



INNOWACYJNA
GOSPODARKA
NARODOWA STRATEGIA SPÓJNOŚCI



MINISTERSTWO
SPRAWIEDLIWOŚCI

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



Dokumentacja Powykonawcza Nadzoru Autorskiego nad Oprogramowaniem Cyfrowej Rejestracji Przebiegu Rozpraw Sądowych Cześć V. Dokumentacja interfejsów programistycznych

Dokumentacja interfejsów programistycznych systemu ReCourt będąca częścią dokumentacji powykonawczej zadań nadzoru autorskiego nad eksploatacją oraz rozbudowy oprogramowania „ReCourt” świadczonego przez Comarch Polska S.A. na zlecenie Ministerstwa Sprawiedliwości w ramach Umowy DIRS nr 51/2012 z dnia 29 października 2012 roku

Projekt „Wdrożenie protokołu elektronicznego w sprawach cywilnych i wykroczeniowych”

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego na podstawie Porozumienia o dofinansowanie nr POIG.07.01.00-00-061/13-00 w ramach 7 osi priorytetowej „Społeczeństwo informacyjne – budowa elektronicznej administracji”, Programu Operacyjnego Innowacyjna Gospodarka 2007-2013

METRYKA DOKUMENTU

Tytuł dokumentu	Dokumentacja Powykonawcza Nadzoru Autorskiego Oprogramowania Cyfrowej Rejestracji Przebiegu Rozpraw Sądowych Cześć V. Dokumentacja interfejsów programistycznych
Zamawiający	Ministerstwo Sprawiedliwości
Wykonawca	Comarch Polska S.A.
Autorzy	Krzysztof Garliński, Mirosław Wajda, Tomasz Meresiński, Comarch Polska S.A.
Wersja	1.5
Data ostatniej modyfikacji	2015-12-31
Ilość stron	975
Nazwa pliku	DIRS 51 - DPNAiR ReCourt - Cz.V - Dokumentacja interfejsów programistycznych.docx

TABELA MODYFIKACJA

Wersja	Data	Zmodyfikował	Opis zmian
1.5	2015-12-31	Magdalena Radoń	Utworzenia dokumentu – skompletowanie specyfikacji interfejsów zewnętrznych systemu ReCourt oraz przekazanie do Zamawiającego do akceptacji

Spis treści

I.	Wstęp.....	74
II.	Interfejsy modułu REcourt 2.....	74
	Dokumentacja API dostarczanego przez moduł AVUnit	74
1	Dokumentacja grup	74
1.1	Wartości globalne	74
1.1.1	Definicje.....	74
1.1.2	Zmienne.....	74
1.1.3	Opis szczegółowy.....	74
1.1.4	Dokumentacja zmiennych	74
1.2	Typy.....	75
1.2.1	Struktury danych	75
1.2.2	Definicje typów.....	75
1.2.3	Wyliczenia.....	76
1.2.4	Opis szczegółowy.....	77
1.2.5	Dokumentacja definicji typów	77
1.2.6	Dokumentacja typów wyliczanych	78
1.3	Funkcje inicjalizacyjne	82
1.3.1	Funkcje	82
1.3.2	Opis szczegółowy.....	83
1.3.3	Dokumentacja funkcji.....	83
1.4	Funkcje modułu audio.....	91
1.4.1	Funkcje	91
1.4.2	Opis szczegółowy.....	93
1.4.3	Dokumentacja funkcji.....	93
1.5	Funkcje modułu wideo.....	100
1.5.1	Funkcje	100
1.5.2	Opis szczegółowy.....	103



1.5.3	Dokumentacja funkcji.....	103
1.6	Funkcje modułu kodującego	122
1.6.1	Funkcje	122
1.6.2	Opis szczegółowy.....	123
1.6.3	Dokumentacja funkcji.....	123
1.7	Funkcje modułu dekodującego	131
1.7.1	Funkcje	131
1.7.2	Opis szczegółowy.....	131
1.7.3	Dokumentacja funkcji.....	132
1.8	Funkcje do przetwarzania plików multimedialnych	136
1.8.1	Moduły	136
1.8.2	Funkcje służące do przekonwertowania nagrania na inny format.	136
1.8.3	Opis szczegółowy.....	138
1.8.4	Dokumentacja funkcji.....	138
1.9	Funkcje ekstrakcji.....	145
1.9.1	Funkcje	145
1.9.2	Opis szczegółowy.....	145
1.9.3	Dokumentacja funkcji.....	145
1.10	Funkcje do przekonwertowania nagrania.....	147
1.10.1	Funkcje	147
1.10.2	Opis szczegółowy.....	148
1.10.3	Dokumentacja funkcji.....	148
1.11	Obsługa urządzeń zewnętrznych	152
1.11.1	Moduły	152
1.11.2	Funkcje	152
1.11.3	Opis szczegółowy.....	152
1.11.4	Dokumentacja funkcji.....	152
1.12	Sterowanie kamerami poprzez interfejs PTZ.....	154
1.12.1	Funkcje	155
1.12.2	Opis szczegółowy.....	155



1.12.3	Dokumentacja funkcji.....	155
1.13	Komunikacja z kartami akwizycji dźwięku	158
1.13.1	Funkcje	158
1.13.2	Opis szczegółowy.....	159
1.13.3	Dokumentacja funkcji.....	159
1.14	Komunikacja ze wzmacniaczem.....	167
1.14.1	Funkcje	167
1.14.2	Opis szczegółowy.....	167
1.14.3	Dokumentacja funkcji.....	168
1.15	Komunikacja z zasilaczem awaryjnym UPS.....	172
1.15.1	Funkcje	173
1.15.2	Opis szczegółowy.....	173
1.15.3	Dokumentacja funkcji.....	174
1.16	Obsługa odtwarzacza monitora wielkoformatowego	178
1.16.1	Funkcje	179
1.16.2	Opis szczegółowy.....	179
1.16.3	Dokumentacja funkcji.....	179
1.17	Obsługa odczytywania tekstu	185
1.17.1	Funkcje	185
1.17.2	Opis szczegółowy.....	185
1.17.3	Dokumentacja funkcji.....	186
1.18	Dostępne miejsca na dysku.....	187
1.18.1	Funkcje	187
1.18.2	Opis szczegółowy.....	188
1.18.3	Dokumentacja funkcji.....	188
1.19	Klient NTP.....	191
1.19.1	Funkcje	191
1.19.2	Opis szczegółowy.....	191
1.19.3	Dokumentacja funkcji.....	191
1.20	Obsługa błędów krytycznych	194



1.20.1	Definicje typów.....	194
1.20.2	Wyliczenia.....	194
1.20.3	Funkcje	194
1.20.4	Opis szczegółowy.....	194
1.20.5	Dokumentacja definicji typów	194
1.20.6	Dokumentacja typów wyliczanych.....	194
1.20.7	Dokumentacja funkcji.....	195
2	Dokumentacja struktur danych	196
2.1	Dokumentacja struktury AVUnitCrop	196
2.1.1	Pola danych	196
2.1.2	Opis szczegółowy.....	196
2.2	Dokumentacja struktury AVUnitParameterInteger	196
2.2.1	Pola danych	196
2.2.2	Opis szczegółowy.....	197
2.3	Dokumentacja struktury AVUnitParameterList	198
2.3.1	Pola danych	198
2.3.2	Opis szczegółowy.....	198
2.4	Dokumentacja struktury AVUnitRect.....	198
2.4.1	Pola danych	198
2.4.2	Opis szczegółowy.....	198
3	Dokumentacja plików	200
3.1	Dokumentacja pliku AVUnit.h.....	200
3.1.1	Struktury danych	200
3.1.2	Definicje.....	200
3.1.3	Definicje typów.....	200
3.1.4	Wyliczenia.....	201
3.1.5	Funkcje	202
3.1.6	Zmienne.....	215
3.1.7	Opis szczegółowy.....	215
3.2	AVUnit.h.....	215



Dokumentacja API dostarczanego przez bibliotekę casmb	233
1 Dokumentacja grup	233
1.1 Wartości globalne	233
1.1.1 Definicje.....	233
1.1.2 Opis szczegółowy.....	233
1.2 Typy.....	233
1.2.1 Struktury danych	233
1.2.2 Definicje typów.....	233
1.2.3 Wyliczenia.....	234
1.2.4 Zmienne.....	235
1.2.5 Opis szczegółowy.....	235
1.2.6 Dokumentacja definicji typów	235
1.2.7 Dokumentacja typów wyliczanych.....	237
1.2.8 Dokumentacja zmiennych.....	241
1.3 Funkcje narzędziowe.....	242
1.3.1 Funkcje	242
1.3.2 Opis szczegółowy.....	242
1.3.3 Dokumentacja funkcji.....	242
1.4 Funkcje inicjalizacyjne.....	242
1.4.1 Funkcje	243
1.4.2 Opis szczegółowy.....	243
1.4.3 Dokumentacja funkcji.....	243
1.5 Funkcje połączeniowe.....	245
1.5.1 Funkcje	245
1.5.2 Opis szczegółowy.....	245
1.5.3 Dokumentacja funkcji.....	245
1.6 Operacje na plikach.....	246
1.6.1 Funkcje	247
1.6.2 Opis szczegółowy.....	247
1.6.3 Dokumentacja funkcji.....	247



1.7	Listowanie zawartości katalogów	252
1.7.1	Funkcje	252
1.7.2	Opis szczegółowy.....	252
1.7.3	Dokumentacja funkcji.....	253
1.8	Operacje na katalogach i atrybutach plików	254
1.8.1	Funkcje	254
1.8.2	Opis szczegółowy.....	255
1.8.3	Dokumentacja funkcji.....	255
2	Dokumentacja struktur danych	259
2.1	Dokumentacja struktury CASMBAttributes	259
2.1.1	Pola danych	259
2.1.2	Opis szczegółowy.....	259
2.2	Dokumentacja struktury CASMBDirectoryEntry.....	260
2.2.1	Pola danych	260
2.2.2	Opis szczegółowy.....	260
3	Dokumentacja plików	260
3.1	Dokumentacja pliku casmb.h.....	260
3.1.1	Struktury danych	260
3.1.2	Definicje.....	260
3.1.3	Definicje typów.....	260
3.1.4	Wyliczenia.....	261
3.1.5	Funkcje	262
3.1.6	Opis szczegółowy.....	264
3.2	casmb.h	264
	Dokumentacja API dostarczanego przez moduł RemoteView.....	270
1	Dokumentacja grup	270
1.1	Wartości globalne	270
1.1.1	Definicje.....	270
1.1.2	Zmienne.....	270
1.1.3	Opis szczegółowy.....	270



1.1.4	Dokumentacja zmiennych	270
1.2	Typy	270
1.2.1	Definicje typów	271
1.2.2	Wyliczenia	271
1.2.3	Opis szczegółowy	271
1.2.4	Dokumentacja definicji typów	271
1.2.5	Dokumentacja typów wyliczanych	271
1.3	Funkcje inicjalizacyjne i połączeniowe	272
1.3.1	Funkcje	272
1.3.2	Opis szczegółowy	273
1.3.3	Dokumentacja funkcji	273
1.4	Funkcje odtwarzacza	282
1.4.1	Funkcje	282
1.4.2	Opis szczegółowy	282
1.4.3	Dokumentacja funkcji	283
2	Dokumentacja plików	285
2.1	Dokumentacja pliku RemoteView.h	285
2.1.1	Definicje	285
2.1.2	Definicje typów	285
2.1.3	Wyliczenia	285
2.1.4	Funkcje	286
2.1.5	Zmienne	287
2.1.6	Opis szczegółowy	287
2.2	RemoteView.h	287
Dokumentacja API dostarczanego przez moduł SpeechSynthesis		291
1	Dokumentacja grup	291
1.1	Typy	291
1.1.1	Definicje typów	291
1.1.2	Wyliczenia	291
1.1.3	Opis szczegółowy	291



1.1.4	Dokumentacja typów wyliczanych	291
1.2	Funkcje inicjalizacyjne	292
1.2.1	Funkcje	292
1.2.2	Opis szczegółowy.....	292
1.2.3	Dokumentacja funkcji.....	292
1.3	Funkcje do kontroli procesu odczytu	294
1.3.1	Funkcje	294
1.3.2	Opis szczegółowy.....	295
1.3.3	Dokumentacja funkcji.....	295
2	Dokumentacja plików	297
2.1	Dokumentacja pliku SpeechSynthesis.h	297
2.1.1	Definicje.....	297
2.1.2	Definicje typów.....	297
2.1.3	Wyliczenia.....	297
2.1.4	Funkcje	298
2.1.5	Opis szczegółowy.....	299
2.1.6	Dokumentacja definicji.....	299
2.2	SpeechSynthesis.h.....	299
Dokumentacja API dostarczanego przez moduł Wideoterminala		302
1	Dokumentacja grup	302
1.1	Wartości globalne	302
1.1.1	Definicje.....	302
1.1.2	Zmienne.....	302
1.1.3	Opis szczegółowy.....	302
1.1.4	Dokumentacja zmiennych	302
1.2	Typy.....	302
1.2.1	Struktury danych	302
1.2.2	Definicje typów.....	302
1.2.3	Opis szczegółowy.....	302
1.3	Inicjalizacja	302



1.3.1	Wyliczenia.....	303
1.3.2	Funkcje	303
1.3.3	Opis szczegółowy.....	303
1.3.4	Dokumentacja typów wyliczanych.....	303
1.3.5	Dokumentacja funkcji.....	303
1.4	Zakończenie działania	305
1.4.1	Definicje typów.....	305
1.4.2	Funkcje	305
1.4.3	Opis szczegółowy.....	305
1.4.4	Dokumentacja definicji typów	305
1.4.5	Dokumentacja funkcji.....	305
1.5	Połączenia	305
1.5.1	Definicje typów.....	306
1.5.2	Wyliczenia.....	306
1.5.3	Funkcje	306
1.5.4	Opis szczegółowy.....	307
1.5.5	Wykonywanie połączeń.....	307
1.5.6	Odbieranie połączeń	307
1.5.7	Zakończenie połączenia.....	307
1.5.8	Dokumentacja definicji typów	307
1.5.9	Dokumentacja typów wyliczanych	308
1.5.10	Dokumentacja funkcji.....	310
1.6	Zarządzanie źródłami	312
1.6.1	Struktury danych	312
1.6.2	Opis źródła.....	312
1.6.3	Wyliczenia.....	312
1.6.4	Funkcje	313
1.6.5	Opis szczegółowy.....	313
1.6.6	Podstawowe pojęcia	313
1.6.7	Strumienie a źródła	313



1.6.8	Aktywność źródeł	313
1.6.9	Reprezentacja strumieni i źródeł w interfejsie	313
1.6.10	Podgląd obrazu	313
1.6.11	Dokumentacja definicji typów	314
1.6.12	Dokumentacja typów wyliczanych	314
1.6.13	Dokumentacja funkcji.....	315
1.7	Rozmiar wideokonferencji	316
1.7.1	Struktury danych	316
1.7.2	Rozdzielczość Funkcje.....	316
1.7.3	Opis szczegółowy.....	316
1.7.4	Dokumentacja funkcji.....	316
1.8	Książki adresowe oraz historia połączeń.....	317
1.8.1	Definicje typów.....	317
1.8.2	Wyliczenia.....	317
1.8.3	Funkcje	317
1.8.4	Opis szczegółowy.....	318
1.8.5	Książki adresowe	318
1.8.6	Historia połączeń	318
1.8.7	Format danych	318
1.8.8	Dokumentacja typów wyliczanych.....	318
1.8.9	Dokumentacja funkcji.....	319
1.9	Status pracy Wideoterminala	323
1.9.1	Funkcje	323
1.9.2	Opis szczegółowy.....	324
1.9.3	Dokumentacja funkcji.....	324
1.10	Inne	325
1.10.1	Definicje typów.....	325
1.10.2	Funkcje	326
1.10.3	Opis szczegółowy.....	326
1.10.4	Dokumentacja definicji typów	326



1.10.5	Dokumentacja funkcji.....	326
2	Dokumentacja struktur danych	326
2.1	Dokumentacja struktury C_Resolution	326
2.1.1	Pola danych	326
2.1.2	Opis szczegółowy.....	326
2.1.3	Dokumentacja pól	326
2.2	Dokumentacja struktury Callbacks	327
2.2.1	Pola danych	327
2.2.2	Opis szczegółowy.....	327
2.2.3	Dokumentacja pól	327
2.3	Dokumentacja struktury Handle.....	331
2.3.1	Opis szczegółowy.....	331
2.4	Dokumentacja struktury SourceDescriptor	331
2.4.1	Pola danych	331
2.4.2	Opis szczegółowy.....	331
2.4.3	Dokumentacja pól	331
2.5	Dokumentacja struktury StatusHandle.....	332
2.5.1	Opis szczegółowy.....	332
3	Dokumentacja plików	332
3.1	Dokumentacja pliku conferenceInterface.h	332
3.1.1	Struktury danych	332
3.1.2	Rozdzielczość Definicje.....	333
3.1.3	Definicje typów.....	333
3.1.4	Wyliczenia.....	333
3.1.5	Funkcje	334
3.1.6	Zmienne.....	336
3.1.7	Opis szczegółowy.....	336
3.2	conferenceInterface.h.....	336
III.	Interfejs komponentu podpisu	340
3.1	Przykład zdefiniowanego parametru dla komponentu:	340



3.2	Przykładowe wywołanie:	341
IV.	Interfejsy RCS dla systemów i programów zewnętrznych.....	342
1	Interfejsy zewnętrzne RCS	342
2	Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base	342
2.1	Interfejsy	342
3	Interfejs ICourtRepositoryCommunicationService	343
3.1	Składnia	343
3.2	Metody	343
3.3	Zobacz również	344
4	Metody ICourtRepositoryCommunicationService.ICourtRepositoryCommunicationService 345	
4.1	Metody	345
4.2	Zobacz również	346
5	Metoda ICourtRepositoryCommunicationService.AddDiscriminant	347
5.1	Składnia	347
5.2	Zobacz również	347
6	Metoda ICourtRepositoryCommunicationService.AddStorage	348
6.1	Składnia	348
6.2	Zobacz również	348
7	Metoda ICourtRepositoryCommunicationService.CheckTrialsExistsForDiscriminant	349
7.1	Składnia	349
7.2	Zobacz również	349
8	Metoda ICourtRepositoryCommunicationService.GetCourtDepartments	350
8.1	Składnia	350
8.2	Zobacz również	350
9	Metoda ICourtRepositoryCommunicationService.GetCourtRoomNames.....	351
9.1	Składnia	351
9.2	Zobacz również	351
10	Metoda ICourtRepositoryCommunicationService.GetCourtRoomsVO	352
10.1	Składnia	352
10.2	Zobacz również	352



11	Metoda ICourtRepositoryCommunicationService.GetCourts	353
11.1	Składnia	353
11.2	Zobacz również	353
12	Metoda ICourtRepositoryCommunicationService.GetDepartments	354
12.1	Składnia	354
12.2	Zobacz również	354
13	Metoda ICourtRepositoryCommunicationService.GetDiscriminantByCode	355
13.1	Składnia	355
13.2	Zobacz również	355
14	Metoda ICourtRepositoryCommunicationService.GetDiscriminantById	356
14.1	Składnia	356
14.2	Zobacz również	356
15	Metoda ICourtRepositoryCommunicationService.GetDiscriminants	357
15.1	Składnia	357
15.2	Zobacz również	357
16	Metoda ICourtRepositoryCommunicationService.GetDiscriminantsVO.....	358
16.1	Składnia	358
16.2	Zobacz również	358
17	Metoda ICourtRepositoryCommunicationService.GetDiscriminantsVOPagger	359
17.1	Składnia	359
17.2	Zobacz również	359
18	Metoda ICourtRepositoryCommunicationService.GetStorages.....	360
18.1	Składnia	360
18.2	Zobacz również	360
19	Metoda ICourtRepositoryCommunicationService.GetStoragesByType.....	361
19.1	Składnia	361
19.2	Zobacz również	361
20	Metoda ICourtRepositoryCommunicationService.GetStoragesPagger	362
20.1	Składnia	362
20.2	Zobacz również	362



21	Metoda ICourtRepositoryCommunicationService.RemoveCourtRoom	363
21.1	Składnia	363
21.2	Zobacz również	363
22	Metoda ICourtRepositoryCommunicationService.RemoveDiscriminant.....	364
22.1	Składnia	364
22.2	Zobacz również	364
23	Metoda ICourtRepositoryCommunicationService.RemoveStorage.....	365
23.1	Składnia	365
23.2	Zobacz również	365
24	Metoda ICourtRepositoryCommunicationService.SetCourtRoomsVO	366
24.1	Składnia	366
24.2	Zobacz również	366
25	Metoda ICourtRepositoryCommunicationService.UpdateDiscriminant	367
25.1	Składnia	367
25.2	Zobacz również	367
26	Metoda ICourtRepositoryCommunicationService.UpdateStorage	368
26.1	Składnia	368
26.2	Zobacz również	368
27	Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base	369
27.1	Interfejsy	369
28	Interfejs IDictionaryRepositoryCommunicationService	370
28.1	Składnia	370
28.2	Metody.....	370
28.3	Zobacz również	372
29	Metody IDictionaryRepositoryCommunicationService.IDictionaryRepositoryCommunicationService...	373
29.1	Metody.....	373
29.2	Zobacz również	375
30	Metoda IDictionaryRepositoryCommunicationService.AddAnnotation	376
30.1	Składnia	376
30.2	Zobacz również	376



