

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Lotus Notes i Domino. Leksykon

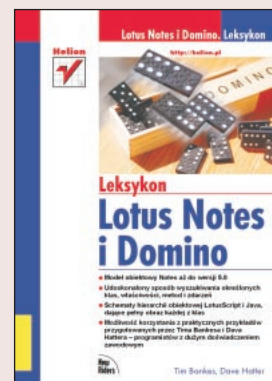
Autorzy: Tim Bankes, Dave Hatter

Tłumaczenie: Zbigniew Gała

ISBN: 83-7197-476-0

Tytuł oryginału: [Lotus Notes and Domino Essential Reference](#)

Format: B5, stron: 688



„Lotus Notes i Domino. Leksykon” jest narzędziem niezastąpionym dla osób pragnących tworzyć aplikacje w sposób szybki i zdecydowany. Zawiera bowiem wszystkie klasy frontendowe (klasy interfejsów) i backendowe (klasy baz danych) zarówno dla języka LotusScript, jak i Javy opatrzone omówieniem ich właściwości, metod oraz zdarzeń. W odróżnieniu od plików pomocy Notesa i innych dokumentacji referencyjnych, omówienie każdej z tych klas zawiera liczne przykłady zastosowań opisywanych zagadnień. Dzięki tej książce można zapomnieć o błędzeniu po meandrach pomocy Notesa w poszukiwaniu odpowiedzi na nurtujące nas pytanie. Opracowany zgodnie ze sposobem myślenia programisty, „Lotus Notes i Domino – leksykon” w znacznym stopniu upraszcza proces wyszukiwania informacji. Można korzystać ze spisu treści i indeksu, ale czyż nie łatwiej korzystać z umieszczonych bezpośrednio w tekście odnośników, tabeli referencyjnych i schematów hierarchii obiektowej? Posiadając kompendium wiedzy w postaci niniejszej książki zamiana ołowiu posiadanej wiedzy na złote aplikacje Notesa będzie dużo prostsze. Spośród innych publikacji na ten temat książka ta jest:

- najdokładniejszym omówieniem modelu obiektowego Notes aż do wersji 5.0;
- usprawnionym sposobem wyszukiwania interesujących nas klas, właściwości metod i zdarzeń;
- rozbudowanym schematem hierarchii obiektowej języków LotusScript i Java dającym pełen obraz każdej z klas;
- pełnym i jednocześnie zwartym zbiorem referencji.

Pozwala także na korzystanie z praktycznych przykładów przygotowanych przez Tima Bankesa i Dava Hattera – programistów o dużym zasobie doświadczeń zawodowych.



## Spis treści

- Button** 4.0 4.1x 4.5x 4.6 5.0 **17**
  - Składnia LotusScript 17
  - Zdarzenia 17
  - Komentarz 17
  - Przykład 18
- Field** 4.0 4.1x 4.5x 4.6x 5.0 **19**
  - Składnia LotusScript 19
  - Zdarzenia 19
  - Komentarz 20
  - Przykład 21
- Navigator** 4.0 4.1x 4.5x 4.6x 5.0 **22**
  - Składnia LotusScript 22
  - Zdarzenia 22
  - Komentarz 22
  - Przykład 22
- NotesUIDatabase** 4.5x 4.6x 5.0 **24**
  - Składnia LotusScript 24
  - Właściwości 24
  - Metody 25
  - Zdarzenia 26
  - Komentarz 29
  - Przykład 30
- NotesUIDocument** 4.0 4.1x 4.5x 4.6x 5.0 **31**
  - Składnia LotusScript 31
  - Właściwości 31
  - Metody 36
  - Zdarzenia 47
  - Komentarz 49
  - Przykład 49
- NotesUIView** 4.5x 4.6x 5.0 **51**
  - Składnia LotusScript 51
  - Właściwości 51
  - Metody 52
  - Zdarzenia 53
  - Komentarz 56
  - Przykład 56
- NotesUIWorkspace** 4.0 4.1x 4.5x 4.6x 5.0 **57**
  - Składnia LotusScript 57
  - Właściwości 57
  - Metody 58
  - Komentarz 70
  - Przykład 70
- NotesACL** 4.0 4.1x 4.5x 4.6x 5.0 **72**
  - Składnia LotusScript 72
  - Właściwości 73
  - Metody 75
  - Komentarz 78
  - Przykład 79
- NotesACLEntry** 4.0 4.1x 4.5x 4.6x 5.0 **80**
  - Składnia LotusScript 80
  - Parametry LotusScript 80
  - Właściwości 81
  - Metody 88
  - Komentarz 89
  - Przykład 89
- NotesAgent** 4.0 4.1x 4.5x 4.6x 5.0 **92**
  - Składnia LotusScript 92
  - Właściwości 92
  - Metody 95
  - Komentarz 95
  - Przykład 96
- NotesDatabase** 4.0 4.1x 4.5x 4.6x 5.0 **97**
  - Składnia LotusScript 97
  - Parametry LotusScript 97
  - Właściwości 98
  - Metody 105
  - Komentarz 120
  - Przykład 121
- NotesDateRange** 4.5x 4.6x 5.0 **122**
  - Składnia LotusScript 122
  - Właściwości 122
  - Przykład 123
- NotesDateTime** 4.0 4.1x 4.5x 4.6x 5.0 **124**
  - Składnia LotusScript 124
  - Parametry LotusScript 124
  - Właściwości 124
  - Metody 126
  - Komentarz 129
  - Przykład 130
- NotesDbDirectory** 4.0 4.1x 4.5x 4.6x 5.0 **131**
  - Składnia LotusScript 131
  - Parametry LotusScript 131
  - Właściwości 132
  - Metody 132
  - Komentarz 133
  - Przykład 133
- NotesDocument** 4.0 4.1x 4.5x 4.6x 5.0 **135**
  - Składnia LotusScript 135
  - Parametry LotusScript 135
  - Właściwości 136
  - Metody 144
  - Komentarz 154
  - Przykład 155
- NotesDocumentCollection** 4.0 4.1x 4.5x 4.6x 5.0 **158**
  - Składnia LotusScript 158
  - Właściwości 159
  - Metody 160
  - Komentarz 166
  - Przykład 167

- NotesEmbeddedObject** 4.0 4.1x 4.5x 4.6x 5.0 **169**
  - Składnia LotusScript 169
  - Właściwości 169
  - Metody 172
  - Komentarz 173
  - Przykład 173
- NotesForm** 4.5x 4.6x 5.0 **177**
  - Składnia LotusScript 177
  - Właściwości 177
  - Metody 179
  - Komentarz 179
  - Przykład 179
- NotesInternational** 4.5x 4.6x 5.0 **181**
  - Składnia LotusScript 181
  - Właściwości 181
  - Przykład 183
- NotesItem** 4.0 4.1x 4.5x 4.6x 5.0 **184**
  - Składnia LotusScript 184
  - Parametry LotusScript 184
  - Właściwości 185
  - Metody 189
  - Komentarz 191
  - Przykład 192
- NotesLog** 4.0 4.5x 4.6x 5.0 **193**
  - Składnia LotusScript 193
  - Parametry LotusScript 193
  - Właściwości 193
  - Metody 194
  - Komentarz 197
  - Przykład 197
- NotesName** 4.0 4.1x 4.5x 4.6x 5.0 **199**
  - Składnia LotusScript 199
  - Parametry LotusScript 199
  - Właściwości 199
  - Metody 202
  - Komentarz 202
  - Przykład 203
- NotesNewsletter** 4.0 4.1x 4.5x 4.6x 5.0 **204**
  - Składnia LotusScript 204
  - Parametry LotusScript 204
  - Właściwości 205
  - Metody 206
  - Komentarz 207
  - Przykład 207
- NotesOutline** 5.0 **209**
  - Składnia LotusScript 209
  - Parametry LotusScript 209
  - Właściwości 209
  - Metody 210
  - Przykład 212
- NotesOutlineEntry** 5.0 **213**
  - Składnia LotusScript 213
  - Właściwości 213
- Metody 215
- Przykład 216
- NotesRegistration** 4.6x 5.0 **217**
  - Składnia LotusScript 217
  - Właściwości 217
  - Metody 219
  - Komentarz 226
  - Przykład 227
- NotesReplication** 5.0 **228**
  - Składnia LotusScript 228
  - Właściwości 228
  - Metody 232
  - Komentarz 233
  - Przykład 233
- NotesRichTextItem** 4.0 4.1x 4.5x 4.6x 5.0 **234**
  - Składnia LotusScript 234
  - Parametry LotusScript 234
  - Właściwości 235
  - Metody 236
  - Komentarz 241
  - Przykład 242
- NotesRichTextParagraphStyle** 5.0 **244**
  - Składnia LotusScript 244
  - Właściwości 244
  - Metody 247
  - Komentarz 249
  - Przykład 249
- NotesRichTextStyle** 4.6x 5.0 **251**
  - Składnia LotusScript 251
  - Właściwości 251
  - Komentarz 254
  - Przykład 254
- NotesRichTextTab** 5.0 **255**
  - Składnia LotusScript 255
  - Parametry LotusScript 255
  - Właściwości 255
  - Metody 256
  - Komentarz 256
  - Przykład 256
- NotesSession** 4.0 4.5x 4.6x 5.0 **258**
  - Składnia LotusScript 258
  - Właściwości 258
  - Metody 263
  - Przykład 269
- NotesTimer** 4.5x 4.6x 5.0 **270**
  - Składnia LotusScript 270
  - Parametry LotusScript 270
  - Właściwości 270
  - Metody 271
  - Zdarzenia 271
  - Komentarz 271
  - Przykład 272

