego kryterium poprawności jest podana lista pól dokumentu, które będą wykorzystywane przez procedurę weryfikacji. Jest to innymi słowy lista argumentów (zmiennych wejściowych) skryptu lub funkcji weryfikacji. W liście są umieszczone następujące informacje:

Typ typ pola rozpoznawanego oznaczony ikoną.

Nazwa pola nazwa pola dokumentu.

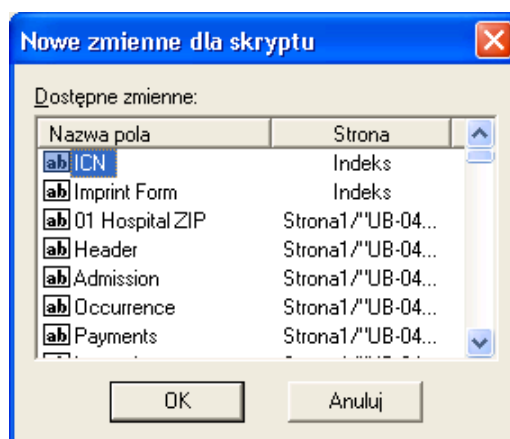
Strona opis strony w szablonie.

Nazwa zmiennej nazwa zmiennej używana w procedurze weryfikacji.

W przypadku skryptu nazwa zmiennej jest początkowo taka sama jak nazwa pola. Można ją zmienić klikając na nazwie wybranego elementu - pole nazwy zmiennej przejdzie do trybu edycji. Dla funkcji weryfikacji nazwa jest pobierana z biblioteki DLL – nie możemy jej zmieniać.

Typ zmiennej typ zmiennej w skrypcie (funkcji).

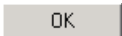
Nowe elementy do listy argumentów dodaje się w oknie nowych. To okno otwiera naciśnięcie



Rys. 12-8 Okno wyboru argumentów skryptu globalnego.

W tabeli Dostępne zmienne są umieszczone nazwy pól dokumentu, które mogą być użyte jako argumenty procedury weryfikacji. W tabeli nie są umieszczane pola już używane jako argumenty skryptu.

W tabeli można zaznaczyć więcej niż jedno pole. Stosuje się tu ogólne zasady wybierania elementów w Windows. Żeby zaznaczyć pojedynczy element trzeba kliknąć na nazwie pola. Klikając z klawiszem Shift zaznaczamy zakres elementów, natomiast z Ctrl dodajemy lub ujmujemy (gdy był zaznaczony) element z listy zaznaczonych.

Naciśnięcie  spowoduje zamknięcie okna nowych zmiennych i dodanie do listy argumentów skryptu wszystkich zaznaczonych pól.  zamyka okno nie zmieniając listy argumentów skryptu.

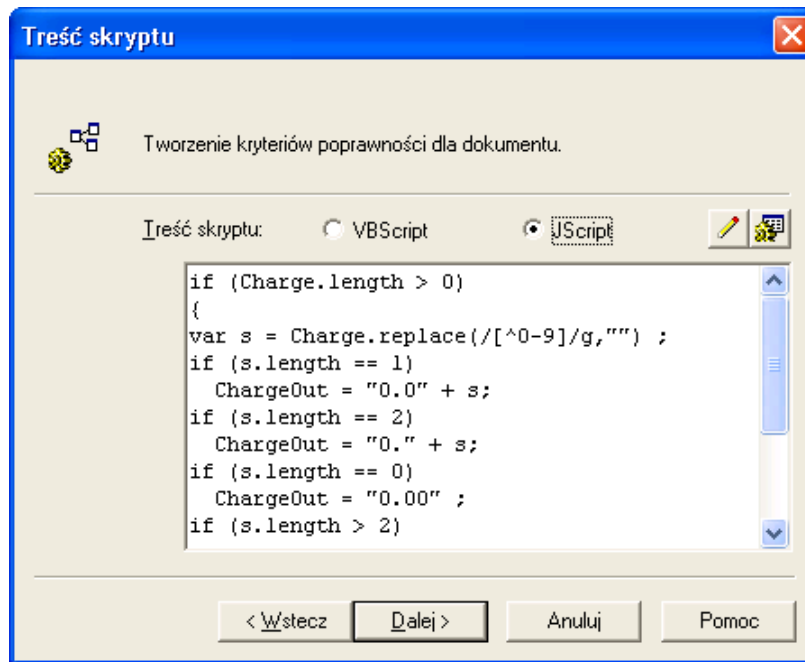
W przypadku funkcji z biblioteki DLL lub WebService, liczba i nazwy argumentów określa definicja odpowiedniej procedury. Musimy wybrać dokładnie tyle pól ile argumentów ma wykorzystywana funkcja. Argumenty muszą być przy tym odpowiednich typów.

Wybierając argumenty dla skryptu mamy większą swobodę - treść skryptu nie jest jeszcze napisana. Możemy, zatem zdefiniować dowolną liczbę parametrów.

Przycisk  służy do usuwania zaznaczonego argumentu procedury weryfikacji. W przypadku, gdy usuwamy argumenty skryptu, którego treść jest już napisana, trzeba pamiętać, że zmiana listy zmiennych wejściowych nie wiąże się ze sprawdzeniem poprawności skryptu. Jeśli usuniemy argument wykorzystywany w skrypcie, to komunikat o błędzie zostanie wyświetlony dopiero przy sprawdzaniu działania skryptu (lub, kiedy działania skryptu nie sprawdziliśmy w FormAnalyzer Designer, w czasie wykonania skryptu).

Jeżeli procedura weryfikacji modyfikuje wartości pól to kolejna strona kreatora umożliwia przyporządkowanie pól szablonu do zmiennych wyjściowych kryterium. Okno służące wskazaniu zmiennych wyjściowych jest analogiczne jak to używane przy wskazaniu argumentów.

W przypadku skryptu, po wybraniu argumentów i zmiennych wyjściowych procedury weryfikacji, musimy napisać jeszcze jego treść. Przechodzimy na ostatnią stronę kreatora skryptów klikając przycisk .



Rys. 12-9 Kreator skryptu – strona treści skryptu.

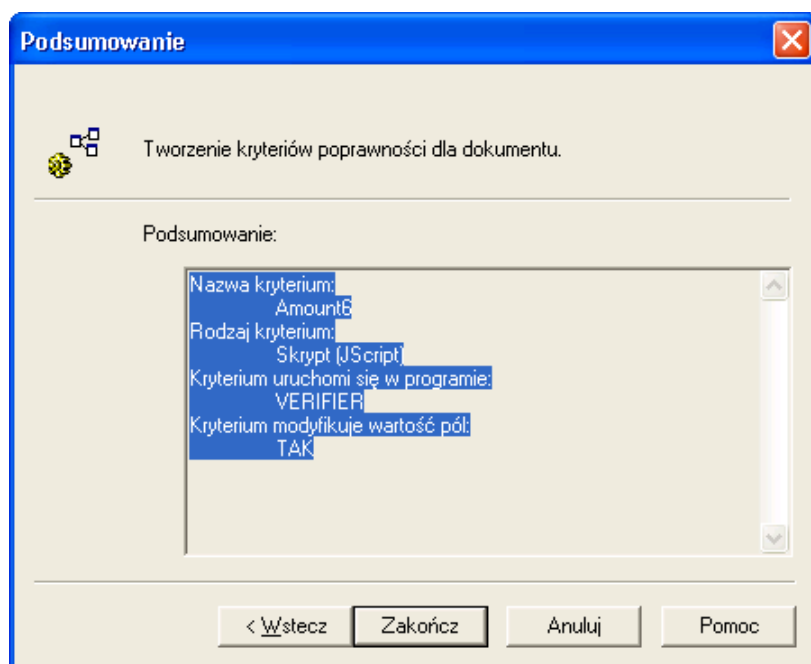
Edytor Treść skryptu jest dla nowego skryptu pusty. Przyjmuje się, że skrypt o pustej treści zawsze kończy się poprawnie.

W celu wprowadzania lub zmiany treści skryptu należy kliknąć przycisk . Pojawi się okno *Edytor Skryptu*. W edytorze wprowadzamy treść skryptu w języku VBScript lub JScript. Bardziej szczegółowe omówienie zasad pisania skryptów dla systemu FormAnalyze znajduje się w kolejnych punktach. Edycję skryptu kończymy wykonując polecenie Plik -> Zapisz z menu okna a następnie zamykając okno *Edytora*.

Po zakończeniu pisania treści skryptu musimy sprawdzić jego działanie.

Sprawdzenie działania skryptu omówiono w następnym punkcie.

Ostatnim oknem kreatora kryterium poprawności jest okno *Podsumowanie*.



Rys. 12-10 Kreator skryptu – strona podsumowania.

Kliknięcie przycisku *Zakończ* kończy definiowanie kryterium poprawności.

12.3. Narzędzie Edytor Skryptu

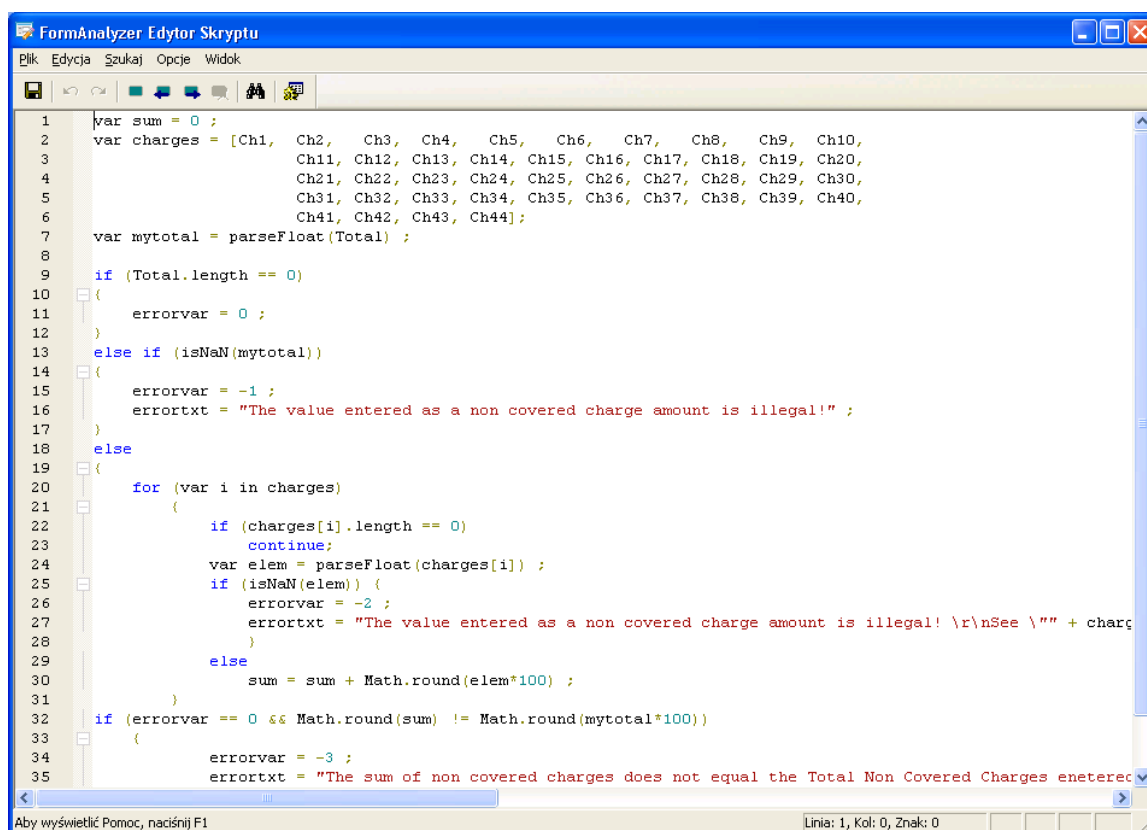
Narzędzie *Edytor Skryptu* służy do edycji skryptów wykorzystywanych do tworzenia kryteriów poprawności danych (skrypty w języku JScript lub VBScript) .

To samo narzędzie służy do tworzenia i edycji skryptów lokalnych, przypisanych do poszczególnych pól formularza, oraz skryptów XSLT wykorzystywanych podczas tworzenia plików eksportu.

Okno *Edytora* dostępne jest po kliknięciu przycisku  w oknie *Kreatora skryptu* lub przycisku  w oknie *Kryteria poprawności*.

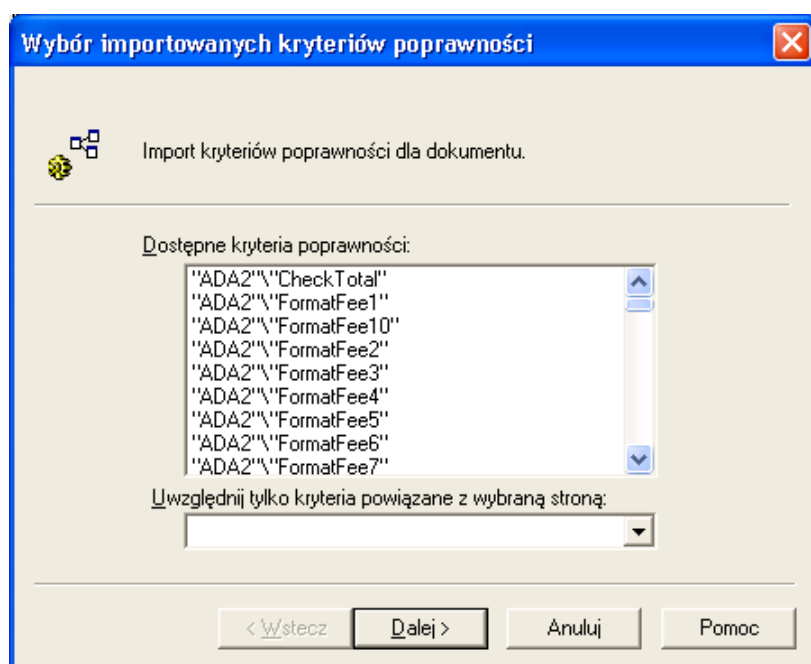
Edytor wspomaga tworzenie i edycję skryptów poprzez:

- podświetlenie składni języków skryptowych (JScript, VBScript, XSLT),
- wyszukiwanie i zamianę ciągów (w tym za pomocą wyrażeń regularnych),
- numerowanie linii,
- umieszczanie i wyszukiwanie znaczników,
- zwijanie – rozwijanie bloków tekstu,
- wielopozycyjną operację cofnij – ponów,
- możliwość uruchomienia skryptu w celu przetestowania poprawności jego działania.

Rys. 12-11 Okno narzędzia *Edytor Skryptu*.

12.4. Importowanie kryteriów poprawności

W przypadku dokumentów składających się z wielu takich samych stron bardzo przydatną funkcją jest importowanie globalnych kryteriów poprawności. Ponieważ globalne kryteria poprawności są związane z szablonem dokumentu, a nie formularzem, powielenie formularza nie powoduje powielenia globalnych kryteriów poprawności nawet, gdy są one związane tylko z polami występującymi na tym formularzu. Importowanie kryteriów poprawności jest wykonywane jedynie w obrębie bieżącego pliku konfiguracji. Importowanie rozpoczynamy naciskając przycisk  w oknie kryteriów poprawności dokumentów. Kryterium poprawności zostanie skopiowane do zaznaczonego szablonu dokumentu. Program wyświetla okno pokazane na poniższym rysunku.

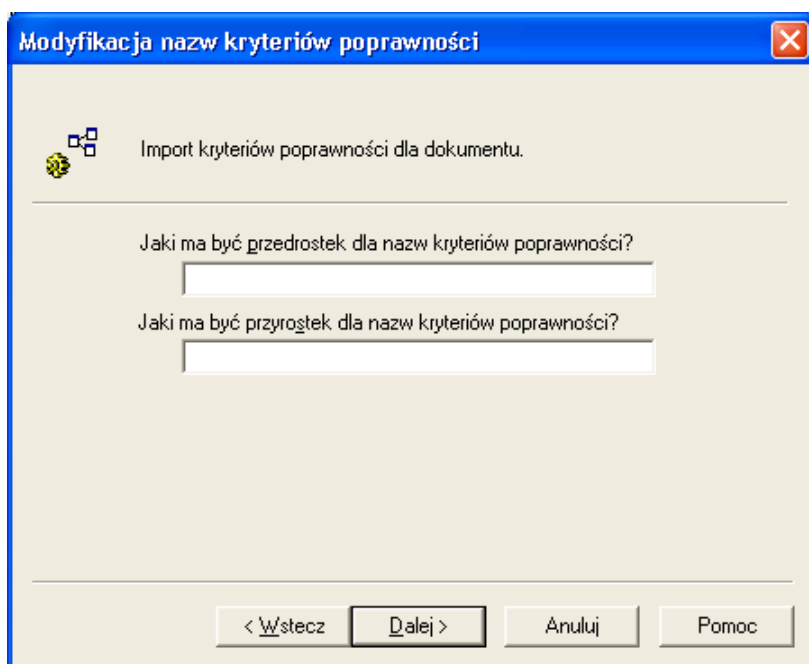


Rys. 12-12 Okno wyboru importowanych kryteriów poprawności.

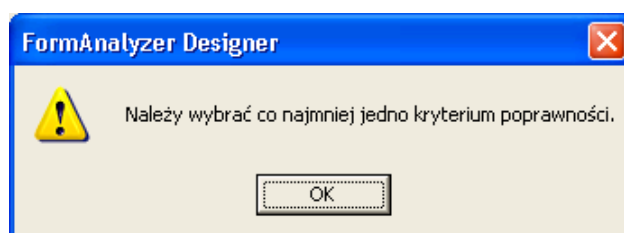
W liście *Dostępne kryteria poprawności* są wyświetlane kryteria poprawności, które mogą być importowane. Możemy ograniczyć liczbę kryteriów wybierając tylko kryteria powiązane z konkretną stroną szablonu dokumentu. W tym celu w liście *Uwzględnij tylko kryteria powiązane z wybraną stroną* wybieramy odpowiednią stronę konkretnego szablonu dokumentu. Na liście dostępnych kryteriów poprawności pozostaną tylko te kryteria, które są związane z polami znajdującymi się na wybranej stronie. Przypomnijmy, że kryterium poprawności jest związane tylko z jednym polem, nawet, gdy korzysta z wielu zmiennych. To pole rozpoznawane zostało wybrane w oknie cech podstawowych kryterium poprawności w polu *Z jakim polem związać kryterium poprawności?*.

Po zaznaczeniu kryteriów poprawności, które mają być importowane, przechodzimy do pokazanego na poniższym rysunku okna modyfikacji nazw importowanych kryteriów naciskając .

Naciśnięcie  przy niewybraniu jakiegokolwiek kryterium spowoduje pojawienie się komunikatu przedstawionego na rysunku 12-15.



Rys. 12-13 Okno modyfikacji nazw kryteriów poprawności.



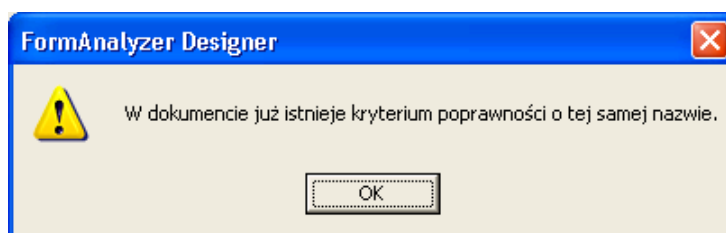
Rys. 12-14 Komunikat o nie zaznaczeniu żadnego kryterium poprawności.

W oknie tym można podać przedrostek i/lub przyrostek, który zostanie dodany do nazw kryteriów w czasie importu. Możliwość modyfikacji nazwy ma podstawowe znaczenie w sytuacji, gdy importujemy kryteria poprawności w obrębie tego samego szablonu dokumentów. Kryteria poprawności muszą mieć w szablonie unikalne nazwy, dlatego bez modyfikacji nazwy nie można byłoby importować kryteriów z tego samego szablonu.

Najczęściej przyjmuje się zasadę, że na pierwszej ze stron, na których używa się importowanych kryteriów poprawności ich nazwy nie mają przedrostków i przyrostków. Na kolejnych stronach, na których będą wykorzystywane kryteria dodaje się przyrostek (albo przedrostek) z oznaczeniem strony, np. _str2.

Przy importowaniu kryteriów z innych szablonów modyfikacja nazwy może nie być konieczna.

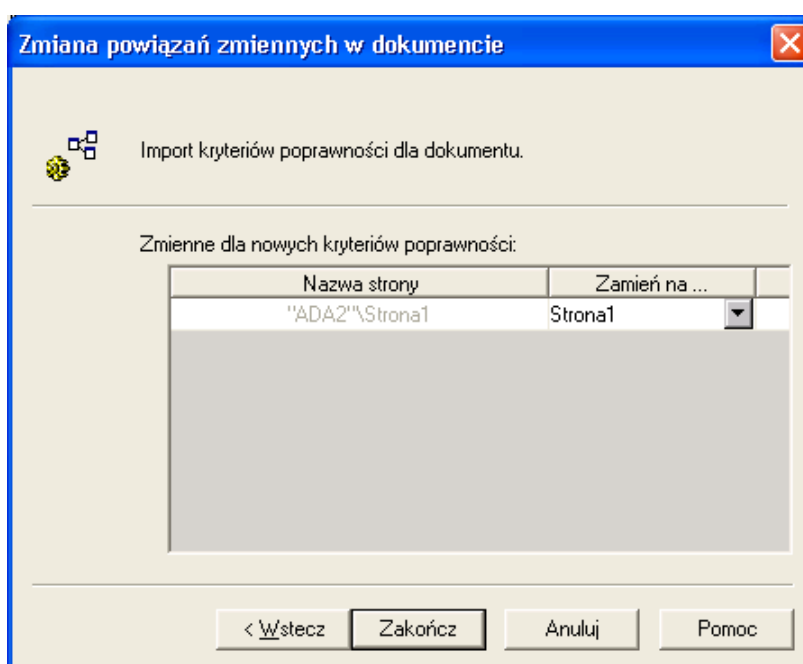
Przed przejściem do ostatniej strony importowania kryteriów program sprawdza, czy jest zachowana unikalność nazw kryteriów. Jeśli nie, jest wyświetlany komunikat pokazany na poniższym rysunku.



Rys. 12-15 Komunikat o konflikcie nazw importowanych kryteriów poprawności.

W tym przypadku pozostajemy na stronie modyfikacji nazwy, żeby podać przedrostek lub przyrostek modyfikujący nazwy kryteriów.

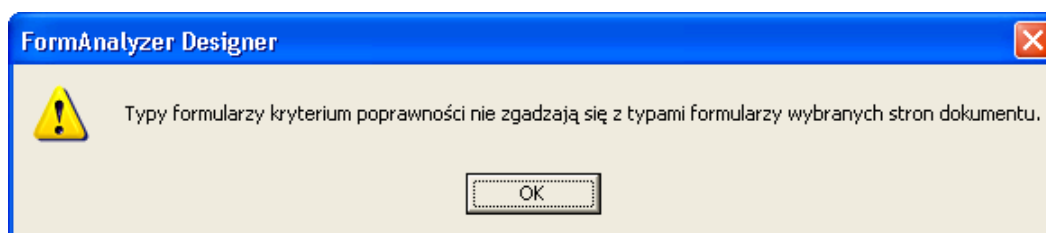
Na ostatniej stronie importowania kryteriów poprawności, pokazanej na, wskazujemy, na której stronie szablonu dokumentu znajdują się zmienne używane przez kryterium poprawności.



Rys. 12-16 Wskazanie strony, na której będą poszukiwane zmienne dla kryterium poprawności.

Bardzo istotne jest to, że obydwie strony: źródłowa, czyli używana przez kryterium importowane, podana w kolumnie *Nazwa strony*, oraz docelowa, czyli ta, która będzie używana przez nowe kryterium po imporcie, podana w kolumnie *Zamień na* muszą oznaczać ten sam formularz.

W kolumnie *Zamień na* wybieramy odpowiednie strony docelowe. Po naciśnięciu  program wykonuje import kryteriów poprawności. Pierwszym krokiem jest sprawdzenie typów formularzy stron źródłowych i docelowych. Jeśli warunek identyczności typów formularzy nie jest spełniony, program wyświetla komunikat pokazany na rysunku poniżej.

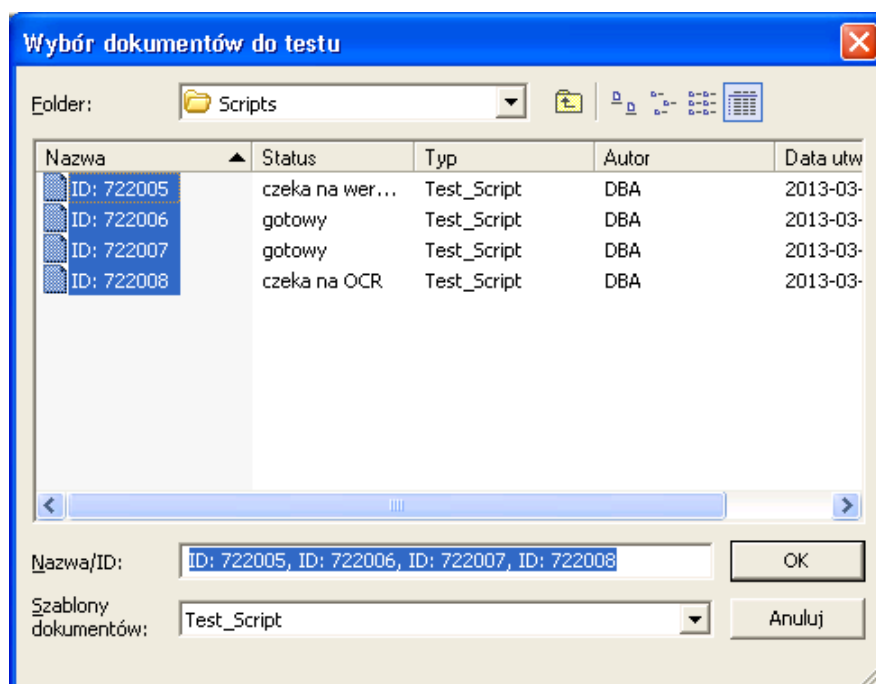


Rys. 12-17 Komunikat o niezgodności typów formularzy źródłowego i docelowego przy imporcie kryteriów poprawności.

Jeśli typy formularzy są takie same program wykonuje import kryteriów poprawności.

12.5. Testowanie zaawansowane

W przypadku kryteriów poprawności zawierających dużą liczbę zmiennych wejściowych/wyjściowych lub w celu sprawdzenia działania skryptów bardzo przydatną funkcją jest testowanie zaawansowane. Funkcja ta umożliwia wybranie dokumentów z bazy danych systemu FormAnalyzer i sprawdzenie na danych zapisanych w rezultatach rozpoznawania/weryfikacji działania kryteriów poprawności. Testowanie zaawansowane rozpoczynamy naciskając przycisk  w oknie kryteriów poprawności dokumentów. Na początku zostanie wyświetlone okno dialogowe do wyboru dokumentów do testu Rys. 12-18. W oknie tym wyświetlone są jedynie dokumenty, które mają typ zgodny z aktualnie edytowanym szablonem dokumentu. Użytkownik może zaznaczyć jeden lub więcej dokumentów do testu.