31	Metoda IDictionaryRepositoryCommunicationService.AddBenchRole	377
31.1	Składnia	377
31.2	Zobacz również	377
32	Metoda IDictionaryRepositoryCommunicationService.AddCaseType	378
32.1	Składnia	378
32.2	Zobacz również	378
33	Metoda IDictionaryRepositoryCommunicationService.AddComplexAnnotation	379
33.1	Składnia	379
33.2	Zobacz również	379
34	Metoda IDictionaryRepositoryCommunicationService.AddDocumentTemplate	380
34.1	Składnia	380
34.2	Zobacz również	380
35	Metoda IDictionaryRepositoryCommunicationService.AddParticipantFunction	381
35.1	Składnia	381
35.2	Zobacz również	381
36	Metoda IDictionaryRepositoryCommunicationService.AddReadingRoom	382
36.1	Składnia	382
36.2	Zobacz również	382
37	Metoda IDictionaryRepositoryCommunicationService.GetAnnotations	383
37.1	Składnia	383
37.2	Zobacz również	383
38	Metoda IDictionaryRepositoryCommunicationService.GetAnnotationsPager	384
38.1	Składnia	384
38.2	Zobacz również	384
39	Metoda IDictionaryRepositoryCommunicationService.GetBenchRoles	385
39.1	Składnia	385
39.2	Zobacz również	385
40	Metoda IDictionaryRepositoryCommunicationService.GetBenchRolesPager	386
40.1	Składnia	386
40.2	Zobacz również	386



41	Metoda IDictionaryRepositoryCommunicationService.GetCaseTypes	387
41.1	Składnia	387
41.2	Zobacz również	387
42	Metoda IDictionaryRepositoryCommunicationService.GetCaseTypesPager	388
42.1	Składnia	388
42.2	Zobacz również	388
43	Metoda IDictionaryRepositoryCommunicationService.GetComplexAnnotations	389
43.1	Składnia	389
43.2	Zobacz również	389
44	Metoda IDictionaryRepositoryCommunicationService.GetComplexAnnotationsPager	390
44.1	Składnia	390
44.2	Zobacz również	390
45	Metoda IDictionaryRepositoryCommunicationService.GetComplexAnnotationUsedShortcuts 391	
45.1	Składnia	391
45.2	Zobacz również	391
46	Metoda IDictionaryRepositoryCommunicationService.GetDocumentTemplate	392
46.1	Składnia	392
46.2	Zobacz również	392
47	Metoda IDictionaryRepositoryCommunicationService.GetDocumentTemplates	393
47.1	Składnia	393
47.2	Zobacz również	393
48	Metoda IDictionaryRepositoryCommunicationService.GetDocumentTemplatesVO	394
48.1	Składnia	394
48.2	Zobacz również	394
49	Metoda IDictionaryRepositoryCommunicationService.GetFreeDays	395
49.1	Składnia	395
49.2	Zobacz również	395
50	Metoda IDictionaryRepositoryCommunicationService.GetMachineInstanceForMainDiscriminant	396
50.1	Składnia	396



50.2	Zobacz również	396
51	Metoda IDictionaryRepositoryCommunicationService.GetMachineInstances.....	397
51.1	Składnia	397
51.2	Zobacz również	397
52	Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctions	398
52.1	Składnia	398
52.2	Zobacz również	398
53	Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctionsPager	399
53.1	Składnia	399
53.2	Zobacz również	399
54	Metoda IDictionaryRepositoryCommunicationService.GetReadingRoom	400
54.1	Składnia	400
54.2	Zobacz również	400
55	Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomNames	401
55.1	Składnia	401
55.2	Zobacz również	401
56	Metoda IDictionaryRepositoryCommunicationService.GetReadingRooms.....	402
56.1	Składnia	402
56.2	Zobacz również	402
57	Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomsPager	403
57.1	Składnia	403
57.2	Zobacz również	403
58	Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomsVO.....	404
58.1	Składnia	404
58.2	Zobacz również	404
59	Metoda IDictionaryRepositoryCommunicationService.IndexDocumentTemplates	405
59.1	Składnia	405
59.2	Zobacz również	405
60	Metoda IDictionaryRepositoryCommunicationService.IsMainDiscriminantCorrect	406
60.1	Składnia	406



60.2	Zobacz również	406
61	Metoda IDictionaryRepositoryCommunicationService.RemoveAnnotation	407
61.1	Składnia	407
61.2	Zobacz również	407
62	Metoda IDictionaryRepositoryCommunicationService.RemoveBenchRole	408
62.1	Składnia	408
62.2	Zobacz również	408
63	Metoda IDictionaryRepositoryCommunicationService.RemoveCaseType	409
63.1	Składnia	409
63.2	Zobacz również	409
64	Metoda IDictionaryRepositoryCommunicationService.RemoveComplexAnnotation	410
64.1	Składnia	410
64.2	Zobacz również	410
65	Metoda IDictionaryRepositoryCommunicationService.RemoveDocumentTemplate	411
65.1	Składnia	411
65.2	Zobacz również	411
66	Metoda IDictionaryRepositoryCommunicationService.RemoveParticipantFunction.....	412
66.1	Składnia	412
66.2	Zobacz również	412
67	Metoda IDictionaryRepositoryCommunicationService.RemoveReadingRoom	413
67.1	Składnia	413
67.2	Zobacz również	413
68	Metoda IDictionaryRepositoryCommunicationService.SetFreeDays.....	414
68.1	Składnia	414
68.2	Zobacz również	414
69	Metoda IDictionaryRepositoryCommunicationService.UpdateAnnotation	415
69.1	Składnia	415
69.2	Zobacz również	415
70	Metoda IDictionaryRepositoryCommunicationService.UpdateBenchRole.....	416
70.1	Składnia	416



70.2	Zobacz również	416
71	Metoda IDictionaryRepositoryCommunicationService.UpdateCaseType	417
71.1	Składnia	417
71.2	Zobacz również	417
72	Metoda IDictionaryRepositoryCommunicationService.UpdateComplexAnnotation	418
72.1	Składnia	418
72.2	Zobacz również	418
73	Metoda IDictionaryRepositoryCommunicationService.UpdateDocumentTemplate	419
73.1	Składnia	419
73.2	Zobacz również	419
74	Metoda IDictionaryRepositoryCommunicationService.UpdateMachineInstances	420
74.1	Składnia	420
74.2	Zobacz również	420
75	Metoda IDictionaryRepositoryCommunicationService.UpdateParticipantFunction	421
75.1	Składnia	421
75.2	Zobacz również	421
76	Metoda IDictionaryRepositoryCommunicationService.UpdateReadingRoom	422
76.1	Składnia	422
76.2	Zobacz również	422
77	Przestrzeń nazw ReCourt.Services.Communication.Documents.Base	423
77.1	Interfejsy	423
78	Interfejs IDocumentsCommunicationService	424
78.1	Składnia	424
78.2	Metody	424
78.3	Zobacz również	424
79	Metody IDocumentsCommunicationService.IDocumentsCommunicationService	425
79.1	Metody	425
79.2	Zobacz również	425
80	Metoda IDocumentsCommunicationService.AreNewDocumentAvailable	426
80.1	Składnia	426



80.2	Zobacz również	426
81	Metoda IDocumentsCommunicationService.GetDocument.....	427
81.1	Składnia	427
81.2	Zobacz również	427
82	Metoda IDocumentsCommunicationService.GetDocumentForReadingRoom.....	428
82.1	Składnia	428
82.2	Zobacz również	428
83	Metoda IDocumentsCommunicationService.GetUserDocuments	429
83.1	Składnia	429
83.2	Zobacz również	430
84	Metoda IDocumentsCommunicationService.GetUserDocumentsVO.....	431
84.1	Składnia	431
84.2	Zobacz również	432
85	Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base	433
85.1	Klasy	433
85.2	Interfejsy	433
85.3	Wyliczenia	433
86	Klasa FileTransferInfo	434
86.1	Hierarchia.....	434
86.2	Składnia	434
86.3	Konstruktory	434
86.4	Właściwości.....	434
86.5	Zobacz również	434
87	Konstruktor FileTransferInfo	435
87.1	Składnia	435
87.2	Zobacz również	435
88	Właściwości FileTransferInfo.FileTransferInfo	436
88.1	Właściwości.....	436
88.2	Zobacz również	436
89	Właściwość FileTransferInfo.Checksum	437

89.1	Składnia	437
89.2	Zobacz również	437
90	Właściwość FileTransferInfo.FileTransferDestination	438
90.1	Składnia	438
90.2	Zobacz również	438
91	Właściwość FileTransferInfo.Length	439
91.1	Składnia	439
91.2	Zobacz również	439
92	Właściwość FileTransferInfo.Name	440
92.1	Składnia	440
92.2	Zobacz również	440
93	Właściwość FileTransferInfo.SubDirName	441
93.1	Składnia	441
93.2	Zobacz również	441
94	Wyliczenie FileType	442
94.1	Składnia	442
94.2	Składowe	442
94.3	Zobacz również	442
95	Interfejs IFileTransferCommunicationService	443
95.1	Składnia	443
95.2	Metody	443
95.3	Zobacz również	443
96	Metody IFileTransferCommunicationService.IFileTransferCommunicationService	444
96.1	Metody	444
96.2	Zobacz również	444
97	Metoda IFileTransferCommunicationService.CalculateFileSize	445
97.1	Składnia	445
97.2	Zobacz również	445
98	Metoda IFileTransferCommunicationService.DropDirectory	446
98.1	Składnia	446



98.2	Zobacz również	446
99	Metoda IFileTransferCommunicationService.DropFiles	447
99.1	Składnia	447
99.2	Zobacz również	447
100	Metoda IFileTransferCommunicationService.GetFiles	448
100.1	Składnia	448
100.2	Zobacz również	448
101	Metoda IFileTransferCommunicationService.GetTextFileContent	449
101.1	Składnia	449
101.2	Zobacz również	449
102	Metoda IFileTransferCommunicationService.SendFile	450
102.1	Składnia	450
102.2	Zobacz również	450
103	Klasa SendFileRequestMessage.....	451
103.1	Hierarchia	451
103.2	Składnia	451
103.3	Konstruktory	451
103.4	Pola.....	451
103.5	Zobacz również	451
104	Konstruktor SendFileRequestMessage.....	452
104.1	Składnia	452
104.2	Zobacz również	452
105	Pola SendFileRequestMessage.SendFileRequestMessage.....	453
105.1	Pola.....	453
105.2	Zobacz również	453
106	Pole SendFileRequestMessage.FileData.....	454
106.1	Składnia	454
106.2	Zobacz również	454
107	Pole SendFileRequestMessage.FileInfo	455
107.1	Składnia	455



107.2 Zobacz również	455
108 Przestrzeń nazw ReCourt.Services.Communication.Logging.Base.....	456
108.1 Interfejsy	456
109 Interfejs ILoggingCommunicationService.....	457
109.1 Składnia	457
109.2 Metody.....	457
109.3 Zobacz również	457
110 Metody ILoggingCommunicationService.ILoggingCommunicationService.....	458
110.1 Metody.....	458
110.2 Zobacz również	458
111 Metoda ILoggingCommunicationService.LogAction	459
111.1 Składnia	459
111.2 Zobacz również	459
112 Metoda ILoggingCommunicationService.LogError.....	460
112.1 Składnia	460
112.2 Zobacz również	460
113 Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base	461
113.1 Interfejsy	461
114 Interfejs IMediaServerConfiguration.....	462
114.1 Składnia	462
114.2 Metody.....	462
114.3 Zobacz również	462
115 Metody IMediaServerConfiguration.IMediaServerConfiguration.....	463
115.1 Metody.....	463
115.2 Zobacz również	463
116 Metoda IMediaServerConfiguration.CheckConfigs.....	464
116.1 Składnia	464
116.2 Zobacz również	464
117 Metoda IMediaServerConfiguration.GetAllMachines.....	465
117.1 Składnia	465



117.2 Zobacz również	465
118 Metoda IMediaServerConfiguration.GetConfig	466
118.1 Składnia	466
118.2 Zobacz również	466
119 Metoda IMediaServerConfiguration.GetMachines	467
119.1 Składnia	467
119.2 Zobacz również	467
120 Metoda IMediaServerConfiguration.MarkConfigAsInvalid	468
120.1 Składnia	468
120.2 Zobacz również	468
121 Metoda IMediaServerConfiguration.MarkConfigAsLoaded	469
121.1 Składnia	469
121.2 Zobacz również	469
122 Metoda IMediaServerConfiguration.SaveConfig	470
122.1 Składnia	470
122.2 Zobacz również	470
123 Metoda IMediaServerConfiguration.SetMachineRoom	471
123.1 Składnia	471
123.2 Zobacz również	471
124 Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base	472
124.1 Interfejsy	472
125 Interfejs IRcsSettingsCommunicationService	473
125.1 Składnia	473
125.2 Metody	473
125.3 Zobacz również	474
126 Metody IRcsSettingsCommunicationService.IRcsSettingsCommunicationService	475
126.1 Metody	475
126.2 Zobacz również	476
127 Metoda IRcsSettingsCommunicationService.AddInformation	477
127.1 Składnia	477



127.2 Zobacz również	477
128 Metoda IRcsSettingsCommunicationService.AddLink.....	478
128.1 Składnia	478
128.2 Zobacz również	478
129 Metoda IRcsSettingsCommunicationService.AddServerInstance	479
129.1 Składnia	479
129.2 Zobacz również	479
130 Metoda IRcsSettingsCommunicationService.AddTrialListView	480
130.1 Składnia	480
130.2 Zobacz również	480
131 Metoda IRcsSettingsCommunicationService.GetActiveCourtDepartmentForDiscriminant	481
131.1 Składnia	481
131.2 Zobacz również	481
132 Metoda IRcsSettingsCommunicationService.GetConfiguredCourtDepartments	482
132.1 Składnia	482
132.2 Zobacz również	482
133 Metoda IRcsSettingsCommunicationService.GetCourtAddress.....	483
133.1 Składnia	483
133.2 Zobacz również	483
134 Metoda IRcsSettingsCommunicationService.GetCurrentTimeFromServer	484
134.1 Składnia	484
134.2 Zobacz również	484
135 Metoda IRcsSettingsCommunicationService.GetDashBoardData	485
135.1 Składnia	485
135.2 Zobacz również	485
136 Metoda IRcsSettingsCommunicationService.GetInformations.....	486
136.1 Składnia	486
136.2 Zobacz również	486
137 Metoda IRcsSettingsCommunicationService.GetLinks.....	487
137.1 Składnia	487



137.2 Zobacz również	487
138 Metoda IRcsSettingsCommunicationService.GetServerInstance.....	488
138.1 Składnia	488
138.2 Zobacz również	488
139 Metoda IRcsSettingsCommunicationService.GetServerInstanceForDepartment	489
139.1 Składnia	489
139.2 Zobacz również	489
140 Metoda IRcsSettingsCommunicationService.GetServerInstanceVos.....	490
140.1 Składnia	490
140.2 Zobacz również	490
141 Metoda IRcsSettingsCommunicationService.GetServerInstanceVosPagger.....	491
141.1 Składnia	491
141.2 Zobacz również	491
142 Metoda IRcsSettingsCommunicationService.GetSignerComponents	492
142.1 Składnia	492
142.2 Zobacz również	492
143 Metoda IRcsSettingsCommunicationService.GetTrialListViews.....	493
143.1 Składnia	493
143.2 Zobacz również	493
144 Metoda IRcsSettingsCommunicationService.RemoveInformation	494
144.1 Składnia	494
144.2 Zobacz również	494
145 Metoda IRcsSettingsCommunicationService.RemoveLink.....	495
145.1 Składnia	495
145.2 Zobacz również	495
146 Metoda IRcsSettingsCommunicationService.RemoveServerInstance	496
146.1 Składnia	496
146.2 Zobacz również	496
147 Metoda IRcsSettingsCommunicationService.RemoveTrialListView.....	497
147.1 Składnia	497



147.2 Zobacz również	497
148 Metoda IRcsSettingsCommunicationService.UpdateInformation	498
148.1 Składnia	498
148.2 Zobacz również	498
149 Metoda IRcsSettingsCommunicationService.UpdateLink	499
149.1 Składnia	499
149.2 Zobacz również	499
150 Metoda IRcsSettingsCommunicationService.UpdateServerInstance	500
150.1 Składnia	500
150.2 Zobacz również	500
151 Metoda IRcsSettingsCommunicationService.UpdateTrialListView	501
151.1 Składnia	501
151.2 Zobacz również	501
152 Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base	502
152.1 Interfejsy	502
153 Interfejs IReadingRoom	503
153.1 Składnia	503
153.2 Metody	503
153.3 Zobacz również	503
154 Metody IReadingRoom.IReadingRoom	504
154.1 Metody	504
154.2 Zobacz również	504
155 Metoda IReadingRoom.GetToken	505
155.1 Składnia	505
155.2 Zobacz również	505
156 Metoda IReadingRoom.IsTokenValid	506
156.1 Składnia	506
156.2 Zobacz również	506
157 Przestrzeń nazw ReCourt.Services.Communication.Reports.Base	507
157.1 Interfejsy	507

158 Interfejs IReportsCommunicationService.....	508
158.1 Składnia	508
158.2 Metody	508
158.3 Zobacz również	510
159 Metody IReportsCommunicationService.IReportsCommunicationService	511
159.1 Metody	511
159.2 Zobacz również	512
160 Metoda IReportsCommunicationService.AddCategory	514
160.1 Składnia	514
160.2 Zobacz również	514
161 Metoda IReportsCommunicationService.AddReportRightToCategory.....	515
161.1 Składnia	515
161.2 Zobacz również	515
162 Metoda IReportsCommunicationService.AddReportRightToReport	516
162.1 Składnia	516
162.2 Zobacz również	516
163 Metoda IReportsCommunicationService.AddReportSchedule	517
163.1 Składnia	517
163.2 Zobacz również	517
164 Metoda IReportsCommunicationService.AreNewReportsAvailable.....	518
164.1 Składnia	518
164.2 Zobacz również	518
165 Metoda IReportsCommunicationService.DeleteCategory	519
165.1 Składnia	519
165.2 Zobacz również	519
166 Metoda IReportsCommunicationService.DeleteReport	520
166.1 Składnia	520
166.2 Zobacz również	520
167 Metoda IReportsCommunicationService.DeleteReportRight	521
167.1 Składnia	521



167.2 Zobacz również	521
168 Metoda IReportsCommunicationService.ExportReports	522
168.1 Składnia	522
168.2 Zobacz również	522
169 Metoda IReportsCommunicationService.GetAllVersionForReport.....	523
169.1 Składnia	523
169.2 Zobacz również	523
170 Metoda IReportsCommunicationService.GetReport	524
170.1 Składnia	524
170.2 Zobacz również	524
171 Metoda IReportsCommunicationService.GetReportByCode	525
171.1 Składnia	525
171.2 Zobacz również	525
172 Metoda IReportsCommunicationService.GetReportCategories	526
172.1 Składnia	526
172.2 Zobacz również	526
173 Metoda IReportsCommunicationService.GetReportHistoryByUser	527
173.1 Składnia	527
173.2 Zobacz również	527
174 Metoda IReportsCommunicationService.GetReportHistoryByUserPagger	528
174.1 Składnia	528
174.2 Zobacz również	528
175 Metoda IReportsCommunicationService.GetReports	529
175.1 Składnia	529
175.2 Zobacz również	529
176 Metoda IReportsCommunicationService.GetReportSchedule.....	530
176.1 Składnia	530
176.2 Zobacz również	530
177 Metoda IReportsCommunicationService.GetReportSchedulerAdminPagger	531
177.1 Składnia	531



177.2 Zobacz również	531
178 Metoda IReportsCommunicationService.GetReportSchedulerByUserPager.....	532
178.1 Składnia	532
178.2 Zobacz również	532
179 Metoda IReportsCommunicationService.GetReportsPager.....	533
179.1 Składnia	533
179.2 Zobacz również	533
180 Metoda IReportsCommunicationService.GetReportWithRights	534
180.1 Składnia	534
180.2 Zobacz również	534
181 Metoda IReportsCommunicationService.GetReportWithRightsPager	535
181.1 Składnia	535
181.2 Zobacz również	535
182 Metoda IReportsCommunicationService.GetRolesRights.....	536
182.1 Składnia	536
182.2 Zobacz również	536
183 Metoda IReportsCommunicationService.GetUserRights	537
183.1 Składnia	537
183.2 Zobacz również	537
184 Metoda IReportsCommunicationService.ImportReports	538
184.1 Składnia	538
184.2 Zobacz również	538
185 Metoda IReportsCommunicationService.PerformQuietReportOnServer.....	539
185.1 Składnia	539
185.2 Zobacz również	539
186 Metoda IReportsCommunicationService.PerformReport.....	540
186.1 Składnia	540
186.2 Zobacz również	540
187 Metoda IReportsCommunicationService.PerformReportOnServer.....	541
187.1 Składnia	541