- NotesView** 4.0 4.1x 4.5x 4.6x 5.0 273  
Składnia LotusScript 273  
Właściwości 274  
Metody 279  
Komentarz 288  
Przykład 289
- NotesViewColumn** 4.6x 5.0 291  
Składnia LotusScript 291  
Właściwości 291  
Komentarz 300  
Przykład 300
- NotesViewEntry** 5.0 302  
Składnia LotusScript 302  
Właściwości 303  
Metody 305  
Przykład 305
- NotesViewEntryCollection** 5.0 307  
Składnia LotusScript 307  
Właściwości 307  
Metody 308  
Komentarz 312  
Przykład 312
- NotesViewNavigator** 5.0 313  
Składnia LotusScript 313  
Właściwości 313  
Metody 314  
Komentarz 318  
Przykład 318
- ACL** 4.6x 5.0 323  
Składnia Javy 323  
Metody 323  
Komentarz 331  
Przykład 331
- ACLEntry** 4.6x 5.0 333  
Składnia Javy 333  
Metody 333  
Komentarz 347  
Przykład 347
- Agent** 4.6x 5.0 350  
Składnia Javy 350  
Metody 350  
Komentarz 355  
Przykład 355
- AgentContext** 4.6x 5.0 357  
Składnia Javy 357  
Metody 357  
Przykład 362
- Database** 4.0 4.1x 4.5x 4.6x 5.0 364  
Składnia Javy 364  
Metody 365  
Komentarz 385  
Przykład 386
- DateRange** 4.5x 4.6x 5.0 387  
Składnia Javy 387  
Parametry Javy 387  
Metody 388  
Przykład 389
- DateTime** 4.6x 5.0 390  
Składnia Javy 390  
Parametry Javy 390  
Metody 391  
Komentarz 396  
Przykład 397
- DbDirectory** 4.6x 5.0 399  
Składnia Javy 399  
Parametry Javy 399  
Metody 400  
Komentarz 403  
Przykład 403
- Document** 4.6x 5.0 405  
Składnia Javy 406  
Metody 406  
Komentarz 424  
Przykład 426
- DocumentCollection** 4.6x 5.0 427  
Składnia Javy 427  
Metody 428  
Komentarz 436  
Przykład 437
- EmbeddedObject** 4.6x 5.0 439  
Składnia Javy 439  
Metody 440  
Komentarz 442  
Przykład 443
- Form** 4.6x 5.0 444  
Składnia Javy 444  
Metody 444  
Przykład 447
- International** 4.6x 5.0 449  
Składnia Javy 449  
Metody 449  
Komentarz 452  
Przykład 452
- Item** 4.6x 5.0 453  
Składnia Javy 453  
Parametry Javy 453  
Metody 454  
Komentarz 463  
Przykład 464
- Log** 4.6x 5.0 466  
Składnia Javy 466  
Parametry Javy 466  
Metody 466  
Komentarz 470  
Przykład 471

**Name** 4.6x 5.0 **473**  
Składnia Javy 473  
Parametry Javy 473  
Metody 473  
Komentarz 479  
Przykład 480

**Newsletter** 4.6x 5.0 **482**  
Składnia Javy 482  
Parametry Javy 482  
Metody 483  
Komentarz 486  
Przykład 486

**NotesFactory** 5.0 **488**  
Składnia Javy 488  
Metody 488  
Przykład 489

**Outline** 5.0 **491**  
Składnia Javy 491  
Parametry Javy 491  
Metody 491  
Przykład 494

**OutlineEntry** 5.0 **496**  
Składnia Javy 496  
Metody 496  
Przykład 500

**Registration** 4.6x 5.0 **503**  
Składnia Javy 503  
Metody 503  
Komentarz 514  
Przykład 515

**Replication** 5.0 **516**  
Składnia Javy 516  
Metody 516  
Komentarz 523  
Przykład 524

**RichTextItem** 4.6x 5.0 **526**  
Składnia Javy 526  
Metody 526  
Komentarz 534  
Przykład 535

**RichTextParagraphStyle** 5.0 **537**  
Składnia Java 537  
Parametry Javy 537  
Metody 537  
Komentarz 544  
Przykład 544

**RichTextStyle** 4.6x 5.0 **546**  
Składnia Javy 546  
Parametry Javy 546  
Metody 546  
Komentarz 552  
Przykład 552

**RichTextTab** 5.0 **554**  
Składnia Javy 554  
Metody 554  
Komentarz 555  
Przykład 556

**Session** 4.6x 5.0 **558**  
Składnia Javy 558  
Metody 559  
Przykład 566

**View** 4.6x 5.0 **567**  
Składnia Javy 567  
Metody 568  
Komentarz 585  
Przykład 585

**ViewColumn** 4.6x 5.0 **588**  
Składnia Javy 588  
Metody 588  
Komentarz 600  
Przykład 600

**ViewEntry** 5.0 **601**  
Składnia Javy 601  
Metody 602  
Przykład 605

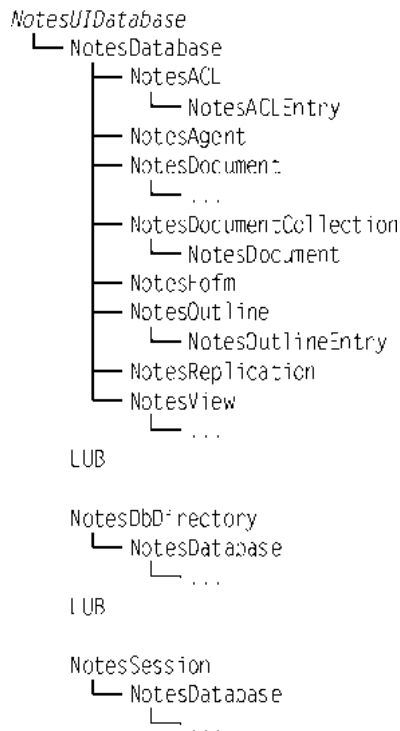
**ViewEntryCollection** 5.0 **606**  
Składnia Javy 606  
Metody 606  
Komentarz 612  
Przykład 612

**ViewNavigator** 5.0 **614**  
Składnia Javy 614  
Metody 614  
Komentarz 622  
Przykład 622

# NotesDatabase

4.0 4.1x 4.5x 4.6x 5.0

Klasa *NotesDatabase* jest jedną z najważniejszych i najbardziej podstawowych klas Notes, ponieważ to właśnie *NotesDatabase* (Notes Storage Facility) jest głównym ośrodkiem przechowywania danych w Notes/Domino. Opanowanie jej licznych metod pozwala na uzyskanie biegłości w projektowaniu aplikacji Notes/Domino.



## Składnia LotusScript

```
Dim NotesDatabase As New NotesDatabase(strServer, strFileName)
```

lub

```
Set NotesDatabase = New NotesDatabase(strServer, strFileName)
```

Istnieje wiele sposobów uzyskiwania dostępu do obiektów *NotesDatabase*. Oto niektóre z nich.

Zupełnie nową bazę danych utworzyć możemy przy użyciu metody *Create* z klasy *NotesDbDirectory*.

Nową bazę danych, w oparciu o już istniejącą, można utworzyć, korzystając z metod `CreateCopy`, `CreateFromTemplate` lub `CreateReplica` z klasy `NotesDatabase`.

Jeśli znamy nazwę serwera bazy danych i nazwę pliku, to dostęp do bazy możemy uzyskać przy użyciu metod: `New` oraz `GetDatabase` klasy `NotesSession` (► 258).

Do dostępu do bazy danych, w której aktualnie działa skrypt, bez wskazywania serwera lub nazwy pliku, służy właściwość `CurrentDatabase` z klasy `NotesSession` (► 259).

Znając serwer i `ReplicaID` można istniejącą bazę danych otworzyć za pomocą metody `OpenDatabaseByReplicaID`.

Znając nazwę serwera, na którym rezyduje baza danych, ale nie znając nazwy jej pliku, można użyć klasy `NotesDbDirectory` (► 131).

Aby określić, które bazy są książkami adresowymi, a następnie otworzyć je, używamy właściwości `AddressBooks` z `NotesSession` (► 258).

Jeśli chcemy otworzyć bazę pocztową bieżącego użytkownika, stosujemy metodę `OpenMail`.

Mając dowolny z obiektów: `NotesView`, `NotesDocument`, `NotesDocumentCollection`, `NotesACL`, `NotesViewEntry` lub `NotesViewEntryCollection` warto spróbować skorzystać z właściwości `Parent` lub `ParentDatabase`.

## Parametry LotusScript

---

<code>strServer</code>	(wmg)	\$
Wartość ciągu zawierająca nazwę serwera, na którym rezyduje baza danych. Aby wskazać bazę na lokalnej maszynie, należy podać pusty ciąg znaków (" ").		
<code>strFileName</code>	(wmg)	\$
Wartość ciągu zawierająca ścieżkę dostępu oraz nazwę pliku bazy Notes w katalogu danych Notes. Jeśli baza nie znajduje się w katalogu danych Notes to, precyzując parametr, należy podać pełną ścieżkę dostępu do pliku.		

## Właściwości

---

<code>ACL</code>	O	O.Ref.
To Get: <code>setnACLCurACL = ndbNotesDatabase.ACL</code> Zwraca obiekt <code>NotesACL</code> (► 72) zawierający listę kontroli dostępu do bazy danych.		
<code>Agents</code>	O	?
To get: <code>varAgents = ndbNotesDatabase.Agents</code> Zwraca tablicę obiektów <code>NotesAgent</code> (► 92). Jeśli dostęp do właściwości następuje ze stacji roboczej, to tablica zawierać będzie publiczne i personalne agenty, należące do bieżącego użytkownika. Jeśli właściwość jest wywołana z serwera, to tablica będzie zawierała tylko agenty publiczne.		

---

<b>AllDocuments</b>	<b>O</b>	<b>O.Ref.</b>
---------------------	----------	---------------

To get: `setndcAllDocuments = ndbNotesDatabase.AllDocuments`

Właściwość zwraca nieposortowany zbiór *NotesDocumentCollection* (► 158) zawierający wszystkie dokumenty z bazy danych. Ze względu na to, że metoda może zwrócić ogromny zbiór właściwości, należy jej używać tylko wtedy, gdy rzeczywiście chcemy uzyskać dostęp do każdego z dokumentów bazy. Chcąc uzyskać dostęp do podzbioru dokumentów w bazie danych, można zastosować metody `getAllDocumentsByKey` z klasy *NotesView* (► 283), `Search` (► 117) lub `FTSearch` (► 108) z klasy *NotesDatabase* albo `getAllEntriesByKey` z klasy *NotesViewEntry*. Poniższy przykład ilustruje zastosowanie właściwości.

```
Dim S As New NotesSession
Dim ndcCollection As NotesDocumentCollection
Set ndcCollection=S.CurrentDatabase.AllDocuments
Call ndcCollection.StampAll("Utworzone przez" , "Wieńczysław Nieszczygólny") Call
ndcCollection.UpdateAll
```

---

<b>Categories</b>	<b>O/Z</b>	<b>?</b>
-------------------	------------	----------

To get: `varCategories = ndbNotesDatabase.Categories`  
 To set: `ndbNotesDatabase.Categories = varCategories`

Zawiera wykaz kategorii, w których baza danych występuje w Notes Database Library. Jeśli zawiera wiele kategorii, to są one rozdzielane przecinkami lub średnikami.

---

<b>Created</b>	<b>O</b>	<b>?</b>
----------------	----------	----------

To get: `varCreatedDate = ndbNotesDatabase.Created`

Właściwość zwraca wariant typu `Date` (7) wskazujący godzinę i datę utworzenia bazy danych.

---

<b>CurrentAccessLevel</b>	<b>O</b>	<b>%</b>
---------------------------	----------	----------

To get: `intCurAccess = ndbNotesDatabase.CurrentAccessLevel`

Zwraca poziom dostępu dla użytkownika bieżącego w postaci liczby o zakresie od 0 (brak dostępu) do 6 (dostęp na poziomie menadżera). Warto pamiętać, że właściwość nie zwraca poziomu dostępu dla użytkownika bieżącego stacji roboczej, lecz poziom dostępu dla osoby, która ostatnia zapisywała agenta na serwerze. Każda z liczb może być przedstawiona jako jedna z następujących stałych.