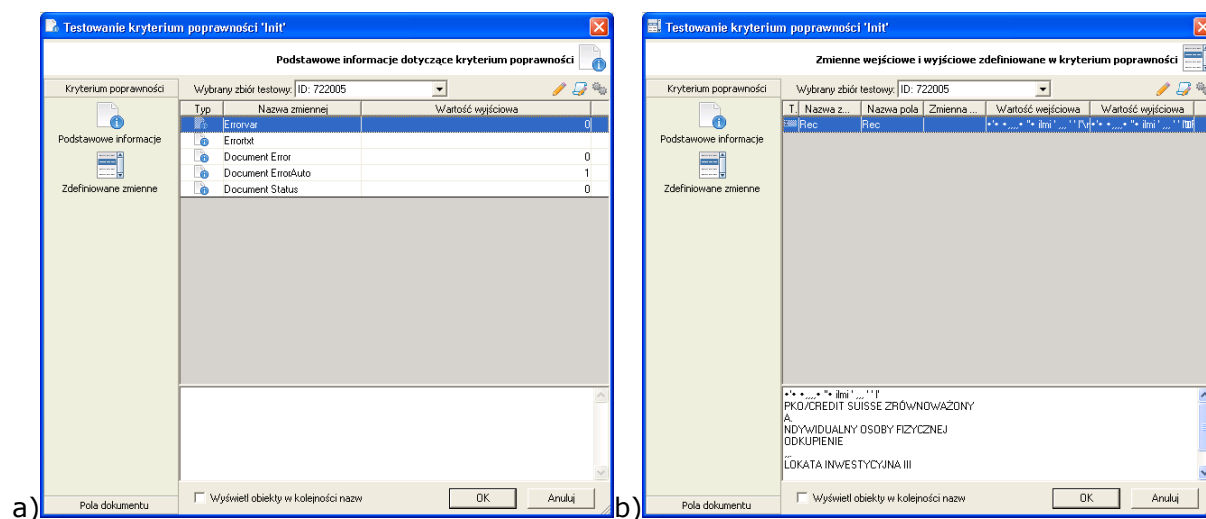
Rys. 12-18 Okno dialogowe z wyborem dokumentów do testu.

Po zaakceptowaniu wyboru poprzez naciśnięcie przycisku , program odczyta wyniki przetwarzania zaznaczonych dokumentów z bazy danych, a następnie wyświetli okno dialogowe przedstawione na Rys. 12-19 i na Rys. 12-20.

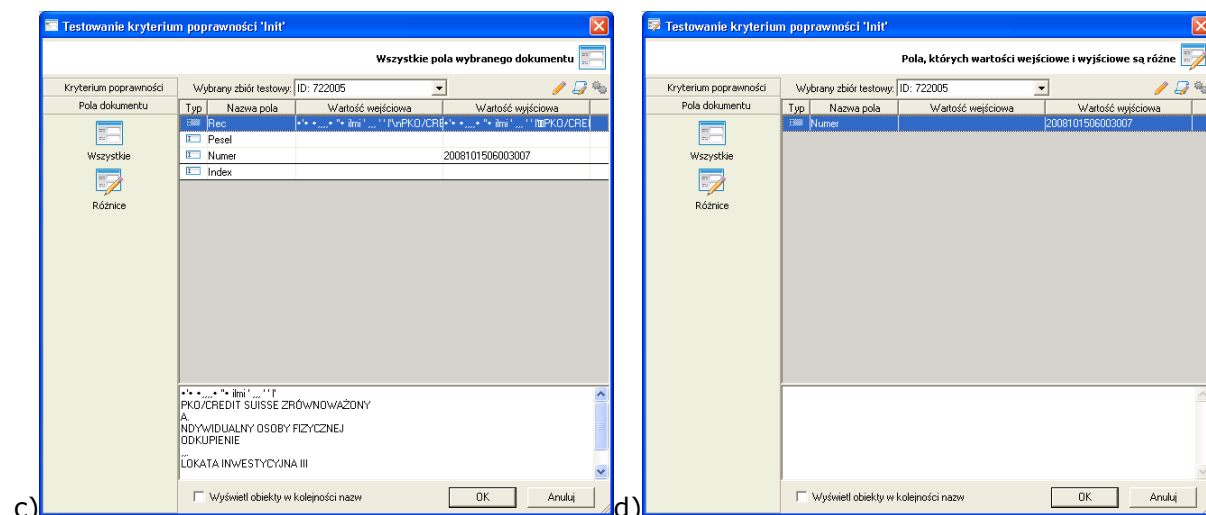
W liście *Wybrany zbiór testowy* są dostępne nazwy wszystkich wybranych dokumentów do testu. Użytkownik może się pomiędzy nimi przełączać, sprawdzając różnice w

działaniu kryterium poprawności dla konkretnych dokumentów – danych wejściowych. W pasku w lewej części okna dialogowego użytkownik może wskazać jakiego rodzaju informacje chce wyświetlać:

- podstawowe informacje o przetwarzaniu: kod wykonania kryterium poprawności - *Errorvar*, ustawiony komunikat o stanie przetwarzania - *Errortxt*, kod błędu dokumentu - *Document Error*, kod automatycznej weryfikacji - *Document ErrorAuto*, status dokumentu - *Document Status* (Rys. 12-19a)
- informacje o zdefiniowanych zmiennych (Rys. 12-19b)
- lista wszystkich pól dokumentu (Rys. 12-20c)
- lista zmienionych pól dokumentu (Rys. 12-20d)



Rys. 12-19 Okno dialogowe Testowanie kryterium poprawności – podstawowe informacje.

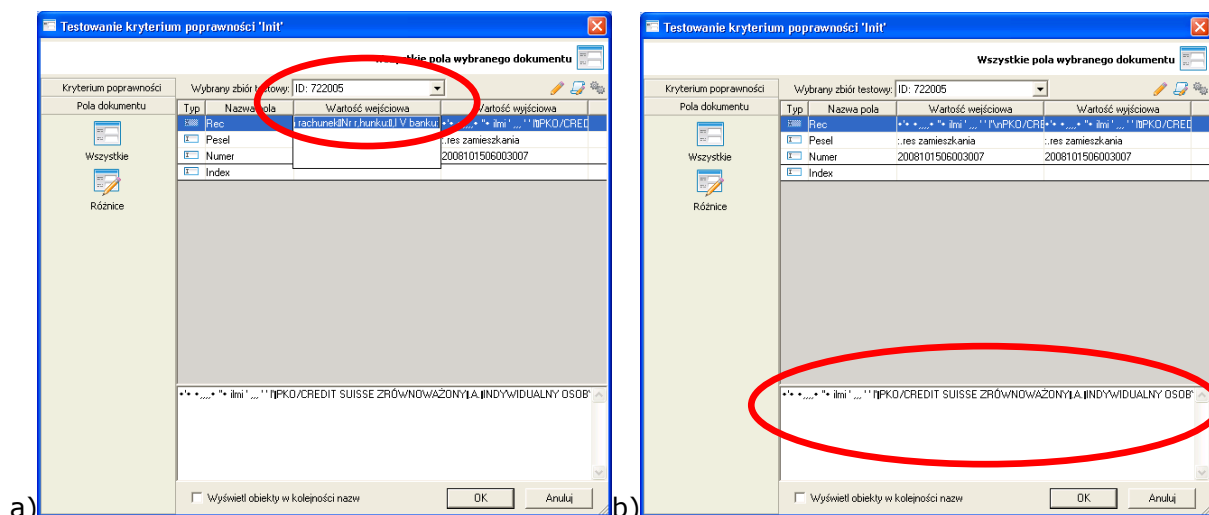


Rys. 12-20 Okno dialogowe Testowanie kryterium poprawności – informacje o dokumencie.

Wartości w kolumnie *Wartość wejściowa* są inicjowane odczytanymi wynikami z bazy danych. Wartości te można zmienić poprzez wprowadzenie w tryb edycji odpowiedniej komórki z danymi (Rys. 12-21a). Istnieje także możliwość zmiany wartości przez dokonanie modyfikacji w oknie edytora, który znajduje się na dole okna dialogowego (Rys. 12-21b).

Modyfikację właściwości kryterium poprawności umożliwia przycisk . Te zmiany są wykonywane w kreatorze kryterium poprawności omówionym w rozdziale Rys. 12-2. Jeżeli mamy do czynienia ze skryptem dostępna jest także funkcja , która wyświetla okno z treścią skryptu. Opis edycji skryptów został przedstawiony w rozdziale 12.3. Dla innych bibliotek DLL i funkcji WebService funkcja ta nie jest dostępna.

Przycisk  uruchamia kryterium poprawności ze zmodyfikowanymi danymi, które wyświetlone są w kolumnie *Wartość wejściowa*. Po zakończeniu tego procesu wyniki przetwarzania wyświetlane są w oknie dialogowym.



Rys. 12-21 Edycja wartości wejściowych.

Rozmiar okna można zmienić, poprzez ciągnięcie prawego-dolnego rogu okna. Rozmiar jest zapisany w rejestrze systemu Windows. Przy kolejnym wyświetleniu tego okna rozmiar jest odtwarzany.

12.6. Skrypty w FormAnalyze Designer

Skrypty są programami pisanymi w języku VBScript lub JScript. Przedstawienie zasad programowania w tych językach wykracza poza zakres tego podręcznika. Istnieje wiele książek poświęconych programowaniu w obu językach.

W tym punkcie zostaną przedstawione jedynie specyficzne dla systemu FormAnalyze właściwości skryptów.

Skrypty w systemie FormAnalyze posiadają poniższe cechy:

- Język skryptu:
 - VBScript,
 - JScript.
- Zasięg skryptu:
 - Skrypty lokalne – są to skrypty związane z pojedynczym polem formularza,
 - Skrypty globalne – są to skrypty odwołujące się do dowolnej liczby pól szablonu.
- Miejsce uruchomienia:
 - Skrypt uruchamiany w programie FA Engine,
 - Skrypt uruchamiany w programie FA Verifier.
- Sposób działania:

- Skrypty weryfikujące dane,
- Skrypty modyfikujące dane.

Zastosowanie niektórych z powyższych cech narzuca stosowanie pozostałych i tak:

- Skrypty weryfikujące dane uruchamiane są tylko w programie FA Verifier.
- Skrypty lokalne, które modyfikują dane uruchamiane są tylko w programie FA Engine.

Przed uruchomieniem, skrypt uzupełniany jest o deklarację zmiennych globalnych. Są to zmienne: *errorvar*, *errortxt*, oraz zmienne przypisane polom formularzy.

12.6.1. Zmienne *errorvar* i *errortxt*.

Reakcja programu Form Analyzer na zakończenie wykonania skryptu zależy od wartości dwóch zmiennych globalnych: *errorvar* i *errortxt*.

Zmienna *errorvar* przyjmuje wartości z zakresu liczb całkowitych (zmienna typu *integer*) natomiast zmienna *errortxt* jest zmienną tekstową (zmienna typu *string*). Przed uruchomieniem skryptu obie zmienne są deklarowane i przypisywane są im wartości. Zmiennej *errorvar* przypisywana jest wartość 0 natomiast zmiennej *errortxt* wartość "" (pusty tekst).

Interpretacja wartości obu zmiennych zależy od sposobu działania i miejsca uruchomienia skryptu.

W przypadku skryptu weryfikującego dane, przypisanie zmiennej *errorvar* wartości równej 0 oznacza, że weryfikowane pola zawierają poprawne wartości (wartość zmiennej *errortxt* jest wtedy ignorowana). Przypisanie zmiennej *errorvar* wartości różnej od 0 oznacza, że wartości weryfikowanych pól są niepoprawne. W tym przypadku wartość przypisana zmiennej *errortxt* traktowana jest jako tekst komunikatu informującego o przyczynie błędu i wyświetlana w postaci komunikatu użytkownikowi końcowemu.

W przypadku skryptu modyfikującego dane przypisanie zmiennej *errorvar* wartości równej 0 oznacza, że skrypt zakończył działanie poprawnie. Wartość różna od 0 oznacza, że podczas wykonywania skryptu wystąpił błąd. Sygnalizacja błędu poprzez przypisanie zmiennej *errorvar* wartości różnej od 0 powoduje, że skrypt nie modyfikuje wartości pól szablonu. Wartość zmiennej *errortxt* traktowana jest jako komunikat informujący o przyczynie wystąpienia błędu.

12.6.2. Zmienne przypisane polom szablonu

Celem działania skryptu jest weryfikacja wartości przypisanych polom szablonu i ewentualna zmiana wartości tych pól. Przekazywanie wartości pomiędzy polami formularza a skryptem odbywa się poprzez zmienne globalne. Tworząc kryterium poprawności typu skrypt, użytkownik wskazuje pola formularza do których skrypt ten będzie się odwoływał. Każdemu polu przypisywana jest zmienna. Wskazywane są zmienne wejściowe (nazywane również argumentami skryptu) i wyjściowe. Zmienne wyjściowe wskazywane są tylko w przypadku skryptów modyfikujących dane.

W przypadku skryptów lokalnych (są to skrypty związane z pojedynczym polem formularza) polu formularza przypisywana jest zmienna globalna o nazwie X.

Zwróćmy uwagę na fakt, że nie zawsze nazwa zmiennej może odpowiadać nazwie pola w definicji szablonu. Nazwy zmiennych nie mogą zaczynać się od cyfry i nie mogą zawierać

w sobie spacji. W przypadku, gdy nazwa pola nie spełnia tych kryteriów, należy zmienić nazwę zmiennej.

Podobnie jak w przypadku zmiennych *errorvar* i *errortxt* zmienne przypisane polom formularza są deklarowane (jako zmienne globalne) i inicjowane przed wykonaniem pierwszej linii skryptu.

12.6.3. Typy zmiennych

Zmienna przypisana do pola formularza może być jednego z 3 typów. Typ zmiennej zależy od rodzaju pola z którym została związana:

Logiczny	Przyjmuje wartości <i>True</i> lub <i>False</i> . Typ logiczny przypisywany jest tylko polom wyboru. W przypadku pojedynczego pola zmienna jest wartością skalarną. W przypadku grupy pól zmienna jest tablicą wartości typu Logicznego. Tablica ta jest indeksowana kolejnymi liczbami całkowitymi. Indeksowanie rozpoczyna się od 0 w przypadku JScript lub 1 w przypadku VBScript. Rozmiar tablicy odpowiada liczbie pól wyboru w grupie.
Numeryczny	Przyjmuje wartości z zakresu liczb całkowitych (integer). Przypisywany jest polom dla których na zakładce <i>Przetwarzanie końcowe</i> zaznaczono opcję <i>Liczba</i> .
Tekstowy	Wartość zmiennej traktowana jest jako ciąg znaków (string). Typ przypisywany wszystkim pozostałym rodzajom pól.

W dalszej części tego punktu podamy bardziej szczegółowe zasady dotyczące sposobów przekazywania wartości argumentów do skryptów. Przedstawione poniżej zasady są stosowane w tym samym stopniu do skryptów lokalnych jak i globalnych.

12.7. Wbudowane funkcje rozszerzające możliwości skryptów

System FormAnalyzer umożliwia wywołanie wbudowanych funkcji podczas wykonania skryptów. Funkcje te umożliwiają następujące operacje:

- Pobranie informacji o dokumencie i stronie
- Zmiana podstawowych informacji w dokumencie
- Pobranie informacji o dowolnym polu z dokumentu
- Zmianę wartości dowolnego pola w dokumencie
- Efektywne wyszukanie w polu dokumentu zadanej frazy tekstu, uwzględniając błędy rozpoznania

Poniżej został zaprezentowana pełna lista funkcji, które można wywoływać podczas rozpoznawania i weryfikacji. Modyfikacja wartości w procesie weryfikacji powinna odbywać się poprzez zmienne wyjściowe.

```
/*
    Funkcja wyświetla okno z wersją skryptów.
*/
void FAHelpAbout(void);

/*
    Funkcja zwraca obiekt z informacjami o bieżącej paczce dokumentów:
    Obiekt zawiera następujące pola:
        boxid - identyfikator paczki w bazie danych
        doccnt - liczba dokumentów w paczce
        docnum - numer aktualnie przetwarzanego dokumentu (liczony od 0)
        name - nazwa box-a
        folder - ścieżka folderów
*/
LPDISPATCH FAGetBoxInfo(void);

/*
    Funkcja zwraca obiekt z informacjami o dokumencie z pozycji 'docnum'
    w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 zwracane informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        obid - identyfikator dokumentu w bazie danych
        docid - identyfikator dokumentu w bazie danych
        pagecnt - liczba stron w dokumencie
        error - kod błędu dokumentu
        status - wejściowy status dokumentu
        type - nazwa szablonu dokumentu
        name - nazwa boxa
        folder - pełna ścieżka boxa
*/
LPDISPATCH FAGetDocInfo(short docnum);

/*
    Funkcja zmienia podstawowe informacje o dokumencie z pozycji 'docnum'
    w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        error - kod błędu dokumentu
        errorauto - kod błędu automatycznej weryfikacji
        status - status dokumentu
*/
void FASetDocInfo(short docnum, short error, short errorauto, short status);

/*
    Funkcja zwraca obiekt z informacjami o stronie z pozycji 'pagenum'
    z dokumentu 'docnum' w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 zwracane informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        pageid - identyfikator strony w bazie danych
        rotate - kąt obrotu
        empty - znacznik pustości strony
        formula - identyfikator formularza
        pagePath - ścieżka do pliku na wolumenie
        processedPath - ścieżka do pliku w trybie rozpoznawania
        endorser - wartość endorsera
*/
LPDISPATCH FAGetPageInfo(short pagenum, short docnum);

/*
    Funkcja zmienia podstawowe dane strony 'pagenum' dokumentu z pozycji 'docnum'
    w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 informacje odnoszą się do bieżącego dokumentu):
        rotate - kąt obrotu (-90, 0, 90, 180)
        empty - znacznik pustości strony (strona pusta gdy wartość <> 0)
        endorser - wartość endorsera
*/
void FASetPageInfo(short pagenum, short docnum, short rotate, short empty, LPCSTR endorser);
```

```
/*
    Funkcja zwraca obiekt z informacjami o pierwszym polu zdefiniowanym
    w kryterium poprawności.
    Obiekt zawiera następujące pola:
        name - nazwa pola
        page - numer strony na której jest pole
        value - wartość pola
        cl - poziom zaufania wyników
        left, top, right, bottom - pozycja pola
        pos - identyfikator pola, który może być wykorzystywany
            w innych wbudowanych funkcjach
*/
LPDISPATCH FAGetCurrField();

/*
    Funkcja zwraca obiekt z informacjami o polu o nazwie 'name' z pozycji 'pagenum'
    z dokumentu 'docnum' w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 zwracane informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        name - nazwa pola
        page - numer strony na której jest pole
        value - wartość pola
        cl - poziom zaufania wyników
        left, top, right, bottom - pozycja pola
        pos - identyfikator pola, który może być wykorzystywany
            w innych wbudowanych funkcjach
*/
LPDISPATCH FAGetField(LPCSTR name, short pagenum, short docnum);

/*
    Funkcja zwraca obiekt z informacjami o pierwszym polu, którego nazwa zawiera ciąg
    'subname' z pozycji 'pagenum' z dokumentu 'docnum' w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 zwracane informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        name - nazwa pola
        page - numer strony na której jest pole
        value - wartość pola
        cl - poziom zaufania wyników
        left, top, right, bottom - pozycja pola
        pos - identyfikator pola, który może być wykorzystywany w innych wbudowanych
            funkcjach (jeżeli 'pos' < 0 oznacza, że pole nie zostało znalezione)
*/
LPDISPATCH FAFindField(LPCSTR subname, short pagenum, short docnum);

/*
    Funkcja zwraca obiekt z informacjami o następnym słowie
    (niepusty ciąg oddzielony białymi znakami)
    w wynikach pola o identyfikatorze 'fieldID'.
    Zmienna 'posWord' definiuje pozycję od której szukane jest kolejne słowo.
    Wartość zmiennej <= 0 oznacza rozpoczęcie przeszukiwania od początku.
    Zmienna 'lpszToken' może zawierać separatory słów, standardowo wykorzystywane są
    białe znaki: spacja, tabulacja znak nowej linii.
    z pozycji 'pagenum' z dokumentu 'docnum' w bieżącej paczce dokumentów.
    (jeżeli 'docnum' < 0 zwracane informacje odnoszą się do bieżącego dokumentu)
    Obiekt zawiera następujące pola:
        name - nazwa pola
        value - wartość pola
        cl - poziom zaufania wyników
        left, top, right, bottom - pozycja pola
        pos - identyfikator słowa
            (jeżeli 'pos' < 0 oznacza, że słowo nie zostało znalezione)
*/
LPDISPATCH FAGetNextWord(long fieldID, long posWord, LPCSTR lpszToken);

/*
    Funkcja zmienia informacje o polu o identyfikatorze 'fieldID':
    wartość, poziom zaufania, współrzędne (jeżeli rLeft < 0 współrzędne nie są zmieniane).
*/
void FASetField(long fieldID, LPCSTR strValue, short nCL,
    long rLeft, long rTop, long rRight, long rBottom);
```

```
/*
    Funkcja zapisuje zmiany dokonane w skrypcie.
    Wszystkie pobrane obiekty są zwalniane.
*/
void FAAcceptFields(void);

/*
    Funkcja ignoruje zmiany dokonane w skrypcie.
    Wszystkie pobrane obiekty są zwalniane.
    Funkcja nie jest obligatoryjna (zwalnianie pamięci odbywa się po wykonaniu skryptu).
*/
void FARollbackFields(void);

/*
    Funkcja ta inicjuje proces wyszukiwania wzorców tekstu w polu
    o identyfikatorze 'fieldID'.
    Wyszukiwanie nie rozróżnia dużych i małych znaków.
    Wyszukiwanie rozpoczyna się od znaku na pozycji 'posChar'
    (jeżeli wartość jest mniejsza od 0 to wyszukiwanie odbywa się od początku pola).
    Poszukiwanie tekstu jest wykonywane w kolejnych poprzednich 'maxLines' tekstu
    poczynawszy od pozycji 'posChar'.
    Wyszukiwane wzorce przekazywane są w parametrze 'lpszPatterns'.
    Wzorce oddzielone są znakiem o kodzie 0.
    Wszystkie wzorce kończą się dwoma znakami o kodzie 0.
    Wyszukiwane są tylko te frazy,
    który poziom zgodności jest większy lub równy 'nCL' (0-100).
    Znaki specjalne, które mogą być wykorzystywane w wyszukiwanych wzorcach:
        # - dowolny numeryczny znak 0-9
        $ - dowolny znak alpha a-z, A-Z
        ? - dowolny znak alphanum
        * - dowolny znak nonwhite
        \c - znak c czyli \# -> znak #, a nie dowolny znak numeryczny
        [abc] - jeden z listy znaków abc
    Funkcja zwraca identyfikator procesu wyszukiwania.
    Wartość mniejsza od 0 informuje, że wystąpił błąd.
    Wartość większa od zera oznacza,
    że funkcja znalazła co najmniej jedno wystąpienie frazy tekstu.
*/
long FAInitSearch1(long fieldID, LPCSTR lpszPatterns, short nCL);
long FAInitSearch2(long fieldID, LPCSTR lpszPatterns, short nCL,
                   long posStart, long cntChars);
long FAInitSearch3(long fieldID, LPCSTR lpszPatterns, short nCL,
                   long posStart, long cntLines);
long FAInitSearch4(long fieldID, LPCSTR lpszPatterns, short nCL, long left,
                   long top, long right, long bottom);

/*
    Funkcja ta zwraca liczbę wyszukanych fraz tekstu
    zgodnie z cechami zdefiniowanymi podczas inicjacji 'searchID'.
*/
long FAGetSearchCount(long searchID);

/*
    Funkcja zwraca obiekt z informacjami o kolejnej frazie tekstu
    spełniającej warunki podane podczas inicjacji wyszukiwania w sesji 'searchID'.
    Frazy posortowane są po 'cl' i 'idxPatt'.
    Obiekt zawiera następujące pola:
        pattern - wyszukany wzorec tekstu
        value - wyszukana fraza tekstu
        cl - poziom zgodności frazy i wzorca
        left, top, right, bottom - współrzędne frazy
        start - początkowa pozycja (znakowa) frazy tekstu w polu
            (jeżeli start < 0 oznacza to, że proces nie znalazł więcej fraz tekstu)
        end - (start + value.length())
        idxPatt - znaleziony indeks wzorca (lub -1 gdy nie znaleziono)
*/
LPDISPATCH FAGetSearch(long searchID, long idx)
```

```
/*
    Funkcja zwraca obiekt z informacjami o części znalezionego ciągu tekstu
    spełniającego warunki podane podczas inicjacji wyszukiwania w sesji 'searchID'.
    Fraza jest tekstem odpowiadającym części wzorca (od pozycji 'start' do 'end').
    Obiekt zawiera następujące pola:
        value - wyszukana fraza tekstu
        cl - poziom zgodności frazy i wzorca
        left, top, right, bottom - współrzędne frazy
        start - początkowa pozycja (znakowa) frazy tekstu w polu
                (jeżeli start < 0 oznacza to, że proces nie znalazł więcej fraz tekstu)
        end - (start + value.length())
*/
LPDISPATCH FAGetSubSearch(long searchID, long idx, int start, int end);

/*
    Funkcja zwalnia obiekt z informacjami dotyczącymi procesu
    wyszukiwania tekstu o identyfikatorze 'searchID'.
    Jeżeli 'searchID' < 0 funkcja zwalnia informację o wszystkich procesach wyszukiwania.
    Funkcja nie jest obligatoryjna (zwalnianie pamięci odbywa się po wykonaniu skryptu).
*/
void FAFreeSearch(long searchID);

/*
    Funkcja dostępna tylko w eksporcie dokumentów.
    Funkcja zwraca wyeksportowane dane dla aktualnie przetwarzanego dokumentu.
    Format danych jest uzależniony od wybranych opcji w FA Designer i
    trybu pracy FA Export.
*/
LPCWSTR FAExpGetText(void);

/*
    Funkcja dostępna tylko w eksporcie dokumentów.
    Funkcja zmienia wyeksportowane dane na lpsz.
    Dane te będą eksportowane do pliku eksportu.
*/
void FAExpSetText(LPCSTR lpsz);

/*
    Funkcja dostępna tylko w eksporcie dokumentów.
    Funkcja zwraca liczbę plików z obrazami stron stworzonymi
    podczas eksportu aktualnego dokumentu.
    Liczba stron może wynieść 1 (dla trybu wszystkie strony w jednym pliku) lub
    tyle ile jest niepustych stron.
*/
long FAExpGetPgCnt(void);

/*
    Funkcja dostępna tylko w eksporcie dokumentów.
    Funkcja zwraca pełną ścieżkę do pliku z obrazami stron.
*/
LPCWSTR FAExpGetPg(long pg);
```

12.8. Przykłady skryptów

12.8.1. Przykład I – Skrypt weryfikujący wartość pól

Skrypt weryfikujący wartość pól. Skrypt porównuje wartość pól PESEL i Płeć. Skrypt kończy się poprawnie jeżeli wartość pola PESEL jest zgodna z wartością pola Płeć w przeciwnym wypadku zwracany jest komunikat o błędzie. Płeć kodowana jest w numerze PESEL na przedostatniej cyfrze. Jeżeli cyfra jest parzysta to mamy do czynienia z kobietą w przeciwnym wypadku z mężczyzną.