187.2 Zobacz również	542
188 Metoda IReportsCommunicationService.RemoveAllReportHistory	543
188.1 Składnia	543
188.2 Zobacz również	543
189 Metoda IReportsCommunicationService.RemoveReportHistory	544
189.1 Składnia	544
189.2 Zobacz również	544
190 Metoda IReportsCommunicationService.RemoveReportSchedule	545
190.1 Składnia	545
190.2 Zobacz również	545
191 Metoda IReportsCommunicationService.SwitchReportRight	546
191.1 Składnia	546
191.2 Zobacz również	546
192 Metoda IReportsCommunicationService.UpdateCategory	547
192.1 Składnia	547
192.2 Zobacz również	547
193 Metoda IReportsCommunicationService.UpdateReport	548
193.1 Składnia	548
193.2 Zobacz również	548
194 Metoda IReportsCommunicationService.UpdateReportSchedule	549
194.1 Składnia	549
194.2 Zobacz również	549
195 Przestrzeń nazw ReCourt.Services.Communication.Settings.Base	550
195.1 Interfejsy	550
196 Interfejs ISettingsCommunicationService	551
196.1 Składnia	551
196.2 Metody	551
196.3 Zobacz również	551
197 Metody ISettingsCommunicationService.ISettingsCommunicationService	552
197.1 Metody	552



197.2 Zobacz również	552
198 Metoda <code>ISettingsCommunicationService.GetCurrentServerName</code>	553
198.1 Składnia	553
198.2 Zobacz również	553
199 Metoda <code>ISettingsCommunicationService.GetCurrentTimeFromServer</code>	554
199.1 Składnia	554
199.2 Zobacz również	554
200 Metoda <code>ISettingsCommunicationService.GetGlobalSettings</code>	555
200.1 Składnia	555
200.2 Zobacz również	555
201 Metoda <code>ISettingsCommunicationService.GetRefCodes</code>	556
201.1 Składnia	556
201.2 Zobacz również	556
202 Metoda <code>ISettingsCommunicationService.GetUserSettings</code>	557
202.1 Składnia	557
202.2 Zobacz również	557
203 Metoda <code>ISettingsCommunicationService.IsInstanceAuthorized</code>	558
203.1 Składnia	558
203.2 Zobacz również	558
204 Metoda <code>ISettingsCommunicationService.SetGlobalSetting</code>	559
204.1 Składnia	559
204.2 Zobacz również	559
205 Metoda <code>ISettingsCommunicationService.SetUserSetting</code>	560
205.1 Składnia	560
205.2 Zobacz również	560
206 Przestrzeń nazw <code>ReCourt.Services.Communication.TaskList.Base</code>	561
206.1 Interfejsy	561
207 Interfejs <code>ITaskListCommunicationService</code>	562
207.1 Składnia	562
207.2 Metody.....	562



207.3 Zobacz również	562
208 Metody ITaskListCommunicationService.ITaskListCommunicationService	563
208.1 Metody	563
208.2 Zobacz również	563
209 Metoda ITaskListCommunicationService.AreNewTasksAvailable	564
209.1 Składnia	564
209.2 Zobacz również	564
210 Metoda ITaskListCommunicationService.GetTask	565
210.1 Składnia	565
210.2 Zobacz również	565
211 Metoda ITaskListCommunicationService.GetTaskParameter	566
211.1 Składnia	566
211.2 Zobacz również	566
212 Metoda ITaskListCommunicationService.GetTaskProgress	567
212.1 Składnia	567
212.2 Zobacz również	567
213 Metoda ITaskListCommunicationService.GetUserTasks	568
213.1 Składnia	568
213.2 Zobacz również	569
214 Metoda ITaskListCommunicationService.RemoveAllTasks	570
214.1 Składnia	570
214.2 Zobacz również	570
215 Metoda ITaskListCommunicationService.RemoveTask.....	571
215.1 Składnia	571
215.2 Zobacz również	571
216 Metoda ITaskListCommunicationService.UpdateTaskParams	572
216.1 Składnia	572
216.2 Zobacz również	572
217 Przestrzeń nazw ReCourt.Services.Communication.TaskManager.Base	573
217.1 Interfejsy	573



218	Interfejs ITaskManagerCommunicationService	574
218.1	Składnia	574
218.2	Metody	574
218.3	Zobacz również	574
219	Metody ITaskManagerCommunicationService.ITaskManagerCommunicationService	575
219.1	Metody	575
219.2	Zobacz również	575
220	Metoda ITaskManagerCommunicationService.GetPendingTaskList	576
220.1	Składnia	576
220.2	Zobacz również	576
221	Metoda ITaskManagerCommunicationService.TryToCancelTask	577
221.1	Składnia	577
221.2	Zobacz również	577
222	Przestrzeń nazw ReCourt.Services.Communication.TransferManager	578
222.1	Interfejsy	578
223	Interfejs ITransferManagerCommunicationService	579
223.1	Składnia	579
223.2	Metody	579
223.3	Zobacz również	579
224	Metody ITransferManagerCommunicationService.ITransferManagerCommunicationService 580	
224.1	Metody	580
224.2	Zobacz również	580
225	Metoda ITransferManagerCommunicationService.CheckIsTrialExists	581
225.1	Składnia	581
225.2	Zobacz również	581
226	Metoda ITransferManagerCommunicationService.FinalizeTransfer	582
226.1	Składnia	582
226.2	Zobacz również	582
227	Metoda ITransferManagerCommunicationService.GetCaseFileHashes	583
227.1	Składnia	583



227.2 Zobacz również	583
228 Metoda ITransferManagerCommunicationService.IsAlive	584
228.1 Składnia	584
228.2 Zobacz również	584
229 Metoda ITransferManagerCommunicationService.RegisterTransfer	585
229.1 Składnia	585
229.2 Zobacz również	587
230 Metoda ITransferManagerCommunicationService.SendContent.....	588
230.1 Składnia	588
230.2 Zobacz również	588
231 Metoda ITransferManagerCommunicationService.SynchronizationLockTrial	589
231.1 Składnia	589
231.2 Zobacz również	589
232 Metoda ITransferManagerCommunicationService.UpdateSynchronizeDestination	590
232.1 Składnia	590
232.2 Zobacz również	590
233 Metoda ITransferManagerCommunicationService.UpdateTranscriptionFiles	591
233.1 Składnia	591
233.2 Zobacz również	591
234 Przestrzeń nazw ReCourt.Services.Communication.TransferRepository.Base	592
234.1 Interfejsy	592
235 Interfejs ITransferRepositoryCommunicationService	593
235.1 Składnia	593
235.2 Metody	593
235.3 Zobacz również	593
236 Metody	
ITransferRepositoryCommunicationService.ITransferRepositoryCommunicationService	594
236.1 Metody	594
236.2 Zobacz również	594
237 Metoda ITransferRepositoryCommunicationService.CancelTransfer	595
237.1 Składnia	595



237.2 Zobacz również	595
238 Metoda <code>ITransferRepositoryCommunicationService.DeleteByType</code>	596
238.1 Składnia	596
238.2 Zobacz również	596
239 Metoda <code>ITransferRepositoryCommunicationService.GetFinishedTransfers</code>	597
239.1 Składnia	597
239.2 Zobacz również	597
240 Metoda <code>ITransferRepositoryCommunicationService.GetFinishedTransfersPager</code>	598
240.1 Składnia	598
240.2 Zobacz również	598
241 Metoda <code>ITransferRepositoryCommunicationService.GetTransfers</code>	599
241.1 Składnia	599
241.2 Zobacz również	599
242 Metoda <code>ITransferRepositoryCommunicationService.GetTransfersPager</code>	600
242.1 Składnia	600
242.2 Zobacz również	600
243 Metoda <code>ITransferRepositoryCommunicationService.IsAlive</code>	601
243.1 Składnia	601
243.2 Zobacz również	601
244 Metoda <code>ITransferRepositoryCommunicationService.TransferCase</code>	602
244.1 Składnia	602
244.2 Zobacz również	604
245 Przestrzeń nazw <code>ReCourt.Services.Communication.TrialAuthoring.Base</code>	605
245.1 Interfejsy	605
246 Interfejs <code>ITrialAuthoringCommunicationService</code>	606
246.1 Składnia	606
246.2 Metody	606
246.3 Zobacz również	606
247 Metody <code>ITrialAuthoringCommunicationService.ITrialAuthoringCommunicationService</code>	607
247.1 Metody	607



247.2 Zobacz również	607
248 Metoda ITrialAuthoringCommunicationService.ChangeDiscriminant	608
248.1 Składnia	608
248.2 Zobacz również	608
249 Metoda ITrialAuthoringCommunicationService.ChangeRoom	609
249.1 Składnia	609
249.2 Zobacz również	609
250 Metoda ITrialAuthoringCommunicationService.ChangeSignature	610
250.1 Składnia	610
250.2 Zobacz również	610
251 Metoda ITrialAuthoringCommunicationService.CopyAudioAfterSignature	611
251.1 Składnia	611
251.2 Zobacz również	612
252 Metoda ITrialAuthoringCommunicationService.EraseFragment	613
252.1 Składnia	613
252.2 Zobacz również	614
253 Metoda ITrialAuthoringCommunicationService.ExtractAudio	615
253.1 Składnia	615
253.2 Zobacz również	616
254 Metoda ITrialAuthoringCommunicationService.ExtractFragment	617
254.1 Składnia	617
254.2 Zobacz również	618
255 Metoda ITrialAuthoringCommunicationService.ExtractSeparatedAudio	619
255.1 Składnia	619
255.2 Zobacz również	619
256 Metoda ITrialAuthoringCommunicationService.ImportAudioWithExportMixed	620
256.1 Składnia	620
256.2 Zobacz również	621
257 Metoda ITrialAuthoringCommunicationService.OpenTrial	622
257.1 Składnia	622



257.2 Zobacz również	623
258 Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base	624
258.1 Interfejsy	624
259 Interfejs ITrialRepositoryCommunicationService.....	625
259.1 Składnia	625
259.2 Metody	625
259.3 Zobacz również	627
260 Metody ITrialRepositoryCommunicationService.ITrialRepositoryCommunicationService .	628
260.1 Metody	628
260.2 Zobacz również	630
261 Metoda ITrialRepositoryCommunicationService.ChangeAuthorizedDepartmentCase	631
261.1 Składnia	631
261.2 Zobacz również	631
262 Metoda ITrialRepositoryCommunicationService.ChangeCapeDiscriminantRcsCase.....	632
262.1 Składnia	632
262.2 Zobacz również	632
263 Metoda ITrialRepositoryCommunicationService.ChangeCapeSignatureCase	633
263.1 Składnia	633
263.2 Zobacz również	633
264 Metoda ITrialRepositoryCommunicationService.ChangeCapeSupplyRcsCase	634
264.1 Składnia	634
264.2 Zobacz również	634
265 Metoda ITrialRepositoryCommunicationService.ChangeCreatorRcsCase	635
265.1 Składnia	635
265.2 Zobacz również	635
266 Metoda ITrialRepositoryCommunicationService.CheckDataIntegration	636
266.1 Składnia	636
266.2 Zobacz również	636
267 Metoda ITrialRepositoryCommunicationService.CloseTrial.....	637
267.1 Składnia	637



267.2 Zobacz również	637
268 Metoda ITrialRepositoryCommunicationService.CloseTrialByData	638
268.1 Składnia	638
268.2 Zobacz również	638
269 Metoda	
ITrialRepositoryCommunicationService.CreateChangeAuthorizedDepartmentGlobalTask	639
269.1 Składnia	639
269.2 Zobacz również	639
270 Metoda	
ITrialRepositoryCommunicationService.CreateChangeCapeDiscriminantRcsGlobalTask	640
270.1 Składnia	640
270.2 Zobacz również	640
271 Metoda ITrialRepositoryCommunicationService.CreateChangeCapeSupplyRcsGlobalTask	641
271.1 Składnia	641
271.2 Zobacz również	641
272 Metoda ITrialRepositoryCommunicationService.CreateChangeCreatorRcsGlobalTask	642
272.1 Składnia	642
272.2 Zobacz również	642
273 Metoda ITrialRepositoryCommunicationService.CreateDataIntegrationCheckTask	643
273.1 Składnia	643
273.2 Zobacz również	643
274 Metoda ITrialRepositoryCommunicationService.DeleteTestTrial	644
274.1 Składnia	644
274.2 Zobacz również	644
275 Metoda ITrialRepositoryCommunicationService.EventData	645
275.1 Składnia	645
275.2 Zobacz również	645
276 Metoda ITrialRepositoryCommunicationService.EventDataForDiscriminantChange	646
276.1 Składnia	646
276.2 Zobacz również	646
277 Metoda ITrialRepositoryCommunicationService.GetCaseIndexingResults	647



277.1 Składnia	647
277.2 Zobacz również	647
278 Metoda ITrialRepositoryCommunicationService.GetDataIntegrationResults	648
278.1 Składnia	648
278.2 Zobacz również	648
279 Metoda ITrialRepositoryCommunicationService.GetDiscriminantsByCode	649
279.1 Składnia	649
279.2 Zobacz również	649
280 Metoda ITrialRepositoryCommunicationService.GetRecordingDataDescription	650
280.1 Składnia	650
280.2 Zobacz również	650
281 Metoda ITrialRepositoryCommunicationService.GetTrial	651
281.1 Składnia	651
281.2 Zobacz również	651
282 Metoda ITrialRepositoryCommunicationService.GetTrialAnnotations	652
282.1 Składnia	652
282.2 Zobacz również	652
283 Metoda ITrialRepositoryCommunicationService.GetTrialAnnotationsHistory	653
283.1 Składnia	653
283.2 Zobacz również	653
284 Metoda ITrialRepositoryCommunicationService.GetTrialCapeStateHistory	654
284.1 Składnia	654
284.2 Zobacz również	654
285 Metoda ITrialRepositoryCommunicationService.GetTrialDocument	655
285.1 Składnia	655
285.2 Zobacz również	655
286 Metoda ITrialRepositoryCommunicationService.GetTrialDocumentsVO	656
286.1 Składnia	656
286.2 Zobacz również	656
287 Metoda ITrialRepositoryCommunicationService.GetTrialFiles	657

287.1 Składnia	657
287.2 Zobacz również	657
288 Metoda <code>ITrialRepositoryCommunicationService.GetTrialProtocol</code>	658
288.1 Składnia	658
288.2 Zobacz również	658
289 Metoda <code>ITrialRepositoryCommunicationService.GetTrialProtocolHistory</code>	659
289.1 Składnia	659
289.2 Zobacz również	659
290 Metoda <code>ITrialRepositoryCommunicationService.GetTrials</code>	660
290.1 Składnia	660
290.2 Zobacz również	660
291 Metoda <code>ITrialRepositoryCommunicationService.GetTrialsForRecording</code>	661
291.1 Składnia	661
291.2 Zobacz również	661
292 Metoda <code>ITrialRepositoryCommunicationService.GetTrialTimeline</code>	662
292.1 Składnia	662
292.2 Zobacz również	662
293 Metoda <code>ITrialRepositoryCommunicationService.IndexTrials</code>	663
293.1 Składnia	663
293.2 Zobacz również	663
294 Metoda <code>ITrialRepositoryCommunicationService.MarkTrialAsMovedToCKIS</code>	664
294.1 Składnia	664
294.2 Zobacz również	664
295 Metoda <code>ITrialRepositoryCommunicationService.ReCloseTrial</code>	665
295.1 Składnia	665
295.2 Zobacz również	665
296 Metoda <code>ITrialRepositoryCommunicationService.ReIndexTrial</code>	666
296.1 Składnia	666
296.2 Zobacz również	666
297 Metoda <code>ITrialRepositoryCommunicationService.RemoveTrialFiles</code>	667



297.1 Składnia	667
297.2 Zobacz również	667
298 Metoda ITrialRepositoryCommunicationService.SetSetCaseCapeStatus	668
298.1 Składnia	668
298.2 Zobacz również	668
299 Metoda ITrialRepositoryCommunicationService.StoreBurnResult	669
299.1 Składnia	669
299.2 Zobacz również	670
300 Metoda ITrialRepositoryCommunicationService.StoreIndexingResult	671
300.1 Składnia	671
300.2 Zobacz również	671
301 Metoda ITrialRepositoryCommunicationService.UpdateDataIntegrationResult	672
301.1 Składnia	672
301.2 Zobacz również	672
302 Metoda ITrialRepositoryCommunicationService.UpdateTrialAnnotations	673
302.1 Składnia	673
302.2 Zobacz również	674
303 Metoda ITrialRepositoryCommunicationService.UpdateTrialData	675
303.1 Składnia	675
303.2 Zobacz również	675
304 Przestrzeń nazw ReCourt.Services.Communication.Update.Base	676
304.1 Klasy	676
304.2 Interfejsy	676
305 Klasa FileTransferInfo	677
305.1 Hierarchia	677
305.2 Składnia	677
305.3 Konstruktory	677
305.4 Właściwości	677
305.5 Zobacz również	677
306 Konstruktor FileTransferInfo	678



306.1 Składnia	678
306.2 Zobacz również	678
307 Właściwości FileTransferInfo.FileTransferInfo	679
307.1 Właściwości	679
307.2 Zobacz również	679
308 Właściwość FileTransferInfo.Checksum	680
308.1 Składnia	680
308.2 Zobacz również	680
309 Właściwość FileTransferInfo.ChunkSize	681
309.1 Składnia	681
309.2 Zobacz również	681
310 Właściwość FileTransferInfo.CreateFile	682
310.1 Składnia	682
310.2 Zobacz również	682
311 Właściwość FileTransferInfo.FileFinished	683
311.1 Składnia	683
311.2 Zobacz również	683
312 Właściwość FileTransferInfo.Name	684
312.1 Składnia	684
312.2 Zobacz również	684
313 Właściwość FileTransferInfo.RunParam	685
313.1 Składnia	685
313.2 Zobacz również	685
314 Interfejs IMachineCatalogueService	686
314.1 Składnia	686
314.2 Metody	686
314.3 Zobacz również	686
315 Metody IMachineCatalogueService.IMachineCatalogueService	687
315.1 Metody	687
315.2 Zobacz również	687



316 Metoda IMachineCatalogueService.CheckMachineRegistration	688
316.1 Składnia	688
316.2 Zobacz również	689
317 Metoda IMachineCatalogueService.ForceUpdateMachine	690
317.1 Składnia	690
317.2 Zobacz również	690
318 Metoda IMachineCatalogueService.GetAllMachines.....	691
318.1 Składnia	691
318.2 Zobacz również	691
319 Metoda IMachineCatalogueService.GetAllMachinesForType	692
319.1 Składnia	692
319.2 Zobacz również	692
320 Metoda IMachineCatalogueService.GetMachine	693
320.1 Składnia	693
320.2 Zobacz również	693
321 Metoda IMachineCatalogueService.GetMachines.....	694
321.1 Przeciążenia.....	694
321.2 Zobacz również	694
322 Metoda IMachineCatalogueService.GetMachines (Int32, Int32, SortCriteria, ICollection(String)).....	695
322.1 Składnia	695
322.2 Zobacz również	695
323 Metoda IMachineCatalogueService.GetMachines (Int32, Int32, SortCriteria, String).....	697
323.1 Składnia	697
323.2 Zobacz również	697
324 Metoda IMachineCatalogueService.GetMachinesWithTypePrefix.....	699
324.1 Przeciążenia.....	699
324.2 Zobacz również	699
325 Metoda IMachineCatalogueService.GetMachinesWithTypePrefix (String).....	700
325.1 Składnia	700
325.2 Zobacz również	700



326 Metoda IMachineCatalogueService.GetMachinesWithTypePrefix (Int32, Int32, SortCriteria, String)	701
326.1 Składnia	701
326.2 Zobacz również	701
327 Metoda IMachineCatalogueService.RegisterMachine	703
327.1 Składnia	703
327.2 Zobacz również	703
328 Interfejs IReCourtUpdateCommunicationService	704
328.1 Składnia	704
328.2 Metody	704
328.3 Zobacz również	704
329 Metody IReCourtUpdateCommunicationService.IReCourtUpdateCommunicationService	705
329.1 Metody	705
329.2 Zobacz również	705
330 Metoda IReCourtUpdateCommunicationService.StatusRequest	706
330.1 Składnia	706
330.2 Zobacz również	706
331 Metoda IReCourtUpdateCommunicationService.UpdateRequest	707
331.1 Składnia	707
331.2 Zobacz również	707
332 Interfejs IUpdateCommunicationService	708
332.1 Składnia	708
332.2 Metody	708
332.3 Zobacz również	710
333 Metody IUpdateCommunicationService.IUpdateCommunicationService	711
333.1 Metody	711
333.2 Zobacz również	713
334 Metoda IUpdateCommunicationService.AddConfigGroup	714
334.1 Składnia	714
334.2 Zobacz również	714
335 Metoda IUpdateCommunicationService.AddUpdateRequest	715