---

Wartość stałej	Poziom dostępu
ACLLEVEL_NOACCESS	Uprawnienia dostępu na poziomie „No Access” (brak dostępu).
ACLLEVEL_DEPOSITOR	Uprawnienia dostępu na poziomie „Depositor” (deponent).
ACLLEVEL_READER	Uprawnienia dostępu na poziomie „Reader” (czytelnik).
ACLLEVEL_AUTHOR	Uprawnienia dostępu na poziomie „Author” (autor).
ACLLEVEL_EDITOR	Uprawnienia dostępu na poziomie „Editor” (redaktor).
ACLLEVEL_DESIGNER	Uprawnienia dostępu na poziomie „Designer” (projektant).
ACLLEVEL_MANAGER	Uprawnienia dostępu na poziomie „Manager” (menadżer).

---

<b>DelayUpdates</b>	<b>O/Z</b>	<b>T/F</b>
---------------------	------------	------------

To get: `intDelayUpdates = ndbNotesDatabase.DelayUpates`  
 To set: `ndbNotesDatabase.DelayUpdates = intDelayUpdates`



Właściwość ta wskazuje, czy operacje zapisu w bazie danych powinny być dokonywane niezwłocznie, czy też przetwarzane wsadowo (co przyspiesza działanie). Ustawienie wartości FALSE (wartość standardowa) powoduje, że aktualizacje bazy danych są dokonywane natychmiast, powodując zatrzymanie skryptu do czasu zakończenia operacji zapisu. W przeciwnym razie (gdy wartość parametru wynosi TRUE), aktualizacje zatrzymywane są w pamięci i wykonywane później, a skrypt kontynuuje swoje działanie. Warto pamiętać, że mimo iż metoda ta przyspiesza działanie, to podczas awarii serwera niezapisane dane zostaną utracone. Właściwość wpływa na operacje zapisu i usuwania dokumentów.

---

DesignTemplateName	O	\$
To get: strTemplateName = ndbNotesDatabase.DesignTemplateName		
Jeśli baza danych dziedziczy swoją strukturę od szablonu, to właściwość zwraca nazwę szablonu. W przeciwnym razie zwraca pusty ciąg znaków (" "). Jeśli baza danych dziedziczy tylko pewne elementy projektu (takie jak widok lub nawigatory), a nie cały projekt, to metoda zwraca pusty ciąg znaków.		
<pre>Dim S as New NotesSession MsgBox "Bieżącym szablonem projektu jest: " &amp; S.CurrentDatabase.DesignTemplateName.64, "informacja..."</pre>		

---

FileName	O	\$
To get: strFileName = ndbNotesDatabase.FileName		
Zwraca rzeczywistą nazwę i rozszerzenie pliku bazy, bez ścieżki dostępu.		
Poniższy przykład demonstruje budowę ogólnej procedury, która na podstawie właściwości serwera klasy <i>NotesDatabase</i> , takich m.in. jak <i>Server</i> , <i>FilePath</i> i <i>FileName</i> , zwraca miejsce lokalizacji bieżącej bazy danych. Stanowić może użyteczny dodatek do biblioteki skryptu, zwłaszcza gdy chcemy tworzyć aplikacje wielobazodanowe i wszystkie bazy danych przechowywać w tym samym katalogu (co jest dobrą praktyką).		
<pre>Public Function ResolveCurrentLocation(ndbCur As NotesDatabase) As Variant 'Jest to procedura ogólna, która zwraca bieżący 'serwer oraz ścieżkę dostępu pliku Dim aLocation(1) As String Dim strFilePath As String strFilePath=Left\$(ndbCur.FilePath, (Len(ndbCur.FilePath)-Len(ndbCur.FileName))) aLocation(0)=ndbCur.Server aLocation(1)=strFilePath ResolveCurrentLocation=aLocation End Function</pre>		

---

FilePath	O	\$
To get: strFilePath = ndbNotesDatabase.FilePath		
Zwraca rzeczywistą nazwę i rozszerzenie pliku bazy oraz jego ścieżkę dostępu. Bazy danych na stacjach roboczych zwracają pełną ścieżkę dostępu (na przykład, <i>C:\Notes\data\definiti\cmgr.nsf</i> ), a bazy danych na serwerze — względną ścieżkę dostępu do katalogu danych Notes (na przykład <i>definiti\cmgr.nsf</i> ).		

---

5.0 FolderRefsEnabled	O/Z	T/F
To get: intFldrRefs = ndbNotesDatabase.FolderRefsEnabled		
To set: ndbNotesDatabase.FolderRefsEnabled = intFldrRefs		

Właściwość można stosować do określania, czy referencje do folderów są dla określonej bazy danych dostępne. Jeśli właściwość ta ma wartość TRUE, to określa folder, w którym dokument się znajduje. Przed rozpoczęciem sprawdzania odniesień folderu upewnij się, że właściwość jest odpowiednio ustawiona. Każda baza używająca odniesień folderu musi mieć uaktywnioną właściwość `FolderRef`.

<b>Forms</b>	<b>O/Z</b>	<b>()</b>
To get: <code>varAllForms = ndbNotesDatabase.Forms</code>		
Metoda zwraca tablicę wszystkich obiektów <i>NotesForm</i> (► 177) z bazy danych.		
<b>5.0 IsDirectoryCatalog</b>	<b>O</b>	<b>T/F</b>
To get: <code>intIsDirCatalog = ndbNotesDatabase.IsDirectoryCatalog</code>		
Zastosowanie właściwości determinuje, czy baza danych jest bazą typu Directory Catalog, znaną również jako Lightweight NAB. Właściwość zwraca wartość FALSE dla wszystkich baz, do których dostęp nie nastąpił przez właściwość <code>AddressBooks</code> klasy <i>NotesSession</i> (► 258) i które są jawnie otwarte (właściwość <code>IsOpen</code> zwraca TRUE; zobacz opis <code>IsOpen</code> ).		
<b>IsFTIndexed</b>	<b>O</b>	<b>T/F</b>
To get: <code>intIsIndexed = ndbNotesDatabase.IsFTIndexed</code>		
Właściwość ta wskazuje, czy baza danych posiada indeks pełnotekstowy. Zwraca wartość TRUE, gdy indeks istnieje i FALSE, w przeciwnym razie.		
<b>IsMultiDbSearch</b>	<b>O/Z</b>	<b>T/F</b>
To get: <code>intMultiDbSearch = ndbNotesDatabase.IsMultiDbSearch</code>		
Właściwość zwraca wartość TRUE, gdy baza jest częścią indeksu wyszukiwania wielobazowego i FALSE, w przeciwnym razie.		
<b>IsOpen</b>	<b>O</b>	<b>T/F</b>
To get: <code>intIsOpen = ndbNotesDatabase.IsOpen</code>		
Metoda wskazuje, czy baza danych została otwarta. Zwraca wartość TRUE, gdy jest ona otwarta i FALSE, w przeciwnym razie.		
<b>IsPrivateAddressBook</b>	<b>O</b>	<b>T/F</b>
To get: <code>intPrivNAB = ndbNotesDatabase.IsPrivateAddressBook</code>		
Właściwość <code>IsPrivateAddressBook</code> zwraca wartość TRUE dla dowolnej bazy danych, będącej Osobistą książką adresową, do której dostęp nastąpił przy użyciu właściwości <code>AddressBooks</code> z klasy <i>NotesSession</i> (► 258). Dla wszelkich baz, z Osobistą książką adresową łącznie, do których dostęp nastąpił inaczej niż za pośrednictwem właściwości <code>AddressBooks</code> z klasy <i>NotesSession</i> , właściwość zwraca wartość FALSE.		
<b>IsPublicAddressBook</b>	<b>O</b>	<b>T/F</b>
To get: <code>intPubNAB = ndbNotesDatabase.IsPublicAddressBook</code>		
Właściwość <code>IsPublicAddressBook</code> zwraca wartość TRUE dla dowolnej bazy danych, będącej Osobistą książką adresową, do której dostęp nastąpił przy użyciu właściwości <code>AddressBooks</code>		

z klasy *NotesSession* (► 258). Dla wszelkich baz, z Osobistą książką adresową włącznie, do których dostęp nastąpił inaczej niż za pośrednictwem właściwości *AddressBooks* z klasy *NotesSession*, właściwość ta zwraca wartość `FALSE`.

---

<b>LastFTIndexed</b>	<b>O</b>	<b>?</b>
----------------------	----------	----------

To get: `varLastIndexDate = ndbNotesDatabase.LastFTIndexed`

Zwraca wariant typu `DATE` zawierający datę i godzinę ostatniego indeksowania pełnotekstowego bazy danych. Jeśli baza danych nie posiada indeksu, właściwość zwraca dość popularną datę 12/30/1899.

---

<b>LastModified</b>	<b>O</b>	<b>?</b>
---------------------	----------	----------

To get: `varModDate = ndbNotesDatabase.LastModified`

Zwraca wariant typu `DATE` zawierający datę i godzinę ostatniej modyfikacji bazy danych.

---

<b>Managers</b>	<b>O</b>	<b>()</b>
-----------------	----------	-----------

To get: `varManagers = NotesDatabase.Managers`

Właściwość zwraca tablicę zawierającą nazwy osób, serwerów oraz grup, którym przyznano dostęp do bazy danych na poziomie „Manager”. Poniższy przykład demonstruje zastosowanie tej właściwości do przesyłania menadżerom baz danych żądań podwyższenia poziomu dostępu do bazy. Może być uruchamiana w postaci agenta wywoływanego przez kliknięcie przycisku.

```
Dim S As New NotesSession
Dim ndocMail As NotesDocument
Set ndocMail=S.CurrentDatabase.CreateDocument
Dim nrtBody As New NotesRichTextItem(ndocMail, "Treść")
ndocMail.Subject="Proszę o rozpatrzenie mojego poziomu dostępu..."
Call nrtBody.AppendText("Sądzę, że mam niewystarczający dostęp do następującej bazy danych:")
Call nrtBody.AddNewLine(2)
Call nrtBody.AppendText("tytuł: " & S.CurrentDatabase.Title)
Call nrtBody.AddNewLine(1)
Call nrtBody.AppendText("Informacja o pliku: " & S.CurrentDatabase.Filepath & " on " &
S.CurrentDatabase.Server)
Call nrtBody.AddNewLine(1)
Call nrtBody.AppendText("RepID: " & S.CurrentDatabase.ReplicaID)
Call nrtBody.AddNewLine(1)
Call nrtBody.AppendText("Proszę o rozpatrzenie mojego poziomu dostępu to tej bazy danych.")
Call ndocMail.Send(False, S.CurrentDatabase.Managers)
```

---

<b>5.0 MaxSize</b>	<b>O</b>	<b>Et</b>
--------------------	----------	-----------

To get: `lngMaxSize = ndbNotesDatabase.MaxSize`

Właściwość zwraca maksymalny dopuszczalny rozmiar bazy danych określony podczas jej utworzenia.