Pole *PESEL* to pole tekstowe. Pole *Płeć* to grupa dwóch pól wyboru.

Polu *PESEL* przyporządkowano zmienną o nazwie *PESEL*, polu *Płeć* zmienną o nazwie *sex*.

Język skryptu: VBScript

```
k=0
If Len(PESEL) <> 11 Then
    errorvar = 1
    errortxt = "Pole PESEL ma niepoprawną długość !"
Else
    k = CInt(Mid(PESEL, 10, 1))
    k = (k mod 2)
    If (sex(k+1)<>TRUE) Then k = -1
End If
If (k<0) Then
    errorvar = 2
    errortxt = "Niezgodność pola PESEL z polem Płeć !"
End If
```

Język skryptu: JScript

```
k=0;
if (PESEL.length != 11)
{
    errorvar = 1;
    errortxt = "Pole PESEL ma niepoprawną długość !";
}
else
{
    k = parseInt(PESEL.charAt(9));
    k %= 2;
    if (sex[k] != true) k = -1;
}
if (k<0)
{
    errorvar = 2
    errortxt = "Niezgodność pola PESEL z polem Płeć !"
}
```

12.8.2. Przykład II - Skrypt modyfikujący wartość pól

Skrypt modyfikujący wartość pól. Skrypt w oparciu o wartość pola *PESEL* ustawia wartość pola *Data urodzenia*. Jeżeli pole *PESEL* posiada niepoprawną długość to pole *Data urodzenia* jest czyszczone (przypisywana jest mu wartość pusta).

Pola *PESEL* i *Data urodzenia* to pola tekstowe. Polu *PESEL* przyporządkowano zmienną wejściową o nazwie *PESEL*, polu *Data urodzenia* zmienną wyjściową o nazwie *DU*.

Język skryptu: VBScript

```
DU=""
If Len(PESEL) = 11 Then
    If Cint(Mid(PESEL,3,1)) > 1 Then
        DU="20" & Mid(PESEL,1,2) & "-" & Cint(Mid(PESEL,3,1))-2
        DU=DU & Mid(PESEL,4,1) & "-" & Mid(PESEL,5,2)
    Else
        DU="19" & Mid(PESEL,1,2) & "-" & Mid(PESEL,3,2) & "-" & Mid(PESEL,5,2)
    End If
End If
```

Język skryptu: Jscript

```
DU="";
if (PESEL.length == 11)
{
    if (parseInt(PESEL.charAt(2)) > 1)
    {
        DU="20";
        DU+=PESEL.substr(0,2);
        DU+="-";
        DU+=parseInt(PESEL.charAt(2))-2;
        DU+=PESEL.charAt(3);
        DU+="-";
        DU+=PESEL.substr(4,2);
    }
    else
    {
        DU="19";
        DU+=PESEL.substr(0,2);
        DU+="-";
        DU+=PESEL.substr(2,2);
        DU+="-";
        DU+=PESEL.substr(4,2);
    }
}
```

12.8.3. Przykład III – Wywołanie metody WebService z poziomu skryptu w języku JScript

Skrypt modyfikujący wartość pól. Skrypt pobiera wartość pola *NrUmowy*. Następnie wywoływana jest metoda WebService o nazwie *GetClient*. Metoda, której argumentem jest *NrUmowy* zwraca związane z danym nr umowy informacje o kliencie (imię i nazwisko). Na podstawie zwróconych wartości skrypt ustawia wartość pól szablonu: *Imię* i *Nazwisko*. Jeżeli wywołanie metody WebService zakończyło się niepowodzeniem skrypt zwraca komunikat błędu (modyfikowane są zmienne globalne *errorvar* i *errortxt*).

Poniżej zamieszczono przykładowe żądanie i odpowiedź SOAP 1.1 dla metody *GetClient*:

Przykład żądania (SOAP 1.1)

```
POST /MyApp/Service1.asmx HTTP/1.1
Host: test
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/GetClient"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClient xmlns="http://tempuri.org/">
      <NrUmowy>string</NrUmowy>
    </GetClient >
  </soap:Body>
</soap:Envelope>
```

Przykład odpowiedzi (SOAP 1.1)

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientResponse xmlns="http://tempuri.org/">
      <GetClientResult>
        <Imie>string</Imie>
        <Nazwisko>string</Nazwisko>
      </GetClientResult>
    </GetClientResponse>
  </soap:Body>
</soap:Envelope>
```

Poniżej zamieszczono skrypt napisany w języku JScript:

Język skryptu: JScript

```
strURLAddress = "http://test/MyApp/Service1.asmx";
strNamespace = "http://tempuri.org/";
strMethod = "GetClient";
strNrUmowy = NrUmowy;

// zmienne inicjowane przez FA
errorvar = 0;
errortxt = "";
```

```
try
{
    callGetUser();
}
catch (objError)
{
    errorvar = 1;
    errortxt = objError.description;
}

function callGetUser()
{
    // Tworzymy obiekty WinHttpRequest i DOMDocument
    var xmlHttp = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
    var Result = new ActiveXObject("MSXML2.DOMDocument");
    var Args = new ActiveXObject("MSXML2.DOMDocument");

    // Tworzymy żądanie SOAP
    var soapStr = "<?xml version='1.0' encoding='utf-8'?'>";
    soapStr += "<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd='http://www.w3.org/2001/XMLSchema\"
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>";
    soapStr += "<soap:Body>";
    soapStr += "<GetClient xmlns='http://tempuri.org/'>";
    soapStr += "<NrUmowy>" + strNrUmowy + "</NrUmowy>";
    soapStr += "</GetClient>";
    soapStr += "</soap:Body>";
    soapStr += "</soap:Envelope>";

    Args.loadXML(soapStr);
    xmlHttp.Open("POST", strURLAddress , false);
    xmlHttp.setRequestHeader("content-Type", "text/xml");
    xmlHttp.setRequestHeader("charset", "utf-8");
    xmlHttp.setRequestHeader("SOAPAction", strNamespace + strMethod);
    xmlHttp.Send(Args.xml);

    // Odczytujemy rezultat
    Result.loadXML(xmlHttp.ResponseText);
    Result.setProperty("SelectionLanguage", "XPath");

    errorInfo = GetResult(Result, "errorInfo");
    if (errorInfo != "")
    {
        errorvar = 1;
        errortxt = errorInfo;
        return;
    }

    Imie = GetResult(Result, "Imie");
    Nazwisko = GetResult(Result, "Nazwisko");
    return ;
}

function GetResult(objResult, strName)
{
    var ret = "";
    var objTmp = objResult.selectSingleNode("//*[local-name() = '");
    objTmp += strName;
    objTmp += "']");
    if (objTmp != null)
        ret = objTmp.text;
    return ret;
}
```

12.8.4. Przykład IV – Pobranie wartości z bazy danych przypisanej do aktualnego dokumentu w języku JScript

Skrypt modyfikujący wartość pola. Skrypt tworzy połączenie z bazą danych. Następnie pobierana jest informacja o bieżącym dokumencie w celu pobrania jego identyfikatora dokumentu. W kolejnym kroku pobierany jest wiersz z tabeli pomocniczej, w której dla każdego dokumentu przechowywane są informacje indeksowe. Na podstawie znalezionych wartości skrypt ustawia wartość zmiennej X. Jeżeli wystąpi błąd skrypt zwraca komunikat błędu (modyfikowane są zmienne globalne *errorvar* i *errortxt*).

```
// połączenie do bazy danych
var conn_str = "DSN=FormAnalyze;UID=indexer;PWD=indexer;";
var conn = new ActiveXObject("ADODB.Connection");
conn.open(conn_str, "", "");

// pobranie informacji o aktualnym dokumencie
var docinfo = FGetDocInfo(-1);

// pobranie indeksu dla bieżącego dokumentu
var rs = new ActiveXObject("ADODB.Recordset");
var strQuery =
    "SELECT VAL_INDEX FROM \"ADMIN\".\"Document_INDEX\""
    "where obid=(select obid from \"ADMIN\".\"Document\""
    "where obid = " + docinfo.docid + ")";
rs.open(strQuery, conn, 2, 3);

// zapisanie indeksu do zmiennej wyjściowej
if (!rs.bof)
{
    rs.MoveFirst();
    errorvar = 0;
    X = rs.fields(0).value + "";
}
else
{
    errortext = "Brak indeksu dla dokumentu.";
    errorvar = -1;
    X = "~";
}

// zwolnienie obiektów
conn.close();
conn = null;
rs = null;
```

12.8.5. Przykład V – Wyrażenia regularne w języku JScript

Skrypt modyfikujący wartość pola. Skrypt pobiera informacje o polu źródłowym. Następnie wyszukuje pole o nazwie „12. Regon”. W wartości pola wejściowego wyszukiwana jest fraza tekstu złożona z 14 cyfr i znaków OS. W kolejnym kroku znaki OS zamieniane są na 05. Następnie wyszukiwana jest pozycja wyniku, aby można było ustawić właściwe współrzędne. Wynik zapamiętywany jest w polu regonu. Jeżeli wystąpi błąd skrypt zwraca komunikat błędu (modyfikowane są zmienne globalne *errorvar* i *errortxt*).

```
// pobranie pola z którego skrypt został wywołany - pole źródłowe
fldIn = FGetCurrField();

// pobranie pola docelowego
fldRegon = FGetField("12. Regon", 0, -1);

// można by było także odwołać się do pola np. poprzez wywołanie
// fldRegon = FFindField("12.", 0, -1);

if (fldRegon == null)
{
    errortext = "Błąd konfiguracji: Nie znaleziono pola regon.";
    errorvar = 1;
}
else
{
    strRegon = "~";
    left = fldIn.left;
    right = fldIn.right;
    top = fldIn.top;
    bottom = fldIn.bottom;

    // znalezienie wzorca regonu 9 lub 14 cyfr
    arr = X.match(/(^|\D|[OS])+([0-9OS]{9}|[0-9OS]{14})(\D|[OS])+/im);
    if (arr != null && arr.length > 2)
    {
        strRegon = arr[2];

        // znalezienie pozycji dla znalezionego słowa,
        // aby w FA Verifier był wyświetlony tylko podgląd znalezionej frazy
        pos = -1;
        do
        {
            word = FGetNextWord(fldIn.pos, pos, "");
            if (word.value.search(strRegon) >= 0)
            {
                left = word.left;
                right = word.right;
                top = word.top;
                bottom = word.bottom;
                break;
            }
            pos = word.pos;
        }
        while (pos >= 0)

        // zamiana złych znaków Ss na 5 i Oo na 0
        strRegon = strRegon.replace(/S/ig, "5");
        strRegon = strRegon.replace(/O/ig, "0");
    }

    // zapisywanie zmian dla regonu
    FSetField(fldRegon.pos, strRegon, 90, left, top, right, bottom);

    // zaakceptowanie zmian
    FAcceptFields();

    // gdyby wywołano FARollbackFields zmiany dokonane poprzez FSetField
    // nie miałyby znaczenia
}
```

12.8.6. Przykład VI – Wyszukiwanie fraz tekstu oraz akceptacja dokumentu w języku JScript

Skrypt modyfikujący wartość pola. Skrypt pobiera informacje o polu „Rec”. Następnie wyszukuje w polu jednej z wartości stałych (liczba błędnych znaków może wynieść 30%): „ Numer zlecenia” lub „ Nr. zlecenia”. Począwszy od końca znalezionych wartości wyszukiwana jest sekwencja rozpoczynająca się od 001 i złożonej z dodatkowych 13 cyfr (liczba błędnych znaków może wynieść 30%). Wartość podstawiana jest do pola „Numer”. Jeżeli wzorzec jest znaleziony ze 100% pewnością status dokumentu jest ustawiany jako gotowy. Jeżeli wystąpi błąd skrypt zwraca komunikat błędu (modyfikowane są zmienne globalne *errorvar* i *errortxt*).

```
cl = 0;

// pobranie pola źródłowego
fldRec = FAGetField("Rec", 0, -1);

// wyszukanie wartości stałej " Numer zlecenia" lub " Nr. zlecenia"
searchID = FAInitSearch1(fldRec.pos, " Numer zlecenia\0 Nr. zlecenia\0", 70);
if (searchID > 0)
{
    // wyszukanie wzorca rozpoczynającego się od 001 a następnie z 13 cyfr
    // począwszy od pozycji końcowej wzorca " Numer zlecenia"
    searchData = FAGetSearch(searchID, 0);
    searchID = FAInitSearch3(fldRec.pos, "001#####", 70,
        searchData.end, 1);

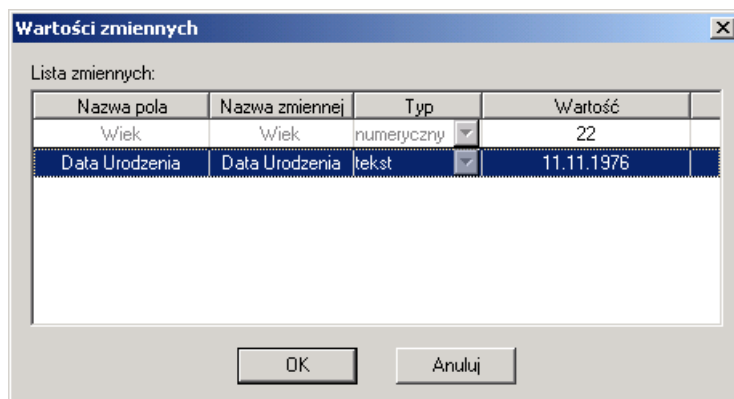
    if (searchID > 0)
    {
        // zapisanie wyszukania w polu Numer
        searchData = FAGetSearch(searchID, 0);
        fld = FAGetField("Numer", 0, -1);
        FASetField(fld.pos, searchData.value.replace(/^\s+|\s+$/g, ''),
            searchData.cl, searchData.left, searchData.top,
            searchData.right, searchData.bottom);
        cl = searchData.cl;
    }
}

// akceptacja dokumentu (ustawianie statusu na gotowy)
// jeżeli wyszukany wzorzec jest bez przekłamań
if (cl == 100)
{
    FASetDocInfo(0, 0, 0, 7);
}

// zaakceptowanie zmian
FAAcceptFields();
```

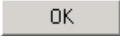
12.9. Sprawdzenie działania skryptu

Sprawdzenie działania skryptu rozpoczyna się naciśnięciem przycisku . Program otwiera okno parametrów skryptu, w którym podajemy wartości argumentów, dla których zostanie wykonany skrypt.



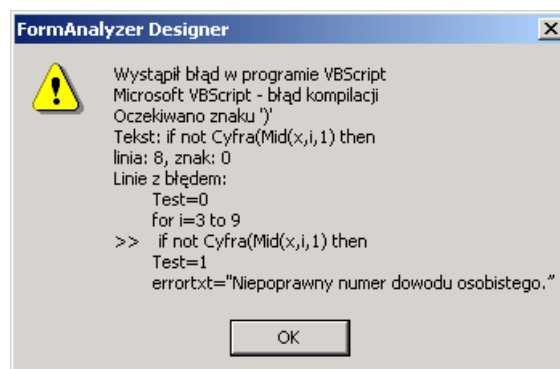
Rys. 12-22 Okno wartości zmiennych otwarte dla skryptu wiek.

W pierwszych trzech kolumnach są podane dane identyfikujące argumenty skryptu. Są to kolejno: nazwa pola (dokumentu), nazwa zmiennej (używana dla identyfikacji wartości pola w skrypcie) oraz typ zmiennej. Te wartości nie mogą być zmieniane. W kolumnie *Wartość* podajemy wartości pól (argumentów), dla których zostanie wykonany skrypt.

Naciśnięcie  powoduje rozpoczęcie wykonywania skryptu.

Pierwszym krokiem jest sprawdzenie poprawności składniowej skryptu. Jeżeli program stwierdza błędy, są one sygnalizowane komunikatem zawierającym opis błędu i fragment kodu, w którym został wykryty błąd.

Przykład takiego komunikatu pokazano na rysunku poniżej.



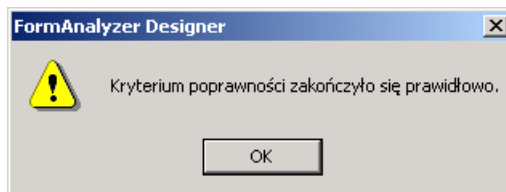
Rys. 12-23 Komunikat o błędzie składniowym VBScript (błąd niezamknięcia otwartego nawiasu).

W tym przykładzie w linii zawierającej warunek (if not ...) zostały otwarte dwa nawiasy a zamknięty tylko jeden. W okienku oprócz komunikatu o błędzie, został wyświetlony fragment treści skryptu zawierający błąd.

Przed ponowną próbą testowania działania skryptu trzeba poprawić wszystkie błędy składniowe.

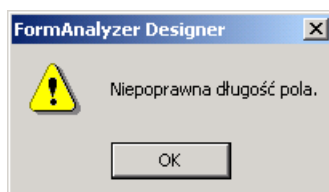
Jeżeli skrypt jest poprawny składniowo, program nadaje zmiennym wejściowym skryptu wpisane przez nas wartości.

Po podstawieniu wartości następuje wykonanie skryptu. W przypadku, gdy skrypt kończy się bez zgłoszenia błędu (zmiennej *errorvar* nadano wartość 0), co oznacza spełnienie kryteriów poprawności, jest wyświetlany komunikat pokazany na rysunku poniżej.



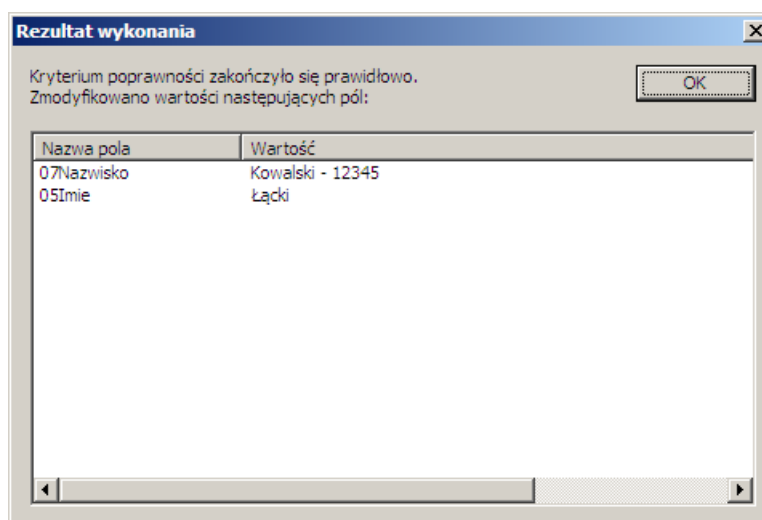
Rys. 12-24 Komunikat o spełnieniu kryteriów poprawności pola (pól).

Zgłoszenie przez skrypt błędu (zmiennej *errorvar* nadano wartość różną od 0) oznacza, że pole (pola) nie spełnia kryteriów weryfikacji. Jest wtedy wyświetlany okno komunikatu o błędzie. Treścią tego okienka jest wartość zmiennej *errortxt* ustawiona w czasie wykonywania skryptu. Na rysunku poniżej przedstawiono przykład komunikatu informującego o błędzie weryfikacji danych.



Rys. 12-25 Komunikat o niepoprawnej weryfikacji pola.

W przypadku gdy testowanym skryptem jest skrypt modyfikujący wartość pól szablonu, wykonanie skryptu kończy się wyświetleniem okna *Rezultat wykonania*. W oknie prezentowana jest lista pól, których wartość została zmieniona w wyniku wykonania skryptu.



Rys. 12-26 Okno *Rezultat wykonania*.

12.10. Tworzenie funkcji weryfikacji w bibliotekach DLL

Wykorzystanie do weryfikacji funkcji z zewnętrznych bibliotek dynamicznych DLL wymaga przygotowania najpierw odpowiedniej biblioteki DLL. Biblioteka DLL używana do weryfikacji musi spełniać pewne ściśle określone wymagania, które zostaną omówione dalej.

Do tworzenia biblioteki DLL możemy użyć dowolnego narzędzia. Jako przykładowe środowisko tworzenia biblioteki weryfikacji będzie używany Microsoft Developer Studio z kompilatorem Microsoft Visual C++ ver. 5.0. Podane w dalszej części przykłady będą pokazywały tworzenie biblioteki w takim środowisku. Należy o tym pamiętać, jeśli korzystamy z innych narzędzi programistycznych.

W systemie FormAnalyze z funkcji weryfikacji zdefiniowanych w bibliotekach DLL korzystają trzy programy. Na etapie projektowania przetwarzania formularza FormAnalyze Designer. W czasie rozpoznawania treści dokumentu FormAnalyze Engine. W czasie weryfikacji FormAnalyze Verifier.

Aby dowolny z tych programów mógł skorzystać z biblioteki weryfikacji DLL muszą być spełnione następujące warunki:

1. Plik biblioteki DLL musi znajdować się w tym samym folderze, co binarny plik odpowiedniej aplikacji.
2. Nazwa pliku biblioteki DLL musi rozpoczynać się od liter .VAL.. Mówiąc inaczej, biblioteki weryfikacji mają nazwy o szablonie .VAL*.DLL..
3. Biblioteka DLL musi posiadać odpowiednią strukturę (opisaną niżej).

Wykorzystanie funkcji weryfikacji z biblioteki DLL wymaga zdefiniowania w niej kilku funkcji organizacyjnych. Informacje dostarczane przez te funkcje są istotne przede wszystkim w trakcie definiowania procedur weryfikacji w FormAnalyze Designer. Pozwalają one na częściowe sprawdzenie poprawności powiązań pomiędzy polami dokumentu a funkcjami weryfikacji z biblioteki DLL. W pewnym stopniu ułatwiają także korzystanie z takich funkcji.

W bibliotece weryfikacji DLL musi być zdefiniowanych pięć funkcji organizacyjnych:

DllInitialize funkcja inicjacji biblioteki

DllReset funkcja wywoływana przed usunięciem biblioteki z pamięci operacyjnej.

DllGetName wartością tej funkcji jest nazwa biblioteki weryfikacji używana w programie FormAnalyze Designer.

DllGetFuns zadaniem tej funkcji jest przekazanie informacji o liczbie i bazowych nazwach funkcji weryfikacji zdefiniowanych w bibliotece weryfikacji.

DllGetError ta funkcja przekazuje wskazanie na komunikat opisujący ostatnio zarejestrowany przez bibliotekę błąd.

Przedstawimy teraz kolejno dokładniej znaczenie i sposób korzystania z tych funkcji.

Funkcja inicjacji biblioteki jest wywoływana przy zmianie połączenia z bazą danych w aplikacjach FormAnalyze Verifier i FormAnalyze Designer. W czasie rozpoznawania w aplikacji FormAnalyze Engine funkcja inicjacji jest wywoływana na początku przetwarzania dokumentu, który korzysta z funkcji weryfikacji z tej biblioteki.

Deklaracja funkcji inicjacji biblioteki jest następująca:

```
BOOL DllInitialize(LPVOID lpvReserved);  
// Wejscie: lpvReserved - zarezerwowane do wykorzystania w przyszłości  
// Wartosc: TRUE - biblioteka została poprawnie zainicjowana  
//          FALSE - błąd inicjacji biblioteki
```

W funkcji inicjacji biblioteki należy inicjować wartości zmiennych.

Funkcja porządkowania biblioteki przed zamknięciem - DllReset - jest wywoływana dwukrotnie: bezpośrednio przed usunięciem biblioteki z pamięci operacyjnej oraz **bezpośrednio po załadowaniu biblioteki do pamięci**. Tak, więc wywołanie DllReset poprzedza pierwsze wywołanie DllInitialize.

Deklaracja funkcji porządkowania biblioteki przed zamknięciem jest następująca:

```
void DllReset();
```

Podstawowym zadaniem DllReset jest zwolnienie zasobów systemowych używanych przez bibliotekę.

Wartością funkcji DllGetName jest wskazanie na nazwę biblioteki. Dzięki temu identyfikacja biblioteki (szczególnie w FormAnalyzer Designer) jest możliwa na podstawie znaczącej nazwy a nie tylko nazwy pliku DLL.

```
LPCSTR DllGetName();  
// Wejscie:  
// Wartosc: wskazanie na nazwę biblioteki weryfikacji
```

Funkcja DllGetFuns służy do pobrania z biblioteki liczby i nazw bazowych funkcji weryfikacji.

```
LPCSTR DllGetFuns();  
// Wartosc: wskazanie na napis składający się z bazowych nazw funkcji weryfikacji oddzielonych  
//          znakiem '\0'. Ciąg znaków jest zakończony dwoma znakami '\0'.
```

Oto bardzo prosta implementacja DllGetFuns dla biblioteki zawierającej trzy funkcje weryfikacji: NrDowodu, Wiek, Znajomosc.

```
LPCSTR DllGetFuns(int& nCount)  
{  
    return "NrDowodu\0Wiek\0Znajomosc\0";  
}
```

Funkcja DllGetError służy do pobrania z biblioteki weryfikacji treści ostatnio zarejestrowanego komunikatu błędu. Ta treść - lub identyfikator - powinna być przechowywana przez bibliotekę do wystąpienia następnego błędu.

```
LPCSTR DllGetError();  
// Wartosc: wskazanie na treść komunikatu ostatnio zarejestrowanego błędu
```

Jak w każdej bibliotece dynamicznej trzeba też zdefiniować funkcję DllMain. Przedstawiony niżej fragment kodu, pokazuje realizację omówionych powyżej funkcji w bibliotece weryfikacji.

Przyjęto następujące założenia:

- biblioteka zawiera funkcje weryfikacji NrDowodu, Wiek i Znajomosc.
- biblioteka nie alokuje dynamicznie zasobów systemowych.

- treść komunikatów błędów (a także nazwa biblioteki) są pobierane z tabeli napisów zasobów biblioteki.
- do skopiowania treści komunikatu o błędzie z tabeli napisów do bufora zostanie użyta pomocnicza funkcja SetLastError.