335.1 Składnia	715
335.2 Zobacz również	715
336 Metoda IUpdateCommunicationService.AddUpdateRequestForAll.....	716
336.1 Składnia	716
336.2 Zobacz również	716
337 Metoda IUpdateCommunicationService.AddUpdateRequestForGroup.....	717
337.1 Składnia	717
337.2 Zobacz również	717
338 Metoda IUpdateCommunicationService.CancelUpdateReCourtRequest.....	718
338.1 Składnia	718
338.2 Zobacz również	718
339 Metoda IUpdateCommunicationService.CheckUpdateReCourtRequest.....	719
339.1 Składnia	719
339.2 Zobacz również	719
340 Metoda IUpdateCommunicationService.GetAgentConfigGroup.....	720
340.1 Składnia	720
340.2 Zobacz również	720
341 Metoda IUpdateCommunicationService.GetAgentInstance.....	721
341.1 Składnia	721
341.2 Zobacz również	721
342 Metoda IUpdateCommunicationService.GetAgentInstanceCurrentModules.....	722
342.1 Składnia	722
342.2 Zobacz również	722
343 Metoda IUpdateCommunicationService.GetAgentInstanceGroupList.....	723
343.1 Składnia	723
343.2 Zobacz również	723
344 Metoda IUpdateCommunicationService.GetAudioDevicesConfig.....	724
344.1 Składnia	724
344.2 Zobacz również	724
345 Metoda IUpdateCommunicationService.GetCodecsConfig.....	725

345.1 Składnia	725
345.2 Zobacz również	725
346 Metoda IUpdateCommunicationService.GetCommandMappingsConfig	726
346.1 Składnia	726
346.2 Zobacz również	726
347 Metoda IUpdateCommunicationService.GetConfigGroupList	727
347.1 Składnia	727
347.2 Zobacz również	727
348 Metoda IUpdateCommunicationService.GetConfigGroupListPager	728
348.1 Składnia	728
348.2 Zobacz również	728
349 Metoda IUpdateCommunicationService.GetConfigListGroup	729
349.1 Składnia	729
349.2 Zobacz również	729
350 Metoda IUpdateCommunicationService.GetCourtApplicationConfig	730
350.1 Składnia	730
350.2 Zobacz również	730
351 Metoda IUpdateCommunicationService.GetCourtDataConfig	731
351.1 Składnia	731
351.2 Zobacz również	731
352 Metoda IUpdateCommunicationService.GetCourtSettingsConfig	732
352.1 Składnia	732
352.2 Zobacz również	732
353 Metoda IUpdateCommunicationService.GetFontConfig	733
353.1 Składnia	733
353.2 Zobacz również	733
354 Metoda IUpdateCommunicationService.GetMediaServerConfig	734
354.1 Składnia	734
354.2 Zobacz również	734
355 Metoda IUpdateCommunicationService.GetModuleItemWithData	735

355.1 Składnia	735
355.2 Zobacz również	735
356 Metoda IUpdateCommunicationService.GetModulesForGroup	736
356.1 Składnia	736
356.2 Zobacz również	736
357 Metoda IUpdateCommunicationService.GetReCourtInstanceList.....	737
357.1 Składnia	737
357.2 Zobacz również	737
358 Metoda IUpdateCommunicationService.GetReCourtInstanceListPager	738
358.1 Składnia	738
358.2 Zobacz również	738
359 Metoda IUpdateCommunicationService.GetReCourtServicesInstanceList	739
359.1 Składnia	739
359.2 Zobacz również	739
360 Metoda IUpdateCommunicationService.GetReCourtServicesInstanceListByGroupId	740
360.1 Składnia	740
360.2 Zobacz również	740
361 Metoda IUpdateCommunicationService.GetReCourtServicesInstanceListPager	741
361.1 Składnia	741
361.2 Zobacz również	741
362 Metoda IUpdateCommunicationService.GetReCourtVersionList	742
362.1 Składnia	742
362.2 Zobacz również	742
363 Metoda IUpdateCommunicationService.GetTvConfig.....	743
363.1 Składnia	743
363.2 Zobacz również	743
364 Metoda IUpdateCommunicationService.GetUpdateHistory	744
364.1 Składnia	744
364.2 Zobacz również	744
365 Metoda IUpdateCommunicationService.GetVideoDevicesConfig.....	745

365.1 Składnia	745
365.2 Zobacz również	745
366 Metoda IUpdateCommunicationService.GetVideoLayoutsConfig	746
366.1 Składnia	746
366.2 Zobacz również	746
367 Metoda IUpdateCommunicationService.LogOnReCourtInstance.....	747
367.1 Składnia	747
367.2 Zobacz również	747
368 Metoda IUpdateCommunicationService.LogOnReCourtServicesInstance	748
368.1 Składnia	748
368.2 Zobacz również	748
369 Metoda IUpdateCommunicationService.RemoveConfigGroup	749
369.1 Składnia	749
369.2 Zobacz również	749
370 Metoda IUpdateCommunicationService.SaveAudioDevicesConfig	750
370.1 Składnia	750
370.2 Zobacz również	750
371 Metoda IUpdateCommunicationService.SaveCodecsConfig	751
371.1 Składnia	751
371.2 Zobacz również	751
372 Metoda IUpdateCommunicationService.SaveCommandMappingsConfig	752
372.1 Składnia	752
372.2 Zobacz również	752
373 Metoda IUpdateCommunicationService.SaveCourtApplicationConfig.....	753
373.1 Składnia	753
373.2 Zobacz również	753
374 Metoda IUpdateCommunicationService.SaveCourtDataConfig	754
374.1 Składnia	754
374.2 Zobacz również	754
375 Metoda IUpdateCommunicationService.SaveCourtSettingsConfig	755

375.1 Składnia	755
375.2 Zobacz również	755
376 Metoda IUpdateCommunicationService.SaveFontConfig.....	756
376.1 Składnia	756
376.2 Zobacz również	756
377 Metoda IUpdateCommunicationService.SaveMediaServerConfig	757
377.1 Składnia	757
377.2 Zobacz również	757
378 Metoda IUpdateCommunicationService.SaveTvConfig	758
378.1 Składnia	758
378.2 Zobacz również	758
379 Metoda IUpdateCommunicationService.SaveVideoDevicesConfig	759
379.1 Składnia	759
379.2 Zobacz również	759
380 Metoda IUpdateCommunicationService.SaveVideoLayoutsConfig	760
380.1 Składnia	760
380.2 Zobacz również	760
381 Metoda IUpdateCommunicationService.UpdateReCourtRequest	761
381.1 Składnia	761
381.2 Zobacz również	761
382 Interfejs IUpdateService	762
382.1 Składnia	762
382.2 Metody.....	762
382.3 Zobacz również	763
383 Metody IUpdateService.IUpdateService	764
383.1 Metody.....	764
383.2 Zobacz również	765
384 Metoda IUpdateService.AddConfigType	766
384.1 Składnia	766
384.2 Zobacz również	766



385 Metoda IUpdateService.ConfigTypes	767
385.1 Składnia	767
385.2 Zobacz również	767
386 Metoda IUpdateService.GetConfigType.....	768
386.1 Składnia	768
386.2 Zobacz również	768
387 Metoda IUpdateService.GetManualUpdatePackagesForMachine	769
387.1 Składnia	769
387.2 Zobacz również	769
388 Metoda IUpdateService.GetUpdatesHistory.....	770
388.1 Składnia	770
388.2 Zobacz również	770
389 Metoda IUpdateService.HasAnyUpdatesForMachine.....	771
389.1 Składnia	771
389.2 Zobacz również	771
390 Metoda IUpdateService.ProductsForConfigType.....	772
390.1 Składnia	772
390.2 Zobacz również	772
391 Metoda IUpdateService.ProductsForOperatingSystem	773
391.1 Składnia	773
391.2 Zobacz również	773
392 Metoda IUpdateService.RegisterDummyPackage.....	774
392.1 Składnia	774
392.2 Zobacz również	774
393 Metoda IUpdateService.RemoveConfigGroup.....	775
393.1 Składnia	775
393.2 Zobacz również	775
394 Metoda IUpdateService.ReportInstallationStatus	776
394.1 Składnia	776
394.2 Zobacz również	776



395 Metoda IUpdateService.RescanDropDirectry	777
395.1 Składnia	777
395.2 Zobacz również	777
396 Metoda IUpdateService.ScheduleAllPackagesInstallOnMachine	778
396.1 Składnia	778
396.2 Zobacz również	778
397 Metoda IUpdateService.SchedulePackageInstallationForConfigType	779
397.1 Przeciężenia.....	779
397.2 Zobacz również	779
398 Metoda IUpdateService.SchedulePackageInstallationForConfigType (String, OperatingSystem, List(Int32))	780
398.1 Składnia	780
398.2 Zobacz również	780
399 Metoda IUpdateService.SchedulePackageInstallationForConfigType (String, OperatingSystem, Int32)	781
399.1 Składnia	781
399.2 Zobacz również	781
400 Metoda IUpdateService.SchedulePackageInstallationOnMachine	782
400.1 Przeciężenia.....	782
400.2 Zobacz również	782
401 Metoda IUpdateService.SchedulePackageInstallationOnMachine (Guid, List(Int32))	783
401.1 Składnia	783
401.2 Zobacz również	783
402 Metoda IUpdateService.SchedulePackageInstallationOnMachine (Guid, Int32)	784
402.1 Składnia	784
402.2 Zobacz również	784
403 Metoda IUpdateService.SetConfigTypeAutoUpdate.....	785
403.1 Składnia	785
403.2 Zobacz również	785
404 Metoda IUpdateService.SetConfigTypeProducts	786
404.1 Składnia	786



404.2 Zobacz również	786
405 Metoda IUpdateService.UpdatesForMachine.....	787
405.1 Składnia	787
405.2 Zobacz również	787
406 Klasa MachineHash.....	788
406.1 Hierarchia.....	788
406.2 Składnia	788
406.3 Konstruktory	788
406.4 Metody.....	788
406.5 Pola.....	788
406.6 Zobacz również	788
407 Konstruktor MachineHash.....	789
407.1 Składnia	789
407.2 Zobacz również	789
408 Metody MachineHash.MachineHash	790
408.1 Metody.....	790
408.2 Zobacz również	790
409 Metoda MachineHash.Calculate	791
409.1 Składnia	791
409.2 Zobacz również	791
410 Metoda MachineHash.HashString.....	792
410.1 Składnia	792
410.2 Zobacz również	792
411 Pola MachineHash.MachineHash.....	793
411.1 Pola.....	793
411.2 Zobacz również	793
412 Pole MachineHash.Empty.....	794
412.1 Składnia	794
412.2 Zobacz również	794
413 Klasa SendFileRequestMessage.....	795



413.1 Hierarchia	795
413.2 Składnia	795
413.3 Konstruktory	795
413.4 Pola.....	795
413.5 Zobacz również	795
414 Konstruktor SendFileRequestMessage.....	796
414.1 Składnia	796
414.2 Zobacz również	796
415 Pola SendFileRequestMessage.SendFileRequestMessage.....	797
415.1 Pola.....	797
415.2 Zobacz również	797
416 Pole SendFileRequestMessage.FileData.....	798
416.1 Składnia	798
416.2 Zobacz również	798
417 Pole SendFileRequestMessage.FileInfo	799
417.1 Składnia	799
417.2 Zobacz również	799
418 Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base	800
418.1 Interfejsy	800
419 Interfejs IUserRepositoryCommunicationService	801
419.1 Składnia	801
419.2 Metody.....	801
419.3 Zobacz również	802
420 Metody IUserRepositoryCommunicationService.IUserRepositoryCommunicationService.....	803
420.1 Metody.....	803
420.2 Zobacz również	804
421 Metoda IUserRepositoryCommunicationService.AddRole	805
421.1 Składnia	805
421.2 Zobacz również	805
422 Metoda IUserRepositoryCommunicationService.AddUser	806

422.1 Składnia	806
422.2 Zobacz również	806
423 Metoda IUserRepositoryCommunicationService.AuthorizeUser	807
423.1 Składnia	807
423.2 Zobacz również	807
424 Metoda IUserRepositoryCommunicationService.EnableUser	808
424.1 Składnia	808
424.2 Zobacz również	808
425 Metoda IUserRepositoryCommunicationService.GetAction	809
425.1 Składnia	809
425.2 Zobacz również	809
426 Metoda IUserRepositoryCommunicationService.GetActions	810
426.1 Składnia	810
426.2 Zobacz również	810
427 Metoda IUserRepositoryCommunicationService.GetActiveDirectoryUserNames	811
427.1 Składnia	811
427.2 Zobacz również	811
428 Metoda IUserRepositoryCommunicationService.GetLastVisitedStatistics	812
428.1 Składnia	812
428.2 Zobacz również	812
429 Metoda IUserRepositoryCommunicationService.GetReportRights	813
429.1 Składnia	813
429.2 Zobacz również	813
430 Metoda IUserRepositoryCommunicationService.GetRole	814
430.1 Składnia	814
430.2 Zobacz również	814
431 Metoda IUserRepositoryCommunicationService.GetRoles	815
431.1 Składnia	815
431.2 Zobacz również	815
432 Metoda IUserRepositoryCommunicationService.GetUserACL	816

432.1 Składnia	816
432.2 Zobacz również	816
433 Metoda IUserRepositoryCommunicationService.GetUserById	817
433.1 Składnia	817
433.2 Zobacz również	817
434 Metoda IUserRepositoryCommunicationService.GetUserByName	818
434.1 Składnia	818
434.2 Zobacz również	818
435 Metoda IUserRepositoryCommunicationService.GetUserNames	819
435.1 Składnia	819
435.2 Zobacz również	819
436 Metoda IUserRepositoryCommunicationService.GetUserPrivilegesVO	820
436.1 Składnia	820
436.2 Zobacz również	820
437 Metoda IUserRepositoryCommunicationService.GetUserRoles	821
437.1 Składnia	821
437.2 Zobacz również	821
438 Metoda IUserRepositoryCommunicationService.GetUsers	822
438.1 Składnia	822
438.2 Zobacz również	822
439 Metoda IUserRepositoryCommunicationService.GetUsersForAction	823
439.1 Składnia	823
439.2 Zobacz również	823
440 Metoda IUserRepositoryCommunicationService.IsAuthorizedUser	824
440.1 Składnia	824
440.2 Zobacz również	824
441 Metoda IUserRepositoryCommunicationService.IsUserInDomain	825
441.1 Składnia	825
441.2 Zobacz również	825
442 Metoda IUserRepositoryCommunicationService.RequestNewDiscriminant	826



442.1 Składnia	826
442.2 Zobacz również	826
443 Metoda IUserRepositoryCommunicationService.RequestNewRole	827
443.1 Składnia	827
443.2 Zobacz również	827
444 Metoda IUserRepositoryCommunicationService.RespondToDiscriminantRequest	828
444.1 Składnia	828
444.2 Zobacz również	828
445 Metoda IUserRepositoryCommunicationService.RespondToRoleRequest	829
445.1 Składnia	829
445.2 Zobacz również	829
446 Metoda IUserRepositoryCommunicationService.UpdateRole	830
446.1 Składnia	830
446.2 Zobacz również	830
447 Metoda IUserRepositoryCommunicationService.UpdateUser	831
447.1 Składnia	831
447.2 Zobacz również	831
448 Metoda IUserRepositoryCommunicationService.UpdateUserLastVisited	832
448.1 Składnia	832
448.2 Zobacz również	832
449 Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base	833
449.1 Interfejsy	833
450 Interfejs IDocumentsReadingRoomCommunicationService	834
450.1 Składnia	834
450.2 Metody	834
450.3 Zobacz również	834
451 Metody	
IDocumentsReadingRoomCommunicationService.IDocumentsReadingRoomCommunicationService	835
451.1 Metody	835
451.2 Zobacz również	835



452 Metoda IDocumentsReadingRoomCommunicationService.AcceptWorkflow	836
452.1 Składnia	836
452.2 Zobacz również	836
453 Metoda IDocumentsReadingRoomCommunicationService.AcceptWorkflowISO	837
453.1 Składnia	837
453.2 Zobacz również	837
454 Metoda IDocumentsReadingRoomCommunicationService.CloseWorkflow	838
454.1 Składnia	838
454.2 Zobacz również	838
455 Metoda IDocumentsReadingRoomCommunicationService.CloseWorkflowISO	839
455.1 Składnia	839
455.2 Zobacz również	839
456 Metoda IDocumentsReadingRoomCommunicationService.CreateWorkflow	840
456.1 Składnia	840
456.2 Zobacz również	840
457 Metoda IDocumentsReadingRoomCommunicationService.CreateWorkflowISO	841
457.1 Składnia	841
457.2 Zobacz również	841
458 Metoda IDocumentsReadingRoomCommunicationService.RejectWorkflow	842
458.1 Składnia	842
458.2 Zobacz również	842
459 Metoda IDocumentsReadingRoomCommunicationService.RejectWorkflowISO	843
459.1 Składnia	843
459.2 Zobacz również	843
460 Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base	844
460.1 Klasy	844
460.2 Interfejsy	844
461 Klasa FileTransferInfo	845
461.1 Hierarchia	845
461.2 Składnia	845



461.3 Konstruktory	845
461.4 Pola.....	845
461.5 Zobacz również	845
462 Konstruktor FileTransferInfo	846
462.1 Składnia	846
462.2 Zobacz również	846
463 Pola FileTransferInfo.FileTransferInfo	847
463.1 Pola.....	847
463.2 Zobacz również	847
464 Pole FileTransferInfo.Document.....	848
464.1 Składnia	848
464.2 Zobacz również	848
465 Pole FileTransferInfo.FileName	849
465.1 Składnia	849
465.2 Zobacz również	849
466 Pole FileTransferInfo.User	850
466.1 Składnia	850
466.2 Zobacz również	850
467 Interfejs IDocumentsTranscriptionCommunicationService	851
467.1 Składnia	851
467.2 Metody.....	851
467.3 Zobacz również	851
468 Metody	
IDocumentsTranscriptionCommunicationService.IDocumentsTranscriptionCommunicationService	852
468.1 Metody.....	852
468.2 Zobacz również	852
469 Metoda IDocumentsTranscriptionCommunicationService.AcceptImportRequest	853
469.1 Składnia	853
469.2 Zobacz również	853
470 Metoda IDocumentsTranscriptionCommunicationService.AcceptWorkflow	854

470.1 Składnia	854
470.2 Zobacz również	854
471 Metoda IDocumentsTranscriptionCommunicationService.CreateWorkflow	855
471.1 Składnia	855
471.2 Zobacz również	855
472 Metoda IDocumentsTranscriptionCommunicationService.RejectImportRequest	856
472.1 Składnia	856
472.2 Zobacz również	856
473 Metoda IDocumentsTranscriptionCommunicationService.RejectWorkflow	857
473.1 Składnia	857
473.2 Zobacz również	857
474 Metoda IDocumentsTranscriptionCommunicationService.UploadTranscriptionFile	858
474.1 Składnia	858
474.2 Zobacz również	858
475 Klasa UploadFileRequest	859
475.1 Hierarchia	859
475.2 Składnia	859
475.3 Konstruktory	859
475.4 Pola	859
475.5 Zobacz również	859
476 Konstruktor UploadFileRequest	860
476.1 Składnia	860
476.2 Zobacz również	860
477 Pola UploadFileRequest.UploadFileRequest	861
477.1 Pola	861
477.2 Zobacz również	861
478 Pole UploadFileRequest.File	862
478.1 Składnia	862
478.2 Zobacz również	862
479 Pole UploadFileRequest.FileInfo	863



479.1 Składnia	863
479.2 Zobacz również	863
480 Interfejsy RCS dla agenta CAPE.....	863
481 Interfejsy RCS dla systemu Portali Informacyjnych	868
481.1 Mechanizm tworzenia pliku webm dla Portali Informacyjnych	869
1 Ekstrakcja ścieżek audio w RCS (na potrzeby transkrypcji)	871
V. Interfejsy CRCS dla systemów lub programów zewnętrznych	871
1 Interfejsy zewnętrzne CRCS	871
2 Przestrzeń nazw ReCourt.Services.Central.Communication.CrcsSettings.Base	871
2.1 Interfejsy	871
3 Interfejs ICrcsSettingsCommunicationService.....	872
3.1 Składnia	872
3.2 Metody.....	872
3.3 Zobacz również	872
4 Metody ICrcsSettingsCommunicationService.ICrcsSettingsCommunicationService	873
4.1 Metody.....	873
4.2 Zobacz również	873
5 Metoda ICrcsSettingsCommunicationService.NeededSetting	874
5.1 Składnia	874
5.2 Zobacz również	874
6 Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base	875
6.1 Interfejsy	875
7 Interfejs IDictionaryRepositoryCommunicationService	876
7.1 Składnia	876
7.2 Metody.....	876
7.3 Zobacz również	878
8 Metody	
IDictionaryRepositoryCommunicationService.IDictionaryRepositoryCommunicationService...	879
8.1 Metody.....	879
8.2 Zobacz również	880
9 Metoda IDictionaryRepositoryCommunicationService.AddAppeal	882