---

<b>MaxSizeV5</b>	<b>O</b>	<b>#</b>
------------------	----------	----------

To get: `db1MaxR5Size = ndbnotesDatabase.MaxSizeV5`

Zwraca minimalny dopuszczalny rozmiar dla bazy danych Notes wersji 5.0.

<b>Parent</b>	O	O.Ref.
<p>To get: <code>setnsSession = ndbNotesDatabase.Parent</code></p> <p>Właściwość zwraca obiekt <i>NotesSession</i> (► 258) przedstawiający aktualną sesję <i>NotesSession</i>.</p>		
<b>PercentUsed</b>	O	#
<p>To get: <code>dblUsed = ndbNotesDatabase.PercentUsed</code></p> <p>Właściwość zwraca rozmiar przestrzeni przeznaczony w bazie do przechowywania danych. Proszę zapoznać się z poniższym przykładem.</p> <pre>Dim S As New NotesSession If S.CurrentDatabase.PercentUsed &lt;90 Then     Call S.CurrentDatabase.Compact End If</pre>		
<b>ReplicaID</b>	O	\$
<p>ReplicaID</p> <p>To get: <code>strRepID = ndbNotesDatabase.ReplicaID</code></p> <p>Właściwość zwraca ID repliki bazy danych (jest to 16-cyfrowy ciąg znaków alfanumerycznych).</p>		
<b>ReplicationInfo</b>	O	O.Ref.
<p>To get: <code>setnrRepInfo = ndbNotesDatabase.ReplicationInfo</code></p> <p>Każda baza <i>Notes</i> zawiera jeden obiekt <i>NotesReplication</i> (► 271), do którego za pomocą tej właściwości można uzyskać dostęp.</p>		
<b>Server</b>	O	\$
<p>To get: <code>strServer = ndbNotesDatabase.Server</code></p> <p>Właściwość zwraca nazwę serwera, na którym rezyduje baza danych. Jeśli baza znajduje się na stacji roboczej, to metoda zwraca pusty ciąg znaków (" ").</p>		
<b>Size</b>	O	#
<p>To get: <code>dblSize = ndbNotesDatabase.Size</code></p> <p>Właściwość zwraca rozmiar bazy danych w bajtach w formacie liczby typu double.</p>		
<b>SizeQuota</b>	O/Z	?
<p>To get: <code>lngQuota = ndbNotesDatabase.SizeQuota</code>  To set: <code>ndbNotesDatabase.SizeQuota = lngQuota</code></p> <p>Właściwość zwraca rozmiar przestrzeni dyskowej przydzielonej bazie danych. Nie wszystkie bazy ograniczone są przydzieloną im przestrzenią. Dla tych baz metoda zwraca wartość 0. Jeśli skrypt (a dokładnie jego właściciel) ma dostęp administracyjny do serwera, to może zmieniać tę właściwość, ale w większości wypadków stanowi ona parametr tylko do odczytu.</p>		
<b>TemplateName</b>	O	\$
<p>To get: <code>strTplName = ndbNotesDatabase.TemplateName</code></p> <p>Właściwość zwraca pusty ciąg (" "), jeśli baza nie jest szablonem. W przeciwnym razie zwraca wartość wprowadzoną dla nazwy szablonu.</p>		

Title	O/Z	\$
To get: <code>strTitle = ndbNotesDatabase.Title</code> To set: <code>ndbNotesDatabase.Title = strTitle</code>		
Właściwość zawiera tytuł bazy danych (wyświetlany na ikonie bazy). Nie można zmienić tytułu bazy danych, przez którą nastąpił dostęp do tej właściwości.		

UnprocessedDocuments	O	O.Ref.
To get: <code>setndcUnprocessed = ndbNotesDatabase.UnprocessedDocuments</code>		
<p>Ta niesamowicie przydatna właściwość obowiązuje jedynie agentów lub skrypty akcji widoków. Może być użyta tylko na obiektach <i>NotesDocument</i> uzyskanych dzięki właściwości <i>CurrentDatabase</i> obiektu <i>NotesSession</i> (► 259). Zwraca obiekt <i>NotesDocumentCollection</i> (► 149) zawierający dokumenty należące do bazy danych, które bieżący agent lub akcja widoku uznaje za „nieprzetworzone”. Gdy dostęp następuje spoza agenta lub akcji widoku, zwrócony zbiór nie będzie zawierać żadnych dokumentów. Natomiast w razie wywołania właściwości <i>UnprocessedDocuments</i> z obiektu <i>NotesDatabase</i>, do którego dostęp nie nastąpił za pomocą właściwości <i>CurrentDatabase</i>, metoda zgłosi błąd.</p> <p>Warto pamiętać, że jeśli używamy agentów działających na nowych i zmodyfikowanych dokumentach, niedawno otrzymanych dokumentach pocztowych, dokumentach wklejonych lub świeżo zmodyfikowanych, to w celu oznaczenia każdego dokumentu zbioru jako „przetworzonego” trzeba wywołać metodę <i>UpdateProcessedDoc</i> z <i>NotesSession</i> (► 268). Zapobiega to wielokrotnemu przetwarzaniu tych samych dokumentów.</p> <p>Gdy nie wywołamy tej metody dla każdego dokumentu, to agent przy każdorazowym uruchomieniu będzie przetwarzał te same dokumenty. Należy pamiętać, że właściwość <i>UpdateProcessedDoc</i> oznacza dokument jako „przetworzony” jedynie dla wyraźnie określonego agenta, w którym została wywołana.</p> <p>Właściwość <i>UnprocessedDocuments</i> użyta w akcji widoku zwraca te same dokumenty, co baza danych uruchomiona na wybranych dokumentach.</p> <p>Poniższa tabela ilustruje dokładnie, czego można oczekiwać, używając tej metody.</p>		

Zadanie (którego wystąpienie uruchamia agenta)	Wynik jest zbiorem dokumentów (Document Collection) spełniającym poniższe kryteria
Wszystkie dokumenty w bazie danych	Spełnione są kryteria wyszukiwania określone w Database Builder (edytorze bazy danych).
Wszystkie dokumenty nowe i zmodyfikowane od czasu ostatniego uruchomienia agenta	Nieprzetwarzane uprzednio przez bazę za pomocą metody <i>UpdateProcessedDoc</i> .
Niedawno utworzone lub zmodyfikowane	Spełnione są kryteria wyszukiwania określone w Database Builder (Edytorze bazy danych).
Wszystkie nie przeczytane dokumenty w widoku	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wszystkie dokumenty w widoku	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Dokumenty zaznaczone / akcja widoku	Dokumenty zaznaczone w widoku, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Uruchomione jeden raz	Dokument bieżący.

Zadanie (którego wystąpienie uruchamia agenta)	Wynik jest zbiorem dokumentów (Document Collection) spełniającym poniższe kryteria
Niedawno wysłane dokumenty	Świeżo wysłane do bazy danych, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Niedawno modyfikowane dokumenty	Dokumenty świeżo zmodyfikowane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).

---

**Views**

O

()

To get: `varViews = ndbNotesDatabase.Views`

Właściwość zwraca tablicę zawierającą wszystkie obiekty *NotesView* (► 273) znajdujące się w bazie danych.

## Metody

---

**Compact**

`LngSizediff = ndbNotesDatabase.Compact`

Metoda nie posiada parametrów, zwraca liczbę (w formacie długim), reprezentującą przestrzeń zwolnioną w wyniku kompaktowania.

---

**Create**

Call `ndbNotesDatabase.Create(strServer, strFilename, intOpenFlag, intMaxSize)`

<code>strServer</code>	(wmg)	\$
Nazwa serwera, na którym chcemy utworzyć bazę danych. Aby utworzyć bazę na komputerze lokalnym, należy podać pusty ciąg znaków ("").		
<code>strFilename</code>	(wmg)	\$
Nazwa pliku, którą chcemy nadać bazie.		
<code>intOpenFlag</code>	(wmg)	T/F
Parametr ten służy do otwierania bazy danych podczas jej tworzenia (gdy jego wartość wynosi TRUE). Jeśli otwarcie nie jest przewidywane, należy parametrowi nadać wartość FALSE. W celu jej późniejszego otwarcia należy zastosować metodę <code>Open</code> lub <code>OpenIfModified</code> .		
<code>intMaxSize</code>	(opc)	%
Maksymalny dopuszczalny rozmiar bazy danych.		

Metoda umożliwi tworzenie nowego obiektu *NotesDatabase*. Oba parametry: `strServer` i `strFileName` mogą zawierać pusty ciąg, o ile zainicjowaliśmy już obiekt *NotesDatabase*, podając nazwę serwera i pliku.

---

**CreateCopy**

Set `ndbNotesDatabase = ndbNotesDatabase.CreateCopy(strServer, strFileName, intMaxSize)`

<code>strServer</code>	(wmg)	\$
Nazwa serwera, na którym chcemy utworzyć nową kopię bazy danych. Jeśli ma nim być komputer lokalny, to podać musimy pusty ciąg znaków ("").		

<code>strFileName</code>	(wmg)	\$
Nazwa pliku, którą chcemy przypisać nowej kopii.		
<code>intMaxSize</code>	(opc)	%
Maksymalny dopuszczalny rozmiar bazy danych.		

---

Metoda `CreateCopy` tworzy nową kopię (nie replikę) aktualnej bazy danych, zwracaną jako obiekt *NotesDatabase*. Warto pamiętać, że w razie gdyby baza danych na serwerze określonym w `strServer` miała taką samą nazwę, jak baza podana w `strFileName`, to zgłoszony zostałby błąd. Jednocześnie do nowej bazy danych skopiowana zostanie lista dostępu ACL.

---

### CreateDocument

Set `ndbNotesDocument` = `ndbNotesDatabase.CreateDocument`

Metoda nie posiada parametrów, zwraca nowy obiekt *NotesDocument* (► 135). Można ją stosować wraz z automatyzacją OLE w celu utworzenia nowego obiektu bez stosowania metody `New`. Należy pamiętać o wywołaniu metody `Save` z obiektu *NotesDocument* w celu zapisania nowego dokumentu na dysku.

---

### CreateFromTemplate

Set `ndbNotesDatabase` = `ndbNotesDatabase.CreateFromTemplate(strServer, strFileName, intInherit, intMaxSize)`

<code>strServer</code>	(wmg)	\$
Nazwa serwera, na którym znajdować się będzie nowa kopia bazy danych. Pozostawienie pustego ciągu znaków (" ") spowoduje utworzenie nowej kopii na komputerze lokalnym.		
<code>strFileName</code>	(wmg)	\$
Nazwa pliku, którą chcemy przypisać nowej kopii.		
<code>intInherit</code>	(wmg)	T/F
Jeśli chcemy, aby nowa baza danych dziedziczyła strukturę bazy bieżącej, to wartość parametru <code>intInherit</code> ustawiamy na <code>TRUE</code> . W przeciwnym razie — na <code>FALSE</code> .		
<code>intMaxSize</code>	(opc)	%
Maksymalny dopuszczalny rozmiar bazy danych.		