```
#include <windows.h>
#include "resource.h"
#include "DllErr.h"           // nagłówki konieczne w bibliotece weryfikacji
#include "VBSscrEx.h"         // nagłówki dolaczone ze względu na funkcje
#include <string.h>           // używane w bibliotece
#include <ctype.h>
#include <time.h>
#include <stdio.h>
HINSTANCE hInst;
char errBuf[256];           // bufor komunikatu błędu

BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    if(fdwReason == DLL_PROCESS_ATTACH)
        hInst = hinstDLL;
    return TRUE;
}

// Dll Interface functions
BOOL DllInitialize(LPVOID lpvReserved)
{
    return TRUE;
}
void DllReset()
{
}
LPCSTR DllGetName()
{
    static char DllName [128] = "";
    if(DllName[0] == '\0')
        LoadString(hInst, IDS_DLL_NAME, DllName, sizeof(DllName));
    return DllName;
}
LPCSTR DllGetFuns()
{
    return "NrDowodu\0Wiek\0Znajomosc\0";
}
LPCSTR DllGetError()
{
    return errBuf;
}

// Pomocnicza funkcja kopiująca do bufora błędu wartość z tabeli napisów biblioteki.
// Wejście: resourceID - identyfikator komunikatu błędu
void SetLastError(int resourceID)
{
    if(!LoadString(hInst, resourceID, errBuf, sizeof(errBuf)))
        strcpy(errBuf, "Błąd wewnętrzny: brak zdefiniowanego napisu.");
}
```

Funkcja weryfikacji wykorzystywana w systemie FormAnalyze jest realizowana w bibliotece jako zestaw czterech funkcji. Wynika to z różnorodności wykorzystania:

- W fazie projektowania interesuje nas liczba i nazwy parametrów wywołania funkcji oraz to, czy wartość pola (pól) jest przez funkcję modyfikowana, czy też nie.
- W czasie weryfikacji wartości pola (pól) chcemy dowiedzieć się czy są spełnione kryteria poprawności.
- Jeżeli funkcja modyfikuje wartość weryfikowanego pola, a są spełnione kryteria poprawności, chcemy pobrać zmodyfikowaną wartość pola (dla pól tekstowych mogą nas także interesować poziomy ufnosci poszczególnych znaków).

Bardzo istotną sprawą jest tworzenie nazw funkcji. Powstają one przez złożenie przedrostka oraz bazowej nazwy funkcji. Przedrostki są następujące:

DllGetVars_ dla funkcji opisującej parametry funkcji weryfikacji.

DllProcess_ dla samej funkcji weryfikacji.

DllGetVal_ dla funkcji pobrania zmodyfikowanych wartości weryfikowanych pól.

DllGetExInfo_ dla funkcji przekazującej nowe wartości poziomów zaufania dla znaków zmodyfikowanych w DllProcess_.

Dla drugiej funkcji weryfikacji z podanego wyżej przykładu - Wiek, w bibliotece muszą znaleźć się definicje funkcji: DllGetVars_Wiek, DllProcess_Wiek,

DllGetVal_Wiek.

W dokładniejszym omówieniu tych funkcji zamiast konkretnej nazwy funkcji będziemy stosować napis Nazwa.

```
LPCSTR DllGetVars_Nazwa();  
// Wartość: wskazanie na listę nazw parametrów funkcji w postaci: "par0\0par1\0...parn\0"
```

Przekazywanie listy nazw parametrów funkcji wydaje się tutaj niepotrzebne, wystarczyłaby liczba parametrów. W istocie te nazwy nie są w żaden sposób wykorzystywane przez oprogramowanie. Służą one przede wszystkim osobie projektującej przetwarzanie dokumentu w FormAnalyze Designer, do powiązania parametrów weryfikacji z odpowiednimi polami dokumentu. Podanie tutaj nazw, które sugerują znaczenie parametru lub nawet pole dokumentu, które powinno zostać związane z tym parametrem, ogromnie ułatwia projektowanie weryfikacji dokumentu.

Funkcja wykonująca sprawdzenie kryteriów poprawności ma następującą deklarację:

```
int DllProcess_Nazwa(LPVOID pVars, int nVars, DWORD dwFlags);  
// Wejście: pVars - wskazanie na tablicę parametrów wejściowych  
//          nVars - liczba parametrów wejściowych  
//          dwFlags - opcje przetwarzania  
// Wartość: ERR_OK - dla prawidłowych danych  
//          ERR_VALIDATE - dla nieprawidłowych danych  
//          ERR_INTERNAL - błąd wewnętrzny biblioteki
```

Omówimy najpierw sposób przekazywania parametrów funkcji weryfikacji. Wskazanie pVars, mimo że zadeklarowane jako typu LPVOID, jest w istocie wskazaniem na pierwszy element tablicy elementów typu varprop. Liczba elementów tej tablicy jest przekazywana do funkcji w nVars.

Definicja struktury varprop (znajduje się w pliku nagłówkowym VBScrEx.h) jest następująca:

```
struct varprop  
{  
    const void *mem; // wskazanie na wartość parametru  
    int vtype; // identyfikator typu wartości  
    int vdim; // rozmiar wartości  
    int vbtype; // zarezerwowane dla VBScript
```

```
const char *vbname;    // zarezerwowane dla VBScript  
};
```

Identyfikator typu wartości vtype może w funkcjach weryfikacji z biblioteki DLL przyjmować następujące wartości (są one zdefiniowane w pliku VBScrEx.h):

vbs_text parametr znakowy (mem wskazuje na pierwszy znak ciągu znaków zakończonoego znakiem .\0.). Jako vbs_text są przekazywane dane tekstowe i **numeryczne**. Wartość vdim ma dla parametru vbs_text wartość nieokreśloną.

vbs_bits parametr bitowy (mem wskazuje na pierwszy element tablicy bajtów, z których każdy odpowiada pojedynczemu polu wyboru). Jako vbs_bist są przekazywane pola wyboru i grupy pól wyboru. Liczba elementów tablicy jest przekazana w vdim. Wartość elementu tablicy mem równa 0, oznacza niezaznaczone pole wyboru; wartość różna od 0 oznacza pole zaznaczone.

Na początku wykonania funkcji weryfikacji należy sprawdzić, czy funkcja została wywołana z poprawną liczbą parametrów oraz to, czy poprawne są typy parametrów. W przypadku parametrów vbs_bits, należy także sprawdzić, czy liczba pól wyboru jest zgodna z oczekiwaniami.

Tak szczegółowe sprawdzenie poprawności danych wejściowych pozwala na ustalenie, czy powiązania parametrów funkcji weryfikacji z polami dokumentu zostały poprawnie zdefiniowane.

Wartość funkcji powinna przyjmować jedną z trzech wartości:

ERR_OK gdy pole (pola) spełniają kryteria poprawności.

ERR_VALIDATE gdy pole (pola) nie spełniają kryteriów poprawności.

ERR_INTERNAL gdy funkcja nie jest w stanie sprawdzić kryteriów poprawności, ze względu na niepoprawną liczbę lub typ parametrów, lub wystąpienie innego błędu.

W dwóch ostatnich przypadkach trzeba zadbać o zarejestrowanie stosownego komunikatu opisującego błąd, który wystąpił i - jeśli to możliwe - jego przyczynę. W podanym wyżej szkielecie biblioteki DLL polegałoby to na wywołaniu funkcji SetLastError z odpowiednim identyfikatorem komunikatu o błędzie.

Parametr dwFlags jest w obecnej wersji używany dla funkcji nie modyfikujących wartości pól. Gdy w dwFlags jest ustawiony bit VF_DISABLE_UI (stała zdefiniowana w VBScrEx.h) funkcja nie powinna korzystać z interfejsu użytkownika.

Poniższy przykład pokazuje implementację funkcji weryfikacji NrDowodu sprawdzającej poprawność serii i numeru dowodu osobistego. Kryteria poprawności są uproszczone: pole zostanie uznane za poprawne, gdy liczy 9 znaków, z których dwa pierwsze są literami a pozostałych 7 jest cyframi.

```
LPCSTR DllGetVars_NrDowodu()  
{  
    return "SeriaNumer\0"; // nazwa zmiennej dla FR Designer  
}  
int DllProcess_NrDowodu(LPVOID pVars, int nVars, DWORD dwFlags)
```

```
{
    varprop* prop = (varprop*) pVars;
    if(nVars != 1 || prop[0].vtype != vbs_text)
    {
        SetLastError(IDS_ERR_BADPARAMETERS);
        return ERR_INTERNAL; // niepoprawna liczba lub typ parametru
    }
    LPCSTR prm = (LPCSTR) prop[0].mem;
    if(strlen(prm) != 9)
    {
        SetLastError(IDS_ERR_NDOW_DŁUGOSC);
        return ERR_VALIDATE; // niepoprawna długość pola
    }
    if(isalpha(prm[0]) && isalpha(prm[1]))
    {
        for(int i = 2; i < 9; i++)
            if(!isdigit(prm[i]))
            {
                SetLastError(IDS_ERR_NDOW_NUMER);
                return ERR_VALIDATE; // nie-cyfra w numerze
            }
        return ERR_OK;
    }
    SetLastError(IDS_ERR_NDOW_SERIA);
    return ERR_VALIDATE; // nie-litera w serii
}

// Dla funkcji nie modyfikującej pola, nie definiuje się funkcji DLLGetVal_NrDowodu
```

W powyższym przykładzie użyto kilku identyfikatorów napisów. Trzeba je zdefiniować w tabeli napisów zasobów biblioteki. Widok kompletnej tabeli napisów (zawierającej wszystkie napisy używane w bibliotece) jest pokazany na rysunku 12-28.

Ponieważ funkcja weryfikacji NrDowodu nie modyfikuje wartości pola, w bibliotece nie należy implementować funkcji DllGetVal_NrDowodu. Gdyby w bibliotece DLL znalazła się funkcja DllGetVal_NrDowodu oznaczałoby to, że DllProcess_NrDowodu modyfikuje wartość.

12.10.1. Modyfikacja wartości pola

Podobnie jak w przypadku skryptów, możemy zdefiniować funkcje weryfikujące modyfikujące wartość pola. Bardzo istotną różnicą, w stosunku do skryptów weryfikacji, jest to, że funkcje weryfikacji z bibliotek DLL mogą modyfikować **wszystkie** wartości przekazane jako parametry funkcji. Jeżeli podstawowa funkcja weryfikacji (DllProcess_Nazwa) ma wartość TRUE - tzn. pola spełniają kryteria poprawności - program, który wywołał funkcję weryfikacji pobiera wartości pól przy użyciu funkcji DllGetVal_Nazwa. (W rzeczywistości, sposób pobierania nowych wartości pól zależy jeszcze od typu pola - zostanie to omówione w dalszej części tego punktu.)

Przypomnijmy, że od tego czy funkcja modyfikuje wartość pola czy też nie, zależy która aplikacja będzie wywoływała daną funkcję. Funkcje modyfikujące wartość pola są wywoływane w czasie rozpoznawania dokumentu przez FormAnalyze Engine. Funkcje, które nie zmieniają wartości pola, są wywoływane w czasie weryfikacji dokumentów, w FormAnalyze Verifier.

W czasie projektowania przetwarzania dokumentów w FormAnalyze Designer, mogą być wywoływane zarówno funkcje modyfikujące, jak i niemodyfikujące wartości. Daje to możliwość przetestowania wszystkich funkcji przed wdrożeniem pliku konfiguracji do użycia w systemie.

Jeżeli dla nazwy bazowej Nazwa istnieje w bibliotece DLL funkcja DllGetVal_Nazwa FormAnalyze Designer przyjmuje, że kryterium poprawności modyfikuje wartość pola.

Innymi słowy obecność DllGetVal_Nazwa w bibliotece decyduje o czasie wykonania DllProcess_Nazwa: w FormAnalyzer Engine (gdy DllGetVal_Nazwa jest zdefiniowana) albo FormAnalyzer Verifier (gdy DllGetVal_Nazwa nie jest zdefiniowana).

Funkcja DllGetVal_Nazwa ma następujący prototyp:

```
LPVOID DllGetVal_Nazwa(int nVar);
// Wejście: nVar - numer parametru (pola), ktorego wartosc należy zwrocic (liczac od zera)
// Wartość: != 0 wskazanie na nowa wartosc
//           == 0 zwracane dla parametrow nie modyfikowanych przez kryterium poprawnosci
//           (0 jest zwracane także dla wartosci pol wyboru (albo grup pol wyboru),
//           ktore sa modyfikowane bezpośrednio w DllProcess_Nazwa.
```

Dla parametrów typu vbs_bits modyfikacja jest dokonywana od razu w tablicy bajtów reprezentujących wartości grupy pól wyboru (lub pojedynczego pola wyboru - tablica ma wtedy długość 1). Ustawienie elementu tablicy na wartość różną od 0 oznacza „zaznaczenie” pola wyboru, a ustawienie 0 - usunięcie zaznaczenia.

Ponieważ nowa wartość pola jest już zapisany w tablicy przekazanej jako parametr, wartość funkcji pobierania zmodyfikowanej wartości (DllGetVal_Nazwa) dla parametrów typu vbs_bits jest ignorowana.

W poniższym przykładzie pokazano implementację funkcji weryfikacji Znajomosc. Realizacja takiej procedury weryfikacji w postaci skryptu została omówiona w punkcie 0 i 12.8.

Przypomnijmy zasady weryfikacji pokazanych na rysunku poniżej grup pól wyboru oznaczających znajomość pojęć.

☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5		
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5		
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5		
☐ znam	☐ nie znam	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5		

Rys. 12-27 Fragment ankiety z grupami pól wyboru określających stopień znajomości określonych pojęć.

Każdy rząd pól wyboru stanowi grupę pól wyboru. Pierwsze dwa z pól określają, czy pojęcie jest respondentowi znane, czy też nie. Pozostałe pola grupy określają stopień znajomości pojęcia. Respondent powinien zaznaczyć jedno z nich tylko w przypadku, gdy zaznaczył pierwsze pole grupy („znam”).

Przyjmujemy następujące kryteria poprawności:

- grupa w której nie zaznaczono żadnych pól wyboru jest poprawna
- jeżeli zaznaczono pierwsze pole wyboru („znam”) to powinno być zaznaczone dokładnie jedno z pól od trzeciego do ostatniego
- jeżeli zaznaczono drugie pole wyboru („nie znam”) to żadne z pozostałych pól nie może być zaznaczone.

Zmiana wartości grupy jest możliwa tylko wtedy, gdy zaznaczono jedno z pierwszych dwóch pól. Dopuszczamy następujące modyfikacje wartości grupy:

- w przypadku zaznaczenia pola „nie znam”, pola stopnia znajomości (od trzeciego do ostatniego) są ustawiane na niezaznaczone
- w przypadku gdy zaznaczono pole „znam”, a w polach stopnia znajomości jest zaznaczone więcej niż jedno pole, pozostawiamy zaznaczone tylko pierwszego z nich.

W przykładzie implementacji takiej funkcji weryfikacji zdefiniowano dwie pomocnicze funkcje:

LiczbaZazn obliczającą ile pól stopnia znajomości zostało zaznaczonych

Zeruj „wyłączając” zaznaczenie pól stopnia znajomości (gdy parametr first ma wartość TRUE, pierwsze zaznaczone pole stopnia znajomości nie jest wyłączane).

```
LPCSTR DllGetVars_Znajomosc()
{
    return "Znajomosc\0"; // nazwa zmiennej dla FR Designer
}
int DllProcess_Znajomosc(LPVOID pVars, int nVars, DWORD dwFlags)
{
    varprop* prop = (varprop*) pVars;
    if(nVars != 1 || prop[0].vtype != vbs_bits)
    {
        SetError(IDS_ERR_BADPARAMETERS);
        return ERR_INTERNAL;
    }
    else if(prop[0].vdim < 3)
    {
        SetError(IDS_ERR_ZNJ_GRUPA);
        return ERR_INTERNAL;
    }
    LPBYTE prm = (LPBYTE) prop[0].mem;
    if(prm[0] ^ prm[1])
    {
        int lba = LiczbaZazn(prm, prop[0].vdim);
        if(prm[0])
        {
            // pojecie znane
            if(lba > 1)
                Zeruj(prm, prop[0].vdim, TRUE);
            else if(lba == 0)
            {
                SetError(IDS_ERR_ZNJ_BRAKSTOPNIA);
                return ERR_VALIDATE;
            }
        }
        else if(lba != 0)
            // pojecie nieznane
            Zeruj(prm, prop[0].vdim, FALSE);
    }
    else if(prm[0] || LiczbaZazn(prm, prop[0].vdim))
    {
        if(prm[0])
            SetError(IDS_ERR_ZNJ_ZNANIEZNA);
        else
            SetError(IDS_ERR_ZNJ_TYLKOSTOPIEN);
        return ERR_VALIDATE;
    }
    return ERR_OK;
}
// Dla funkcji modyfikującej tylko grupe pol wyboru, wartosc tej
// funkcji jest ignorowana - przyjmujemy null
LPVOID DllGetVal_Znajomosc(int nVar)
{
    return 0;
}
void Zeruj(LPBYTE tb, int size, BOOL first)
{
    for(int i = 2; i < size; i++)
```

```
        if(!first)
            tb[i] = 0;
        else
            first = !tb[i];
    }
    int LiczbaZazn(LPBYTE tb, int size)
    {
        int lba = 0;
        for(size--; size >= 2; size--)
            if(tb[size])
                lba++;
        return lba;
    }
```

Jak widać, zmiana zaznaczenia pól wyboru dokonywana w funkcji Zeruj, jest bez dalszych zabiegów „widziana” przez aplikację wywołującą funkcję weryfikacji.

W przypadku modyfikacji wartości pól przekazywanych jako parametry typu vbs_text (pola tekstowe i numeryczne) potrzebne jest inne podejście. Ponieważ nowa wartość parametru tekstowego może mieć inną długość, niż parametr wejściowy, nie należy w ogóle modyfikować wartości parametrów tekstowych.

Zmodyfikowane wartości tekstowe, należy zapamiętać w buforze w bibliotece DLL. Jeżeli zmienianych jest więcej wartości, każda z nich powinna zostać zapamiętana w odpowiednim buforze.

Parametr nVar, z którym jest wywoływana funkcja DllGetVal_Nazwa, oznacza numer parametru wywołania funkcji weryfikacji, którego nową wartość trzeba przekazać. Jeśli dla danej wartości nVar, funkcja DllGetVal_Nazwa zwraca wartość NULL (0) oznacza to, że ta wartość nie została przez funkcję weryfikacji zmodyfikowana.

Jeżeli nVar odpowiada wartości modyfikowanej przez funkcję weryfikacji, DllGetVal_Nazwa powinna zwrócić wskazanie na bufor, w którym funkcja weryfikacji zapamiętała nową wartość.

W kolejnym przykładzie funkcja weryfikacji Wiek, oblicza wiek respondenta na podstawie jego daty urodzenia. Funkcja DllProces_Wiek ma dwa parametry typu vbs_text. Pierwszy z nich jest wyznaczonym wiekiem, natomiast drugi datą urodzenia w formacie dd.mm.rrrr.

Dla przechowywania nowej wartości Wiek został użyty statyczny bufor wiek_znak. Zauważmy, że funkcja DllProces_Wiek nie sprawdza w ogóle jaka jest początkowa wartość Wiek - zostaje ona zastąpiona przez wartość wyznaczoną w tej funkcji.

```

void DllGetVars_Wiek()
{
    return "Wiek\0Data urodzenia\0"; // nazwy parametrow
}
char wiek_znak[16]; // bufor na zmodyfikowana wartosc Wiek
int DllProcess_Wiek(LPVOID pVars, int nVars, DWORD dwFlags)
{
    varprop* prop = (varprop*) pVars;
    if(nVars != 2 || prop[0].vtype != vbs_text || prop[1].vtype != vbs_text)
    {
        SetError(IDS_ERR_BADPARAMETERS);
        return ERR_INTERNAL;
    }
    struct tm *aczas;
    time_t czas;
    time(&czas); // aktualny czas
    aczas = localtime(&czas); // konwersja do dogodniejszej postaci

    int du_rok, du_dzien, du_miesiac;
    sscanf((LPCSTR) prop[1].mem, "%d.%d.%d", &du_dzien, &du_miesiac, &du_rok);
    int wiek = aczas->tm_year + 1900 - du_rok;
    if(aczas->tm_mon < du_miesiac ||
        aczas->tm_mon == du_miesiac && aczas->tm_mday < du_dzien)
        wiek--;
    sprintf(wiek_znak, "%d", wiek); // nowa wartosc
    return ERR_OK;
}
LPVOID DllGetVal_Wiek(int nVar)
{
    if(nVar == 0)
        return wiek_znak;
    return 0;
}

```

Funkcja DllGetVal_Wiek sprawdza wartość nVar: gdy jest ona równa 0, a zatem chodzi o pobranie nowej wartości pierwszego parametru (Wiek), z którym została wywołana funkcja weryfikacji, wartością DllGetVal_Wiek jest wskazanie na bufor przechowujący tę nową wartość. Dla pozostałych wartości nVar jest zwracane 0, co oznacza, że odpowiednie parametry nie zostały przez funkcję weryfikacji zmodyfikowane.

Zwróćmy uwagę na fakt, że w powyższym przykładzie można było sprawdzać, jaka jest wejściowa wartość parametru Wiek. Jeżeli dane były poprawne - tzn. zgodne z wyznaczonymi w funkcji weryfikacji - DllGetVal_Wiek powinna mieć wartość 0, dla wszystkich nVar. Wskazanie na bufor nowej wartości przy takim podejściu powinno być zwracane przez DllGetVal_Wiek, jedynie wtedy, gdy nowa wartość różniła się od wartości wejściowej Wiek.

Trzeba zwrócić uwagę na jeszcze jedną sprawę. Wartość Wiek jest w istocie wartością numeryczną. Jednakże jej nową wartość przekazujemy w postaci **znakowej**.

Modyfikacja wartości rozpoznawanego pola przez kryterium poprawności ma zasadniczy wpływ na poziom ufności dla poszczególnych znaków w polu. Domyślnie wszystkie znaki w polu zmodyfikowanym uzyskują maksymalny poziom ufności. Oznacza to, że dla operatorów weryfikacji to pole będzie wyglądało na poprawne (bez znaków wyświetlanych na czerwono). Takie rozwiązanie jest wystarczające jedynie dla prostych przypadków. Istnieje możliwość ustawienia nowych wartości poziomu ufności dla poszczególnych znaków. FormAnalyze Engine wywołuje w tym celu (o ile jest zdefiniowana w DLL) funkcję DllGetExInfo_Nazwa o następującym prototypie:

```

BOOL DllGetExInfo_Name (struct excharinfo* pInfo, int nVar);
// Wejscie: pInfo - wskazanie na pierwszy element tablicy rozszerzonych informacji o znakach

```

```
//      nVar - numer zmiennej (pola), dla którego ma
//      zostac pobrana rozszerzona informacja (liczac od 0)
// Wyjscie: pInfo - zmodyfikowana tablica informacji rozszerzonej
// Wartosc: TRUE - pInfo zawiera zmodyfikowane dane
//      FALSE - zmienna nie byla modyfikowana: pInfo nie bylo takze modyfikowane
```

Dla kryterium poprawności Wiek, poziom ufności znaków w wyznaczonej wartości powinien być ustawiony na minimalną wartość, ponieważ poprawność wyznaczonego Wiek zależy od poprawności Daty Urodzenia. Jeśli Data Urodzenia została rozpoznana niepoprawnie, to niepoprawny będzie także wyznaczony Wiek.

Poniżej przedstawiono implementację funkcji DllGetExInfo_Wiek:

```
BOOL DllGetExInfo_Wiek(struct excharinfo* pInfo, int nVar)
{
    if(nVar == 0)
    {
        for(int i = strlen(wiek_znak)-1; i >= 0; i--)
            pInfo[i].cl = 0;
        return TRUE;
    }
    return FALSE;
}
```

12.10.2. Integracja biblioteki

Przed kompilacją i konsolidacją biblioteki musimy zadbać jeszcze o dwie sprawy. Pierwszą z nich jest zdefiniowanie stosownych komunikatów błędów w tabeli napisów zasobów biblioteki. Drugą, przygotowanie pliku definicji interfejsu biblioteki zawierającego nazwy funkcji, które mogą być wywoływane spoza biblioteki.

Na rysunku poniżej przedstawiono widok tabeli napisów zdefiniowanych dla przedstawionej powyżej implementacji trzech funkcji weryfikacji.

ID	Value	Caption
IDS_DLL_NAME	1	Weryfikacja Ubezpieczeń
IDS_ERR_BADPARAMETERS	2	Niepoprawne parametry wywołania funkcji weryfikacji.
IDS_ERR_NDOW_DLUKOSC	3	Niepoprawna długość serii i numeru dowodu osobistego.
IDS_ERR_NDOW_NUMER	4	Niepoprawny numer dowodu osobistego.
IDS_ERR_NDOW_SERIA	5	Niepoprawna seria dowodu osobistego.
IDS_ERR_ZNJ_GRUPA	6	W grupie jest za mało pól wyboru.
IDS_ERR_ZNJ_BRAKSTOPN	7	Brak zaznaczenia stopnia znajomości pojęcia.
IDS_ERR_ZNJ_ZNANIEZNA	8	Zaznaczono jednocześnie pola 'Znam' i 'Nie znam'.
IDS_ERR_ZNJ_TYLKOSTOPN	9	Zaznaczony stopień znajomości bez pola 'Znam'.

Rys. 12-28 Tabela napisów zasobów biblioteki zalubezp.dll.

Treść pliku definicji interfejsu biblioteki jest następująca:

```
; valubezp.def : Declares the module parameters for the DLL.