9.1	Składnia	882
9.2	Zobacz również	882
10	Metoda IDictionaryRepositoryCommunicationService.AddBenchRole	883
10.1	Składnia	883
10.2	Zobacz również	883
11	Metoda IDictionaryRepositoryCommunicationService.AddCaseType	884
11.1	Składnia	884
11.2	Zobacz również	884
12	Metoda IDictionaryRepositoryCommunicationService.AddCourtDepartments	885
12.1	Składnia	885
12.2	Zobacz również	885
13	Metoda IDictionaryRepositoryCommunicationService.AddCourts.....	886
13.1	Składnia	886
13.2	Zobacz również	886
14	Metoda IDictionaryRepositoryCommunicationService.AddDistrict.....	887
14.1	Składnia	887
14.2	Zobacz również	887
15	Metoda IDictionaryRepositoryCommunicationService.AddParticipantFunction	888
15.1	Składnia	888
15.2	Zobacz również	888
16	Metoda IDictionaryRepositoryCommunicationService.GetAppeals	889
16.1	Składnia	889
16.2	Zobacz również	889
17	Metoda IDictionaryRepositoryCommunicationService.GetAppealsPager	890
17.1	Składnia	890
17.2	Zobacz również	890
18	Metoda IDictionaryRepositoryCommunicationService.GetBenchRoles	891
18.1	Składnia	891
18.2	Zobacz również	891
19	Metoda IDictionaryRepositoryCommunicationService.GetBenchRolesPager	892



19.1	Składnia	892
19.2	Zobacz również	892
20	Metoda IDictionaryRepositoryCommunicationService.GetCaseTypes	893
20.1	Składnia	893
20.2	Zobacz również	893
21	Metoda IDictionaryRepositoryCommunicationService.GetCaseTypesPager	894
21.1	Składnia	894
21.2	Zobacz również	894
22	Metoda IDictionaryRepositoryCommunicationService.GetCourtDepartments	895
22.1	Składnia	895
22.2	Zobacz również	895
23	Metoda IDictionaryRepositoryCommunicationService.GetCourtDepartmentsPager	896
23.1	Składnia	896
23.2	Zobacz również	896
24	Metoda IDictionaryRepositoryCommunicationService.GetCourts	897
24.1	Składnia	897
24.2	Zobacz również	897
25	Metoda IDictionaryRepositoryCommunicationService.GetCourtsPager	898
25.1	Składnia	898
25.2	Zobacz również	898
26	Metoda IDictionaryRepositoryCommunicationService.GetCourtsWithAppealDistrict	899
26.1	Składnia	899
26.2	Zobacz również	899
27	Metoda IDictionaryRepositoryCommunicationService.GetDistricts	900
27.1	Składnia	900
27.2	Zobacz również	900
28	Metoda IDictionaryRepositoryCommunicationService.GetDistrictsPager	901
28.1	Składnia	901
28.2	Zobacz również	901
29	Metoda IDictionaryRepositoryCommunicationService.GetMachineDiscriminants	902

29.1	Składnia	902
29.2	Zobacz również	902
30	Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctions	903
30.1	Składnia	903
30.2	Zobacz również	903
31	Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctionsPager	904
31.1	Składnia	904
31.2	Zobacz również	904
32	Metoda IDictionaryRepositoryCommunicationService.RemoveAppeal	905
32.1	Składnia	905
32.2	Zobacz również	905
33	Metoda IDictionaryRepositoryCommunicationService.RemoveBenchRole	906
33.1	Składnia	906
33.2	Zobacz również	906
34	Metoda IDictionaryRepositoryCommunicationService.RemoveCaseType	907
34.1	Składnia	907
34.2	Zobacz również	907
35	Metoda IDictionaryRepositoryCommunicationService.RemoveDistrict	908
35.1	Składnia	908
35.2	Zobacz również	908
36	Metoda IDictionaryRepositoryCommunicationService.RemoveParticipantFunction	909
36.1	Składnia	909
36.2	Zobacz również	909
37	Metoda IDictionaryRepositoryCommunicationService.UpdateAppeal	910
37.1	Składnia	910
37.2	Zobacz również	910
38	Metoda IDictionaryRepositoryCommunicationService.UpdateBenchRole	911
38.1	Składnia	911
38.2	Zobacz również	911
39	Metoda IDictionaryRepositoryCommunicationService.UpdateCaseType	912

39.1	Składnia	912
39.2	Zobacz również	912
40	Metoda IDictionaryRepositoryCommunicationService.UpdateCourtDepartments	913
40.1	Składnia	913
40.2	Zobacz również	913
41	Metoda IDictionaryRepositoryCommunicationService.UpdateCourts	914
41.1	Składnia	914
41.2	Zobacz również	914
42	Metoda IDictionaryRepositoryCommunicationService.UpdateDistrict	915
42.1	Składnia	915
42.2	Zobacz również	915
43	Metoda IDictionaryRepositoryCommunicationService.UpdateParticipantFunction	916
43.1	Składnia	916
43.2	Zobacz również	916
44	Przestrzeń nazw ReCourt.Services.Central.Communication.Statistics.Base	917
44.1	Interfejsy	917
45	Interfejs IStatisticsCommunicationService	918
45.1	Składnia	918
45.2	Metody	918
45.3	Zobacz również	918
46	Metody IStatisticsCommunicationService.IStatisticsCommunicationService	919
46.1	Metody	919
46.2	Zobacz również	919
47	Metoda IStatisticsCommunicationService.GetCaseStatistics	920
47.1	Składnia	920
47.2	Zobacz również	920
48	Metoda IStatisticsCommunicationService.GetTransferStatistics	921
48.1	Składnia	921
48.2	Zobacz również	921
49	Przestrzeń nazw ReCourt.Services.Central.Communication.TrialRepository.Base	922

49.1	Interfejsy	922
50	Interfejs <code>ITrialRepositoryCommunicationService</code>	923
50.1	Składnia	923
50.2	Metody.....	923
50.3	Zobacz również	923
51	Metody <code>ITrialRepositoryCommunicationService.ITrialRepositoryCommunicationService</code> .	924
51.1	Metody.....	924
51.2	Zobacz również	924
52	Metoda <code>ITrialRepositoryCommunicationService.GetTrial</code>	925
52.1	Składnia	925
52.2	Zobacz również	925
53	Metoda <code>ITrialRepositoryCommunicationService.GetTrials</code>	926
53.1	Składnia	926
53.2	Zobacz również	926
54	Przestrzeń nazw <code>ReCourt.Services.Central.Communication.UserRepository.Base</code>	927
54.1	Interfejsy	927
55	Interfejs <code>IUserRepositoryCommunicationService</code>	928
55.1	Składnia	928
55.2	Metody.....	928
55.3	Zobacz również	929
56	Metody <code>IUserRepositoryCommunicationService.IUserRepositoryCommunicationService</code> .	930
56.1	Metody.....	930
56.2	Zobacz również	931
57	Metoda <code>IUserRepositoryCommunicationService.AddRole</code>	932
57.1	Składnia	932
57.2	Zobacz również	932
58	Metoda <code>IUserRepositoryCommunicationService.AddUser</code>	933
58.1	Składnia	933
58.2	Zobacz również	933
59	Metoda <code>IUserRepositoryCommunicationService.AuthorizeUser</code>	934

59.1	Składnia	934
59.2	Zobacz również	934
60	Metoda IUserRepositoryCommunicationService.EnableUser	935
60.1	Składnia	935
60.2	Zobacz również	935
61	Metoda IUserRepositoryCommunicationService.GetAction.....	936
61.1	Składnia	936
61.2	Zobacz również	936
62	Metoda IUserRepositoryCommunicationService.GetActions	937
62.1	Składnia	937
62.2	Zobacz również	937
63	Metoda IUserRepositoryCommunicationService.GetActiveDirectoryUserNames	938
63.1	Składnia	938
63.2	Zobacz również	938
64	Metoda IUserRepositoryCommunicationService.GetReportRights	939
64.1	Składnia	939
64.2	Zobacz również	939
65	Metoda IUserRepositoryCommunicationService.GetRole	940
65.1	Składnia	940
65.2	Zobacz również	940
66	Metoda IUserRepositoryCommunicationService.GetRoles	941
66.1	Składnia	941
66.2	Zobacz również	941
67	Metoda IUserRepositoryCommunicationService.GetUserACL	942
67.1	Składnia	942
67.2	Zobacz również	942
68	Metoda IUserRepositoryCommunicationService.GetUserById	943
68.1	Składnia	943
68.2	Zobacz również	943
69	Metoda IUserRepositoryCommunicationService.GetUserByName	944



69.1	Składnia	944
69.2	Zobacz również	944
70	Metoda IUserRepositoryCommunicationService.GetUserNames	945
70.1	Składnia	945
70.2	Zobacz również	945
71	Metoda IUserRepositoryCommunicationService.GetUserPrivilegesVO	946
71.1	Składnia	946
71.2	Zobacz również	946
72	Metoda IUserRepositoryCommunicationService.GetUserRoles	947
72.1	Składnia	947
72.2	Zobacz również	947
73	Metoda IUserRepositoryCommunicationService.GetUsers	948
73.1	Składnia	948
73.2	Zobacz również	948
74	Metoda IUserRepositoryCommunicationService.GetUsersForAction	949
74.1	Składnia	949
74.2	Zobacz również	949
75	Metoda IUserRepositoryCommunicationService.IsAuthorizedUser	950
75.1	Składnia	950
75.2	Zobacz również	950
76	Metoda IUserRepositoryCommunicationService.IsUserInDomain	951
76.1	Składnia	951
76.2	Zobacz również	951
77	Metoda IUserRepositoryCommunicationService.UpdateRole	952
77.1	Składnia	952
77.2	Zobacz również	952
78	Metoda IUserRepositoryCommunicationService.UpdateUser	953
78.1	Składnia	953
78.2	Zobacz również	953
79	Metoda IUserRepositoryCommunicationService.UpdateUserLastVisited	954

79.1	Składnia	954
79.2	Zobacz również	954
80	Przestrzeń nazw ReCourt.Services.Communication.Registration.Base	955
80.1	Interfejsy	955
81	Interfejs IRegistrationCommunicationService.....	956
81.1	Składnia	956
81.2	Metody.....	956
81.3	Zobacz również	956
82	Metody IRegistrationCommunicationService.IRegistrationCommunicationService	957
82.1	Metody.....	957
82.2	Zobacz również	957
83	Metoda IRegistrationCommunicationService.GetMachineDiscriminantStatus.....	958
83.1	Składnia	958
83.2	Zobacz również	958
84	Metoda IRegistrationCommunicationService.GetMachines.....	959
84.1	Składnia	959
84.2	Zobacz również	959
85	Metoda IRegistrationCommunicationService.GetRcsGuid	960
85.1	Składnia	960
85.2	Zobacz również	960
86	Metoda IRegistrationCommunicationService.PingRcsDictionaryCorrectness	961
86.1	Składnia	961
86.2	Zobacz również	961
87	Metoda IRegistrationCommunicationService.SetMachineDictionaryCorrectness	962
87.1	Składnia	962
87.2	Zobacz również	962
88	Metoda IRegistrationCommunicationService.SetMachineRegistrationStatus	963
88.1	Składnia	963
88.2	Zobacz również	963
89	Metoda IRegistrationCommunicationService.ValidateDiscriminatDictionary	964



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



89.1 Składnia	964
89.2 Zobacz również	964
90 Interfejs CRCS dla agenta CAPE	964

I. Wstęp

Interfejsy systemu ReCourt można sklasyfikować w dwóch kategoriach:

- Interfejsy wewnętrzne – wykorzystywane do komunikacji między modułami systemu ReCourt lub w między serwerem a klientem w obrębie systemu ReCourt
- Interfejsy zewnętrzne – umożliwiające wymianę danych i komunikację z systemami zewnętrznymi względem systemu ReCourt takimi jak archiwum e-protokołu lub sprzętowa jednostka rejestrująca

W niniejszym dokumencie opisana jest grupa druga interfejsów, specyfikacja grupy pierwszej zamieszczona została w Załączniku nr 1 do dokumentacji systemowej.

Opisane w dokumencie interfejsy zewnętrzne opisują

II. Interfejsy modułu REcourt 2

Dokumentacja API dostarczanego przez moduł AVUnit

Moduł AVUnit obsługuje sprzęt dostarczony na potrzeby rejestracji, zarządza nim oraz umożliwia rejestrację do pliku. Dodatkowo, moduł posiada możliwość strumieniowania na żywo podglądu rejestrowanego materiału, dzięki współpracy z modułem RemoteView.

Moduł posiada także obsługę dedykowanego sprzętu, takiego jak karty akwizycji audio, wzmacniacz audio czy odtwarzacz monitora wielkoformatowego.

Biblioteka jest wymagana przez projekt MediaServer. Wymaga on następujących plików:

- na systemie Windows wymagany jest 64 bitowy plik DLL o nazwie AVUnit.dll
- na systemie Linux wymagany jest 64 bitowy plik SO o nazwie libAVUnit.so

1 Dokumentacja grup

1.1 Wartości globalne

Wartości globalne wykorzystywane w module.

1.1.1 Definicje

- `#define RE_API`
*Makro eksportujące funkcje API w bibliotece *.dll/*.so.*

1.1.2 Zmienne

- `const uint32_t AVUNIT_API_VERSION = 11`
Zmienna określająca wersję API.

1.1.3 Opis szczegółowy

Wartości globalne wykorzystywane w module.

1.1.4 Dokumentacja zmiennych

1.1.4.1 AVUNIT_API_VERSION = 11

Zmienna określająca wersję API.

Wersję tę należy porównać z wartością zwracaną przez funkcję [avunit_api_version\(\)](#). Jeśli wartości te różnią się, nie należy używać tego modułu.

Zobacz również:

[avunit_api_version](#)

Definicja w linii [219](#) pliku [AVUnit.h](#).

1.2 Typy

Typy wykorzystywane w module.

1.2.1 Struktury danych

- struct [AVUnitRect](#)
Struktura, która pozwala na ustawienie położenia oraz rozmiaru klatek wideo w mikserze.
- struct [AVUnitCrop](#)
Struktura, która pozwala na ustawienie kadrowania klatki źródłowej w mikserze wideo.
- struct [AVUnitParameterList](#)
Struktura, która opisuje parametr typu lista.
- struct [AVUnitParameterInteger](#)
Struktura, która opisuje parametr typu liczbowego.

1.2.2 Definicje typów

- typedef void * [AVUnitHandle](#)
Uchwyt do głównego obiektu AVUnit.
- typedef void * [AVUnitCallbackStateHandle](#)
Uchwyt do obiektu umożliwiającego ustawienie callbacków dla instancji AVUnit.
- typedef void * [EncoderControlHandle](#)
Uchwyt do kontroli procesu enkodowania.
- typedef void * [DecoderControlHandle](#)
Uchwyt do kontroli procesu dekodowania.
- typedef void * [MediaProcessingControlHandle](#)
Uchwyt do obiektu umożliwiającego przetwarzanie plików nagrań
- typedef void * [AudioDriverControlHandle](#)
Uchwyt do kontroli oraz odczytu stanu przetwarzania danych audio.
- typedef void * [VideoDriverControlHandle](#)
Uchwyt do kontroli oraz odczytu stanu przetwarzania danych wideo.
- typedef void * [FreeSpaceControlHandle](#)
Uchwyt do modułu umożliwiającego uzyskanie informacji o dostępnym miejscu na dysku.
- typedef void * [TVControlHandle](#)
Uchwyt umożliwiający sterowanie zewnętrznym odtwarzaczem.
- typedef void * [SpeechSynthesisControlHandle](#)
Uchwyt umożliwiający sterowanie syntezą mowy.
- typedef void * [ExternalDevicesHandle](#)



Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi.

- typedef void * [PTZControlHandle](#)
Uchwyt umożliwiający sterowanie kamerami poprzez interfejs PTZ.
- typedef void * [CarecMicControlHandle](#)
Uchwyt umożliwiający pobranie dodatkowych informacji z karty akwizycji danych audio.
- typedef void * [AmplifierControlHandle](#)
Uchwyt umożliwiający sterowanie wzmacniaczem dźwięku.
- typedef void * [UPSControlHandle](#)
Uchwyt umożliwiający sterowanie zasilaczem awaryjnym UPS.
- typedef void * [NTPClientHandle](#)
Uchwyt umożliwiający sterowanie klientem NTP.
- typedef void(* [StateChangedCallback](#)) (int32_t state)
Callback, który jest wołany podczas każdej zmiany stanu.
- typedef void(* [ProgressCallback](#)) (float progress)
Callback, który jest wykorzystywany do informowania o postępie procesu długotrwałego.
- typedef void(* [CarecMicrophoneStateChanged](#)) (const char *code, int32_t connected)
Callback, który jest wołany przy zmianie stanu mikrofonu podłączonego do karty akwizycji dźwięku.
- typedef void(* [VideoSourceStateChanged](#)) (const char *code, int32_t connected)
Callback, który jest wołany przy zmianie stanu źródła wideo.
- typedef void(* [DeviceConnectedChanged](#)) (int32_t connected)
Callback, który jest wołany przy zmianie stanu podłączenia do wzmacniacza dźwięku lub do zasilacza awaryjnego.

1.2.3 Wyliczenia

- enum [AVUnitStatus](#) { [AVUNIT_SUCCESS](#) = 0, [AVUNIT_ERROR_INITIALIZATION_FAILED](#) = -1, [AVUNIT_ERROR_NOT_INITIALIZED](#) = -2, [AVUNIT_ERROR_INTERNAL](#) = -3, [AVUNIT_ERROR_OUT_OF_MEMORY](#) = -4, [AVUNIT_ERROR_INVALID_PARAMETER](#) = -5, [AVUNIT_ERROR_INVALID_HANDLE](#) = -6, [AVUNIT_ERROR_INVALID_STATE](#) = -7, [AVUNIT_ERROR_OPERATION_FAILED](#) = -8, [AVUNIT_ERROR_INVALID_TYPE](#) = -9, [AVUNIT_ERROR_FAILED_TO_OPEN](#) = -100, [AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) = -200, [AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) = -201, [AVUNIT_ERROR_INVALID_AUDIO_MIXER_INDEX](#) = -202, [AVUNIT_ERROR_INVALID_AUDIO_MIXER_CODE](#) = -203, [AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) = -300, [AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) = -301, [AVUNIT_ERROR_INVALID_VIDEO_MIXER_INDEX](#) = -302, [AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) = -303, [AVUNIT_ERROR_FAILED_TO_OPEN_FILE](#) = -400, [AVUNIT_ERROR_INVALID_AUDIO_CODEC](#) = -401, [AVUNIT_ERROR_INVALID_VIDEO_CODEC](#) = -402, [AVUNIT_ERROR_SERIALIZATION_FAILED](#) = -403, [AVUNIT_ERROR_DESERIALIZATION_FAILED](#) = -404, [AVUNIT_ERROR_INVALID_CONTAINER](#) = -405, [AVUNIT_ERROR_INVALID_STREAM_INDEX](#) = -600, [AVUNIT_ERROR_INVALID_STREAM_TYPE](#)



= -601, [AVUNIT_ERROR_PROCESSING_CANCELED](#) = -602, [AVUNIT_ERROR_INVALID_LICENSE](#) = -1000, [AVUNIT_WARNING_NO_SUCH_CONTROL](#) = -10000 }Enumeracja możliwych wartości zwracanych z funkcji API.

- enum [AVUnitAmplifierSource](#) { [AVUNIT_AMPLIFIER_SOURCE_COMMON](#) = 0, [AVUNIT_AMPLIFIER_SOURCE_DEDICATED](#) = 1 }Wartości dla źródła dźwięku we wzmacniaczu audio.
- enum [AVUnitUPSPowerSource](#) { [AVUNIT_UPS_POWER_SOURCE_MAINS](#) = 0, [AVUNIT_UPS_POWER_SOURCE_BATTERY](#) = 1, [AVUNIT_UPS_POWER_SOURCE_BATTERY_LOW](#) = 2 }Wartości dla źródła zasilania w zasilaczu awaryjnym UPS.
- enum [AVUnitParameterType](#) { [AVUNIT_PARAMETER_TYPE_BOOL](#) = 0, [AVUNIT_PARAMETER_TYPE_INTEGER](#) = 1, [AVUNIT_PARAMETER_TYPE_LIST](#) = 2 }Wartości dla typu parametrów obsługiwanych przez urządzenia.
- enum [AVUnitEncoderState](#) { [AVUNIT_ENCODER_STATE_CLOSED](#) = 0, [AVUNIT_ENCODER_STATE_OPENED](#) = 1, [AVUNIT_ENCODER_STATE_RECORDING](#) = 2, [AVUNIT_ENCODER_STATE_PAUSED](#) = 3 }Stan enkodera.
- enum [AVUnitDecoderState](#) { [AVUNIT_DECODER_STATE_CLOSED](#) = 0, [AVUNIT_DECODER_STATE_PLAYING](#) = 1, [AVUNIT_DECODER_STATE_RECORDING](#) = 2, [AVUNIT_DECODER_STATE_PAUSED](#) = 3 }Stan dekodera.