---

Metoda stosowana jest do tworzenia nowej bazy z już istniejącej. Nowa baza będzie zawierać wszystkie elementy projektu i dokumenty bazy źródłowej. Zwraca obiekt *NotesDatabase* przedstawiający nową bazę danych.

---

### CreateOutline

Set `noeNotesOutline` = `ndbNotesDatabase.CreateOutline(StrName, intDefaultOutline)`

<code>strName</code>	(wmg)	\$
Nazwa nowego szkicu.		
<code>intDefaultOutline</code>	(opc)	T/ F
Wartość <code>TRUE</code> tworzy nowy szkic z domyślnymi wpisami.		

---

Metoda `CreateOutline` pozwala na utworzenie nowego szkicu w bieżącej bazie danych.

---

### CreateReplica

Set `ndbNotesDatabase` = `ndbNotesDatabase.CreateReplica(strServer, strFileName)`

---

**strServer** (wmg) \$  
 Nazwa serwera, na którym znajdować się będzie nowa kopia bazy danych. Pozostawienie pustego ciągu znaków (" ") spowoduje utworzenie nowej kopii na komputerze lokalnym.

**strFileName** (wmg) \$  
 Nazwa pliku, którą chcemy przypisać nowej kopii.

Metoda zwraca obiekt *NotesDatabase* przedstawiający nową replikę kopii bieżącej bazy danych. Lista dostępu ACL aktualnej bazy danych kopiowana jest do nowej repliki. Podobnie jak w metodzie *CreateCopy*, w razie gdyby baza danych na serwerze określonym w *strServer* miała taką samą nazwę, jak baza podana w *strFileName*, to zgłoszony zostałby błąd. Jednocześnie do nowej bazy danych skopiowana zostanie lista dostępu ACL.

### EnableFolder

Call `ndbNotesDatabase.EnableFolder(strFolderName)`

**strFolderName** (wmg) \$  
 Nazwa udostępnianego foldera.

Metoda `EnableFolder` pozwala na udostępnianie folderów.

### 5.0 FTDomainSearch

Set `docNotesDocument = ndbNotesDatabase.FTDomainSearch(strQuery, intMaxDocs, strEntryForm, intSortOptions, intOtherOptions, intStart, intCount)`

**strQuery** (wmg) \$  
 Zapytanie, które chcemy wykonać.

**intMaxDocs** (wmg) %  
 Określa maksymalną liczbę dokumentów, które mają zostać zwrócone w odpowiedzi na zapytanie. Aby otrzymać wszystkie dokumenty spełniające kryterium, należy parametrowi `intMaxDocs` nadać wartość 0.

**strEntryForm** (wmg) \$  
 Nazwa formularza wyszukiwania w katalogu dziedziny.

**intSortOptions** (opc) %  
 Wskazuje, którą z trzech opcji sortowania chcemy zastosować. Można posłużyć się jedną z następujących stałych.

Stała	Rezultat
FT_SCORES	Sortowanie w oparciu o trafność wyniku. Jest to ustawienie domyślne.
FT_DATE_DES	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności malejącej.
FT_DATE_ASC	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności rosnącej.

**intOtherOptions** (opc) %  
 Wskazuje dodatkowe opcje wyszukiwania. Można zastosować jedną z następujących stałych: `FT_STEMS`, `FT_DATABASE`, `FT_FILESYSTEM` lub `FT_FUZZY`.

**intStart** (opc) %  
 Pierwsza zwracana strona.



intCount (opc) %  
Liczba zwracanych stron.

---

Jest to bardzo użyteczna metoda, która dokonuje przeszukania pełnotekstowego wszystkich dokumentów w domenie. Zwraca *NotesDocument* zawierający wykaz formatowanych dokumentów spełniających warunki zapytania. Aby użyć metody, trzeba najpierw skonfigurować katalog domeny (Domain Catalog).

---

## FTSearch

Set ndcNotesDocumentCollection = ndbNotesDatabase.FTSearch(strQuery, intMaxDocs, intSortOptions, intOtherOptions)

---

strQuery (wmg) \$  
Zapytanie, które chcemy wykonać.

intMaxDocs (wmg) %  
Określa maksymalną liczbę dokumentów, które mają zostać zwrócone w odpowiedzi na zapytanie. Aby otrzymać wszystkie dokumenty spełniające kryterium, należy parametrowi intMaxDocs nadać wartość 0.

intSortOptions (opc) %  
Wskazuje, którą z trzech opcji sortowania chcemy zastosować. Można posłużyć się jedną z następujących stałych.

Stała	Rezultat
FT_SCORES	Sortowanie w oparciu o trafność wyniku. Jest to ustawienie domyślne.
FT_DATE_DES	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności malejącej.
FT_DATE_ASC	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności rosnącej.

---

intOtherOptions (opc) %

Wskazuje dodatkowe opcje wyszukiwania. Można zastosować jedną z następujących stałych: FT\_STEMS, FT\_DATABASE, FT\_FILESYSTEM lub FT\_FUZZY.

---

Metoda FTSearch wyszukuje wszystkie dokumenty znajdujące się w bazie danych i zwraca zbiór *NotesDocumentCollection* zawierający wszystkie dokumenty spełniające kryterium zapytania pełnotekstowo. Dokumenty sortowane są w oparciu o parametr intSortOptions. Jeśli nie podano żadnego typu opcji, dokumenty są zwracane w porządku według trafności ich wyniku. Jeśli wyniki są sortowane według daty, nie będą zwracane wyniki trafności. Ważne jest, aby pamiętać, że metoda ta będzie działać nawet wtedy, gdy baza danych nie jest indeksowana pełnotekstowo, jednak działanie będzie dużo wolniejsze.

Jeśli nie chcemy przeszukiwać wszystkich dokumentów w bazie danych, możemy użyć metody FTSearch klasy *NotesView* (► 281) w celu wyszukiwania tylko dokumentów zawartych w określonym widoku. Możemy również użyć FTSearch z *NotesDocumentCollection*.

Poniższy przykład ukazuje tworzenie ogólnej funkcji wyszukiwania pełnotekstowego.

```
Public Function ExecuteSearch(strQuery As String, intmaxDocs As Integer) As
NotesDocumentCollection
Dim S As New NotesSession
If Not S.CurrentDatabase.IsFTIndexed Then
If S.CurrentDatabase.Server="" Then
Call S.CurrentDatabase.UpdateFTIndex(True)
Else
```

```

    If MsgBox("Baza ta NIE jest zindeksowana pełnotekstowo! Można jednak dokonać
    ↵wyszukiwania pełnotekstowego, ale będzie ono znacznie wolniejsze...Czy mimo to chcesz
    ↵wykonać wyszukiwanie?", 32+4, "Db Not FTIndexed")=7 Then
        Exit Function
    End If
End If
End If
Set ExecuteSearch=S.CurrentDatabase.FTSearch(strQuery, intMaxDocs)
End Function

```

---

### GetAgent

```
Set nagtNotesAgent = ndbNotesDatabase.GetAgent(strAgentName)
```

---

<b>strAgentName</b>	(wmg)	\$
---------------------	-------	----

Nazwa agenta, którego chcemy otrzymać.

---

Metoda zwraca obiekt *NotesAgent* (► 89) przedstawiający podanego agenta.

---

### GetDocumentByID

```
Set ndocNotesDocument = ndbNotesDatabase.GetDocumentByID(strNoteID)
```

---

<b>strNoteID</b>	(wmg)	\$
------------------	-------	----

NoteID dokumentu.

---

Metoda zwraca obiekt *NotesDocument* (► 135) przedstawiający dokument, którego NoteID odpowiada wartości podanej w strNoteID. Proszę zapoznać się z poniższym przykładem.

```

Dim S As New NotesSession
Dim ndocDocument As NotesDocument
Set ndocDocument=S.CurrentDatabase.getDocumentById("57CA")
'w tym miejscu można wstawić inny kod

```

---

### GetDocumentByUNID

```
Set ndocNotesDocument = ndbNotesDatabase.GetDocumentByUNID(strNoteUNID)
```

---

<b>strNoteUNID</b>	(wmg)	\$
--------------------	-------	----

Universal ID dokumentu.

---

Metoda zwraca obiekt *NotesDocument* (► 135) przedstawiający dokument, którego Universal ID odpowiada wartości podanej w strUNID. Proszę zapoznać się z poniższym przykładem.

```

Dim S As New NotesSession
Dim ndocDocument As NotesDocument
Set ndocDocument=S.CurrentDatabase.getDocumentByUNID
("B46F3758FF0ZB05A7852565F60044A3E5")
'w tym miejscu można wstawić inny kod

```

---

### GetDocumentByURL

```

Set ndocNotesDocument = ndbNotesDatabase.GetDocumentByURL(strURL, intReload,
↵intUrllist, strCharset, strWebusername, strWebpassword, strProxywebusername,
↵strProxywebpassword, intReturnimmediately)

```

---

<b>strURL</b>	(wmg)	\$
---------------	-------	----

Adres URL strony, którą chcemy uzyskać. Można wprowadzić do 15 kB.

intReload (opc) %

Jeśli ustawiona zostanie wartość 0 (standardowa), to strona pobierana będzie z hosta w Internecie tylko wtedy, gdy nie ma jej w bazie danych Web Navigator Notes. Podanie wartości 1 sprawia, że strona pobierana jest z internetowego hosta do pamięci za każdym razem. Wartość 2 sprawia, że strona ładowana jest z hosta tylko wtedy, gdy została zmodyfikowana po zapisaniu jej w bazie danych Web Navigatora.

intUrllist (opc) %

Parametr ten pozwala określić, czy chcemy zachować odnośniki z pobranych stron w polu o nazwie URLLinksn w dokumencie Notes (Web Navigator tworzy nowe pole URLLinksn za każdym razem, gdy rozmiar pola osiągnie 64 kB). Podanie wartości TRUE spowoduje zapisanie adresów URL w polu (polach) URLLinksn, a wartość FALSE oznacza, że nie będą zapisane.