LIBRARY      "valubezp"
DESCRIPTION  'valubezp Windows Dynamic Link Library'
EXPORTS
DllInitialize
DllReset
DllGetName
DllGetFuns
DllGetError
; zestaw dla funkcji weryfikacji NrDowodu
DllProcess_NrDowodu
DllGetVal_NrDowodu
DllGetVars_NrDowodu
```

```
; zestaw dla funkcji weryfikacji Wiek
DllProcess_Wiek
DllGetVal_Wiek
DllGetVars_Wiek
DllGetExInfo_Wiek
; zestaw dla funkcji weryfikacji Znajomosc
DllProcess_Znajomosc
DllGetVal_Znajomosc
DllGetVars_Znajomosc
```

Jeżeli kompilacja i konsolidacja biblioteki zakończyła się powodzeniem dysponujemy biblioteką funkcji weryfikacji valubezp.dll. Przed skorzystaniem z niej w definiowaniu przetwarzania dokumentów musimy skopiować bibliotekę do foldera, w którym znajduje się plik designer.exe (FormAnalyzer Designer).

Sposób powiązania funkcji weryfikacji z polami dokumentu został opisany w punktach 12.2 - Definiowanie nowego kryterium poprawności i 7.8.6 - Funkcje weryfikacji z zewnętrznych bibliotek DLL.

Po zdefiniowaniu, które pola dokumentu będą weryfikowane przy użyciu funkcji weryfikacji, należy przetestować działanie funkcji. Sposób testowania funkcji weryfikacji z biblioteki DLL jest analogiczny jak w przypadku testowania działania skryptów. Przy funkcjach weryfikacji nie dostaniemy, co prawda komunikatów o niepoprawności składniowej. Mogą natomiast pojawić się komunikaty o niepoprawnym typie lub rozmiarze pól użytych jako parametry weryfikacji.

Jeśli testowanie funkcji potwierdziło, że działają one zgodnie z założeniami, musimy udostępnić bibliotekę weryfikacji dla wszystkich instalacji FormAnalyzer Engine i FormAnalyzer Verifier. Na każdym komputerze, na którym może działać Engine lub Verifier, do foldera, w którym znajduje się engine.exe lub ver.exe, trzeba skopiować bibliotekę weryfikacji (w naszym przykładzie valubezp.dll).

12.11. Tworzenie funkcji weryfikacji przy użyciu WebService.

Jak już wspomniano wcześniej, w systemie FormAnalyzer korzystanie z WebService umożliwia użytkownikowi utworzenie zestawu funkcji umożliwiających sprawdzania poprawności lub modyfikację wartości pól dokumentu.

12.11.1. Definicja usługi WebService.

Prawidłowe metody WebService wykorzystywane w systemie FormAnalyzer, powinny być skonstruowane w następujący sposób:

- Metoda musi posiadać co najmniej 1 argument. Wszystkie argumenty muszą być typu string.
- Metoda musi zwracać zmienną typu string lub sekwencję zmiennych typu string.
- Wartości pól wyboru przekazywane są poprzez ciąg znaków zawierających wartości „0” lub „1”.

Poniżej zaprezentowano przykładowy plik WSDL opisujący metodę GetClient.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="GetClient">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="NrUmowy" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetClientResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetClientResult"
type="tns:RespGetClient" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="RespGetClient">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="Imie" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Nazwisko" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetClientSoapIn">
    <wsdl:part name="parameters" element="tns:GetClient" />
  </wsdl:message>
  <wsdl:message name="GetClientSoapOut">
    <wsdl:part name="parameters" element="tns:GetClientResponse" />
  </wsdl:message>
  <wsdl:portType name="RService1Soap">
    <wsdl:operation name="GetClient">
```

```

        <wsdl:input message="tns:GetClientSoapIn" />
        <wsdl:output message="tns:GetClientSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="RService1Soap" type="tns:RService1Soap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetClient">
        <soap:operation soapAction="http://tempuri.org/GetClient" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="RService1Soap12" type="tns:RService1Soap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetClient">
        <soap12:operation soapAction="http://tempuri.org/GetClient" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="RService1">
    <wsdl:port name="RService1Soap" binding="tns:RService1Soap">
        <soap:address location="http://localhost:49268/RService1.asmx" />
    </wsdl:port>
    <wsdl:port name="RService1Soap12" binding="tns:RService1Soap12">
        <soap12:address location="http://localhost:49268/RService1.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

12.11.2. Definicja zmiennych wyjściowych funkcji

WebService wywoływany przez kryterium poprawności, które modyfikuje wartości pól szablonu musi zwracać sekwencję wartości typu string. Podczas definiowania kryterium wartości zwracane przez funkcję są przyporządkowywane polom szablonu.

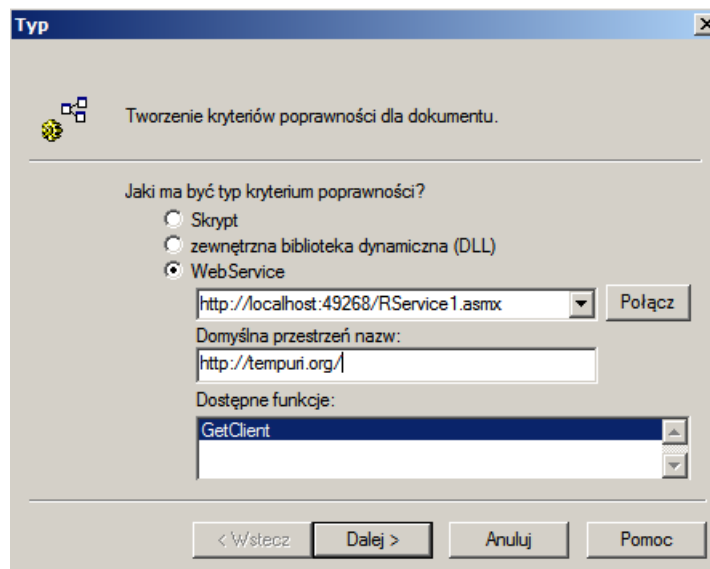
Jeśli funkcja WebService zwraca zmienną o nazwie *ErrorInfo* jest ona traktowana w sposób szczególny. Zmiennej tej nie można przypisać do pola szablonu. Zmienna wykorzystywana jest do zwracania informacji o poprawności wykonania funkcji. Wartość "" (pusty tekst) przypisana zmiennej *ErrorInfo* oznacza, że funkcja zakończyła się poprawnie. W tym przypadku wartość pozostałych zwróconych zmiennych przypisywana jest do pól szablonu. Przypisanie zmiennej *ErrorInfo* wartości różnej od "" oznacza, że wykonanie funkcji zakończyło się błędem. W tym przypadku wartość pól szablonu nie jest modyfikowana a wartość zmiennej *ErrorInfo* traktowana jest jako informacja o przyczynie wystąpienia błędu.

12.11.3. Tworzenie kryterium poprawności typu WebService

W celu utworzenia kryterium poprawności typu WebService należy:

- Z menu programu wybrać polecenie *Dokument->Kryteria poprawności...*
- W oknie *Kryteria poprawności dokumentów* kliknąć ikonę . Pojawi się okno kreatora kryterium poprawności.

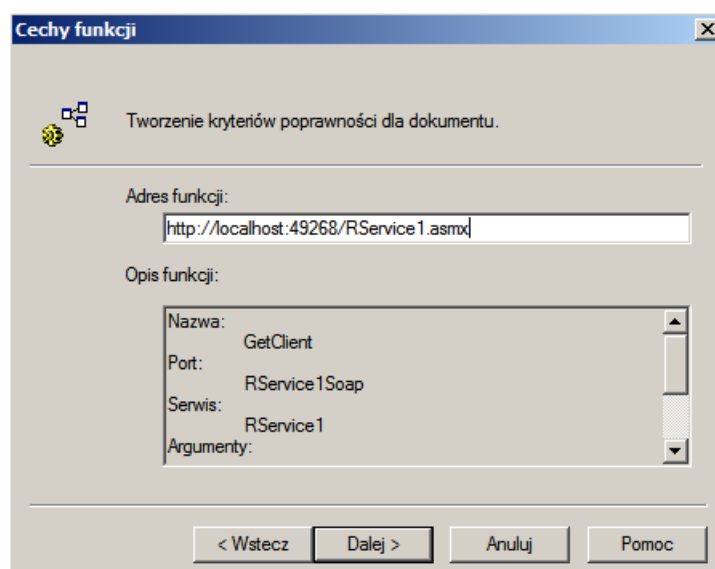
W oknie *Typ* kreatora należy wybrać opcję *WebService*. Udostępnione zostaną 3 pola: *Adres URL*, *Domyślna przestrzeń nazw* i *Dostępne funkcje*. W polu *Adres URL* (rozwijalna lista na lewo od przycisku *Połącz*) należy wprowadzić adres URL pod którym udostępniana jest definicja WebService (WSDL). Możemy bezpośrednio wskazać adres URL usługi np.: <http://www.arhat.com.pl/Wservices> lub odwołać się do definicji zapisanej w postaci pliku dyskowego np.: `file:///C:/Temp/GetClient.wsdl`. Kliknięcie przycisku *Połącz* powoduje wczytanie definicji usługi (WSDL) i wyświetlenie listy dostępnych funkcji. Jeżeli domyślna przestrzeń nazw (namespace) używana przez WebService jest inna niż <http://tempuri.org/> przed kliknięciem przycisku *Połącz* w polu *Domyślna przestrzeń nazw* należy wpisać odpowiednią wartość. W obszarze *Dostępne funkcje*, wyświetlone zostaną wszystkie, poprawnie rozpoznane funkcje zdefiniowane we wskazanej definicji WSDL.



Rys. 12-29 Okno *Typ* kreatora kryterium poprawności.

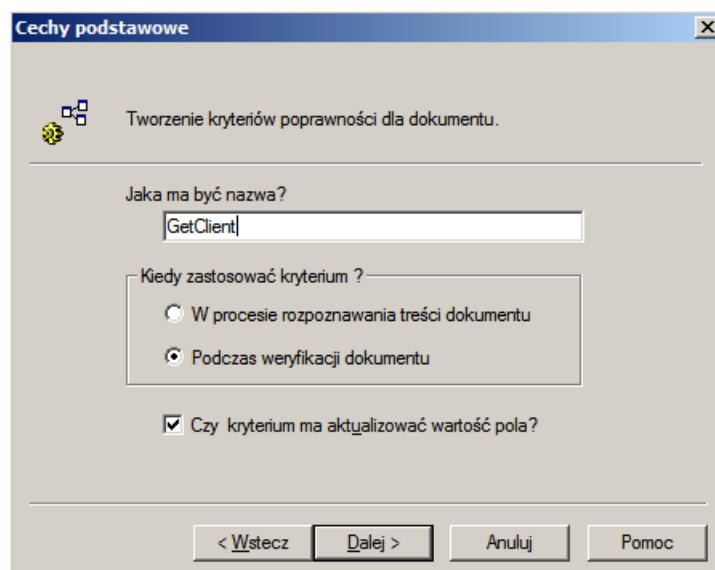
Na liście *Dostępne funkcje* należy wskazać odpowiednią funkcję i kliknąć przycisk *Dalej >*.

Kolejnym oknem kreatora jest okno *Cechy funkcji*. W polu *Adres funkcji* pojawia się adres pod którym dostępna jest wybrana funkcja WebService. Adres pobierany jest z definicji WSDL i w obecnej wersji programu nie można go zmienić. W polu *Opis funkcji* wymienione są pozostałe cechy funkcji odczytane z definicji WSDL.



Rys. 12-30 Okno *Cechy funkcji* kreatora kryterium poprawności.

Następna strona kreatora: *Cechy podstawowe* umożliwia wprowadzenie nazwy dla tworzonego kryterium poprawności, wskazanie programu w którym kryterium ma zostać uruchomione, oraz określenie czy kryterium ma aktualizować pola szablonu. Znaczenie opcji wskazywanych na tej stronie jest analogiczne jak dla skryptów i bibliotek DLL.



Rys. 12-31 Okno *Cechy podstawowe* kreatora kryterium poprawności

Dwa kolejne okna kreatora umożliwiają związanie pól szablonu z argumentami (zmiennymi wejściowymi) i rezultatami (zmiennymi wyjściowymi) funkcji Webservice. Okno zmiennych wyjściowych pojawia się tylko w przypadku gdy kryterium modyfikuje wartość pól szablonu.

Na liście są umieszczone następujące informacje:

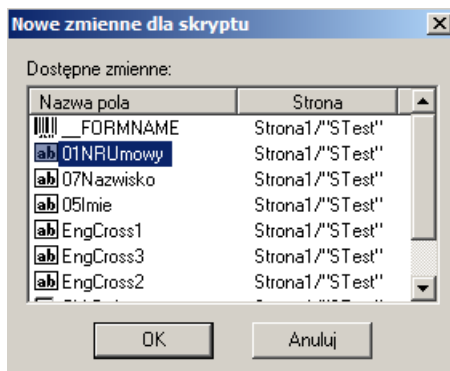
Typ	typ pola rozpoznawanego oznaczony ikoną.
Nazwa pola	nazwa pola dokumentu.

Strona opis strony w szablonie.

Nazwa zmiennej nazwa argumentu używana w Webservice.

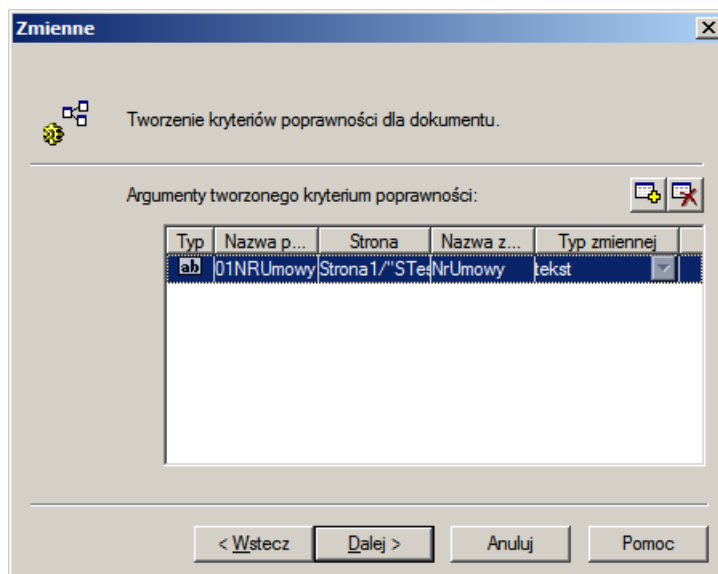
Typ zmiennej typ argumentu w Webservice.

Nowe elementy do listy argumentów (lub listy rezultatów) dodaje się wskazując pola szablonu w oknie *Nowe zmienne dla skryptu*. Okno otwiera kliknięcie przycisku .

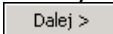


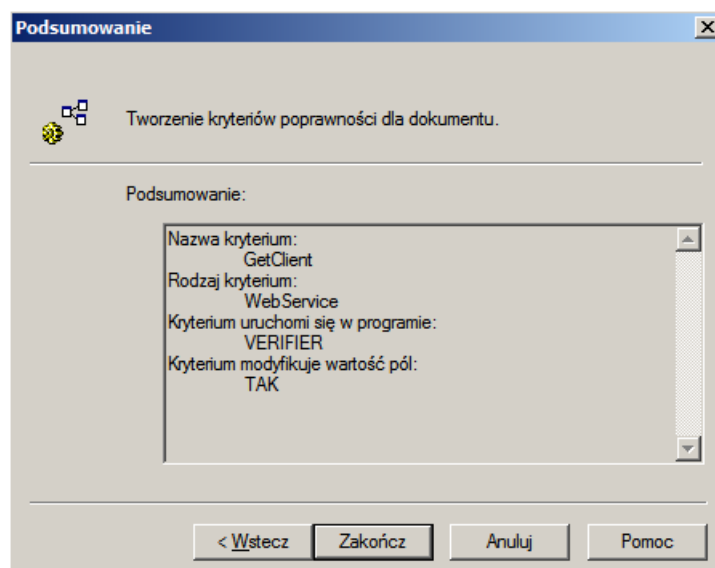
Rys. 12-32 Okno *Nowe zmienne dla skryptu*.

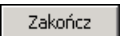
Należy pamiętać, że kolejne pola muszą odpowiadać kolejności parametrów zdefiniowanych w funkcji Webservice.



Rys. 12-33 Okno *Zmienne kreatora*.

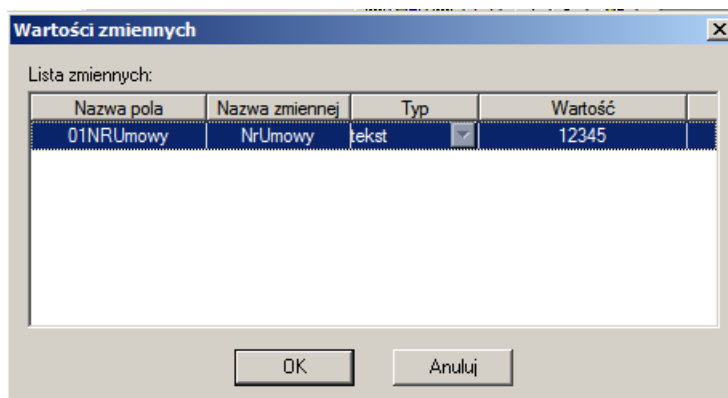
Wskazanie zmiennych wejściowych (argumentów) i ewentualnie zmiennych wyjściowych jest ostatnim krokiem tworzenia kryterium poprawności. Kliknięcie przycisku  powoduje wyświetlenie okna *Podsumowanie*.

Rys. 12-34 Okno *Podsumowanie* kreatora.

Kliknięcie przycisku  powoduje dopisanie definicji kryterium do aktualnego skryptu konfiguracyjnego.

12.11.4. Sprawdzanie poprawności usługi WebService

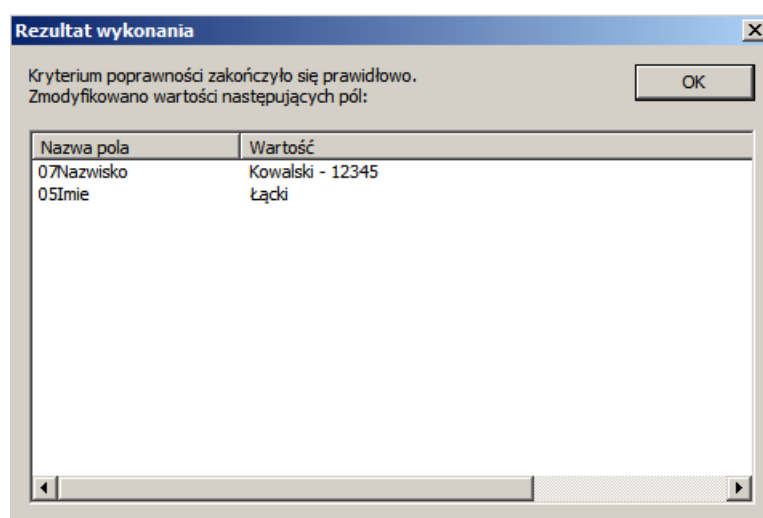
Sprawdzenie działania funkcji WebService rozpoczyna się naciśnięciem przycisku . Program otwiera okno parametrów, w którym podajemy wartości argumentów, dla których zostanie wykonana odpowiednia funkcja WebService.

Rys. 12-35 Okno *Wartości zmiennych*

Kliknięcie przycisku OK powoduje wywołanie funkcji WebService.

Jeżeli funkcja należy do kryterium weryfikującego poprawność danych to zakończenie wykonania funkcji zostanie potwierdzone wyświetleniem komunikatu zwróconego przez funkcję.

Jeżeli funkcja należy do kryterium modyfikującego wartość pól to zakończenie jej wywołania zostanie potwierdzone pojawieniem się okna *Rezultat wykonania*. W oknie prezentowana jest lista pól zmodyfikowanych w wyniku wywołania funkcji.

Rys. 12-36 Okno *Rezultat wykonania*.

13. Słowniki dynamiczne.

13.1. Co to są słowniki dynamiczne?

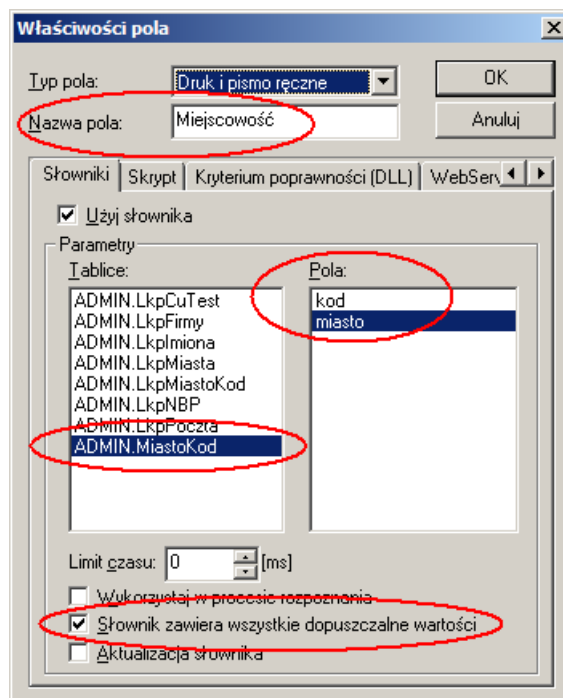
Słowniki dynamiczne są rozszerzeniem funkcjonalności słowników wartości i umożliwiają wybór wartości na podstawie zawartości innych pól formularza. Mogą być wykorzystywane zarówno w procesie weryfikacji, jak i/lub w procesie rozpoznawania.

Przypomnijmy, że słowniki tworzymy przy pomocy narzędzi administracyjnych dostępnych w Sybase SQLCentral. Słownik dynamiczny jest tabelą na DBSpace Lookup.

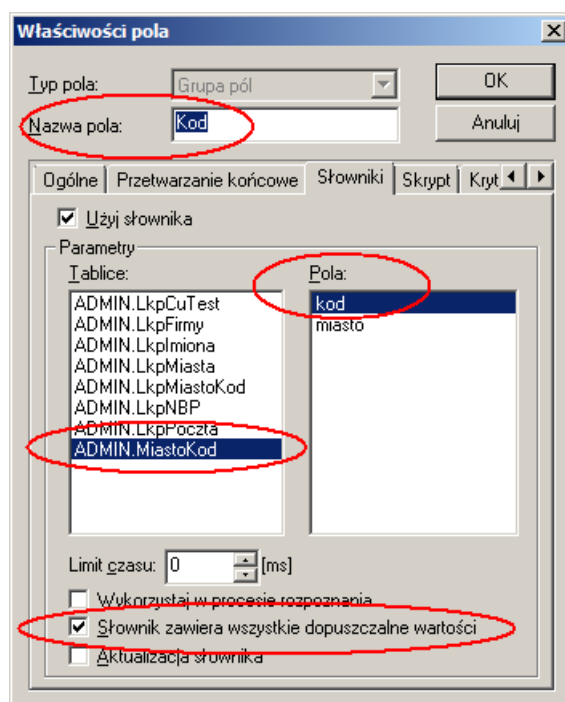
Szczegółowy opis słowników wartości przedstawiony został w punkcie 7.7.4 „Słowniki wartości”.

13.2. Czynności przedwstępne

W celu skorzystania ze słowników dynamicznych, użytkownik musi najpierw, podczas przypisywania tabeli i kolumn do odpowiednich pól, w oknie *Właściwości pola*, w zakładce *Słowniki* wybrać opcję *Słownik zawiera wszystkie dopuszczalne wartości*. Zaznaczenie tego pola wyboru spowoduje, że słownik zawierać będzie kompletną liczbę wartości. Czynność tę należy powtórzyć w przypadku wszystkich pól, względem których użytkownik będzie chciał skorzystać ze słowników dynamicznych. W zaprezentowanym niżej przykładzie będą to pola formularza *Miejscowość* i *Kod*, do których przypisana została tablica *dba.MiastoKod*, a z niej kolejno pola: *miasto* i *kod*. Wstępne ustawienia w oknie dialogowym *Właściwości pola* prezentują dwa poniższe rysunki:



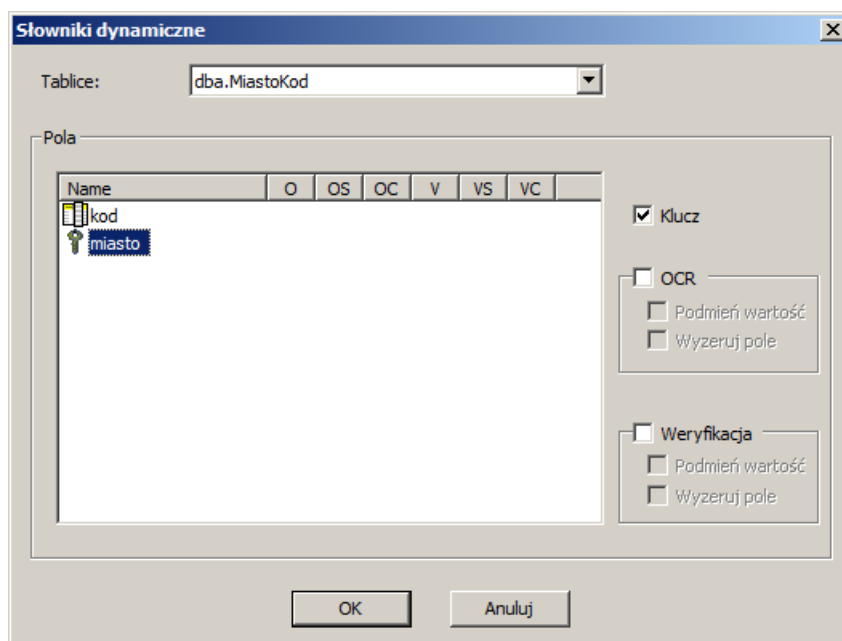
Rys. 13-1 Okno dialogowe Właściwości pola dla pola formularza *Miejscowość*



Rys. 13-2 Okno dialogowe Właściwości pola dla pola formularza Kod

13.3. Używanie słowników dynamicznych w procesie przetwarzania dokumentów

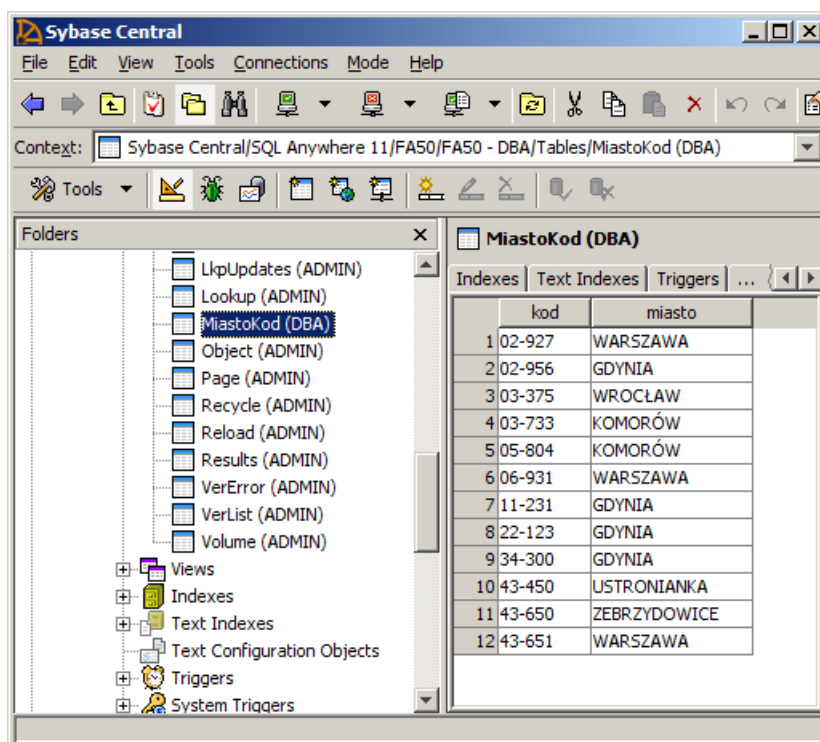
W celu zdefiniowania ustawień słowników dynamicznych, należy z menu *Plik* wybrać polecenie *Słowniki dynamiczne*. Wyświetli się okno dialogowe *Słowniki dynamiczne*.



Rys. 13-3 Okno dialogowe Słowniki dynamiczne

W oknie dialogowym *Słowniki dynamiczne*, z rozwijalnej listy *Tablice*, należy wskazać tabelę, z której pola przypisaliśmy do odpowiednich pól formularza. Wybór odpowiedniej tabeli spowoduje wyświetlenie w obszarze *Pola* listy dostępnych pól. W naszym

przykładzie po wybraniu tabeli *dba.MiastoKod*, w obszarze Pola pojawiły się pola: *kod* i *miasto*. W następnej kolejności należy wskazać, które z dostępnych pól będzie „Kluczem”. W tym celu należy zaznaczyć w obszarze *Pola* jedno z pól i kliknąć na opcję *Klucz*, która znajduje się po prawej stronie w obszarze *Pola*. Przy nazwie pola wyświetli się symbol . Wskazanie jednego z dostępnych pól jako *Klucz* będzie miało bezpośredni wpływ na wyświetlanie wartości w pozostałych polach (polach zależnych). Poniżej zaprezentowano przykładowy słownik wartości dla dwóch rodzajów pól: Miasto i Kod.

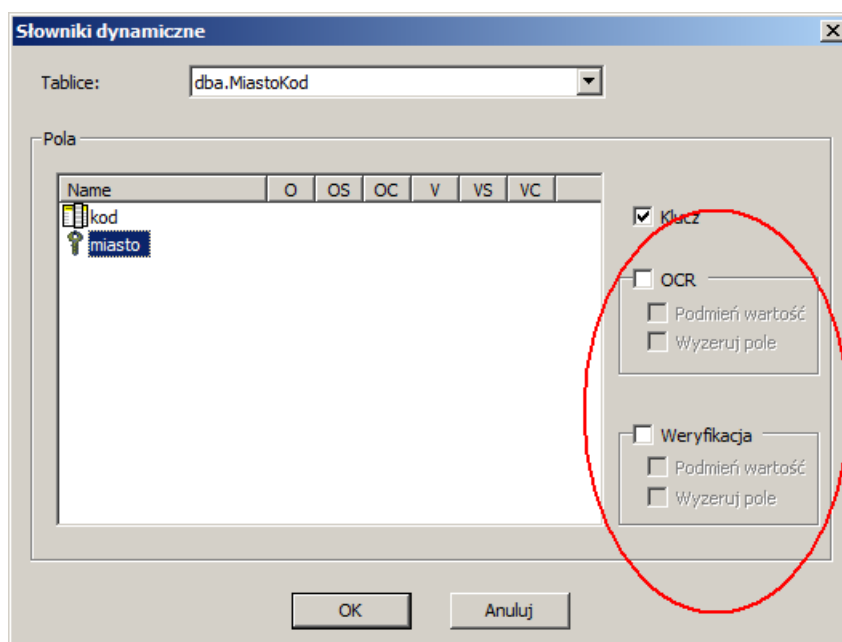


Rys. 13-4 Przykładowy słownik wartości: *MiastoKod(DBA)*

W przypadku wskazania pola *Miasto*, jako *Klucza*, przykładowo wprowadzenie wartości WARSZAWA, w takie pole spowoduje, że dla pola zależnego, którym stanie się automatycznie pole *Kod*, dostępne będą wartości kodów pocztowych tylko związanych z Warszawą.

Poprawność działania słowników dynamicznych zależy jest od odpowiedniego przygotowania przez użytkownika słowników wartości, a także wskazania odpowiedniego pola kluczowego, które powinno być punktem odniesienia dla pozostałych kolumn w słowniku wartości.

Słowniki dynamiczne można wykorzystywać w procesach weryfikacji i/lub rozpoznawania. W tym celu w oknie dialogowym *Słowniki dynamiczne*, należy zdefiniować odpowiednie ustawienia używając grup opcji znajdujących się w prawej części obszaru Pola.



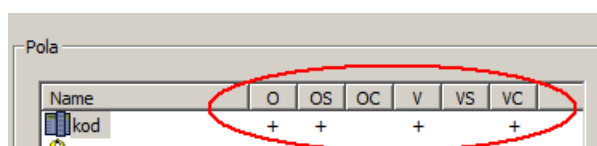
Rys. 13-5 Obszar definiowania ustawień dla słowników dynamicznych.

Użytkownik może zdefiniować następujące ustawienia:

1. Dla procesu weryfikacji (V):
 - Podmień wartość (VS);
 - Wyzeruj pole (VC);
2. Dla procesu rozpoznawania (O):
 - Podmień wartość (OS);
 - Wyzeruj pole (OC).

W zależności od zaznaczonego ustawienia, w odpowiedniej kolumnie („O”, „OS”, „OC”, „V”, „VS”, „VC”) znajdującej się w obszarze Pola, pojawi się znaczek „+”.

Wszystkich ustawień należy dokonywać na polach zależnych od klucza.



Rys. 13-6 Przykładowe ustawienia zdefiniowane dla pola zależnego Kod.

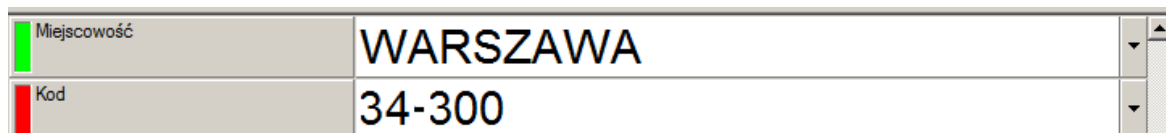
Na rysunku powyżej użytkownik zdefiniował dla przykładowego pola Kod następujące ustawienia:

1. Dla procesu rozpoznawania (OCR):
 - Podmień wartość (OS);
2. Dla procesu weryfikacji (V):
 - Wyzeruj pole (VC).

13.3.1. Słowniki dynamiczne w procesie weryfikacji

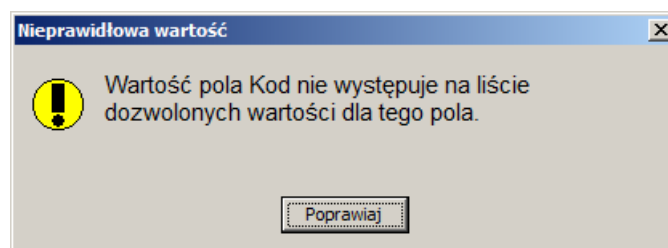
W celu wykorzystania słowników dynamicznych w procesie weryfikacji, należy w oknie dialogowym *Słowniki dynamiczne*, zaznaczyć pole wyboru przy opcji Weryfikacja. Spowoduje to uaktywnienie pozostałych dwóch opcji: *Podmień wartość* oraz *Wyzeruj pole*.

Pozostawienie takiego ustawienia spowoduje, że w procesie weryfikacji system dokona kontroli poprawności pola, jednakże nie dokona żadnych modyfikacji wartości. Jeżeli wartość, w którymś z pól, wobec których zastosowano słowniki dynamiczne, będzie nieprawidłowa, znacznik poprawności takiego pola będzie miał kolor czerwony.



Rys. 13-7 Nieprawidłowa wartość w polu Kod

Przy próbie zaakceptowania takiego pola, system wyświetli odpowiednie okno komunikatu o błędzie.



Rys. 13-8 Okno błędu Nieprawidłowa wartość.

W celu wprowadzenia prawidłowej wartości, należy rozwinąć listę wartości klikając przycisk  znajdujący się poprawnej stronie pola. Dostępne wartości na liście zależą od pola kluczowego.

The screenshot shows a window titled "Obraz" containing a large text area with the handwritten number "34 300". Below this is a form with several fields. The "Kod" field is highlighted in yellow and shows "34-300". A red circle highlights the "Telefon" field, which contains a list of phone numbers: "02-927", "06-931", and "43-651". Other fields include "Miejscowość" (Warszawa), "Ulica", "NrDomu", and "Nr Mieszkania".

Rys. 13-9 Dostępne wartości przykładowego pola *Kod*, przy prawidłowej wartości pola kluczowego *Miasto*.

W sytuacji, gdyby pole kluczowe nie miało wprowadzonej wartości, lub miało wartość niepoprawną, rozwinięcie listy pola zależnego spowoduje wyświetlenie wszystkich wartości zawartych w słowniku wartości, na podstawie którego zdefiniowano ustawienia słowników dynamicznych.

The screenshot shows a window titled "Obraz" containing a large text area with the handwritten number "34 300". Below this is a form with several fields. The "Kod" field is highlighted in yellow and shows a list of values: "02-927", "02-956", "03-375", "03-733", "05-804", "06-931", "11-231", "22-123", "34-300", and "43-450". A red circle highlights the "Telefon" field, which contains a list of phone numbers: "02-927", "02-956", "03-375", "03-733", "05-804", "06-931", "11-231", "22-123", "34-300", and "43-450". Other fields include "Miejscowość", "Ulica", "NrDomu", and "Nr Mieszkania".

Rys. 13-10 Dostępne wartości przykładowego pola *Kod*, przy niewypełnionym polu kluczowym *Miasto*.

13.3.1.1 Podmień wartość

Zaznaczenie opcji *Podmień wartość* spowoduje:

1. Jeżeli wartość w polu zależnym nie zawiera się w słowniku lub takie pole jest puste, wtedy system podstawia jedną z wartości ze słownika powiązaną z wartością pola zdefiniowanego, jako *Klucz*, (np. jeżeli w naszym polu *Miasto*, znajdowałaby się wartość WARSZAWA, a w polu *Kod*, system nie odnalazłby wartości, lub odnalazłby wartość niezgodną ze słownikiem, wtedy podstawiałby jedną z wartości kodu pocztowego ze słownika, bezpośrednio związaną z Warszawą).
2. Jeżeli wartość w polu zdefiniowanym, jako *Klucz* nie zawiera się w słowniku, wtedy system nie przypisze żadnych wartości do pól zależnych.
3. W obydwu przypadkach przy próbie akceptacji któregośkolwiek z nieprawidłowo wypełnionych pól, system wyświetli okno komunikatu o błędzie nieprawidłowej wartości w polu.
4. Jeżeli wartość w polu zależnym zgadza się będzie z wartością znajdującą się w słowniku, wtedy system nie dokona żadnych zmian.

13.3.1.2 Wyzeruj pole

Zaznaczenie opcji *Wyzeruj pole* spowoduje:

1. W przypadku, gdy wartość w polu zależnym nie zawiera się w słowniku lub takie pole jest puste, wtedy system usuwa nieprawidłową wartość z pola lub pozostawia je pustym.
2. Jeżeli wartość w polu zdefiniowanym, jako *Klucz* nie zawiera się w słowniku, wtedy system nie przypisze żadnych wartości do pól zależnych.
3. W obydwu przypadkach przy próbie akceptacji któregośkolwiek z nieprawidłowo wypełnionych pól, system wyświetli okno komunikatu o błędzie nieprawidłowej wartości w polu.
4. Jeżeli wartość w polu zależnym zgadza się będzie z wartością znajdującą się w słowniku, wtedy system nie dokona żadnych zmian.

13.3.1.3 Zaznaczone jednocześnie obydwu opcji: Podmień wartość i Wyzeruj pole

W przypadku, gdy użytkownik zaznaczy obydwie opcje:

1. Jeżeli w słowniku wartość jest jednoznaczna, to zostanie podstawiona.
2. Jeżeli w słowniku znajduje się wiele wartości, to zostanie usunięta. Jedynym odstępstwem od tego działania, jest przypadek, gdy w słowniku znajduje się umieszczona w polu wartość. Wtedy system pozostawi taką wartość bez zmian.

13.3.2. Słowniki dynamiczne w procesie rozpoznawania.

W przypadku zastosowania słowników dynamicznych w procesie rozpoznawania, działanie opcji: *Podstaw wartość* i *Wyzeruj pole* jest analogicznie do funkcjonalności opisanej przy okazji procesu weryfikacji. Po zakończeniu procesu rozpoznawania system, sekwencyjnie

dokonywa działań zgodnych z ustawieniami zdefiniowanymi przez użytkownika w oknie dialogowym *Słowniki dynamiczne*: podstawia wartość ze słownika lub zeruje pole. Proces ten odbywa się bezobsługowo, po zakończeniu którego dokument zapisywany jest do dalszego przetwarzania.

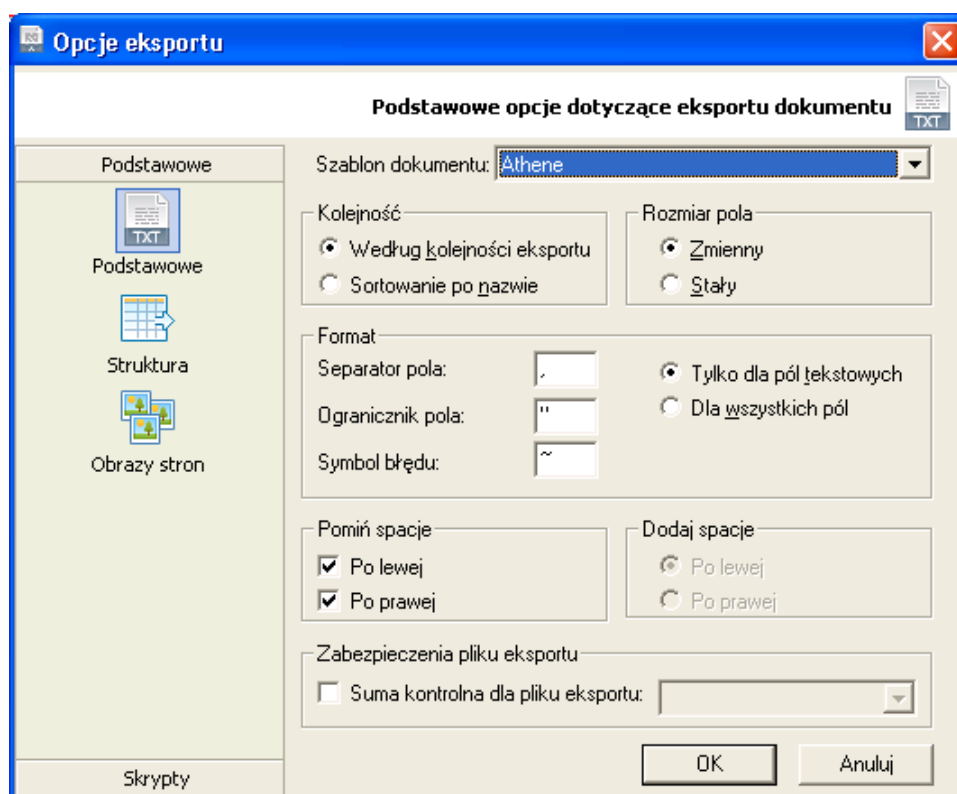
14. Format danych wyjściowych

W systemie FormAnalyzer zweryfikowane dane są zapisywane do plików wyjściowych, z których są pobierane przez system docelowy. Pliki wyjściowe są tworzone przez program FormAnalyzer Export.

Dla każdego typu (szablону) dokumentu zdefiniowanego w pliku konfiguracji można zaprojektować postać danych przekazywanych do systemu docelowego.

Projektowanie formatu danych wyjściowych umożliwia okno dialogowe *Opcje eksportu*, otwierane za pomocą polecenia *Opcje eksportu* z menu *Dokument*. Polecenie jest aktywne tylko w przypadku, kiedy w aktualnym pliku konfiguracji został zdefiniowany przynajmniej jeden szablon dokumentu.

Na poniższym rysunku przedstawiono okno parametrów eksportu bezpośrednio po otwarciu.



Rys. 14-1 Okno opcji eksportu po otwarciu (grupa opcji Podstawowe).

Okno eksportu składa się z trzech części: paska wyboru szablonu, listy grup opcji oraz formatki z opcjami. W pasku wyboru szablonu dostępna jest lista wszystkich szablonów zdefiniowanych w bieżącym pliku konfiguracyjnym. W liście tej występuje także pozycja *wszystkie szablony dokumentów*, która umożliwia jednoczesną zmianę ustawień dla całego pliku konfiguracyjnego. Po otwarciu okna zaznaczony jest szablon, w którego skład wchodzi aktualnie edytowany formularz w głównym oknie programu. W formacie z opcjami wyświetlane są cechy należące do zaznaczonej grupy oraz dla wybranego szablonu. Zmiany zapamiętywane są dopiero po naciśnięciu  i zamknięcie okna z

opcjami eksportu. Zamknięcie okna przyciskiem  powoduje, że jest przywracany stan opcji eksportu sprzed otwarcia okna.

Należy nadmienić, że każde weryfikowane pole posiada dodatkowe cechy związane z eksportem danych: znacznik czy pole ma być eksportowane, format eksportu pola i rozmiar pola w pliku eksportu. Wartości tych cech są zmieniane we właściwościach pól. Są one dostępne po wyborze polecenie *Pola* z menu *Dokument*. Dokładny opis znajduje się w rozdziale 9.5 - Opcje eksportu.

14.1. Grupa opcji *Podstawowe*

W grupie *Kolejność* jest ustawiana kolejność, w jakiej wartości pól będą umieszczane w pliku danych wyjściowych. Zaznaczenie opcji *Według kolejności eksportu* oznacza, że wartości pól będą umieszczone w pliku wyjściowym w takiej kolejności, jaką zdefiniowano dla konkretnego szablonu dokumentu (ustawianie kolejności eksportu omówiono w punkcie 10.3 - Kolejność eksportu pól).

Przy wybranej opcji *Sortowanie po nazwie*, o kolejności zapisu wartości pól do pliku wyjściowego decyduje pozycja pola w posortowanej leksykograficznie (względem nazwy) liście pól.

Grupa *Rozmiar pola* pozwala na ustawienie rozmiaru pojedynczego pola. Można tu wybrać jedną z dwóch dostępnych opcji. Zaznaczenie opcji *Zmienny* oznacza, że pola w pliku wyjściowym będą mogły mieć różną długość. Będzie ona zależała od długości wartości pola rozpoznanej na konkretnym dokumencie. W przypadku wybrania opcji *Stały*, długość pola będzie taka sama dla wszystkich dokumentów. W przypadku dokumentów, dla których rozpoznana wartość jest krótsza niż zadana długość pola, pole w pliku wyjściowym zostanie dopełnione spacjami.

W grupie *Format* jest ustawiany sposób separacji danych w pliku wyjściowym. Pole *Separator pola* zawiera znak oddzielający wartości pól dokumentu. Domyślną wartością ciągu oddzielającego pola jest znak przecinka (,). Szczególnym przypadkiem jest ustawienie pustego ciągu oddzielającego. W takim przypadku wartości pól dokumentu będą umieszczane jedna po drugiej, bez dodatkowych znaków oddzielających pola. Takie rozwiązanie jest stosowane najczęściej przy stałej szerokości pól danych wyjściowych.

Pole *Ogranicznik pola* zawiera znak ograniczających wartość pola. Znaki ograniczające są umieszczane zarówno przed, jak i po wartości pola. Domyślną wartością tego pola jest cudzysłów (").

Ogranicznik może być stosowany do wartości pól tekstowych lub wartości wszystkich pól dokumentu. Decyduje o tym odpowiednio zaznaczenie opcji *Tylko dla pól tekstowych* lub *Dla wszystkich pól*.

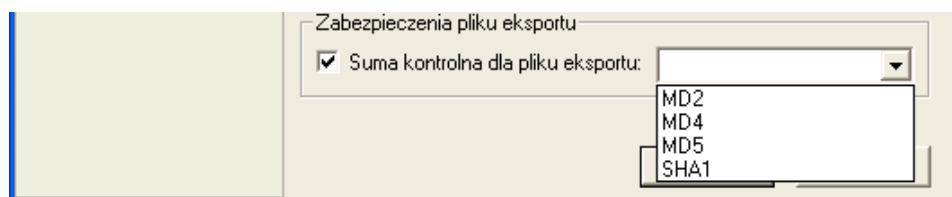
Grupa *Pomiń spacje* pozwala na ustawienie usuwania wiodących i kończących spacji. Opcje *Po lewej* i *Po prawej* są zaznaczane niezależnie. Zaznaczenie *Po lewej* oznacza, że z wartości pola będą usuwane spacje wiodące. Zaznaczenie *Po prawej* oznacza, że z wartości pola będą usuwane spacje kończące.

Grupa *Dodaj spacje* jest wykorzystywana przy zapisywaniu pól o stałej długości. Jeżeli długość wartości pola dokumentu jest mniejsza niż ustawiona dla tego pola, to jest ono uzupełniane spacjami do właściwej długości. Spacje mogą być dodane na początku (zaznaczona opcja *Po lewej*) lub na końcu wartości pola (zaznaczona opcja *Po prawej*).

Przed uzupełnianiem wartości pola spacjami może być wykonywane usuwanie spacji wiodących i/lub kończących (zgodnie z ustawieniami grupy *Pomiń spacje*).

W grupie *Zabezpieczenia pliku eksportu* można ustawić sumę kontrolną pliku eksportu. Jest to funkcja opcjonalna i domyślnie pozostaje wyłączona. W celu ustawienia sumy kontrolnej pliku eksportu należy zaznaczyć pole wyboru *Suma kontrolna dla pliku eksportu*. Uaktywniona zostanie lista rozwijalna, z której należy wskazać jeden z czterech standardowych algorytmów generujących sumę kontrolną:

- MD2 – (128bit);
- MD4 – (128bit);
- MD5 – (128bit);
- SHA1 – (160bit).



Rys. 14-2 Rozwinięta lista Suma kontrolna dla pliku eksportu.

W zależności od wybranej opcji moduł FormAnalyzer Export będzie dodawał na końcu pliku eksportu linie tekstu wygenerowaną sumą kontrolną. Będzie ona zapisana szesnastkowo. Dla algorytmów MD2, MD4 i MD5 długość tego ciągu będzie wynosiła 32 znaki, a dla SHA1 – 40 znaków. Poniżej zaprezentowano struktury dwóch przykładowych plików zawierających wyeksportowane dane dla ustawionych opcji SHA1 i MD5.

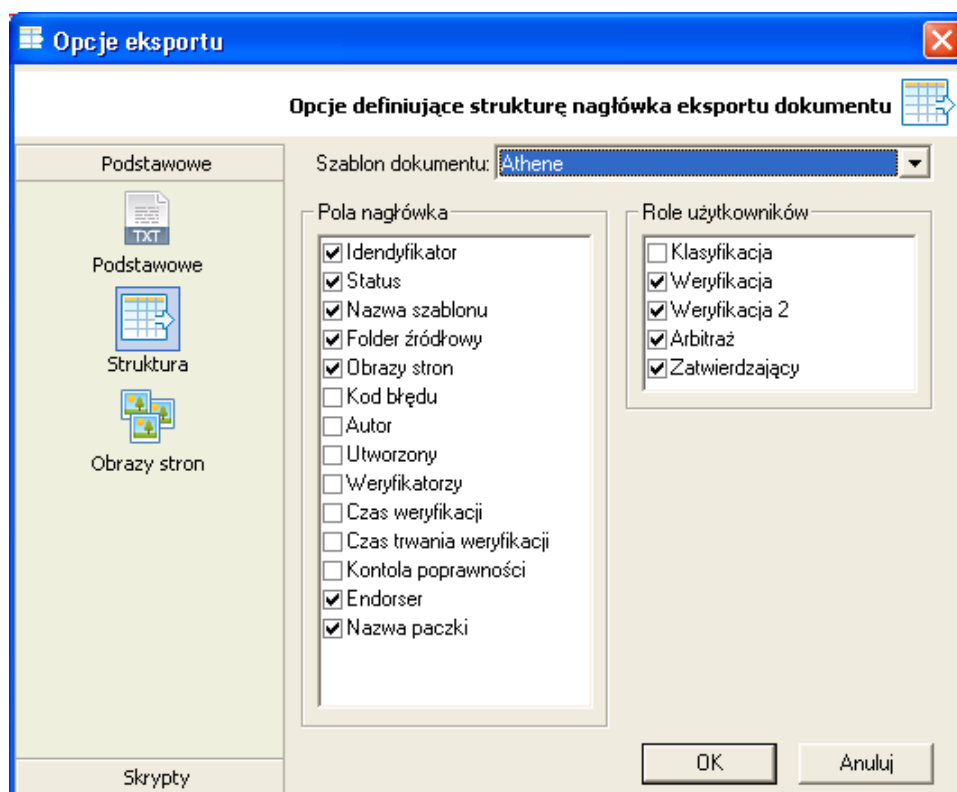
```
183
Verified (exported)
RTS
$\Nowy folder\
1
000000c3.tif
"570-92-5062","ROBERTS PATRICIA",1,12,"03-14-54"
184
Verified (exported)
RTS
$\Nowy folder\
1
000000c4.tif
"570-92-5060","ROBERTS PATRICIA",1,12,"03-14-54"
54998a24601e5953fdb614ca2d790eb099696957
```

Rys. 14-3 Przykładowy plik eksportu dla ustawionej opcji SHA1