1.2.4 Opis szczegółowy

Typy wykorzystywane w module.

1.2.5 Dokumentacja definicji typów

1.2.5.1 typedef void* [TVControlHandle](#)

Uchwyt umożliwiający sterowanie zewnętrznym odtwarzaczem.

Odtwarzacz jest podłączony do monitora wielkoformatowego i umożliwia wyświetlanie na żywo podglądu rejestrowanego obrazu i dźwięku.

Definicja w linii [579](#) pliku [AVUnit.h](#).

1.2.5.2 typedef void(* StateChangedCallback)(int32_t state)

Callback, który jest wołany podczas każdej zmiany stanu.

Callback jest wykorzystywany między innymi przez proces enkodera oraz dekodera.

Parametry:

<i>state</i>	aktualny stan
--------------	---------------

Definicja w linii [622](#) pliku [AVUnit.h](#).

1.2.5.3 typedef void(* ProgressCallback)(float progress)

Callback, który jest wykorzystywany do informowania o postępie procesu długotrwałego.

Parametry:

<i>progress</i>	aktualny postęp, wyrażony jako liczba z przedziału [0.0, 1.0]
-----------------	---

Definicja w linii [628](#) pliku [AVUnit.h](#).

1.2.5.4 `typedef void(* CarecMicrophoneStateChanged)(const char *code, int32_t connected)`

Callback, który jest wołany przy zmianie stanu mikrofonu podłączonego do karty akwizycji dźwięku.

Parametry:

<i>code</i>	kod kanału audio, którego mikrofon zmienił stan
<i>connected</i>	wartość logiczna (true = 1, false = 0) informująca, czy mikrofon jest podłączony

Definicja w linii [635](#) pliku [AVUnit.h](#).

1.2.5.5 `typedef void(* VideoSourceStateChanged)(const char *code, int32_t connected)`

Callback, który jest wołany przy zmianie stanu źródła wideo.

Parametry:

<i>code</i>	kod klatki wideo, dla której stan podłączenia źródła został zmieniony
<i>connected</i>	wartość logiczna (true = 1, false = 0) informująca, czy źródło jest podłączone

Definicja w linii [642](#) pliku [AVUnit.h](#).

1.2.5.6 `typedef void(* DeviceConnectedChanged)(int32_t connected)`

Callback, który jest wołany przy zmianie stanu podłączenia do wzmacniacza dźwięku lub do zasilacza awaryjnego.

Parametry:

<i>connected</i>	wartość logiczna (true = 1, false = 0) informująca, czy nawiązano lub utracono połączenie z urządzeniem
------------------	---

Definicja w linii [649](#) pliku [AVUnit.h](#).

1.2.6 Dokumentacja typów wyliczanych

1.2.6.1 `enum AVUnitStatus`

Enumeracja możliwych wartości zwracanych z funkcji API.

Wartości błędów zawsze są mniejsze od 0, dzięki czemu możliwe jest odróżnienie wystąpienia błędu od poprawy wartości. W opisie każdej z funkcji jest dostępna lista błędów możliwych do wystąpienia.

Wartości wyliczeń

AVUNIT_SUCCESS Funkcja zakończyła się poprawnie.

Wartość zwracana jest, jeśli funkcja nie zwraca innej wartości dodatniej

AVUNIT_ERROR_INITIALIZATION_FAILED Inicjalizacja biblioteki nie powiodła się

AVUNIT_ERROR_NOT_INITIALIZED Próbowano użyć funkcji bez inicjalizacji biblioteki.

AVUNIT_ERROR_INTERNAL Wewnętrzny błąd biblioteki.

AVUNIT_ERROR_OUT_OF_MEMORY Nie można zaalokować wystarczającej ilości pamięci, aby wykonać operację

AVUNIT_ERROR_INVALID_PARAMETER Przekazany parametr jest niepoprawny.

AVUNIT_ERROR_INVALID_HANDLE Przekazany uchwyt do obiektu AVUnit jest niepoprawny.

AVUNIT_ERROR_INVALID_STATE Obiekt znajduje się w stanie uniemożliwiającym wykonanie akcji.

AVUNIT_ERROR_OPERATION_FAILED Żądana operacja zakończyła się niepowodzeniem.

AVUNIT_ERROR_INVALID_TYPE Wykonana akcja jest niepoprawna dla tego typu danych.

AVUNIT_ERROR_FAILED_TO_OPEN Otwarcie obiektu AVUnit nie powiodło się

W celu ustalenia przyczyny awarii, należy przeanalizować logi. Prawdopodobną przyczyną niepowodzenia jest zła konfiguracja lub awaria jednego z urządzeń.

AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX Przekazany indeks kanału audio jest niepoprawny.

AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE Przekazany kod kanału audio jest niepoprawny.

AVUNIT_ERROR_INVALID_AUDIO_MIXER_INDEX Przekazany indeks miksera audio jest niepoprawny.

AVUNIT_ERROR_INVALID_AUDIO_MIXER_CODE Przekazany kod miksera audio jest niepoprawny.

AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX Przekazany indeks klatki video jest niepoprawny.

AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE Przekazany kod klatki video jest niepoprawny.

AVUNIT_ERROR_INVALID_VIDEO_MIXER_INDEX Przekazany indeks miksera video jest niepoprawny.

AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE Przekazany kod miksera wideo jest niepoprawny.

AVUNIT_ERROR_FAILED_TO_OPEN_FILE Otwarcie pliku nie powiodło się.

Przyczyną błędu jest brak pliku o podanej ścieżce lub niewystarczające uprawnienia do jego otwarcia.

AVUNIT_ERROR_INVALID_AUDIO_CODEC Przekazany kodek audio jest niepoprawny.

AVUNIT_ERROR_INVALID_VIDEO_CODEC Przekazany kodek wideo jest niepoprawny.

AVUNIT_ERROR_SERIALIZATION_FAILED Proces serializacji stanu enkodera zakończył się niepowodzeniem.

AVUNIT_ERROR_DESERIALIZATION_FAILED Proces deserializacji stanu enkodera zakończył się niepowodzeniem.

AVUNIT_ERROR_INVALID_CONTAINER Przekazany kontener jest niepoprawny.

AVUNIT_ERROR_INVALID_STREAM_INDEX Przekazany indeks strumienia jest niepoprawny.

AVUNIT_ERROR_INVALID_STREAM_TYPE Wykonana akcja jest niepoprawna dla tego typu strumienia.

Wartość ta zwracana jest wtedy, gdy zostanie wykonana akcja dla strumienia wideo na strumieniu audio (i odwrotnie).

AVUNIT_ERROR_PROCESSING_CANCELED Operacja przetwarzania została przerwana.

AVUNIT_ERROR_INVALID_LICENSE Brak ważnej licencji na produkt.

Aplikacja nie może kontynuować.

AVUNIT_WARNING_NO_SUCH_CONTROL Aplikacja nie udostępnia w tej chwili wymaganej kontrolki.

Brak kontrolki jest spowodowany brakiem konfiguracji i/lub sprzętu dla danego elementu.

Definicja w linii [388](#) pliku [AVUnit.h](#).

1.2.6.2 enum [AVUnitAmplifierSource](#)

Wartości dla źródła dźwięku we wzmacniaczu audio.

Wartości wyliczeń

AVUNIT_AMPLIFIER_SOURCE_COMMON Źródło wspólne wzmacniacza.

Wzmacniacz odtwarza na tym kanale dźwięk ze wspólnego wejścia audio.

AVUNIT_AMPLIFIER_SOURCE_DEDICATED Źródło dedykowane wzmacniacza.

Wzmacniacz odtwarza na tym kanale dźwięk z dedykowanego dla tego kanału wejścia audio.

Definicja w linii [444](#) pliku [AVUnit.h](#).

1.2.6.3 enum [AVUnitUPSPowerSource](#)

Wartości dla źródła zasilania w zasilaczu awaryjnym UPS.

Wartości wyliczeń

AVUNIT_UPS_POWER_SOURCE_MAINS Zasilacz korzysta z zasilania sieciowego.

AVUNIT_UPS_POWER_SOURCE_BATTERY Zasilacz korzysta z zasilania akumulatorowego.

AVUNIT_UPS_POWER_SOURCE_BATTERY_LOW Zasilacz korzysta z zasilania akumulatorowego; bateria jest na wyczerpaniu.

Definicja w linii [462](#) pliku [AVUnit.h](#).

1.2.6.4 enum [AVUnitParameterType](#)

Wartości dla typu parametrów obsługiwanych przez urządzenia.

Wartości wyliczeń

AVUNIT_PARAMETER_TYPE_BOOL Parametr jest typu logicznego i przyjmuje wartości 0 (fałsz) lub 1 (prawda)

AVUNIT_PARAMETER_TYPE_INTEGER Parametr jest typu liczbowego i przyjmuje wartość z podanego przedziału.

AVUNIT_PARAMETER_TYPE_LIST Parametr jest listą wyboru i przyjmuje indeks wybranej opcji.

Definicja w linii [481](#) pliku [AVUnit.h](#).

1.2.6.5 enum [AVUnitEncoderState](#)

Stan enkodera.

Wartości wyliczeń

AVUNIT_ENCODER_STATE_CLOSED Proces zapisu do pliku jest zamknięty.

AVUNIT_ENCODER_STATE_OPENED Plik został otwarty do zapisu.

AVUNIT_ENCODER_STATE_RECORDING Zapis do pliku jest aktywny.

AVUNIT_ENCODER_STATE_PAUSED Zapis do pliku jest wstrzymany.

Definicja w linii [503](#) pliku [AVUnit.h](#).

1.2.6.6 enum [AVUnitDecoderState](#)

Stan dekodera.

Wartości wyliczeń

AVUNIT_DECODER_STATE_CLOSED Nie jest odtwarzany żaden plik.

AVUNIT_DECODER_STATE_PLAYING Plik jest odtwarzany.

AVUNIT_DECODER_STATE_RECORDING Odtwarzany plik jest jednocześnie rejestrowany.

AVUNIT_DECODER_STATE_PAUSED Odtwarzanie pliku jest zatrzymane.

Definicja w linii [526](#) pliku [AVUnit.h](#).

1.3 Funkcje inicjalizacyjne

Podstawowe funkcje służące do inicjalizacji modułu oraz obiektu AVUnit.

1.3.1 Funkcje

- [RE API](#) void [avunit_version](#) (uint32_t *major, uint32_t *minor, uint32_t *patch, uint32_t *build)
Zwrócenie wersji AVUnit.
- [RE API](#) uint32_t [avunit_api_version](#) ()
Zwrócenie aktualnej wersji API, używanej przez moduł.
- [RE API](#) int32_t [avunit_module_initialize](#) (const char *data_dir)
Zainicjalizowanie biblioteki.
- [RE API](#) int32_t [avunit_module_shutdown](#) ()
Zamknięcie modułu AVUnit.
- [RE API](#) int32_t [avunit_machine_serial_number](#) (const char **serial_number)
Pobranie numeru seryjnego maszyny, na którym działa AVUnit.
- [RE API](#) int32_t [avunit_create](#) (AVUnitHandle *handle)
Utworzenie głównego obiektu AVUnit.
- [RE API](#) int32_t [avunit_destroy](#) (AVUnitHandle handle)
Zniszczenie obiektu utworzonego przez funkcję [avunit_create\(\)](#).
- [RE API](#) int32_t [avunit_allocate_callback_state](#) (AVUnitCallbackStateHandle *handle)
Utworzenie obiektu ze stanem wywołań zwrotnych.
- [RE API](#) int32_t [avunit_free_callback_state](#) (AVUnitCallbackStateHandle handle)
Zniszczenie obiektu ze stanem wywołań zwrotnych.
- [RE API](#) int32_t [avunit_open](#) (AVUnitHandle handle)
Otwarcie oraz inicjalizacja obiektu AVUnit.
- [RE API](#) int32_t [avunit_close](#) (AVUnitHandle handle)
Zamknięcie obiektu AVUnit.
- [RE API](#) int32_t [avunit_get_encoder_control](#) (AVUnitHandle handle, [EncoderControlHandle](#) *encoder_handle)
Pobranie uchwytu do kontroli procesu enkodowania.



- [RE_API int32_t avunit_get_decoder_control](#) ([AVUnitHandle](#) handle, [DecoderControlHandle](#) *decoder_handle)
Pobranie uchwytu do kontroli procesu dekodowania.
- [RE_API int32_t avunit_get_audio_driver_control](#) ([AVUnitHandle](#) handle, [AudioDriverControlHandle](#) *audio_driver_handle)
Pobranie uchwytu do kontroli przetwarzania danych audio.
- [RE_API int32_t avunit_get_video_driver_control](#) ([AVUnitHandle](#) handle, [VideoDriverControlHandle](#) *video_driver_handle)
Pobranie uchwytu do kontroli przetwarzania danych wideo.
- [RE_API int32_t avunit_get_free_space_control](#) ([AVUnitHandle](#) handle, [FreeSpaceControlHandle](#) *free_space_handle)
Pobranie uchwytu do pobierania informacji o dostępności miejsca na dysku.
- [RE_API int32_t avunit_get_tv_control](#) ([AVUnitHandle](#) handle, [TVControlHandle](#) *tv_handle)
Pobranie uchwytu do kontroli odtwarzacza znajdującego się przy monitorze wielkoformatowym.
- [RE_API int32_t avunit_get_speech_synthesis_control](#) ([AVUnitHandle](#) handle, [SpeechSynthesisControlHandle](#) *speech_synthesis_handle)
Pobranie uchwytu do kontroli syntezy mowy.
- [RE_API int32_t avunit_get_external_devices_control](#) ([AVUnitHandle](#) handle, [ExternalDevicesHandle](#) *external_devices_handle)
Pobranie uchwytu do kontroli urządzeń zewnętrznych.

1.3.2 Opis szczegółowy

Podstawowe funkcje służące do inicjalizacji modułu oraz obiektu AVUnit.

Funkcje te umożliwiają także pobranie uchwytów do kontroli całego procesu rejestracji oraz sprzętu zewnętrznego. Funkcje inicjalizujące i alokujące obiekty mają w grupie odpowiadające im funkcjonalności odpowiedzialne za zamykanie biblioteki i sprzątanie pamięci.

1.3.3 Dokumentacja funkcji

1.3.3.1 [RE_API void avunit_version](#) (uint32_t * major, uint32_t * minor, uint32_t * patch, uint32_t * build)

Zwrócenie wersji AVUnit.

Parametry:

out	<i>major</i>	Główny numer wersji
out	<i>minor</i>	Mniejszy numer wersji
out	<i>patch</i>	Numer poprawkowy
out	<i>build</i>	Numer kompilacji

1.3.3.2 [RE_API uint32_t avunit_api_version \(\)](#)

Zwrócenie aktualnej wersji API, używanej przez moduł.

Funkcja pozwala na wczesne wykrycie zmian w API i odmowę działania, jeśli aplikacja obsługuje API w wersji innej niż moduł.

Zwraca:

Aktualna wartość stałej [AVUNIT_API_VERSION](#) (w chwili kompilacji)

Zobacz również:

[AVUNIT_API_VERSION](#)

1.3.3.3 [RE_API int32_t avunit_module_initialize \(const char * data_dir\)](#)

Zainicjalizowanie biblioteki.

Parametry:

in	<i>data_dir</i>	Pełna ścieżka do katalogu, w którym zostaną zapisane dane aplikacji. Do danych aplikacji należą między innymi logi oraz ewentualne zrzuty pamięci. Przekazanie wartości NULL spowoduje wykorzystanie domyślnej ścieżki.
----	-----------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - inicjalizacja przebiegła pomyślnie

[AVUNIT_ERROR_INITIALIZATION_FAILED](#) - inicjalizacja nie powiodła się

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka jest niepoprawna

[AVUNIT_ERROR_INVALID_LICENSE](#) - brak ważnej licencji na produkt

Spostrzeżenia:

Funkcja musi być wywołana przed wywołaniem dowolnej innej funkcji. Po zawołaniu [avunit_module_shutdown\(\)](#) funkcja musi być wywołana ponownie. Funkcja może być wywołana więcej niż raz. Aby poprawnie zakończyć pracę należy wywołać funkcję [avunit_module_shutdown\(\)](#) dokładnie tyle samo razy.

Zobacz również:

[avunit_module_shutdown](#)

1.3.3.4 [RE_API int32_t avunit_module_shutdown \(\)](#)

Zamknięcie modułu AVUnit.

Zwraca:

[AVUNIT_SUCCESS](#) - zamknięcie przebiegło pomyślnie

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Spostrzeżenia:

Funkcja musi być wywołana jeden raz dla każdego poprawnego wywołania funkcji [avunit_module_initialize\(\)](#), Przed wywołaniem funkcji należy zniszczyć wszystkie utworzone obiekty.

Zobacz również:

[avunit_module_initialize](#)

1.3.3.5 [RE_API](#) int32_t avunit_machine_serial_number (const char ** serial_number)

Pobranie numeru seryjnego maszyny, na którym działa AVUnit.

Parametry:

out	serial_number	Wskaźnik, do którego zostanie zapisany numer seryjny
-----	---------------	--

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano numer seryjny maszyny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik ma wartość NULL

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Spostrzeżenia:

Pamięć, w której zapisana jest wartość kodu klatki wideo jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_module_shutdown\(\)](#).

1.3.3.6 [RE_API](#) int32_t avunit_create ([AVUnitHandle](#) * handle)

Utworzenie głównego obiektu AVUnit.

Funkcja ta jedynie alokuje obiekt. Aby przeprowadzić inicjalizację należy wywołać funkcję [avunit_open\(\)](#).

Parametry:

out	handle	Wskaźnik na uchwyt do obiektu AVUnit
-----	--------	--------------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie utworzono nowy obiekt AVUnit

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację

Spostrzeżenia:

W przypadku pomyślnego wywołania tej funkcji obiekt należy zniszczyć za pomocą funkcji [avunit_destroy\(\)](#).

Zobacz również:

[avunit_open](#), [avunit_destroy](#)

1.3.3.7 [RE_API int32_t avunit_destroy \(AVUnitHandle handle\)](#)

Zniszczenie obiektu utworzonego przez funkcję [avunit_create\(\)](#).

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit, który zostanie zniszczony.
----	---------------	--

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zniszczono obiekt AVUnit

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Zobacz również:

[avunit_create](#)

1.3.3.8 [RE_API int32_t avunit_allocate_callback_state \(AVUnitCallbackStateHandle * handle\)](#)

Utworzenie obiektu ze stanem wywołań zwrotnych.

Obiekt służy do ustawiania callbacków w obiekcie AVUnit.

Parametry:

out	<i>handle</i>	Wskaźnik na uchwyt do obiektu ze stanem wywołań zwrotnych
-----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie utworzono nowy obiekt ze stanem wywołań zwrotnych

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci, aby wykonać tę operację

Spostrzeżenia:

W przypadku pomyślnego wywołania tej funkcji obiekt należy zniszczyć za pomocą funkcji [avunit_free_callback_state\(\)](#).

Zobacz również:

[avunit_free_callback_state](#)

1.3.3.9 [RE_API int32_t avunit_free_callback_state \(AVUnitCallbackStateHandle handle\)](#)

Zniszczenie obiektu ze stanem wywołań zwrotnych.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu, który zostanie zniszczony.
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zniszczono obiekt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Zobacz również:

[avunit_allocate_callback_state](#)

1.3.3.10 [RE_API](#) int32_t avunit_open ([AVUnitHandle](#) *handle*)

Otwarcie oraz inicjalizacja obiektu AVUnit.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit, który ma zostać zainicjalizowany
----	---------------	--

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zainicjalizowano obiekt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_FAILED_TO_OPEN](#) - inicjalizacja nie powiodła się

1.3.3.11 [RE_API](#) int32_t avunit_close ([AVUnitHandle](#) *handle*)

Zamknięcie obiektu AVUnit.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit, który zostanie zamknięty.
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zamknięto obiekt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

Spostrzeżenia:

Po pomyślnym wykonaniu tej funkcji obiekt może zostać ponownie otwarty za pomocą funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_open](#)

1.3.3.12 [RE API](#) `int32_t avunit_get_encoder_control` ([AVUnitHandle](#) `handle`,
[EncoderControlHandle](#) * `encoder_handle`)

Pobranie uchwytu do kontroli procesu enkodowania.