Wartość 1 (TRUE) podajemy, gdy adresy URL chcemy zapisać w polu (polach) URLLinksn. Jeśli nie chcemy zapisywać adresów URL w polu (polach) URLLinksn wystarczy podać wartość 0 (FALSE) lub też pominąć ten parametr. Jeśli zapiszemy adresy URL, możemy stosować je w agentach. Można, na przykład, utworzyć agenta, który otwiera witryny Web w bazie Web Navigatora, a następnie ładuje wszystkie witryny Web zapisane w każdym z pól URLLinksn.

strCharset (opc) \$

Określa kodowanie znaków MIME (na przykład, ISO 8859-1 dla U.S.A.), które chcemy zastosować do przetwarzania stron internetowych przez Web Navigator. Parametr powinien zostać określony, tylko gdy Web Navigator nie jest w stanie poprawnie określić zestawu znaków MIME.

strWebusername (opc) \$

Jeśli host wymaga identyfikacji, można użyć tego parametru, by podać hostowi nazwę użytkownika.

strWebpassword (opc) \$

Jeśli host wymaga identyfikacji, można użyć tego parametru, by podać hostowi hasło.

strProxywebusername (opc) \$

Jeśli serwer proxy wymaga identyfikacji, można użyć tego parametru, by podać serwerowi nazwę użytkownika.

strProxywebpassword (opc) \$

Jeśli serwer proxy wymaga identyfikacji, można użyć tego parametru, by podać serwerowi hasło.

intReturnImmediately (opc) %

Parametr wskazuje, czy kontynuować skrypt, nie czekając na zakończenie procesu pobierania. Podanie wartości TRUE oznacza natychmiastowy zwrot wartości przez skrypt, FALSE (wartość standardowa) oznacza oczekiwanie. Należy pamiętać, że podanie wartości TRUE nie powoduje zwrotu obiektu *NotesDocument*, reprezentującego stronę o określonym adresie URL.

---

Metoda `GetDocumentByUrl` tworzy dokument w bazie danych nawigatora sieci i zwraca dla niego obiekt *NotesDocument* (pod warunkiem, że wartością parametru `intReturnImmediately` nie jest FALSE). Metoda działa zarówno na bazach Server Web Navigator, jak i Personal Web Navigator.

```
Dim S As New NotesSession
```

```
Dim ndocDocument As NotesDocument
```

```
Set ndocDocument=S.CurrentDatabase.GetDocumentByUrl (http://www.definiti.com, 1)
```

```
'w tym miejscu można wstawić inny kod
```

---

**GetForm**

```
Set nfrmNotesForm = ndbNotesDatabase.GetForm(strName)
```

---

<b>strName</b>	(wmg)	\$
----------------	-------	----

Nazwa albo alias formularza, do którego chcemy uzyskać dostęp.

---

Metoda zwraca obiekt *NotesForm* (► 177) przedstawiający formularze określone w parametrze strName.

---

**GetOutline**

```
Set noeNotesOutline = ndbNotesDatabase.GetOutline(strName)
```

---

<b>strName</b>	(wmg)	\$
----------------	-------	----

Nazwa szkicu, do którego chcemy uzyskać dostęp.

---

Metoda pozwala uzyskać dostęp do istniejących obiektów szkicu.

---

**GetProfileDocCollection**

```
Set ndcNotesDocumentCollection = ndbNotesDatabase.GetProfileDocCollection (strName)
```

---

<b>strName</b>	(wmg)	\$
----------------	-------	----

Nazwa zbioru, którą chcemy pobrać.

---

Metoda zwraca zbiór dokumentów profilowanych.

---

**GetProfileDocument**

```
Set ndocNotesDocument = ndbNotesDatabase.GetProfileDocument(strProfileName, strUsername)
```

---

<b>strProfileName</b>	(wmg)	\$
-----------------------	-------	----

Nazwa albo alias *profilu* dokumentu, który chcemy utworzyć lub do którego chcemy uzyskać dostęp.

---

<b>strUsername</b>	(opc)	\$
--------------------	-------	----

Nazwa użytkownika lub klucz dokumentu profilu, który chcemy pobrać bądź utworzyć.

---

Jeśli żaden dokument profilu nie odpowiada podanym parametrom, to metoda utworzy nowy dokument profilu z nazwą i nazwą użytkownika określoną przez strProfileName i strUserName. W przeciwnym wypadku zwrócony zostanie obiekt *NotesDocument* (► 135), przedstawiający dokument profilu odpowiadający parametrom. Proszę zapoznać się z poniższym przykładem.

```
Dim S As New NotesSession
Dim ndocDocument As NotesDocument
Set ndocDocument=S.CurrentDatabase.getProfileDocument("Przechowywanie danych ", "0 S.UserName)
' tu można wstawić inny kod
```

---

**GetURLHeaderInfo**

```
strHeader = NotesDatabase.GetURLHeaderInfo(strURL, strHeaderName, strWebusername, strWebpassword, strProxywebusername, strProxywebpassword)
```

---

<b>strURL</b>	(wmg)	\$
---------------	-------	----

Adres URL strony, którą chcemy pobrać. Można wprowadzić do 15 kB.

<code>strHeadername</code>	(wmg)	\$
Określa ciąg nagłówka dla adresu URL, który chcemy pobrać. Więcej informacji o dopuszczalnych ciągach nagłówków można znaleźć w specyfikacji HTTP ( <a href="http://www.w3.org/">http://www.w3.org/</a> ).		
<code>strWebusername</code>	(opc)	\$
Jeśli host wymaga identyfikacji, można użyć tego parametru, by podać hostowi nazwę użytkownika.		
<code>strWebpassword</code>	(opc)	\$
Jeśli host wymaga identyfikacji, można użyć tego parametru, by podać hostowi hasło.		
<code>strProxywebusername</code>	(opc)	\$
Jeśli serwer proxy wymaga identyfikacji, można użyć tego parametru, by podać serwerowi nazwę użytkownika.		
<code>strProxywebpassword</code>	(opc)	\$
Jeśli serwer proxy wymaga identyfikacji, można użyć tego parametru, by podać serwerowi hasło.		

---

Metoda zwraca wartość żadanego nagłówka w postaci ciągu. Jeśli zwrócony zostanie pusty ciąg znaków (" "), to albo pod podanym adresem URL nie znaleziono wartości żadanego nagłówka, albo podany adres URL nie został znaleziony w ogóle.

---

## GetView

```
Set nvwNotesView = ndbNotesDatabase.GetView(strViewName)
```

<code>strViewName</code>	(wmg)	\$
Parametr ten określa nazwę lub alias widoku bądź foldera, do których chcemy uzyskać dostęp. Stosować należy albo pełną nazwę widoku lub foldera (z ukośnikami włącznie), albo pełne aliasy, ale nigdy nazw i aliasów jednocześnie.		

Wywołanie metody z lokalnej bazy danych powoduje zwrócenie publicznych i osobistych widoków i folderów. Dla baz danych opartych na serwerze metoda zwróci tylko widoki i foldery publiczne.

Metoda `GetView` zwraca obiekt *NotesView* (► 273) przedstawiający widok albo folder określony przez parametr `strViewName`.

---

## GrantAccess

```
Call ndbNotesDatabase.GrantAccess(strName, intLevel)
```

<code>strName</code>	(wmg)	\$
Parametr ten służy do wskazania nazwy osoby, grupy lub serwera, dla których chcemy ustawić poziom dostępu.		

<code>intLevel</code>	(wmg)	%
Wskazuje poziom dostępu, który chcemy przydzielić. Poniższa tabela przedstawia dopuszczalne stałe dla tego parametru.		

Stała	Opis
<code>ACLLEVEL_NOACCESS</code>	Uprawnienia dostępu na poziomie „No Access” (brak dostępu).
<code>ACLLEVEL_DEPOSITOR</code>	Uprawnienia dostępu na poziomie „Depositor” (deponent).

Stała	Opis
ACLEVEL_READER	Uprawnienia dostępu na poziomie „Reader” (czytelnik).
ACLEVEL_AUTHOR	Uprawnienia dostępu na poziomie „Author” (autor).
ACLEVEL_EDITOR	Uprawnienia dostępu na poziomie „Editor” (redaktor).
ACLEVEL_DESIGNER	Uprawnienia dostępu na poziomie „Designer” (projektant).
ACLEVEL_MANAGER	Uprawnienia dostępu na poziomie „Manager” (menadżer).

Metoda pozwala ustawić w bazie ACL odpowiedni poziom dostępu dla dowolnego serwera, osoby lub grupy. Jeśli serwer, osoba lub grupa określone w `strName` istnieją w ACL, to ich poziomy dostępu aktualizowane są zgodnie z wartością parametru `intLevel`. W innej sytuacji, podana nazwa dodawana jest do listy ACL wraz z określonym poziomem dostępu. Wywołanie metody `GrantAccess` powoduje nadanie rola ACL ich wartości domyślnych.

## New

```
Dim ndbNotesDatabase As New NotesDatabase(strServer, strFileName)
Set ndbNotesDatabase = New NotesDatabase(strServer, strFileName)
```

<b>strServer</b>	(wmg)	\$
------------------	-------	----

Nazwa serwera, na którym chcemy utworzyć bazę danych. Aby utworzyć bazę na komputerze lokalnym, należy podać pusty ciąg znaków (" ").

<b>strFileName</b>	(wmg)	\$
--------------------	-------	----

Rzeczywista nazwa pliku bazy danych, do której chcemy uzyskać dostęp.

Metoda `New` zwraca obiekt *NotesDatabase* przedstawiający bazę o podanej lokalizacji i nazwie. Jeśli to możliwe, otwiera również tę bazę. Warto pamiętać, że metoda ta (w przeciwieństwie do metod `New` innych klas) nie tworzy nowej bazy na dysku. W tym celu skorzystać należy z jednej z trzech następujących metod: `Create`, `CreateCopy` lub `CreateReplica`.

## Open

### Open

```
intflag = ndbNotesDatabase.Open(strServer, strFileName)
```

<b>strServer</b>	(wmg)	\$
------------------	-------	----

Wskazuje nazwę serwera, na którym znajduje się baza danych. Jeśli baza istnieje na komputerze lokalnym, to podać wystarczy pusty ciąg znaków (" ").

<b>strFileName</b>	(wmg)	\$
--------------------	-------	----

Określa faktyczną nazwę pliku bazy danych, dostęp do której chcemy uzyskać. Jeśli obiekt *NotesDatabase* został już zainicjalizowany, to jako parametr podać należy pusty ciąg (" ").

Metoda `Open` pozwala na takie otwarcie bazy danych, aby skrypt miał dostęp do jej właściwości. Zwraca wartość `TRUE`, gdy znalezienie i otwarcie określonej bazy danych kończy się powodzeniem i wartość `FALSE`, jeśli określonej bazy danych nie można było otworzyć. W celu zastosowania tej metody, wywołujący ją skrypt musi mieć prawa dostępu do bazy na poziomie co najmniej czytelnika.

Istnieją dwa podstawowe sposoby użycia metody. Pierwszy polega na otwarciu już zainicjalizowanej bazy danych. Wówczas, jako wartość każdego z dwu parametrów wystarczy

podać pusty ciąg znaków (" "). Sposób drugi polega na otwarciu już istniejącej bazy danych, która nie została jeszcze zainicjalizowana, kiedy to trzeba podać zarówno nazwę serwera, jak i nazwę pliku.

---

### OpenByReplicaID

```
intFlag = ndbNotesDatabase.OpenByReplicaID(strServer, strRepID)
```

---

strServer	(wmg)	\$
-----------	-------	----

Parametr wskazuje nazwę serwera, na którym znajduje się baza danych. Jeśli baza istnieje na komputerze lokalnym, należy zastosować pusty ciąg znaków.

strRepID	(wmg)	\$
----------	-------	----

Określa replica ID bazy danych, którą chcemy otworzyć.