```
183
Verified (exported)
RTS
$\Nowy folder\
1
000000c3.tif
"570-92-5062","ROBERTS PATRICIA",1,12,"03-14-54"
184
Verified (exported)
RTS
$\Nowy folder\
1
000000c4.tif
"570-92-5060","ROBERTS PATRICIA",1,12,"03-14-54"
61eb7714336f386639571c17ab084360
```

Rys. 14-4 Przykładowy plik eksportu dla ustawionej opcji MD5

14.2. Grupa opcji *Struktura*



Rys. 14-5 Okno ogólnych właściwości formatu danych wyjściowych, grupa opcji *Struktura*.

W grupie *Nagłówki dokumentu* można zaznaczyć eksport dodatkowych informacji związanych z dokumentem, które nie stanowią jednak treści dokumentu. Dostępne są następujące informacje o dokumencie:

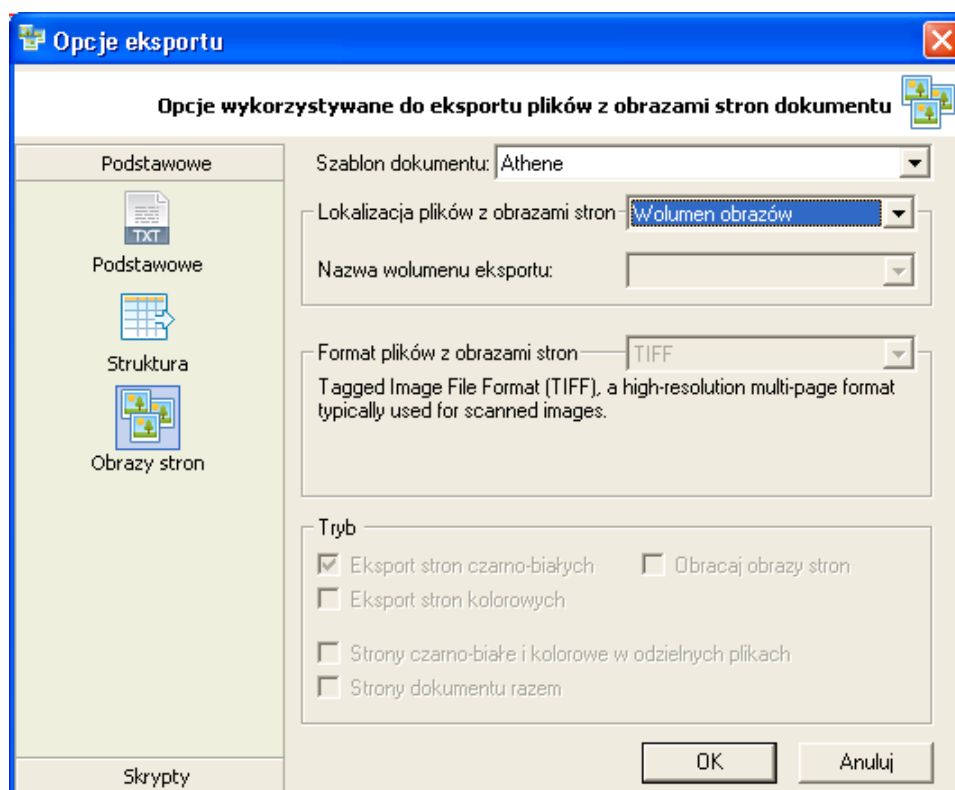
- *Identyfikator* dokumentu w bazie danych systemu FormAnalyze
- *Status* przetwarzania dokumentu
- *Nazwa szablonu* dokumentu zgodnie, z którym była rozpoznawana treść dokumentu
- *Folder źródłowy* w bazie danych, w którym znajduje się dokument
- *Obrazy stron* dokumentu
- *Kod błędu* ustawiony dla dokumentu w czasie rozpoznawania jego treści lub jej weryfikacji
- *Autor* - identyfikator użytkownika, który wprowadził dokument do bazy danych FormAnalyze
- *Utworzony* - data i czas utworzenia dokumentu w bazie danych FormAnalyze
- *Weryfikatorzy* - identyfikator użytkowników, którzy weryfikowali dokument
- *Czas weryfikacji* - czas rozpoczęcia procesu weryfikacji.
- *Czas trwania weryfikacji* - długość weryfikacji dokumentu w sekundach
- *Kontrola poprawności* - identyfikator użytkownika, który kontrolował poprawność dokumentu

- *Endorser* – numer lub pieczęć nadrukowywana przez skaner podczas procesu skanowania, która może zostać pobrana przez system FormAnalyzer i umieszczona w pliku eksportu.
- *Nazwa paczki* – nazwa paczki dokumentów.

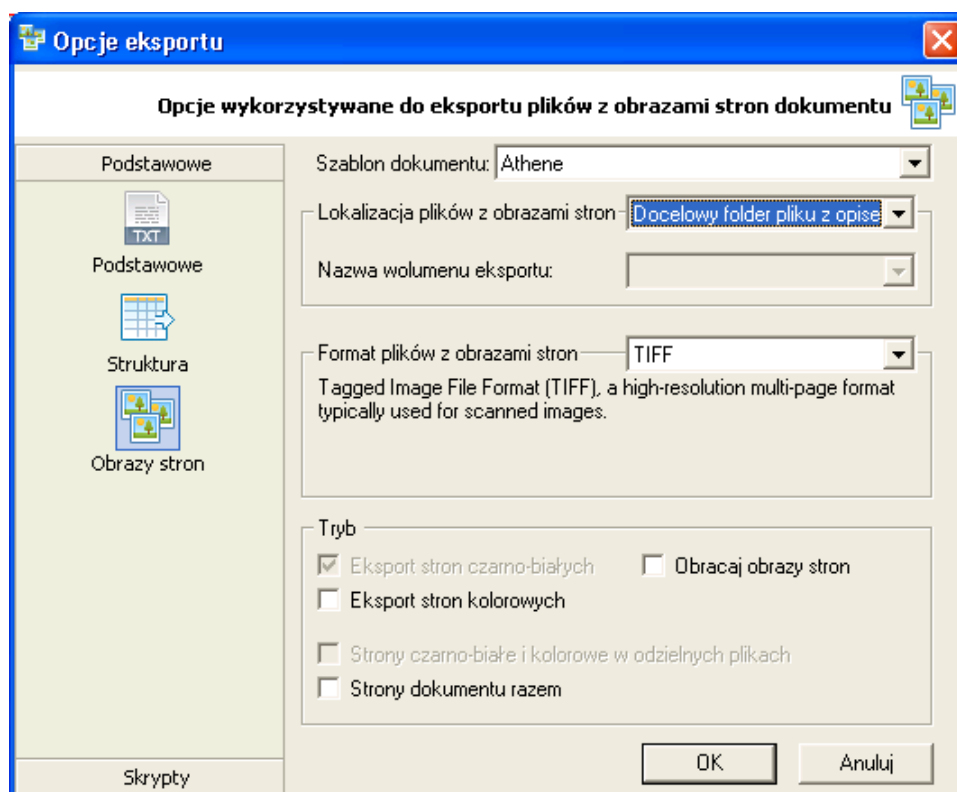
Dla opcji z obszaru *Pola nagłówka*: Weryfikatorzy, Czas weryfikacji oraz Czas trwania weryfikacji, można dodatkowo wskazać użytkowników posiadających odpowiednie uprawnienia, których dane dotyczące weryfikacji dokumentów zostaną wyeksportowane do pliku. W grupie *Role użytkowników* należy zaznaczyć odpowiednie pole wyboru:

- Klasyfikacja;
- Weryfikacja;
- Weryfikacja 2;
- Arbitraż;
- Zatwierdzający.

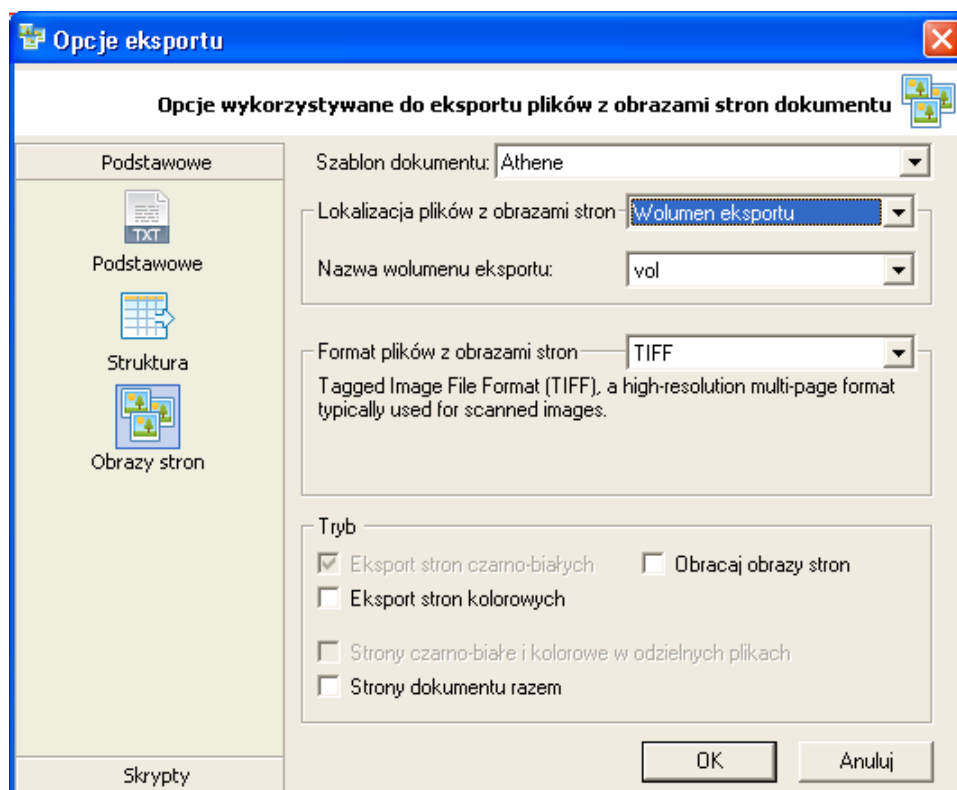
14.3. Grupa opcji *Obrazy stron*



Rys. 14-6 Okno właściwości formatu obrazów stron dokumentu (pobieranie plików z wolumenu stron dokumentu), grupa opcji *Obrazy stron*.



Rys. 14-7 Okno właściwości formatu obrazów stron dokumentu (tworzenie plików stron dokumentu w folderze docelowym pliku z opisem), grupa opcji *Obrazy stron*.



Rys. 14-8 Okno właściwości formatu obrazów stron dokumentu (pobieranie plików z wolumenu eksportu stron dokumentu), grupa opcji *Obrazy stron*.

W grupie *Lokalizacja plików z obrazami stron* można wybrać tryb eksportu plików zawierających obrazy stron dokumentu. Dostępne są następujące tryby eksportu plików z obrazami:

- *Wolumen obrazów* – w pliku eksportu zapisywana jest ścieżka dostępu do zeskanowanych stron z obrazami dokumentu. Są one składowane na wolumenach obrazów zdefiniowanych w module FormAnalyzer Scan&Administrator w formacie TIFF. Po wyborze tej opcji nie są dostępne pozostałe opcje z tej zakładki.
- *Docelowy folder pliku z opisem* – w pliku eksportu zapisywana jest nazwa pliku zawierającego strony z obrazami dokumentu. Plik tworzony jest w tym samym folderze, w którym zapisany jest plik z opisem eksportu. Po wyborze tej opcji możliwy jest wybór formatu pliku z obrazami oraz inne szczegółowe opcje dotyczące tworzenia pliku z obrazami stron dokumentu.
- *Wolumen eksportu* – w pliku eksportu zapisywana jest ścieżka dostępu do wytworzonych plików zawierających strony dokumentu. Są one składowane na wolumenie, który został wskazany z listy *Nazwa wolumenu eksportu*. Po wyborze tej opcji możliwy jest wybór formatu pliku z obrazami oraz inne szczegółowe opcje dotyczące tworzenia pliku z obrazami stron dokumentu. Stworzone podczas eksportu pliki zapisywane są na wskazanym wolumenie. Są one usuwane wraz z dokumentami w module FormAnalyzer Scan&Administrator oraz podczas usuwania sesji archiwizacji w module FormAnalyzer Archive.

Zaznaczenie opcji *Docelowy folder pliku z opisem* lub *Wolumen eksportu* uaktywni pięć następujących opcji:

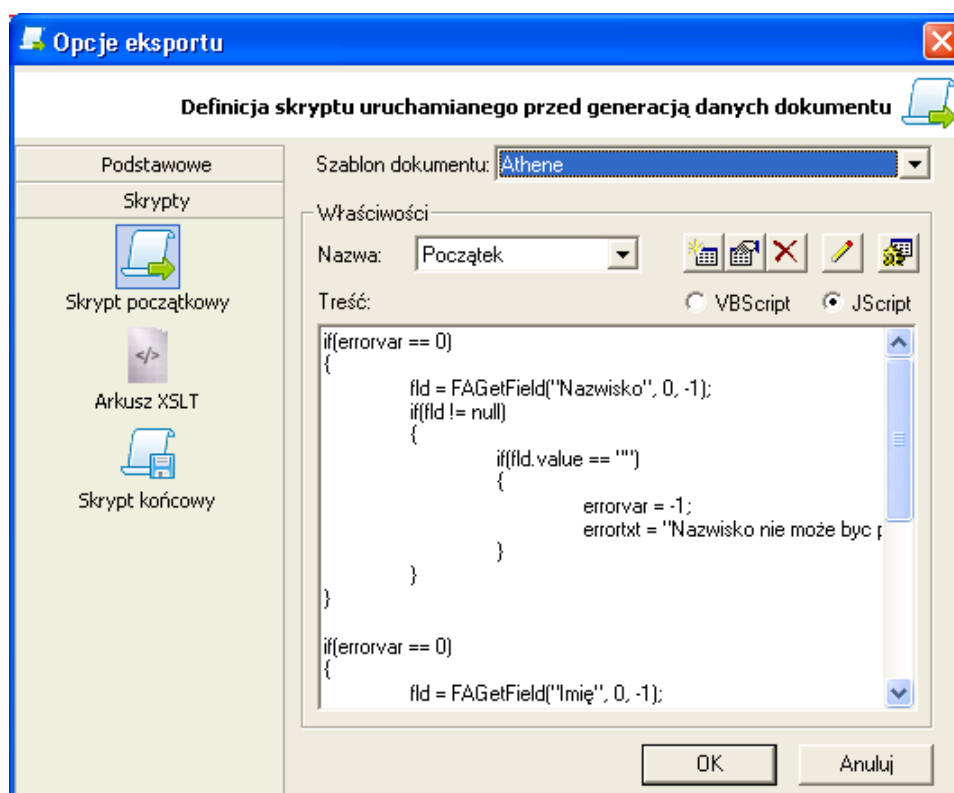
- Czarno-białe – zaznaczenie tej opcji spowoduje, że kopiowane będą tylko obrazy czarno-białe, a wszystkie dokumenty zawierające tylko strony kolorowe zostaną oznaczone jako błędne;
- Kolorowe – zaznaczenie tej opcji spowoduje, że kopiowane będą tylko obrazy kolorowe, a dokumenty zawierające tylko strony czarno-białe zostaną oznaczone jako błędne.
- Kolorowe i czarno-białe w oddzielnych plikach – opcja ta będzie dostępna tylko w przypadku zaznaczenia opcji czarno-białe i kolorowe. Zaznaczenie tej opcji spowoduje, że obrazki czarno-białe i kolorowe trafią do różnych plików. Nie zaznaczenie tej opcji spowoduje, że zarówno czarno-białe, jak i kolorowe trafią do jednego pliku.
- Strony dokumentu razem – w przypadku dokumentów składających się z wielu stron, zaznaczenie tej opcji spowoduje, że wszystkie strony trafią do jednego pliku. Nie zaznaczenie, że każda strona trafi do osobnego pliku.
- Obracaj obrazy stron – funkcja ta będzie obracać obraz strony zgodnie z ustawieniami dokonanymi we wcześniejszych etapach przetwarzania dokumentu.

Dostępne są następujące formaty pliku z obrazami stron dokumentu:

Rozszerzenie	Opis	Plik wielostronicowy	Kolor	Klucz sprzętowy
BMP	Standardowy format bitmapy Windows. Nie używa kompresji.	Nie	Tak	Wymagany
CALS	Format pliku binarnego opracowane przez Computer Aided Acquisition i wsparcie logistyczne (CALS) urząd w Stanach Zjednoczonych Departament Obrony.	Nie	Nie	Wymagany
DCX	Wielostronicowy PCX. Binarny format pliku zazwyczaj używany do faksów.	Tak	Nie	Wymagany
GIF	CompuServe Graphics Interchange Format (GIF) - format często wykorzystywane do zdjęć na stronach internetowych. (Ograniczenie do 256 kolorów).	Nie	Tak	Wymagany
JBG	Joint Bilevel Image Group (JBIG) format.	Nie	Nie	Wymagany
JPG	JPEG 2000 - nowa wersja JPEG, która zapewnia lepszą kompresję i dodatkowe funkcje.	Nie	Tak	Wymagany
JPG	JPEG (Joint Photographic Experts Group), format często używany w fotografii. Obsługuje kolorowe zdjęcia, ale zapisuje je przy użyciu kompresji stratnej.	Nie	Tak	Wymagany
MDA	IBM MO: DCA format, wielostronicowy format zgodnym z systemami obrazowania firmy IBM.	Tak	Tak	Wymagany
PCX	Format PCX używany we wczesnych wersjach systemu Windows.	Nie	Tak	Wymagany
PDA	Calera Processed Document Architecture format.	Nie	Tak	Wymagany
PDF	Adobe Portable Document Format, szeroko stosowane zarówno w przypadku obrazów i dokumentów, które zawierają tekst.	Tak	Tak	Wymagany
PDF	PDF/A jest formatem plików elektronicznych dokumentów do długotrwałego przechowywania.	Tak	Tak	Wymagany
PDF	PDF/Web format PDF, który został zoptymalizowany dla szybszego dostępu i przeglądania podczas pobierania plików z Internetu lub sieci.	Tak	Tak	Wymagany
TIFF	Format TIFF (podzbiór funkcji) zgodny z systemem obrazowania Plexus.	Nie	Nie	Wymagany
PNG	Portable Network Graphics Format, standard World Wide Web Consortium (W3C) dla obrazów rastrowych w sieci.	Nie	Tak	Wymagany
TIFF	Tagged Image File Format (TIFF), wielostronicowy format obrazów najczęściej używany do zeskanowanych obrazów w wysokiej rozdzielczości.	Tak	Tak	Nie wymagany

Zapis obrazu stron dokumentu w formacie TIFF nie wymaga podłączenia programu FormAnalyze Export do klucza sprzętowego. Pozostałe formaty wymagają dostępu do klucza sprzętowego.

14.4. Grupa opcji *Skrypt początkowy*



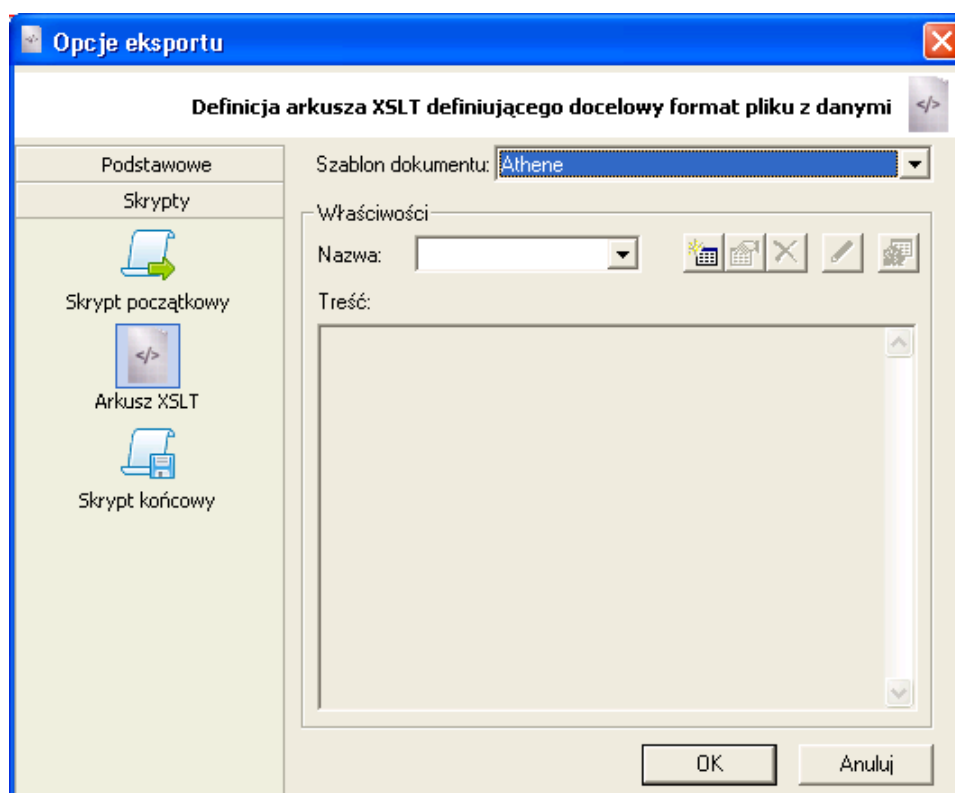
Rys. 14-9 Okno właściwości skryptu wywoływanego przed eksportem danych, grupa opcji *Skrypt początkowy*

W grupie *Skrypt początkowy* można stworzyć skrypt napisany w VBScript lub JScript, który może być wykorzystany do modyfikacji danych dokumentu lub sprawdzania poprawności treści dokumentu. Skrypt wywołany jest bezpośrednio po pobraniu dokumentu do eksportu. Jeżeli skrypt modyfikuje treść dokumentu, zmiany będą uwzględnione podczas tworzenia danych w pliku eksportu. W kodzie skryptu można wykorzystać funkcje opisane w rozdziale 12.7 Wbudowane funkcje rozszerzające możliwości skryptów. Umożliwiają one odwołanie do treści paczki, dokumentu, stron i pól. Skrypt może ustawić wartość zmiennej *errorvar*:

- Wartość 0 (wartość domyślna) oznacza, że dokument może zostać wyeksportowany
- Wartość > 0 oznacza, że dokument nie będzie eksportowany, zmienna *errortxt* zawiera komunikat o błędzie, który zostanie wyświetlony i zapisany w bazie danych. Użytkownik będzie decydował czy proces eksportu będzie kontynuowany.
- Wartość < 0 oznacza, że dokument nie będzie eksportowany, zmienna *errortxt* zawiera komunikat o błędzie, który zostanie wyświetlony i zapisany w bazie danych. Proces eksportu będzie zatrzymany.

Na tym etapie eksportu dokumentu funkcja *FAExpGetText* zwraca pusty ciąg, a funkcja *FAExpGetPgCnt* zwraca zero. Wywołanie funkcji *FAExpSetText* spowoduje, że dane eksportowe będą poprzedzone ciągiem przekazanym w parametrze tej funkcji.

14.5. Grupa opcji *Arkusz XSLT*



Rys. 14-10 Okno właściwości transformaty pliku eksportu, grupa opcji *Arkusz XSLT*.

XSLT (*ang. XSL Transformations, Extensible Stylesheet Language Transformation*) to język przekształceń dokumentów XML. Pozwala na przetłumaczenie dokumentów z jednego formatu XML na inny format zgodny ze składnią XML-a, jak również na zwykły HTML i czysty tekst.

Istnieje wiele źródeł (tak w formie drukowanej jak i elektronicznej) opisujących język XSLT. Niniejszy rozdział nie opisuje języka XSLT a jedynie sposób jego wykorzystania do przekształceń składni plików eksportu. Zainteresowanych szybkim poznaniem podstaw tego języka odsyłamy do strony: <http://www.w3schools.com/xsl/>

W poprzednich rozdziałach opisano sposób konfigurowania składni pliku eksportu. Skrypt XSLT umożliwia nam przekształcenie powstałego w ten sposób pliku eksportu do praktycznie dowolnego formatu (opartego o 8-bitowe kodowanie znaków w tym UTF-8).

Skrypt XSLT należy stosować gdy:

- Chcemy uzyskać plik eksportu o składni niemożliwej do zdefiniowania za pomocą opcji eksportu,
- Wymagamy zmiany sposobu kodowania znaków w pliku eksportu (domyślnie jest to Windows-1250 dla plików w formacie TXT i UTF-8 dla plików w formacie XML).

W celu przygotowania skryptu XSLT na potrzeby programu FA Export zaleca się wykonanie następujących działań:

1. Korzystając z polecenia *Opcje eksportu...* programu FA Designer zdefiniować zakres i format danych które mają znaleźć się w pliku eksportu.

2. Przy pomocy programu FA Export dokonać eksportu kilku przykładowych dokumentów do pliku w formacie XML.
3. Przeglądać powstały plik eksportu pod kątem kompletności potrzebnych informacji.
4. Utworzyć skrypt XSLT i przetestować jego działanie na otrzymanym wcześniej pliku eksportu.
5. Jeżeli testy skryptu XSLT zakończą się pomyślnie dodać skrypt.

14.5.1. Tworzenie arkusza XSLT

W celu dodania skryptu XSLT do pliku konfiguracyjnego należy:

1. Z menu programu wybrać polecenie *Dokument -> Opcje Eksportu*.
2. W oknie *Opcje eksportu* z listy *Szablon dokumentu* wybrać szablon z którym chcemy związać skrypt XSLT.
3. Przejść do grupy opcji *Skrypt XSLT*.
4. Kliknąć przycisk . W oknie *Nowy* wpisać nazwę skryptu i kliknąć przycisk *OK*.
5. Kliknąć przycisk  i w oknie *Edytor Skryptu* wpisać treść skryptu.
6. W oknie *Edytor Skryptu* wykonać polecenie *Plik > Zapisz* a następnie polecenie *Plik > Zamknij*.
7. Kliknąć lewym klawiszem myszy poza obszarem okna *Właściwości eksportu*. Okno zostanie zamknięte.

Treść skryptu można edytować bezpośrednio na zakładce XSLT jednak jest to niewskazane ze względu na mały rozmiar okna edytora, oraz brak funkcji wspomagających pisanie skryptów. Okno to ma jedynie charakter poglądowy.

14.5.2. Przykład arkusza XSLT

W szablonie dokumentu zdefiniowano pola tekstowe: *Nr Umowy*, *Podpis*, *Nazwisko*, *Imie*, *Data Podpisania*.

Naszym celem jest uzyskanie pliku eksportu w formacie TXT z kodowaniem UTF-8.

Składnia pliku eksportu:

- Każdemu dokumentowi odpowiada pojedyncza linia w pliku eksportu,
- Wartości poszczególnych pól oddzielone są znakiem tabulacji,
- Oprócz wartości pól formularza każdy dokument opisany jest dodatkowo przez: Nazwę pliku zawierającego wyeksportowane obrazy, Liczbę zeskanowanych stron dokumentu, Datę i godzinę skanowania dokumentu.
- Kolejność indeksów opisujących dokument: Nazwa pliku, Liczba stron, Nr Umowy, Imię, Nazwisko, Data Podpisania, Podpis, Data Skanowania.

Poniżej przedstawiono: plik eksportu w formacie XML, skrypt XSLT, oraz plik wynikowy w formacie TXT.

Plik eksportu w formacie XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<export>
  <document file="00000011.tif" created="2011-03-04 09:41:04">
    <page number="INDEX">
      <zone name="Nr Umowy">12345678</zone>
      <zone name="Podpis">Tak</zone>
      <zone name="Nazwisko">Kowalski</zone>
      <zone name="Imie">Jan</zone>
      <zone name="Data Podpisania">01-03-2011</zone>
    </page>
    <page number="1"/>
    <page number="2"/>
  </document>
  <document file="00000012.tif" created="2011-03-04 09:41:13">
    <page number="INDEX">
      <zone name="Nr Umowy">87654321</zone>
      <zone name="Podpis">Tak</zone>
      <zone name="Nazwisko">Nowak</zone>
      <zone name="Imie">Tadeusz</zone>
      <zone name="Data Podpisania">02-03-2011</zone>
    </page>
    <page number="1"/>
    <page number="2"/>
  </document>
</export>
```