Parametry:

in	<code>handle</code>	Uchwyt do obiektu AVUnit
out	<code>encoder_handle</code>	Wskaźnik na uchwyt do kontroli enkodera

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwytu przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.13 [RE API](#) `int32_t avunit_get_decoder_control` ([AVUnitHandle](#) `handle`,
[DecoderControlHandle](#) * `decoder_handle`)

Pobranie uchwytu do kontroli procesu dekodowania.

Parametry:

in	<code>handle</code>	Uchwyt do obiektu AVUnit
out	<code>decoder_handle</code>	Wskaźnik na uchwyt do kontroli dekodowania

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwytu przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.14 [RE API](#) `int32_t avunit_get_audio_driver_control` ([AVUnitHandle](#) `handle`,
[AudioDriverControlHandle](#) * `audio_driver_handle`)

Pobranie uchwytu do kontroli przetwarzania danych audio.



Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
out	<i>audio_driver_handle</i>	Wskaźnik na uchwyt do kontroli przetwarzania danych audio

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwyty przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.15 [RE_API](#) int32_t avunit_get_video_driver_control ([AVUnitHandle](#) handle, [VideoDriverControlHandle](#) * video_driver_handle)

Pobranie uchwyty do kontroli przetwarzania danych wideo.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
out	<i>video_driver_handle</i>	Wskaźnik na uchwyt do kontroli przetwarzania danych wideo

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwyty przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.16 [RE_API](#) int32_t avunit_get_free_space_control ([AVUnitHandle](#) handle, [FreeSpaceControlHandle](#) * free_space_handle)

Pobranie uchwyty do pobierania informacji o dostępności miejsca na dysku.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
----	---------------	--------------------------



out	<i>free_space_handle</i>	Wskaźnik na uchwyt do kontroli miejsca na dysku
-----	--------------------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwytu przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.17 [RE_API](#) int32_t avunit_get_tv_control ([AVUnitHandle](#) handle, [TVControlHandle](#) * tv_handle)

Pobranie uchwytu do kontroli odtwarzacza znajdującego się przy monitorze wielkoformatowym.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
out	<i>tv_handle</i>	Wskaźnik na uchwyt do kontroli odtwarzacza monitora wielkoformatowego

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwytu przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.3.3.18 [RE_API](#) int32_t avunit_get_speech_synthesis_control ([AVUnitHandle](#) handle, [SpeechSynthesisControlHandle](#) * speech_synthesis_handle)

Pobranie uchwytu do kontroli syntezy mowy.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
----	---------------	--------------------------



out	<i>speech_synthesis_handle</i>	Wskaźnik na uchwyt do kontroli syntezy mowy
-----	--------------------------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_WARNING_NO_SUCH_CONTROL](#) - synteza mowy nie jest obsługiwana

Spostrzeżenia:

Uchwyt można pobrać dopiero po zwołaniu funkcji [avunit_open\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_close\(\)](#). Próba pobrania uchwyty przed wywołaniem funkcji [avunit_open\(\)](#) zawsze zwróci wartość [AVUNIT_WARNING_NO_SUCH_CONTROL](#).

1.3.3.19 [RE_API](#) int32_t [avunit_get_external_devices_control](#) ([AVUnitHandle](#) handle, [ExternalDevicesHandle](#) * external_devices_handle)

Pobranie uchwyty do kontroli urządzeń zewnętrznych.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu AVUnit
out	<i>external_device_handle</i>	Wskaźnik na uchwyt do kontroli urządzeń zewnętrznych

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zwrócono uchwyt

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu AVUnit jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

Spostrzeżenia:

Uchwyt można pobrać już po zwołaniu funkcji [avunit_create\(\)](#) i jest on ważny do momentu zwołania funkcji [avunit_destroy\(\)](#). Używanie uchwyty przed zwołaniem funkcji [avunit_open\(\)](#) jest zachowaniem niezdefiniowanym.

1.4 Funkcje modułu audio

Funkcje służące do pobierania informacji o konfiguracji modułu audio oraz umożliwiające sterowanie parametrami rejestracji audio.

1.4.1 Funkcje

- [RE_API](#) int32_t [avunit_audio_driver_control_get_channel_count](#) ([AudioDriverControlHandle](#) handle)

Pobranie ilości wszystkich obsługiwanych kanałów audio.



- [RE_API int32_t avunit audio driver control get mixer count \(AudioDriverControlHandle handle\)](#)
Pobranie ilości wszystkich obsługiwanych mikserów audio.
- [RE_API int32_t avunit audio driver control get channel index \(AudioDriverControlHandle handle, const char *channel_code\)](#)
Pobranie indeksu kanału audio na podstawie jego kodu.
- [RE_API int32_t avunit audio driver control get channel code \(AudioDriverControlHandle handle, uint32_t channel_index, const char **channel_code\)](#)
Pobranie kodu kanału audio na podstawie jego indeksu.
- [RE_API int32_t avunit audio driver control get mixer code \(AudioDriverControlHandle handle, uint32_t mixer_index, const char **mixer_code\)](#)
Pobranie kodu miksera audio na podstawie jego indeksu.
- [RE_API int32_t avunit audio driver control get channel name by code \(AudioDriverControlHandle handle, const char *channel_code, const char **channel_name\)](#)
Pobranie opisowej nazwy kanału audio na podstawie jego kodu.
- [RE_API int32_t avunit audio driver control get channel name by index \(AudioDriverControlHandle handle, uint32_t channel_index, const char **channel_name\)](#)
Pobranie opisowej nazwy kanału audio na podstawie jego indeksu.
- [RE_API int32_t avunit audio driver control get mixer channel count \(AudioDriverControlHandle handle, const char *mixer_code\)](#)
Pobranie ilości kanałów audio miksowanych w danym mikserze.
- [RE_API int32_t avunit audio driver control does mixer contain channel \(AudioDriverControlHandle handle, const char *mixer_code, uint32_t channel_index\)](#)
Sprawdzenie, czy dany kanał audio jest miksowany w danym mikserze.
- [RE_API int32_t avunit audio driver control get master volume \(AudioDriverControlHandle handle, float *volume\)](#)
Pobranie wartości głośności dźwięku.
- [RE_API int32_t avunit audio driver control get channel volume \(AudioDriverControlHandle handle, const char *channel_code, float *volume\)](#)
Pobranie wartości głośności dźwięku dla danego kanału audio.
- [RE_API int32_t avunit audio driver control set master volume \(AudioDriverControlHandle handle, float volume\)](#)
Ustawienie wartości głośności dźwięku.
- [RE_API int32_t avunit audio driver control set channel volume \(AudioDriverControlHandle handle, const char *channel_code, float volume\)](#)
Ustawienie wartości głośności dźwięku dla danego kanału audio.
- [RE_API int32_t avunit audio driver control set channels volume \(AudioDriverControlHandle handle, float *volume, uint32_t volume_count\)](#)
Ustawienie wartości głośności dźwięku dla wszystkich kanałów audio jednocześnie.

1.4.2 Opis szczegółowy

Funkcje służące do pobierania informacji o konfiguracji modułu audio oraz umożliwiające sterowanie parametrami rejestracji audio.

Moduł audio odpowiada za obsługę rejestracji dźwięku z dostępnych źródeł, wstępne przetwarzanie oraz miksowanie kanałów audio. Funkcje dostępne w tym module pozwalają na dostęp do informacji o kanałach audio, ich kodach i głośności oraz konfiguracji mikserów.

1.4.3 Dokumentacja funkcji

1.4.3.1 [RE API](#) `int32_t` `avunit_audio_driver_control_get_channel_count` ([AudioDriverControlHandle](#) *handle*)

Pobranie ilości wszystkich obsługiwanych kanałów audio.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość skonfigurowanych kanałów audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.4.3.2 [RE API](#) `int32_t` `avunit_audio_driver_control_get_mixer_count` ([AudioDriverControlHandle](#) *handle*)

Pobranie ilości wszystkich obsługiwanych mikserów audio.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość skonfigurowanych mikserów audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.4.3.3 [RE API](#) `int32_t` `avunit_audio_driver_control_get_channel_index` ([AudioDriverControlHandle](#) *handle*, `const char * channel_code`)

Pobranie indeksu kanału audio na podstawie jego kodu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>channel_code</i>	Kod kanału audio, dla którego chcemy pobrać indeks

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to indeks kanału audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału audio ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - przekazany kod kanału audio nie istnieje

1.4.3.4 [RE API](#) `int32_t` `avunit_audio_driver_control_get_channel_code`
[\(AudioDriverControlHandle](#) `handle`, `uint32_t` `channel_index`, `const char **`
`channel_code)`

Pobranie kodu kanału audio na podstawie jego indeksu.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli przetwarzania danych audio
in	<code>channel_index</code>	Indeks kanału audio, dla którego chcemy pobrać kod
out	<code>channel_code</code>	Wskaźnik, do którego zostanie zapisany kod kanału audio

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano kod kanału audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod kanału ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) - przekazany indeks kanału audio nie istnieje

Spostrzeżenia:

Indeksy kanałów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_audio_driver_control_get_channel_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu kanału audio jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

Zobacz również:

`avunit_audio_driver_control_get_channel_count`

1.4.3.5 [RE API](#) `int32_t` `avunit_audio_driver_control_get_mixer_code`
[\(AudioDriverControlHandle](#) `handle`, `uint32_t` `mixer_index`, `const char **` `mixer_code)`

Pobranie kodu miksera audio na podstawie jego indeksu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>mixer_index</i>	Indeks miksera audio, dla którego chcemy pobrać kod
out	<i>mixer_code</i>	Wskaźnik, do którego zostanie zapisany kod miksera audio

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano kod miksera audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_MIXER_INDEX](#) - przekazany indeks miksera audio nie istnieje

Spostrzeżenia:

Indeksy mikserów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_audio_driver_control_get_mixer_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu miksera audio jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

Zobacz również:

avunit_audio_driver_control_get_mixer_count

1.4.3.6 [RE_API](#) int32_t avunit_audio_driver_control_get_channel_name_by_code
([AudioDriverControlHandle](#) handle, const char * channel_code, const char **
channel_name)

Pobranie opisowej nazwy kanału audio na podstawie jego kodu.

Nazwa ta zostaje wyświetlona użytkownikowi w celu identyfikacji źródła dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>channel_code</i>	Kod kanału audio, dla którego chcemy pobrać nazwę
out	<i>channel_name</i>	Wskaźnik, do którego zostanie zapisana nazwą kanału audio

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano nazwę kanału audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału ma wartość NULL



[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę kanału ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - kanał audio o przekazanym kodzie nie istnieje

Spostrzeżenia:

Pamięć, w której zapisana jest wartość nazwy kanału audio jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

1.4.3.7 [RE API](#) `int32_t avunit_audio_driver_control_get_channel_name_by_index(AudioDriverControlHandle handle, uint32_t channel_index, const char ** channel_name)`

Pobranie opisowej nazwy kanału audio na podstawie jego indeksu.

Nazwa ta zostaje wyświetlona użytkownikowi w celu identyfikacji źródła dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>channel_index</i>	Indeks kanału audio, dla którego chcemy pobrać nazwę
out	<i>channel_name</i>	Wskaźnik, do którego zostanie zapisana nazwa kanału audio

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano nazwę kanału audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę kanału ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) - kanał audio o przekazanym indeksie nie istnieje

Spostrzeżenia:

Indeksy kanałów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_audio_driver_control_get_channel_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu kanału audio jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

Zobacz również:

`avunit_audio_driver_control_get_channel_count`

1.4.3.8 [RE API](#) `int32_t avunit_audio_driver_control_get_mixer_channel_count(AudioDriverControlHandle handle, const char * mixer_code)`

Pobranie ilości kanałów audio miksowanych w danym mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>mixer_code</i>	Kod miksera audio, dla którego chcemy pobrać ilość miksowanych przez niego kanałów

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość kanałów audio miksowanych w danym mikserze. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_MIXER_CODE](#) - mikser audio o przekazanym kodzie nie istnieje

1.4.3.9 [RE_API](#) `int32_t avunit_audio_driver_control_does_mixer_contain_channel(AudioDriverControlHandle handle, const char * mixer_code, uint32_t channel_index)`

Sprawdzenie, czy dany kanał audio jest miksowany w danym mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>mixer_code</i>	Kod miksera audio, dla którego chcemy sprawdzić, czy dany kanał do niego należy
in	<i>channel_index</i>	Indeks kanału audio

Zwraca:

Wartość 1 oznacza, że kanał należy do miksera.

Wartość 0 oznacza, że kanał nie należy do miksera.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) - kanał audio o przekazanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_AUDIO_MIXER_CODE](#) - mikser audio o przekazanym kodzie nie istnieje

Spostrzeżenia:

Indeksy kanałów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_audio_driver_control_get_channel_count\(\)](#) - 1.

Zobacz również:

avunit_audio_driver_control_get_channel_count

1.4.3.10 [RE_API](#) `int32_t` `avunit_audio_driver_control_get_master_volume`
([AudioDriverControlHandle](#) *handle*, float * *volume*)

Pobranie wartości głośności dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
out	<i>volume</i>	Wskaźnik, do którego zostanie zapisana wartość głośności dźwięku

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość głośności audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik ma wartość NULL

Spostrzeżenia:

Poziom głośności jest wartością z przedziału [0.0, 1.0]

Zobacz również:

avunit_audio_driver_control_set_master_volume

1.4.3.11 [RE_API](#) `int32_t` `avunit_audio_driver_control_get_channel_volume`
([AudioDriverControlHandle](#) *handle*, const char * *channel_code*, float * *volume*)

Pobranie wartości głośności dźwięku dla danego kanału audio.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>channel_code</i>	Kod kanału audio, dla którego chcemy pobrać głośność dźwięku
out	<i>volume</i>	Wskaźnik, do którego zostanie zapisana wartość głośności dźwięku

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość głośności audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału audio ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik ma wartość NULL



[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - kanał audio o przekazanym kodzie nie istnieje

Spostrzeżenia:

Poziom głośności jest wartością z przedziału [0.0, 1.0]

Zobacz również:

avunit_audio_driver_control_set_channel_volume

1.4.3.12 [RE_API](#) int32_t avunit_audio_driver_control_set_master_volume
([AudioDriverControlHandle](#) handle, float volume)

Ustawienie wartości głośności dźwięku.

Główna głośność audio jest przemnażana przez głośności poszczególnych kanałów, przez co umożliwia jednoczesne przyciszenie wszystkich rejestrowanych dźwięków.

Parametry:

in	handle	Uchwyt do kontroli przetwarzania danych audio
in	volume	Nowa wartość dla głównej głośności dźwięku

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość głośności audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

Spostrzeżenia:

Poziom głośności jest wartością z przedziału [0.0, 1.0]

1.4.3.13 [RE_API](#) int32_t avunit_audio_driver_control_set_channel_volume
([AudioDriverControlHandle](#) handle, const char * channel_code, float volume)

Ustawienie wartości głośności dźwięku dla danego kanału audio.

Głośność kanału audio pozwala na obniżenie poziomu dźwięku rejestrowanego na jednym kanale audio.

Parametry:

in	handle	Uchwyt do kontroli przetwarzania danych audio
in	channel_code	Kod kanału audio, dla którego chcemy zmienić głośność dźwięku
in	volume	Nowa wartość głośności dźwięku dla kanału

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość głośności audio



[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału audio ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - kanał audio o przekazanym kodzie nie istnieje

Spostrzeżenia:

Poziom głośności jest wartością z przedziału [0.0, 1.0]

1.4.3.14 [RE_API](#) `int32_t` `avunit_audio_driver_control_set_channels_volume`
([AudioDriverControlHandle](#) *handle*, float * *volume*, uint32_t *volume_count*)

Ustawienie wartości głośności dźwięku dla wszystkich kanałów audio jednocześnie.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych audio
in	<i>volume</i>	Tablica wartości głośności dźwięku dla wszystkich kanałów audio
in	<i>volume_count</i>	Ilość przekazanych argumentów w tablicy volume

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość głośności audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica z głośnością kanałów ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość kanałów audio nie zgadza się z ilością zwróconą przez [avunit_audio_driver_control_get_channel_count\(\)](#)

Spostrzeżenia:

Poziom głośności jest wartością z przedziału [0.0, 1.0]

Zobacz również:

`avunit_audio_driver_control_get_channel_count`

1.5 Funkcje modułu wideo

Funkcje służące do pobierania informacji o konfiguracji modułu wideo oraz umożliwiające sterowanie parametrami rejestracji wideo.

1.5.1 Funkcje

- [RE_API](#) `int32_t` [avunit_video_driver_control_set_state_changed_callback](#)
([VideoDriverControlHandle](#) *handle*, [AVUnitCallbackStateHandle](#) *callback_state*,
[VideoSourceStateChanged](#) *callback*)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu źródła wideo.



- [RE API int32_t avunit video driver control remove state changed callback](#) ([VideoDriverControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit video driver control set state changed callback](#).
- [RE API int32_t avunit video driver control get frame count](#) ([VideoDriverControlHandle](#) handle)
Pobranie ilości wszystkich obsługiwanych klatek wideo.
- [RE API int32_t avunit video driver control get mixer count](#) ([VideoDriverControlHandle](#) handle)
Pobranie ilości wszystkich obsługiwanych mikserów wideo.
- [RE API int32_t avunit video driver control get frame index](#) ([VideoDriverControlHandle](#) handle, const char *frame_code)
Pobranie indeksu klatki wideo na podstawie jej kodu.
- [RE API int32_t avunit video driver control get frame code](#) ([VideoDriverControlHandle](#) handle, uint32_t frame_index, const char **frame_code)
Pobranie kodu klatki wideo na podstawie jej indeksu.
- [RE API int32_t avunit video driver control get frame name by code](#) ([VideoDriverControlHandle](#) handle, const char *frame_code, const char **frame_name)
Pobranie opisowej nazwy klatki wideo na podstawie jej kodu.
- [RE API int32_t avunit video driver control get frame name by index](#) ([VideoDriverControlHandle](#) handle, uint32_t frame_index, const char **frame_name)
Pobranie opisowej nazwy klatki wideo na podstawie jej indeksu.
- [RE API int32_t avunit video driver control state](#) ([VideoDriverControlHandle](#) handle, const char *frame_code)
Pobranie stanu podłączenia źródła wideo.
- [RE API int32_t avunit video driver control get mixer code](#) ([VideoDriverControlHandle](#) handle, uint32_t mixer_index, const char **mixer_code)
Pobranie kodu miksera wideo na podstawie jego indeksu.
- [RE API int32_t avunit video driver control get mixer frame count](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code)
Pobranie ilości klatek wideo miksowanych w danym mikserze.
- [RE API int32_t avunit video driver control get mixer supported size](#) ([VideoDriverControlHandle](#) handle, uint32_t *min_width, uint32_t *min_height, uint32_t *max_width, uint32_t *max_height)
Pobranie zakresu rozdzielczości wyjściowych obsługiwanych przez mikser wideo.
- [RE API int32_t avunit video driver control get mixer output size](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *width, uint32_t *height)
Pobranie rozmiaru klatki wideo produkowanej przez dany mikser.



- [RE_API int32_t avunit video driver control set mixer output size](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t width, uint32_t height)
Ustawienie rozmiaru klatki video produkowanej przez dany mikser.
- [RE_API int32_t avunit video driver control does mixer contain frame](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t frame_index, uint32_t *internal_index)
Sprawdzenie, czy dana klatka video jest miksowana w danym mikserze.
- [RE_API int32_t avunit video driver control set mixer frame order](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *order, uint32_t order_count)
Ustawienie kolejności klatek (z-indeks) w mikserze.
- [RE_API int32_t avunit video driver control get mixer frame order](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *order, uint32_t order_count)
Pobranie kolejności klatek (z-indeks) w mikserze.
- [RE_API int32_t avunit video driver control set frame enabled in mixer](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, int32_t enabled)
Włączenie lub wyłączenie przetwarzania klatki w mikserze.
- [RE_API int32_t avunit video driver control enable frames in mixer](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const int8_t *enabled, uint32_t enabled_count)
Włączenie lub wyłączenie przetwarzania dla wszystkich klatek w mikserze.
- [RE_API int32_t avunit video driver control set mixer frame area](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitRect](#) *area)
Ustawienie obszaru klatki w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control set mixer frame areas](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitRect](#) *areas, uint32_t areas_count)
Ustawienie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control get mixer frame area](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitRect](#) *area)
Pobranie obszaru klatki w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control get mixer frame areas](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitRect](#) *areas, uint32_t areas_count)
Pobranie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.



- [RE API int32_t avunit video driver control set mixer frame crop](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, const [AVUnitCrop](#) *crop)
Ustawienie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control set mixer frame crops](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const [AVUnitCrop](#) *crops, uint32_t crop_count)
Ustawienie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control get mixer frame crop](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitCrop](#) *crop)
Pobranie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control get mixer frame crops](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitCrop](#) *crops, uint32_t crop_count)
Pobranie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.

1.5.2 Opis szczegółowy

Funkcje służące do pobierania informacji o konfiguracji modułu wideo oraz umożliwiające sterowanie parametrami rejestracji wideo.