---

Metoda stosuje replica ID do próby otwarcia bazy danych. Zwraca wartość TRUE, gdy otwarcie zakończyło się powodzeniem oraz wartość FALSE w przeciwnym przypadku. Proszę zapoznać się z poniższym przykładem.

```
Dim S As New NotesSession  
Dim ndb As New NotesDatabase (" ")  
intflag = ndb.OpenByReplicaID(" ", "8525664B:005100C1")
```

---

### OpenIfModified

```
intFlag = ndbNotesDatabase.OpenIfModified(strServer, strDbfile, ndtNotesDateTime)
```

---

strServer	(wmg)	\$
-----------	-------	----

Parametr wskazuje nazwę serwera, na którym znajduje się baza danych. Jeśli baza istnieje na komputerze lokalnym, to jako wartość parametru podać należy pusty ciąg znaków.

strDbfile	(wmg)	\$
-----------	-------	----

Parametr ten określa nazwę pliku bazy danych, którą chcemy otworzyć.

ndtNotesDateTime	(wmg)	O.Ref.
------------------	-------	--------

Obiekt *NotesDateTime* (► 124) przedstawiający datę graniczną. Jeśli jakkolwiek dokument bazy był po tej dacie modyfikowany, to baza danych zostaje otwarta. Jeśli żaden dokument w bazie nie został po tej dacie zmodyfikowany, to baza danych pozostaje zamknięta.

---

Metoda pozwala na warunkowe otwieranie bazy danych w oparciu o to, czy dokonywano modyfikacji zawartych w niej dokumentów. Zwraca wartość TRUE, gdy otwarcie podanej bazy zakończyło się powodzeniem oraz wartość FALSE w przeciwnym przypadku.

---

### OpenMail

```
Call ndbNotesDatabase.OpenMail
```

Metoda nie posiada parametrów. Podejmuje próbę lokalizacji oraz otwarcia bazy pocztowej bieżącego użytkownika. Jeśli metoda wywoływana jest przez skrypt uruchomiony na stacji roboczej, to próbuje określić plik poczty użytkownika na podstawie zapisów w pliku *Notes.ini*. Jeśli metoda *OpenMail* wywołana jest na serwerze, to agent próbuje ustalić serwer pocztowy oraz plik poczty właściciela tegoż skryptu na podstawie zamieszczonej na serwerze Publicznej książki adresowej. Warto pamiętać, że skrypty działające na serwerze nie posiadają prawa dostępu do innych serwerów, tak więc, wywołując tę metodę na serwerze, plik poczty powinien znajdować się na serwerze, na którym uruchomiony zostanie agent. W innym razie wygenerowany zostanie komunikat o błędzie.

---

**OpenURLDb**

```
intFlag = ndbNotesDatabase.OpenURLDb
```

Metoda próbuje otworzyć bazę danych Web Navigator. Zwraca wartość TRUE, gdy otwarcie bazy zakończyło się powodzeniem oraz wartość FALSE w przeciwnym przypadku.

---

**OpenWithFailover**

```
intFlag = ndbNotesDatabase.OpenWithFailover(strServer, strFileName)
```

---

<b>strServer</b>	(wmg)	\$
------------------	-------	----

Nazwa głównego serwera, w którego klastrze rezyduje baza danych.

---

<b>strFileName</b>	(wmg)	\$
--------------------	-------	----

Faktyczna nazwa pliku bazy danych, którą chcemy otworzyć.

---

Metoda pozwala skorzystać z zalet stosowanego w Domino klastrowania serwerów i otworzyć bazę danych znajdującą się na serwerze wchodzącym w skład klastra. Zwraca wartość TRUE, gdy otwarcie bazy zakończyło się pomyślnie oraz wartość FALSE w przeciwnym przypadku. Jeśli baza nie może zostać otwarta na wybranym serwerze, to metoda automatycznie odnajduje w klastrze jej replikę, po znalezieniu której, otwiera bazę danych. Właściwość Server wskazuje wówczas serwer, na którym replika bazy została znaleziona.

---

**QueryAccess**

```
intLevel = ndbNotesDatabase.QueryAccess(strName)
```

---

<b>strName</b>	(wmg)	\$
----------------	-------	----

Parametr ten określa nazwy osoby, grupy lub serwera, dla których chcemy znać poziom dostępu.

---

Metoda QueryAccess szuka w liście dostępu ACL bazy danych wartości określonych w parametrze strName oraz zwraca stałą reprezentującą poziom dostępu dla podanej nazwy. Zestawienie możliwych stałych przedstawiamy w tabeli poniżej.

---

<b>Stala</b>	<b>Poziom dostępu</b>
ACLLLEVEL_NOACCESS	Uprawnienia dostępu na poziomie „No Access” (brak dostępu).
ACLLLEVEL_DEPOSITOR	Uprawnienia dostępu na poziomie „Depositor” (deponent).
ACLLLEVEL_READER	Uprawnienia dostępu na poziomie „Reader” (czytelnik).
ACLLLEVEL_AUTHOR	Uprawnienia dostępu na poziomie „Author” (autor).
ACLLLEVEL_EDITOR	Uprawnienia dostępu na poziomie „Editor” (redaktor).
ACLLLEVEL_DESIGNER	Uprawnienia dostępu na poziomie „Designer” (projektant).
ACLLLEVEL_MANAGER	Uprawnienia dostępu na poziomie „Manager” (menadżer).

---

A oto sposób, w jaki omawiana metoda wyprowadza poziom dostępu.

Jeśli podana nazwa (osoby, grupy lub serwera) jest ujęta na liście dostępu i jest również członkiem grup (grupy) wchodzących w skład listy ACL, to zwracany jest najwyższy poziom dostępu.

Jeśli podana nazwa odnosi się do członka grupowego, to zwracany jest poziom dostępu dla tej grupy.



Jeśli podana nazwa odnosi się do członka kilku grup listy ACL, to zwracany jest poziom dostępu grupy o najwyższym jej poziomie.

Jeśli podana nazwa nie odnosi się do członka żadnej grupy, to zwracany jest standardowy poziom dostępu.

Warto pamiętać, że metoda korzysta z podstawowej książki adresowej urządzenia, na którym wykonywany jest skrypt. Na stacji roboczej używa osobistej książki adresowej, a na serwerze — publicznej książki adresowej.

---

## Remove

Call `ndbNotesDatabase.Remove`

Metoda nie przyjmuje parametrów. Trwale usuwa bazę danych z przestrzeni dyskowej.

---

## Replicate

`intFlag = ndbNotesDatabase.Replicate(strServerName)`

---

<code>strServerName</code>	(wmg)	\$
----------------------------	-------	----

Określa nazwę serwera, względem którego replikacja ma być wykonywana. Jeśli istnieje więcej niż jedna replika bazy danych na podanym serwerze, to zreplikowane zostaną wszystkie.

---

Metoda daje możliwość automatyzacji procesu replikacji. Zwraca wartość `TRUE`, jeśli proces replikacji zakończy się pomyślnie oraz wartość `FALSE`, w razie wygenerowania jakichkolwiek błędów. Poniżej przedstawiamy przykład jej zastosowania.

```
Dim S As New NotesSession
S.CurrentDatabase.Replicate("Gonzo/Definiti")
```

---

## RevokeAccess

Call `ndbNotesDatabase.RevokeAccess(strName)`

---

<code>strName</code>	(wmg)	\$
----------------------	-------	----

Parametr ten używany jest do określania nazwy serwera, osoby lub grupy, których prawa dostępu odwołujemy.

---

Metoda pozwala usunąć wskazany w parametrze `strName` serwer, osobę lub grupę z listy ACL bazy danych, skutecznie nadając mu dostęp o poziomie domyślnym. Warto pamiętać, że tym właśnie różni się od przyznania dostępu o poziomie brak dostępu przy użyciu metody `GrantAccess`, co całkowicie zamyka dostęp serwera, osoby lub grupy do bazy danych. Jeśli serwer, osoba lub grupa podana w parametrze `strName` nie może być znaleziona, to metoda `RevokeAccess` generuje komunikat o błędzie.

---

## Search

Set `ndcNotesDocumentCollection = ndbNotesDatabase.Search(strFormula, ndtNotesDateTime, intMaxDocs)`

---

<code>strFormula</code>	(wmg)	\$
-------------------------	-------	----

Parametr stosowany jest do przekazania formuły funkcji `@` określającej kryteria wyboru.

<code>ndtNotesDateTime</code>	(wmg)	O.Ref.
-------------------------------	-------	--------

Parametr jest obiektem `NotesDateTime` (► 124) stosowanym jako data graniczna, tak aby przeszukiwane były tylko dokumenty utworzone lub modyfikowane po tej dacie.

intMaxDocs (wmg) %

Parametr określa liczbę dokumentów, która powinna być zwracana w zbiorze. Nadając temu parametrowi wartość 0, sprawimy, że zbiór dokumentów spełniających kryterium wyboru będzie obejmował wszystkie dokumenty.

Metoda zwraca niesortowany zbiór *NotesDocumentCollection* zawierający wszystkie dokumenty spełniające kryterium wyboru. Zawsze jest zdecydowanie wolniejsza od wyszukiwania pełnotekstowego. Jeśli zatem istnieje alternatywa, zawsze warto korzystać z wyszukiwania pełnotekstowego.

### UnprocessedFTSearch

Set ndcNotesDocumentCollection = ndbNotesDatabase.UnprocessedFTSearch (strQuery, intMaxDocs, intSortOptions, intOtherOptions)

strQuery (wmg) \$

Zapytanie, które chcemy wykonać.

intMaxDocs (wmg) %

Określa maksymalną liczbę dokumentów, które chcemy otrzymać w odpowiedzi na zapytanie. Aby otrzymać wszystkie dokumenty spełniające kryterium, parametrowi temu nadać należy wartość 0.

intSortOptions (opc) %

Wskazuje, którą z trzech opcji sortowania chcemy zastosować. Można posłużyć się jedną z następujących stałych.

Stała	Rezultat
FT_SCORES	Sortowanie w oparciu o trafność wyniku. Jest to ustawienie domyślne.
FT_DATE_DES	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności malejącej.
FT_DATE_ASC	Sortuje wyniki w oparciu o datę utworzenia dokumentu w kolejności rosnącej.

intOtherOptions (opc) %

Określa dodatkowe opcje wyszukiwania. Można zastosować jedną z następujących stałych: FT\_STEMS, FT\_DATABASE, FT\_FILESYSTEM lub FT\_FUZZY.