Arkusz XSLT:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="utf-8"/>
  <xsl:template match="/">
    <xsl:for-each select="export/document">
      <xsl:value-of select="@file"/>
      <xsl:text>&#09;</xsl:text>
      <xsl:value-of select="count(descendant::page[not(@number='INDEX')])"/>
      <xsl:text>&#09;</xsl:text>
      <xsl:for-each select="page[@number='INDEX']">
        <xsl:value-of select="zone[@name='Nr Umowy']"/>
        <xsl:text>&#09;</xsl:text>
        <xsl:value-of select="zone[@name='Imie']"/>
        <xsl:text> </xsl:text>
        <xsl:value-of select="zone[@name='Nazwisko']"/>
        <xsl:text>&#09;</xsl:text>
        <xsl:value-of select="zone[@name='Data Podpisania']"/>
        <xsl:text>&#09;</xsl:text>
        <xsl:value-of select="zone[@name='Podpis']"/>
        <xsl:text>&#09;</xsl:text>
      </xsl:for-each>
      <xsl:value-of select="substring(@created,0,20)"/>
      <xsl:text>&#xa;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

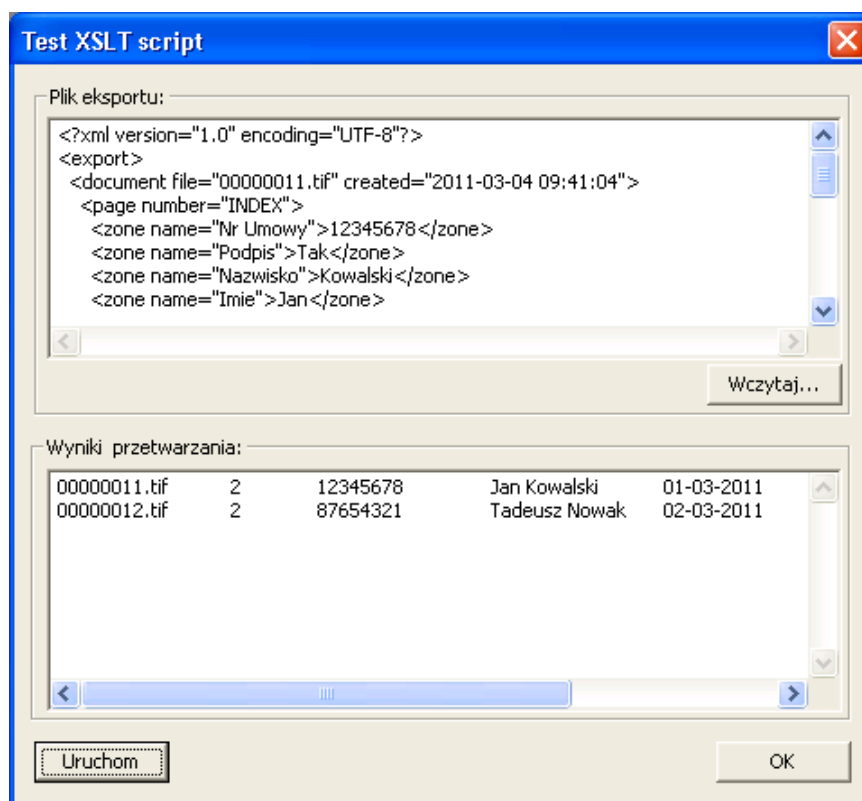
Docelowy plik eksportu w formacie TXT:

00000011.tif	2	12345678	Jan Kowalski	01-03-2011	Tak	2011-03-04 09:41:04
00000012.tif	2	87654321	Tadeusz Nowak	02-03-2011	Tak	2011-03-04 09:41:13

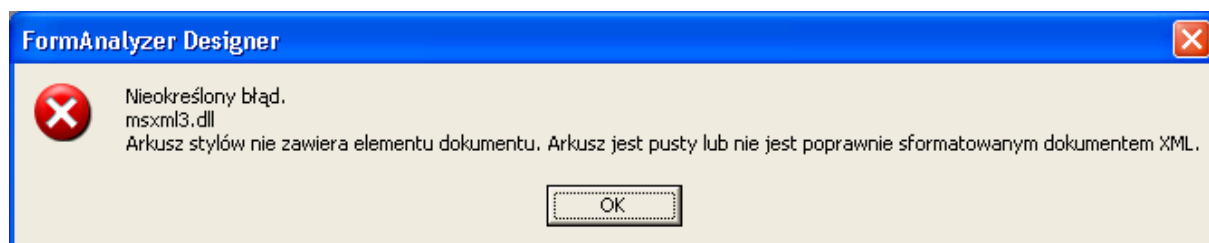
14.5.3. Sprawdzanie poprawności arkusza XSLT

W celu przetestowania działania skryptu XSLT należy:

1. W oknie *Właściwości eksportu* przejść do zakładki *XSLT*.
2. Z listy *Skrypt* wybrać nazwę odpowiedniego skryptu.
3. Kliknąć przycisk . Pojawi się okno *Test skryptu XSLT*.
4. W polu *Plik eksportu* umieścić treść pliku eksportu wykonanego w formacie XML (można posłużyć się poleceniem kopiuj-wklej, lub wczytać treść pliku klikając przycisk *Wczytaj...*).
5. Kliknąć przycisk *Uruchom*. W polu *Wynik przetwarzania* pojawi się treść przetworzonego pliku eksportu.