Metoda *UnprocessedFTSearch* dotyczy jedynie agentów i skryptów akcji widoków. Można z niej korzystać tylko dla obiektów *NotesDocument* uzyskanych dzięki właściwości *CurrentDatabase* z obiektu *NotesSession* (► 259). Zwraca ona obiekt *NotesDocumentCollection* (► 158) będący zbiorem dokumentów bazy danych, które bieżący agent lub akcja widoku uznaje za „nieprzetworzone”. Gdy dostęp następuje spoza agenta lub akcji widoku, zwrócony zbiór nie będzie zawierał żadnych dokumentów. Natomiast wywołanie metody z obiektu *NotesDatabase*, do którego dostęp nastąpił inaczej niż poprzez właściwość *CurrentDatabase* spowoduje zgłoszenie błędu.

Warto pamiętać, że jeśli używamy agentów działających na nowych i zmodyfikowanych dokumentach, niedawno otrzymanych dokumentach pocztowych, dokumentach wklejonych lub świeżo zmodyfikowanych, to w celu oznaczenia każdego dokumentu zbioru jako „przetworzonego” trzeba wywołać metodę *UpdateProcessedDoc* z *NotesSession* (► 268). Zapobiega to wielokrotnemu przetwarzaniu tych samych dokumentów.

Gdy nie wywołamy tej metody dla każdego dokumentu, to agent przy każdorazowym uruchomieniu będzie przetwarzał te same dokumenty. Należy pamiętać, że właściwość `UpdateProcessedDoc` oznacza dokument jako „przetworzony” jedynie dla wyraźnie określonego agenta, w którym została wywołana.

Właściwość `UnprocessedDocuments` użyta w akcji widoku zwraca te same dokumenty, co baza danych uruchomiona na wybranych dokumentach.

Poniższa tabela ilustruje dokładnie, czego można oczekiwać, używając tej metody.

Zadanie (na którego wystąpienie agent zadziała)	Wynik jest zbiorem dokumentów (Document Collection), który spełnia poniższe kryteria
Wszystkie dokumenty w bazie danych.	Spełnione są kryteria wyszukiwania określone w Database Builder (Edytorze bazy danych).
Wszystkie nowe i zmodyfikowane od czasu ostatniego uruchomienia dokumenty.	Nieprzetwarzane poprzednio przez tego agenta za pomocą metody <code>UpdateProcessedDoc</code> .
Niedawno utworzone lub modyfikowane.	Spełnione są kryteria wyszukiwania określone w Database Builder (Edytorze bazy danych).
Wszystkie nieczytane dokumenty w widoku.	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wszystkie dokumenty w widoku.	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wybrane dokumenty.	Dokumenty wskazane (wybrane) w widoku, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Uruchomione jeden raz.	Bieżący dokument.
Niedawno wysłane dokumenty.	Świeżo wysłane do bazy danych, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Niedawno modyfikowane dokumenty.	Świeżo modyfikowane dokumenty, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wklejone dokumenty.	Świeżo wklejone dokumenty, spełniające kryteria podane w Database Builder (Edytorze bazy danych).

Jeśli warunki te są spełnione, metoda wykonuje pełnotekstowe wyszukiwanie dokumentów uważanych przez agenta za nieprzetworzone i zwraca rezultaty w postaci posortowanego zbioru dokumentów.

### UnprocessedSearch

```
Set ndcNotesDocumentCollection = ndbNotesDatabase.UnprocessedSearch (strFormula,
    &ndtNotesDateTime, intMaxDocs)
```

`strFormula`

(wmg)

\$

Parametr stosowany jest do przekazania formuły funkcji @ określającej kryteria wyboru.

`ndtNotesDateTime`

(wmg)

O.Ref.

Parametr jest obiektem `NotesDateTime` (► 124) określającym datę graniczną, tak aby przeszukiwane były tylko dokumenty utworzone lub modyfikowane po tej dacie.

intMaxDocs (wmg) %

Parametr określa liczbę dokumentów, które chcemy otrzymać w wyniku zapytania. Aby otrzymać wszystkie dokumenty spełniające kryterium, należy w parametrze wpisać 0.

Metoda zwraca niesortowany *NotesDocumentCollection* zawierający wszystkie dokumenty spełniające kryterium wyboru. Metoda ta jest zawsze znacznie wolniejsza niż wykonywanie wyszukiwania pełnotekstowego. Powinna być zatem unikana, gdy istnieje możliwość wyszukiwania pełnotekstowego.

Metoda dotyczy jedynie agentów lub skryptów akcji widoków. Może być użyta tylko na obiektach *NotesDocument* uzyskanych dzięki właściwości *CurrentDatabase* z obiektu *NotesSession*. Zwraca obiekt *NotesDocumentCollection* zawierający dokumenty z bazy danych, które bieżący agent lub akcja widoku uznaje za „nieprzetworzone”. Gdy dostęp następuje spoza agenta lub akcji widoku, zwrócony zbiór nie będzie zawierał żadnych dokumentów. Natomiast, wywołując metodę z obiektu *NotesDatabase*, do którego dostęp nie nastąpił z *CurrentDatabase*, metoda zgłosi błąd.

Warto pamiętać, że jeśli używamy agentów działających na nowych i zmodyfikowanych dokumentach, niedawno otrzymanych dokumentach pocztowych, dokumentach wklejonych lub świeżo zmodyfikowanych, to w celu oznaczenia każdego dokumentu zbioru jako „przetworzonego” trzeba wywołać metodę *UpdateProcessedDoc* z *NotesSession* (► 268). Zapobiega to wielokrotnemu przetwarzaniu tych samych dokumentów.

Gdy nie wywołamy tej metody dla każdego dokumentu, to agent przy każdorazowym uruchomieniu będzie przetwarzał te same dokumenty. Należy pamiętać, że właściwość *UpdateProcessedDoc* oznacza dokument jako „przetworzony” jedynie dla wyraźnie określonego agenta, w którym została wywołana.

Właściwość *UnprocessedDocuments* użyta w akcji widoku zwraca te same dokumenty, co baza danych uruchomiona na wybranych dokumentach.

Poniższa tabela ilustruje dokładnie, czego można oczekiwać, używając tej metody.

Zadanie (na którego wystąpienie agent zadziała)	Wynik jest zbiorem dokumentów (Document Collection), który spełnia poniższe kryteria
Wszystkie dokumenty w bazie danych.	Spełnione są kryteria wyszukiwania określone w Database Builder (Edytorze bazy danych).
Wszystkie nowe i zmodyfikowane od czasu ostatniego uruchomienia dokumenty.	Nieprzetwarzane poprzednio przez tego agenta za pomocą metody <i>UpdateProcessedDoc</i> .
Niedawno utworzone lub modyfikowane.	Spełnione są kryteria wyszukiwania określone w Database Builder (Edytorze bazy danych).
Wszystkie nieczytane dokumenty w widoku.	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wszystkie dokumenty w widoku.	Dokumenty z widoku, nieprzeczytane, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wybrane dokumenty / widoki akcji.	Dokumenty wskazane (wybrane) w widoku, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Uruchomione jeden raz.	Bieżący dokument.

Zadanie (na którego wystąpienie agent zadziała)	Wynik jest zbiorem dokumentów (Document Collection), który spełnia poniższe kryteria
Niedawno wysłane dokumenty.	Świeżo wysłane do bazy danych. spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Niedawno modyfikowane dokumenty.	Świeżo modyfikowane dokumenty, spełniające kryteria podane w Database Builder (Edytorze bazy danych).
Wklejone dokumenty.	Świeżo wklejone dokumenty, spełniające kryteria podane w Database Builder (Edytorze bazy danych).

Jeśli warunki te są spełnione, metoda wykonuje pełnotekstowe wyszukiwanie dokumentów uważanych przez agenta za nieprzetworzone i zwraca rezultaty w postaci posortowanego zbioru dokumentów.

## UpdateFTIndex

Call `ndbNotesDatabase.UpdateFTIndex(intCreateFlag)`

intCreateFlag	(wmg)	T/F
Jeśli nie istnieje żaden indeks, to podanie wartości TRUE spowoduje utworzenie nowego indeksu (dotyczy to tylko baz lokalnych). W przeciwnym razie, istniejący indeks zostanie zaktualizowany. Jeśli baza nie jest lokalna, to jako wartość parametru <code>intCreateFlag</code> podać należy FALSE.		

Metoda pozwala na programowe tworzenie (tylko dla lokalnych baz danych) lub aktualizację indeksów pełnotekstowych.

## Komentarz

Podczas wykonywania działań na klasie *NotesDatabase* pamiętać należy o następujących zasadach.

Przed uzyskaniem dostępu do większości właściwości bazy danych przy użyciu LotusScript, bazę trzeba najpierw otworzyć.

W celu sprawdzenia, czy baza danych jest otwarta można skorzystać z właściwości `IsOpen`. Jeśli nie jest, to można ją otworzyć, wywołując metodę `Open` lub `OpenIfModified`.

Aby otworzyć bazę danych, użytkownik lub agent użytkownika (a dokładnie właściciel agenta) musi posiadać prawa dostępu o poziomie co najmniej typu „Reader”.

Żaden skrypt działający na serwerze nie może uzyskać dostępu do bazy danych na innym serwerze. Próba uzyskania takiego dostępu spowoduje wygenerowanie błędu. Podobnie, jeśli skrypt nie posiada prawa dostępu o poziomie odpowiednim do wykonywanej funkcji, to jego wykonanie zostanie zaniechane.

Jeśli dostęp do obiektu *NotesDatabase* następuje przez obiekt *NotesDbDirectory*, to obiekt taki nie jest uważany za otwarty. Za otwarte nie można uważać również obiektów *NotesDatabase*, do których dostęp uzyskujemy przez właściwość `AddressBooks` z klasy *NotesSession*. Także obiekty *NotesDatabase* utworzone za pomocą metody `New` nie są otwarte, o ile baza nie istnieje na podanym w parametrze `strServer` serwerze i nie posiada nazwy określonej w `strFileName`.

Zamknięte bazy udostępniają jednak następujące właściwości: FileName, FilePath, IsOpen, IsPrivateAddressBook, IsPublicAddressBook, Parent oraz Server.

## Przykład

Poniższy przykład sprawdza, czy aktualny użytkownik znajduje się na liście ACL bieżącej bazy danych oraz czy posiada prawa dostępu na poziomie przynajmniej autora. Jeśli tak, to udostępniany jest mu dostęp na poziomie „Editor”. Jeśli nie, to przyznawany jest mu dostęp na poziomie „Reader”.

```
Dim S as New NotesSession
Dim naclCur As NotesACL
Dim naclEntry As NotesACLEntry
Dim initFound As Integer
Set naclCur=S.CurrentDatabase.ACL
Set naclEntry=naclCur.GetFirstEntry
Do While Not naclEntry Is Nothing
  If S.UserName=naclEntry.Name Then
    If naclEntry.Level<=ACLELEVEL_AUTHOR Then
      naclEntry.Level=ACLELEVEL_EDITOR
    End If
    Exit Sub
  End If
Loop
Call S.CurrentDatabase.GrantAccess(S.UserName, ACLELEVEL_READER)
```