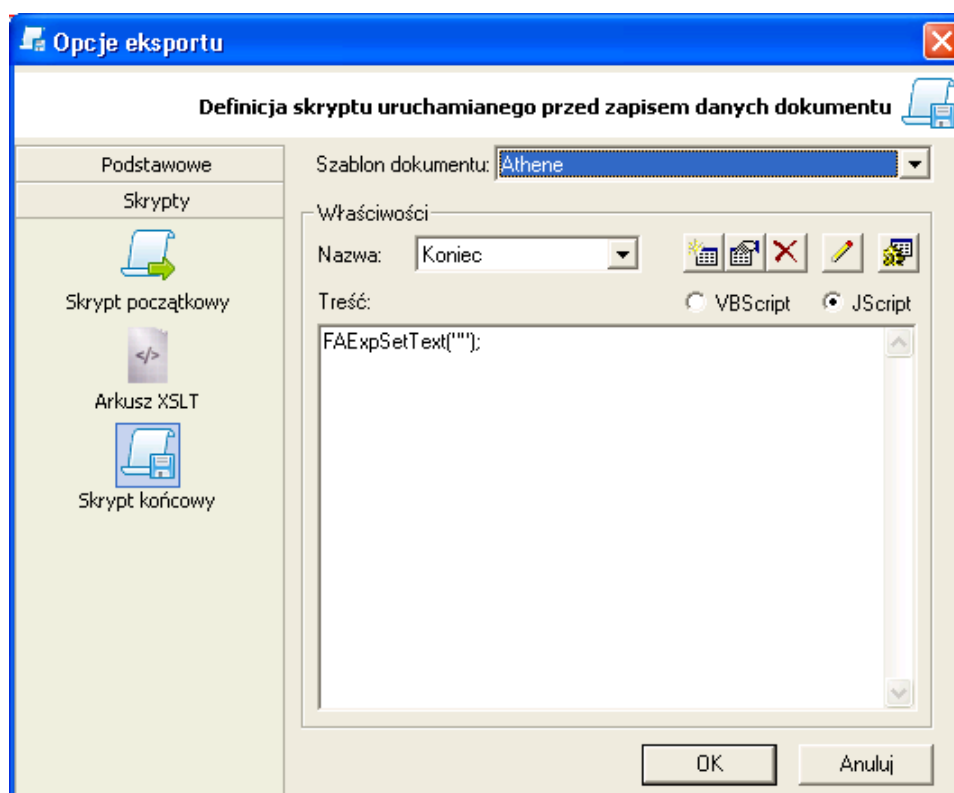
Rys. 14-11 Okno *Test arkusza XSLT*.

Jeżeli podczas przetwarzania pliku eksport wystąpił błąd zostanie on zasygnalizowany odpowiednim komunikatem np.:



Rys. 14-12 Komunikat informujący o błędzie arkusza XSLT.

14.6. Grupa opcji *Skrypt końcowy*



Rys. 14-13 Okno właściwości skryptu wywoływanego przed zapisem danych do pliku eksportu, grupa opcji *Skrypt końcowy*

W grupie *Skrypt końcowy* można stworzyć skrypt napisany w VBScript lub JScript, który może być wykorzystany do modyfikacji danych dokumentu lub sprawdzania poprawności treści dokumentu. Skrypt wywołany jest po stworzeniu danych eksportu i obrazów stron dokumentu. Treść dokumentu nie jest jeszcze zapisana do pliku eksportu. W kodzie skryptu można wykorzystać funkcje opisane w rozdziale 12.7 Wbudowane funkcje rozszerzające możliwości skryptów. Umożliwiają one odwołanie do treści paczki, dokumentu, stron i pól. Skrypt może ustawić wartość zmiennej *errorvar*:

- Wartość 0 (wartość domyślna) oznacza, że dokument może zostać wyeksportowany
- Wartość > 0 oznacza, że dokument nie będzie eksportowany, zmienna *errortxt* zawiera komunikat o błędzie, który zostanie wyświetlony i zapisany w bazie danych. Użytkownik będzie decydował czy proces eksportu będzie kontynuowany.
- Wartość < 0 oznacza, że dokument nie będzie eksportowany, zmienna *errortxt* zawiera komunikat o błędzie, który zostanie wyświetlony i zapisany w bazie danych. Proces eksportu będzie zatrzymany.

Na tym etapie eksportu dokumentu funkcja `FAExpGetText` zwraca ciąg, który powstał na podstawie treści dokumentu i opcji ustawionych w modułach FA Designer i FA Export. Funkcja `FAExpGetPgCnt` zwraca liczbę plików z obrazami stron dokumentu, które zostały stworzone w procesie eksportu. Wywołanie funkcji `FAExpSetText` spowoduje zastąpienie danych eksportu wywołanym ciągiem.

15. Filtry

15.1. Filtry w procesie skanowania i rozpoznawania.

W podręczniku do aplikacji FormAnalyze Scan & Administrator opisane zostało stosowanie filtrów do obróbki obrazu zeskanowanej strony tuż po zakończeniu procesu skanowania. Finalnie do bazy danych trafiał obraz strony zmodyfikowany przez zastosowane filtry.

Istnieje także możliwość wykorzystania filtrów w procesie rozpoznawania obrazu w aplikacji FormAnalyze Engine. W tym przypadku ustawienia związane z zastosowaniem odpowiednich filtrów definiuje się w aplikacji FormAnalyze Designer.

Jeżeli użytkownik zdefiniował ustawienia filtrów, to w procesie rozpoznawania aplikacja FormAnalyze Engine tworzy tymczasowy obraz strony dokumentu zmodyfikowany przez zastosowane filtry. Po zakończeniu procesu rozpoznawania, przy zapisywaniu wyników w bazie danych, tymczasowy obraz zostaje usunięty.

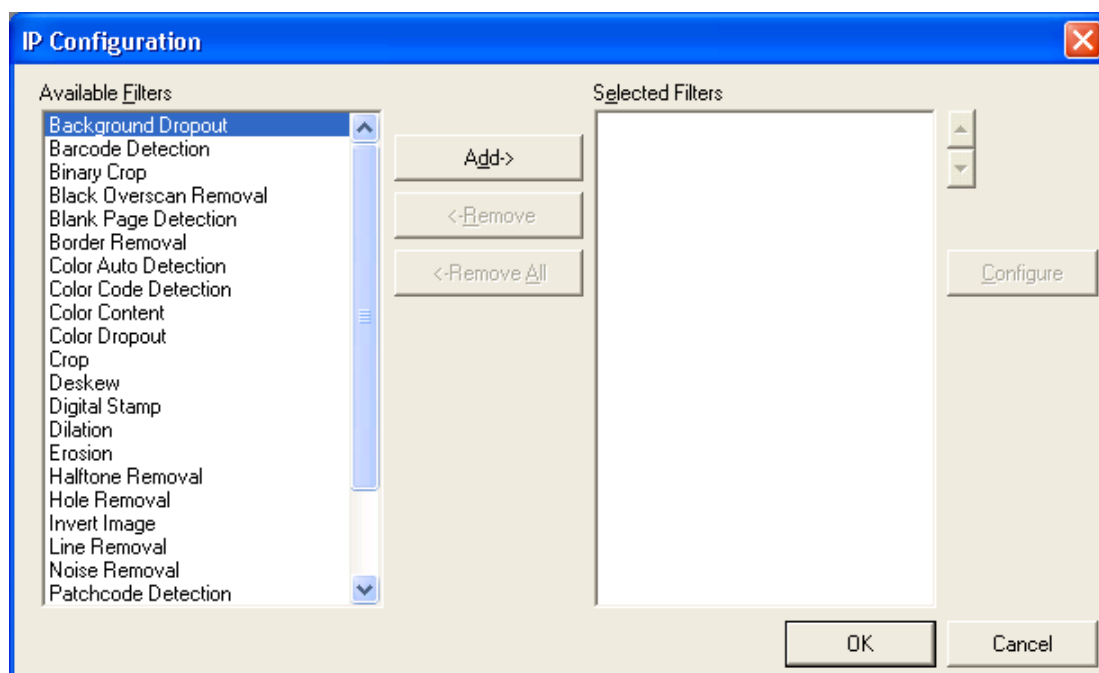
Ustawienia filtrów używanych w procesie rozpoznawania będą zastosowane dla wszystkich szablonów dokumentów zapisanych w danym pliku konfiguracyjnym.

15.2. Zastosowanie filtrów w procesie rozpoznawania dokumentów

W systemie FormAnalyze rozpoznawane są tylko i wyłącznie obrazy czarno-białe. Stosowanie filtrów w procesie rozpoznawania jest przydatne w szczególności dla użytkowników, którzy chcą posiadać zapisane w bazie kolorowe obrazy dokumentów. Wykorzystanie filtrów w procesie rozpoznawania umożliwi im rozpoznanie takiego dokumentu, pozostawiając w bazie danych oryginalny, kolorowy obraz. Obraz kolorowy dostępny będzie także w kolejnych etapach przetwarzania tego dokumentu.

Wszystkie ustawienia dotyczące stosowania filtrów w procesie rozpoznawania dokumentów definiuje się w aplikacji FormAnalyze Designer.

W celu zdefiniowania takich ustawień należy z menu *Plik* wybrać polecenie *Przetwarzanie obrazu...* Spowoduje to wyświetlenie okna dialogowego *IP Configuration*.



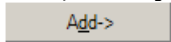
Rys. 15-1 Okno definiowania ustawień filtrów w FormAnalyzer Designer.

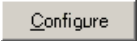
W prawej części okna w obszarze *Available Filters* znajdują się wszystkie filtry, których można użyć w procesie rozpoznawania dokumentów. Szczegółowy opis działania tych filtrów znajduje się w podręczniku FormAnalyzer Scan & Administrator.

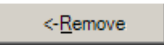
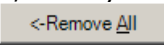
W procesie rozpoznawania należy wykorzystać filtry modyfikujące obraz, tj.: Background Dropout, Binary Crop, Black Overscan Removal, Border Removal, Color Dropout, Crop, Deskew, Dilation, Erosion, Halftone Removal, Hole Removal, Invert Image, Line Removal, Noise Removal, Rotation, Scaling, Skeleton, Smoothing, Threshold.

Większość wymienionych powyżej filtrów, które można stosować podczas procesu rozpoznawania dokumentów przekształca obraz tymczasowy z kolorowego na czarno-biały. Dwa filtry: Color Dropout i Deskew pozostawiają obraz kolorowym, dlatego też do tych filtrów zaleca się stosowanie dodatkowo filtra umożliwiającego przekształcenie obrazu na czarno-biały.

Nie zaleca się stosowania filtrów nie mających wpływu na modyfikację obrazu tj.: Barcode Detection, Black Page Detection, Color Auto Detection, Color Code Detection, Color Content, Digital Stamp, Patchcode Detection,

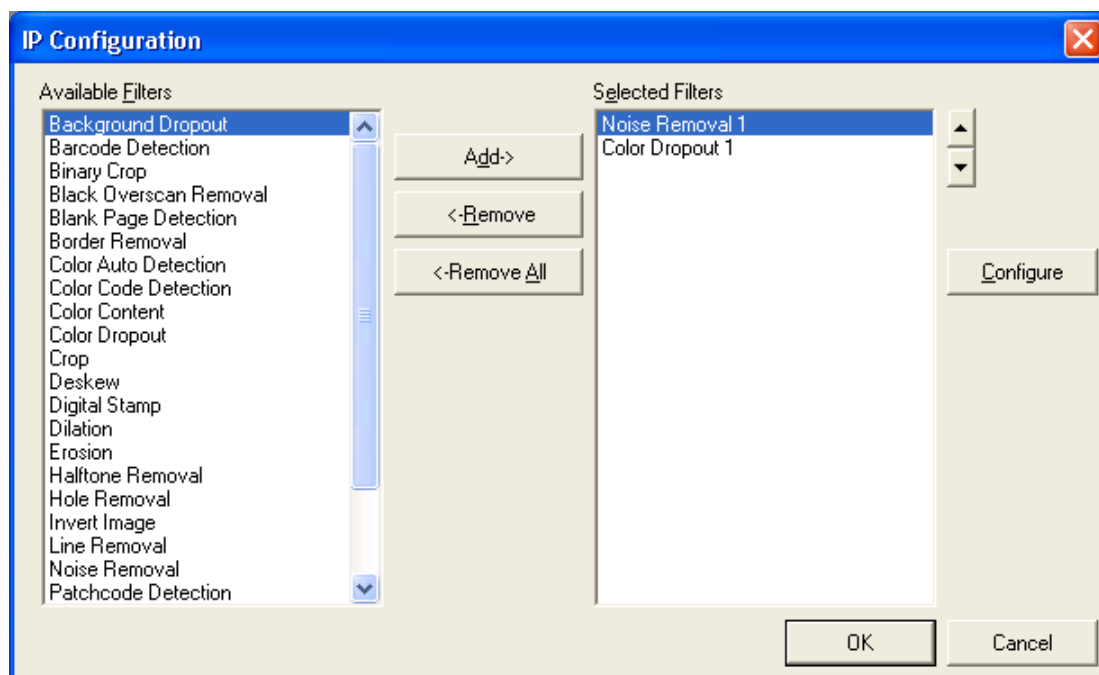
W celu zdefiniowania ustawień filtrów, należy w oknie dialogowym *IP Configuration*, na liście *Available Filters* zaznaczyć odpowiedni filtr, klikając na jego nazwie lewym przyciskiem myszy, a następnie kliknąć przycisk . Wskazane filtry pojawiają się w obszarze *Selected filters*.

Istnieje możliwość definiowania dodatkowych ustawień dla pojedynczych filtrów. Spośród filtrów zalecanych do stosowania w procesie rozpoznawania dokumentów nie można zdefiniować dodatkowych ustawień w następujących filtrach: Binary Crop, Halftone Removal, Hole Removal, Invert Image, Skeleton. W przypadku pozostałych filtrów zaznaczenie ich nazwy w obszarze *Selected Filters* spowoduje uaktywnienie przycisku . Kliknięcie tego przycisku spowoduje wyświetlenie odpowiedniego okna dialogowego.

W celu usunięcia filtra bądź kilku filtrów z obszaru *Selected Filters*, należy zaznaczyć je a następnie kliknąć przycisk . Kliknięcie przycisku  spowoduje usunięcie z obszaru wszystkich dodanych tam filtrów.

Użytkownik może zmienić kolejność wybranych filtrów w obszarze *Selected Filters*. W tym celu należy zaznaczyć odpowiedni filtr lub filtry, a następnie posługując się przyciskami  i  przesunąć odpowiednio zaznaczone elementy o jedną pozycję w górę lub w dół.

Poniżej zaprezentowano okno dialogowe, w którym używanymi filtrami w procesie rozpoznawania będą: Color Dropout oraz Noise Remover.

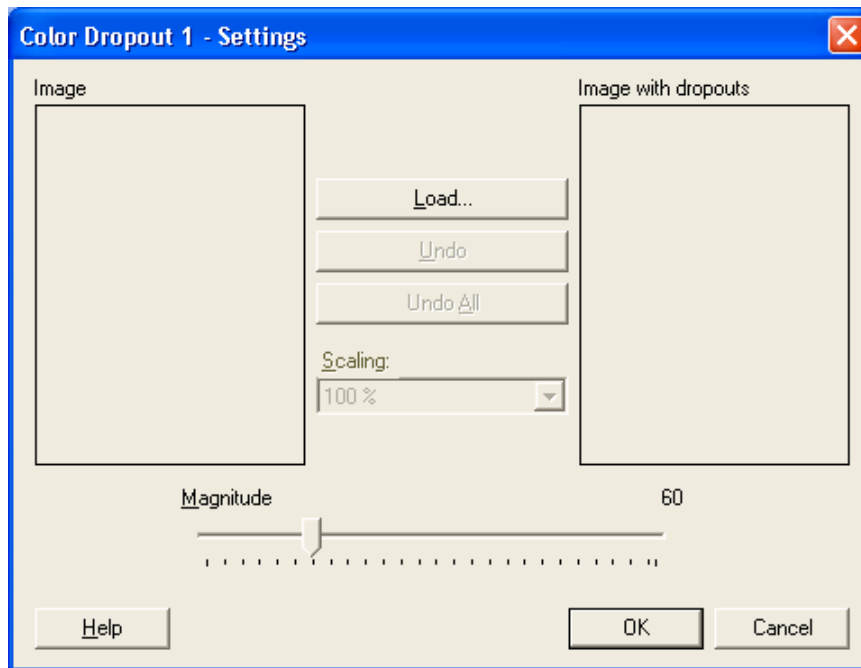


Rys. 15-2 Okno IP Configuration, w którym w obszarze *Selected Filters* wskazano filtry Color Dropout oraz Noise Removal

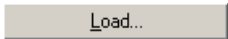
Filtr Color Dropout umożliwia zmianę wskazanego koloru na obrazie dokumentu na biały. Filtr Noise Removal służy do usuwania szumu z obrazu, poza tym jest to filtr automatycznie przetwarzający obraz z kolorowego w czarno-biały. Użycie tylko filtra Color Dropout, przy nieprawidłowej konfiguracji jego ustawień mógłby doprowadzić do niepełnego przerobienia obrazu na czarno-biały, a w konsekwencji do uniemożliwienia prawidłowego rozpoznania takiego dokumentu.

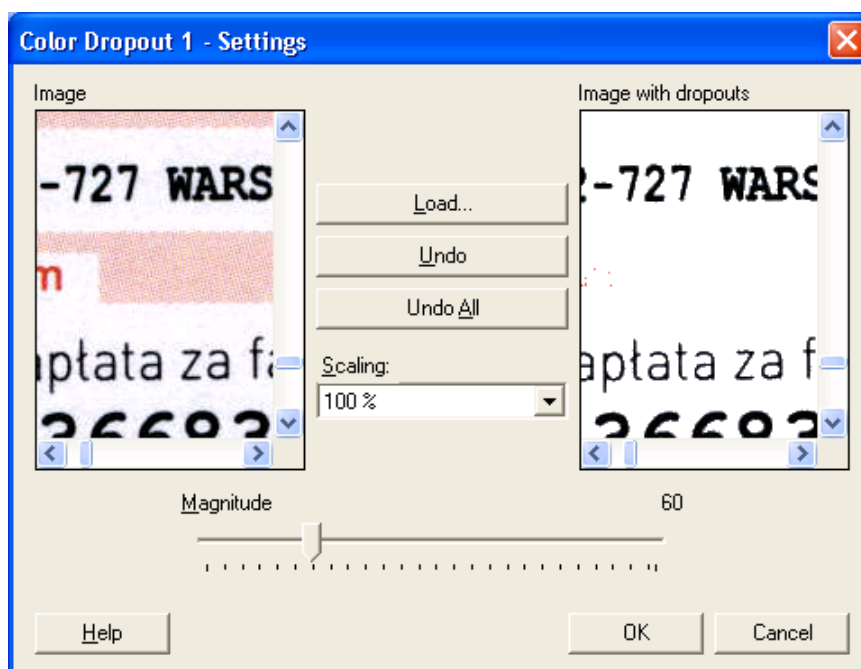
Definiowanie dodatkowych ustawień dla filtrów zostanie zaprezentowane na przykładzie filtra Color Dropout:

1. Kliknąć na nazwie filtra Color Dropout 1 w obszarze *Selected Filters*, a następnie kliknąć przycisk . Na ekranie wyświetli się okno konfigurowania ustawień dla filtra Color Dropout.



Rys. 15-3 Okno dialogowe definiowania ustawień filtra *Color Dropout 1*

4. Kliknąć przycisk , a następnie w wyświetlonym oknie dialogowym wskazać odpowiedni plik zawierający obraz dokumentu.
5. W obszarze *Image* kliknąć lewym przyciskiem na kolor, który ma zostać zamieniony na biały. W obszarze *Image with dropouts* można obserwować obraz po wprowadzonych zmianach. W obszarze *Scaling* można z rozwijalnej listy zmniejszyć lub zwiększyć widok obrazu w obszarach *Image* i *Image with dropouts*. Funkcja ta jest przydatna do precyzyjnego wskazywania odpowiedniego koloru w dokumencie. Suwak *Magnitude* umożliwia ustawienie tolerancji nasycenia wybranego koloru – im większa wartość tym tolerancja jest większa.
6. Kliknąć przycisk , w celu zapamiętania wprowadzonych ustawień.
7. Poniżej znajduje się okno dialogowe *Color Dropout 1 – Settings*, w którym na przykładowym dokumencie pokazano działanie filtra *Color Dropout*.



Rys. 15-4 Przykładowe okno Color Dropout 1- Settings.

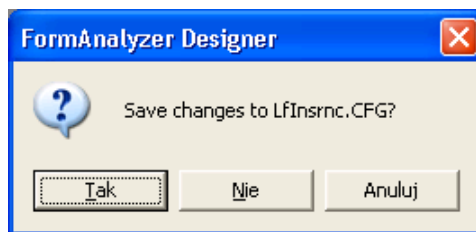
UWAGA: Zaleca się nie wskazywania zbyt wielu kolorów, gdyż może mieć to negatywny wpływ na wydajność rozpoznawania dokumentów.

16. Inne funkcje

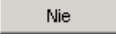
16.1. Zamknięcie programu

W celu zamknięcia programu można posłużyć się jedną z metod: kliknąć przycisk Zamknij  na pasku tytułowym okna programu, wybrać polecenie Zakończ z menu Plik, lub dwukrotnie kliknąć ikonę  znajdującą się w prawym górnym rogu pasku tytułowym okna.

Jeżeli polecenie zamknięcia programu zostanie wydane w czasie, gdy jest otwarty dokument i nie został uprzednio zapisany, wyświetlony zostanie stosowny komunikat.



Rys. 16-1 Komunikat o zamykaniu programu, jeżeli nie zostały zapisane zmiany w aktualnie wybranym dokumencie.

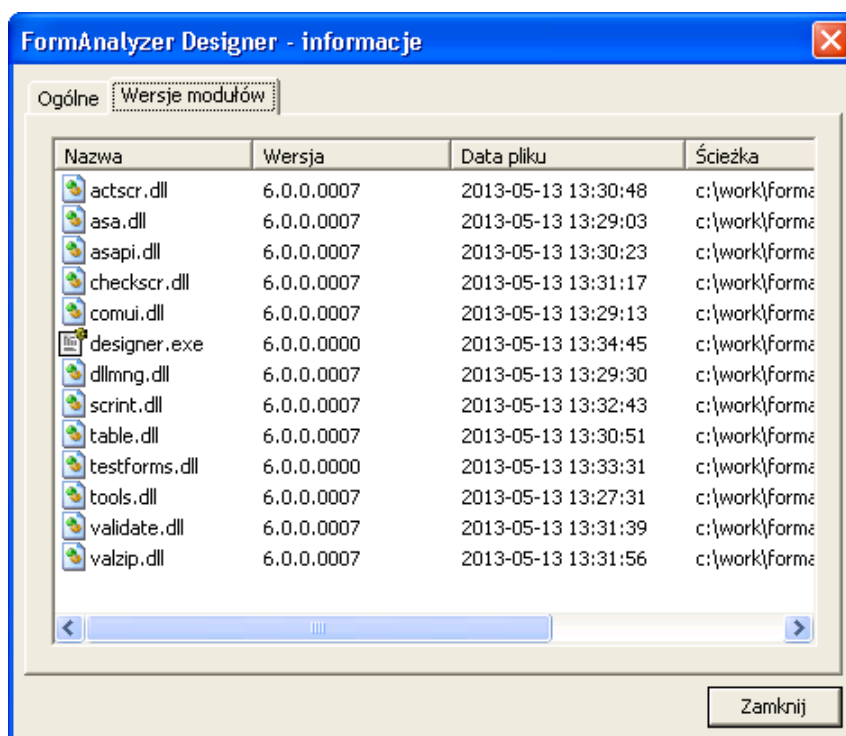
Kliknięcie przycisku  oznacza rezygnację z zamknięcia dokumentu. Kliknięcie przycisku  spowoduje zamknięcie programu bez zapisywania zmian dokonanych w dokumencie. Kliknięcie przycisku , spowoduje, że przed zamknięciem programu zmiany dokonane w dokumencie zostaną zapisane w bazie danych.

16.2. Informacje o programie

Istnieje możliwość wyświetlenia informacji o programie. W tym celu należy kliknąć przycisk  umieszczony na pasku narzędzi lub wybierać z menu *Help* polecenie *FormAnalyze Designer*. Okno informacji posiada dwie zakładki: *Ogólne* oraz *Wersje modułów*.

Pierwsza zakładka zawiera ogólne informacje o programie.

W drugiej zakładce wyszczególnione są biblioteki potrzebne do działania FormAnalyze Designer. Przy każdej bibliotece podana jest jej zainstalowana wersja, data utworzenia pliku na dysku oraz ścieżka do niego.



Rys. 16-2 Okno informacji – wersje modułów.

W razie konieczności podania wersji zainstalowanych modułów przy kłopotach z FormAnalyzer Designer ta zakładka okazuje się niezbędną.