Moduł obsługi wideo odpowiada za akwizycję danych wideo z dostępnych źródeł oraz za miksowanie klatek z różnych źródeł zgodnie z konfiguracją mikserów wideo. Dostępne funkcje pozwalają na pobranie informacji dotyczących klatek: ich liczby, kodu, nazwy opisowej, a także informacji dotyczących mikserów i miksowanych przez dany mikser klatek. Sterowanie parametrami rejestracji pozwala na włączanie i wyłączanie poszczególnych klatek w mikserze a także daje możliwość ustawiania położenia klatek źródłowych w obrazie wynikowym miksera.

1.5.3 Dokumentacja funkcji

1.5.3.1 [RE API int32_t avunit video driver control set_state_changed_callback](#)
([VideoDriverControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [VideoSourceStateChanged](#) callback)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu źródła wideo.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit allocate callback state
in	<i>callback</i>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana przed wywołaniem funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.5.3.2 [RE_API](#) int32_t [avunit_video_driver_control_remove_state_changed_callback](#)
([VideoDriverControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_video_driver_control_set_state_changed_callback](#).

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

[avunit_video_driver_control_set_state_changed_callback](#)

1.5.3.3 [RE API](#) `int32_t` `avunit_video_driver_control_get_frame_count`
([VideoDriverControlHandle](#) *handle*)

Pobranie ilości wszystkich obsługiwanych klatek wideo.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość skonfigurowanych klatek wideo. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.5.3.4 [RE API](#) `int32_t` `avunit_video_driver_control_get_mixer_count`
([VideoDriverControlHandle](#) *handle*)

Pobranie ilości wszystkich obsługiwanych mikserów wideo.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość skonfigurowanych mikserów wideo. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.5.3.5 [RE API](#) `int32_t` `avunit_video_driver_control_get_frame_index`
([VideoDriverControlHandle](#) *handle*, `const char * frame_code`)

Pobranie indeksu klatki wideo na podstawie jej kodu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy pobrać indeks

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to indeks klatki wideo. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki wideo ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - przekazany kod klatki wideo nie istnieje



1.5.3.6 [RE API](#) `int32_t` `avunit_video_driver_control_get_frame_code`
([VideoDriverControlHandle](#) *handle*, `uint32_t` *frame_index*, `const char **` *frame_code*)

Pobranie kodu klatki wideo na podstawie jej indeksu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>frame_index</i>	Indeks klatki wideo, dla której chcemy pobrać kod
out	<i>frame_code</i>	Wskaźnik, do którego zostanie zapisany kod klatki wideo

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano kod klatki wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) - przekazany indeks klatki wideo nie istnieje

Spostrzeżenia:

Indeksy klatek są numerowane od 0 do wartości zwróconej przez funkcję [avunit_video_driver_control_get_frame_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu klatki wideo jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

Zobacz również:

`avunit_video_driver_control_get_frame_count`

1.5.3.7 [RE API](#) `int32_t` `avunit_video_driver_control_get_frame_name_by_code`
([VideoDriverControlHandle](#) *handle*, `const char *` *frame_code*, `const char **` *frame_name*)

Pobranie opisowej nazwy klatki wideo na podstawie jej kodu.

Nazwa ta zostaje wyświetlona użytkownikowi w celu identyfikacji źródła obrazu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy pobrać nazwę
out	<i>frame_name</i>	Wskaźnik, do którego zostanie zapisana nazwa klatki wideo

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano nazwę klatki wideo



[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - klatka wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Pamięć, w której zapisana jest wartość nazwy klatki wideo jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

1.5.3.8 [RE_API](#) `int32_t avunit_video_driver_control_get_frame_name_by_index (VideoDriverControlHandle handle, uint32_t frame_index, const char ** frame_name)`

Pobranie opisowej nazwy klatki wideo na podstawie jej indeksu.

Nazwa ta zostaje wyświetlona użytkownikowi w celu identyfikacji źródła obrazu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>frame_index</i>	Indeks klatki wideo, dla której chcemy pobrać nazwę
out	<i>frame_name</i>	Wskaźnik, do którego zostanie zapisana nazwa klatki wideo

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano nazwę klatki wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) - klatka wideo o przekazanym indeksie nie istnieje

Spostrzeżenia:

Indeksy klatek są numerowane od 0 do wartości zwróconej przez funkcję [avunit_video_driver_control_get_frame_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość nazwy klatki wideo jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

1.5.3.9 [RE_API](#) `int32_t avunit_video_driver_control_state (VideoDriverControlHandle handle, const char * frame_code)`

Pobranie stanu podłączenia źródła wideo.

Funkcja pozwala na pobranie aktualnej informacji, czy źródło wideo produkujące obraz dla podanej klatki jest podłączone.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy pobrać stan

Zwraca:

Wartość 1 oznacza, że źródło produkujące dane dla podanej klatki wideo jest podłączone.

Wartość 0 oznacza, że źródło produkujące dane dla podanej klatki wideo jest odłączone.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - klatka wideo o przekazanym kodzie nie istnieje

1.5.3.10 [RE_API](#) `int32_t avunit_video_driver_control_get_mixer_code(
VideoDriverControlHandle handle, uint32_t mixer_index, const char ** mixer_code)`

Pobranie kodu miksera wideo na podstawie jego indeksu.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_index</i>	Indeks miksera wideo, dla którego chcemy pobrać kod
out	<i>mixer_code</i>	Wskaźnik, do którego zostanie zapisany kod miksera wideo

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano kod miksera wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_INDEX](#) - przekazany indeks miksera wideo nie istnieje

Spostrzeżenia:

Indeksy mikserów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_video_driver_control_get_mixer_count\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu miksera wideo jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

Zobacz również:

`avunit_video_driver_control_get_mixer_count`

1.5.3.11 [RE_API](#) `int32_t` `avunit_video_driver_control_get_mixer_frame_count`
([VideoDriverControlHandle](#) *handle*, `const char * mixer_code`)

Pobranie ilości klatek wideo miksowanych w danym mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać ilość miksowanych przez niego klatek

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość klatek wideo miksowanych w danym mikserze. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

1.5.3.12 [RE_API](#) `int32_t` `avunit_video_driver_control_get_mixer_supported_size`
([VideoDriverControlHandle](#) *handle*, `uint32_t * min_width`, `uint32_t * min_height`,
`uint32_t * max_width`, `uint32_t * max_height`)

Pobranie zakresu rozdzielczości wyjściowych obsługiwanych przez mikser wideo.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
out	<i>min_width</i>	Wskaźnik, do którego zostanie zapisana minimalna szerokość klatki
out	<i>min_height</i>	Wskaźnik, do którego zostanie zapisana minimalna wysokość klatki
out	<i>max_width</i>	Wskaźnik, do którego zostanie zapisana maksymalna szerokość klatki
out	<i>max_height</i>	Wskaźnik, do którego zostanie zapisana maksymalna wysokość klatki

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wielkość wynikowej klatki wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na minimalną szerokość klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na minimalną wysokość klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na maksymalną szerokość klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na maksymalną wysokość klatki ma wartość NULL

1.5.3.13 [RE_API](#) `int32_t` `avunit_video_driver_control_get_mixer_output_size`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, `uint32_t *` *width*, `uint32_t *` *height*)

Pobranie rozmiaru klatki wideo produkowanej przez dany mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać wielkość wynikowej klatki
out	<i>width</i>	Wskaźnik, do którego zostanie zapisana szerokość klatki
out	<i>height</i>	Wskaźnik, do którego zostanie zapisana wysokość klatki

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wielkość wynikowej klatki wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na szerokość klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na wysokość klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

1.5.3.14 [RE_API](#) `int32_t` `avunit_video_driver_control_set_mixer_output_size`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, `uint32_t` *width*, `uint32_t` *height*)

Ustawienie rozmiaru klatki wideo produkowanej przez dany mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać wielkość wynikowej klatki
in	<i>width</i>	Nowa szerokość klatki; musi być wielokrotnością 16
in	<i>height</i>	Nowa wysokość klatki; musi być wielokrotnością 8

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wielkość wynikowej klatki wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana szerokość jest spoza obsługiwanego zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana wysokość jest spoza obsługiwanego zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana szerokość nie jest wielokrotnością 16

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana wysokość nie jest wielokrotnością 8

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Przekazany rozmiar wyjścia musi zawierać się w przedziale zwróconym przez funkcję [avunit_video_driver_control_get_mixer_supported_size\(\)](#).

1.5.3.15 [RE_API](#) `int32_t` `avunit_video_driver_control_does_mixer_contain_frame`
(`VideoDriverControlHandle handle`, `const char * mixer_code`, `uint32_t frame_index`,
`uint32_t * internal_index`)

Sprawdzenie, czy dana klatka wideo jest miksowana w danym mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy sprawdzić, czy dana klatka do niego należy
in	<i>frame_index</i>	Indeks klatki wideo
out	<i>internal_index</i>	Wskaźnik, do którego zostanie zapisany wewnętrzny indeks danej klatki w mikserze. Wartość jest uzupełniana tylko wtedy, gdy klatka należy do miksera.

Zwraca:

Wartość 1 oznacza, że klatka należy do miksera.

Wartość 0 oznacza, że klatka nie należy do miksera.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na wewnętrzny indeks ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) - klatka wideo o przekazanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Indeksy klatek są numerowane od 0 do wartości zwróconej przez funkcję [avunit_video_driver_control_get_frame_count\(\)](#) - 1.

Wewnętrzny indeks klatki w mikserze jest ważny przy określaniu kolejności klatek w mikserze (użycie funkcji [avunit_video_driver_control_set_mixer_frame_order\(\)](#)) a także przy ustawianiu obszaru wszystkich klatek w mikserze (funkcja [avunit_video_driver_control_set_mixer_frame_areas\(\)](#)). Indeks ten jest numerowany od 0 do wartości zwróconej przez [avunit_video_driver_control_get_mixer_frame_count\(\)](#) - 1. Parametry dla poszczególnych klatek do powyższych funkcji należy podawać według kolejności wskazywanej przez ten parametr.

Zobacz również:

`avunit_video_driver_control_get_frame_count`,
`avunit_video_driver_control_set_mixer_frame_order`,
`avunit_video_driver_control_set_mixer_frame_areas`,
`avunit_video_driver_control_enable_frames_in_mixer`

1.5.3.16 [RE_API](#) `int32_t` `avunit_video_driver_control_set_mixer_frame_order`
([VideoDriverControlHandle](#) *handle*, `const char * mixer_code`, `uint32_t * order`, `uint32_t order_count`)

Ustawienie kolejności klatek (z-indeks) w mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić kolejność



in	<i>order</i>	Tablica, zawierająca wartość indeksu kolejności (w zakresie od 0 do avunit_video_driver_control_get_mixer_frame_count() - 1). Wartość 0 oznacza, że klatka będzie się znajdować na samym spodzie i zostanie zastąpiona przez dowolną inną klatkę z wyższym indeksem. Indeksy muszą być unikalne.
in	<i>order_count</i>	Ilość elementów w tablicy <i>order</i>

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono kolejność klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica wartości indeksu ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość przekazanych wartości nie zgadza się z ilością klatek w mikserze

[AVUNIT_ERROR_INVALID_PARAMETER](#) - indeksy są nieprawidłowe

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek należy przekazywać według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

`avunit_video_driver_control_does_mixer_contain_frame`

1.5.3.17 [RE_API](#) `int32_t avunit_video_driver_control_get_mixer_frame_order(VideoDriverControlHandle handle, const char * mixer_code, uint32_t * order, uint32_t order_count)`

Pobranie kolejności klatek (z-indeks) w mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić kolejność

out	<i>order</i>	Tablica, w której zostaną zapisane aktualne wartości indeksu kolejności
in	<i>order_count</i>	Ilość elementów w tablicy order

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie odczytano kolejność klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica wartości indeksu ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość przekazanych wartości nie zgadza się z ilością klatek w mikserze

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

`avunit_video_driver_control_set_mixer_frame_order`,
`avunit_video_driver_control_does_mixer_contain_frame`

1.5.3.18 [RE_API](#) `int32_t` `avunit_video_driver_control_set_frame_enabled_in_mixer`
([VideoDriverControlHandle](#) *handle*, `const char * mixer_code`, `const char * frame_code`,
`int32_t enabled`)

Włączenie lub wyłączenie przetwarzania klatki w mikserze.

Funkcja umożliwia włączenie lub wyłączenie przetwarzania klatki należącej do miksera. Umożliwia to ukrycie nieaktywnych źródeł oraz włączenie ich, gdy będą potrzebne na obrazie wynikowym.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić włączyć/wyłączyć ramkę
in	<i>frame_code</i>	Kod klatki wideo, którą chcemy włączyć/wyłączyć w mikserze
in	<i>enabled</i>	Wartość logiczna określająca, czy daną klatkę chcemy włączyć (1) czy też wyłączyć (0)

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie włączono/wyłączono klatkę

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - mikser nie posiada klatki o przekazanym kodzie

Spostrzeżenia:

Klatka musi należeć do danego miksera, aby móc nią sterować - nie ma możliwości dodania do miksera klatek, których on nie posiada. Domyślnie wszystkie klatki są włączone.

1.5.3.19 [RE_API](#) `int32_t` `avunit_video_driver_control_enable_frames_in_mixer`
([VideoDriverControlHandle](#) *handle*, `const char * mixer_code`, `const int8_t * enabled`,
`uint32_t enabled_count`)

Włączenie lub wyłączenie przetwarzania dla wszystkich klatek w mikserze.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić włączyć/wyłączyć ramkę
in	<i>enabled</i>	Tablica wartości logicznych, czy daną klatkę chcemy włączyć (1) czy też wyłączyć (0)
in	<i>enabled_count</i>	Ilość elementów tablicy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie włączono/wyłączono klatki w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość elementów w tablicy jest różna od ilości klatek w mikserze

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

`avunit_video_driver_control_set_frame_enabled_in_mixer`

1.5.3.20 [RE_API](#) `int32_t` `avunit_video_driver_control_set_mixer_frame_area`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, `const char *` *frame_code*,
[AVUnitRect * *area*\)](#)

Ustawienie obszaru klatki w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy zmienić ustawienia klatki
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy zmienić obszar
in,out	<i>area</i>	Na wejściu jest to obszar klatki, który chcemy ustawić; na wyjściu znajduje się tu obszar klatki, który został ustawiony

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono obszar klatki w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obszar klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - mikser nie posiada klatki o przekazanym kodzie

Spostrzeżenia:

Klatka musi należeć do danego miksera, aby móc nią sterować - nie ma możliwości dodania do miksera klatek, których on nie posiada.

1.5.3.21 [RE API](#) `int32_t` `avunit_video_driver_control_set_mixer_frame_areas`
([VideoDriverControlHandle](#) `handle`, `const char * mixer_code`, [AVUnitRect](#) * `areas`,
`uint32_t areas_count`)

Ustawienie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli przetwarzania danych wideo
in	<code>mixer_code</code>	Kod miksera wideo, dla którego chcemy zmienić ustawienia klatek
in,out	<code>areas</code>	Tablica, w której na wejściu znajdują się obszary klatek, które chcemy ustawić; na wyjściu zostaną w niej zapisane ustawione obszary klatek
in	<code>areas_count</code>	Ilość elementów w tablicy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zmieniono obszary klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość elementów w tablicy jest różna od ilości klatek w mikserze

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit video driver control does mixer contain frame](#).

Zobacz również:

`avunit_video_driver_control_set_mixer_frame_area`

1.5.3.22 [RE API](#) `int32_t` `avunit_video_driver_control_get_mixer_frame_area`
([VideoDriverControlHandle](#) `handle`, `const char * mixer_code`, `const char * frame_code`,
[AVUnitRect](#) * `area`)

Pobranie obszaru klatki w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli przetwarzania danych wideo
----	---------------------	---



in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać ustawienia klatki
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy pobrać obszar
out	<i>area</i>	Aktualnie ustawiony obszar klatki

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano obszar klatki w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obszar klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - mikser nie posiada klatki o przekazanym kodzie

Zobacz również:

avunit_video_driver_control_set_mixer_frame_area

1.5.3.23 [RE_API](#) int32_t avunit_video_driver_control_get_mixer_frame_areas
([VideoDriverControlHandle](#) handle, const char * mixer_code, [AVUnitRect](#) * areas,
uint32_t areas_count)

Pobranie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać ustawienia klatek
out	<i>areas</i>	Tablica, do której zostaną zapisane obszary wszystkich klatek w mikserze
in	<i>areas_count</i>	Ilość elementów w tablicy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano obszary klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość elementów w tablicy jest różna od ilości klatek w mikserze

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

`avunit_video_driver_control_get_mixer_frame_area`

1.5.3.24 [RE_API](#) `int32_t` `avunit_video_driver_control_set_mixer_frame_crop`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, `const char *` *frame_code*,
`const AVUnitCrop *` *crop*)

Ustawienie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić parametry kadrowania klatki
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy ustawić parametry kadrowania klatki
out	<i>crop</i>	Parametry kadrowania klatki, które chcemy ustawić

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono parametry kadrowania klatki w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany parametr kadrowania ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - wartości w przekazanym parametrze kadrowania są spoza dopuszczalnego zakresu

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - mikser nie posiada klatki o przekazanym kodzie

Zobacz również:

avunit_video_driver_control_get_mixer_frame_crop

1.5.3.25 [RE_API](#) int32_t avunit_video_driver_control_set_mixer_frame_crops
([VideoDriverControlHandle](#) handle, const char * mixer_code, const [AVUnitCrop](#) * crops,
uint32_t crop_count)

Ustawienie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy ustawić parametry kadrowania klatek
out	<i>crops</i>	Tablica zawierająca parametry kadrowania wszystkich klatek w mikserze, które chcemy ustawić
in	<i>crop_count</i>	Ilość elementów w tablicy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono parametry kadrowania klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość elementów w tablicy jest różna od ilości klatek w mikserze

[AVUNIT_ERROR_INVALID_PARAMETER](#) - wartości w przekazanych parametrach kadrowania są spoza dopuszczalnego zakresu

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

avunit_video_driver_control_set_mixer_frame_crop

1.5.3.26 [RE API](#) `int32_t` `avunit_video_driver_control_get_mixer_frame_crop`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, `const char *` *frame_code*,
[AVUnitCrop](#) * *crop*)

Pobranie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać parametry kadrowania klatki
in	<i>frame_code</i>	Kod klatki wideo, dla której chcemy pobrać parametry kadrowania klatki
out	<i>crop</i>	Aktualnie ustawione parametry kadrowania klatki

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano parametry kadrowania klatki w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod klatki ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany parametr kadrowania ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) - mikser nie posiada klatki o przekazanym kodzie

Zobacz również:

`avunit_video_driver_control_set_mixer_frame_crop`

1.5.3.27 [RE API](#) `int32_t` `avunit_video_driver_control_get_mixer_frame_crops`
([VideoDriverControlHandle](#) *handle*, `const char *` *mixer_code*, [AVUnitCrop](#) * *crops*,
`uint32_t` *crop_count*)

Pobranie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania danych wideo
in	<i>mixer_code</i>	Kod miksera wideo, dla którego chcemy pobrać parametry kadrowania klatek



out	<i>crops</i>	Tablica, do której zostaną zapisane parametry kadrowania wszystkich klatek w mikserze
in	<i>crop_count</i>	Ilość elementów w tablicy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano parametry kadrowania klatek w mikserze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod miksera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość elementów w tablicy jest różna od ilości klatek w mikserze

[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) - mikser wideo o przekazanym kodzie nie istnieje

Spostrzeżenia:

Wartości dla poszczególnych klatek przekazywane są według kolejności wskazywanej przez parametr `internal_index` otrzymany w funkcji [avunit_video_driver_control_does_mixer_contain_frame](#).

Zobacz również:

`avunit_video_driver_control_get_mixer_frame_crop`

1.6 Funkcje modułu kodującego

Funkcje służące do zarządzania procesem zapisu nagrania do pliku.

1.6.1 Funkcje

- [RE_API int32_t avunit_encoder_control_set_state_changed_callback](#) ([EncoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [StateChangedCallback](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu nagrywania.
- [RE_API int32_t avunit_encoder_control_remove_state_changed_callback](#) ([EncoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_encoder_control_set_state_changed_callback](#).
- [RE_API int32_t avunit_encoder_control_open](#) ([EncoderControlHandle](#) handle, const char *file_name, const char *audio_codec, const char *video_codec, float audio_quality, float video_quality, int32_t video_enabled)
Otwarcie procesu nagrywania z podanymi parametrami.
- [RE_API int32_t avunit_encoder_control_finalize_and_close](#) ([EncoderControlHandle](#) handle)
Finalizacja pliku, zamknięcie procesu nagrywania oraz zwolnienie zasobów.
- [RE_API int32_t avunit_encoder_control_close](#) ([EncoderControlHandle](#) handle)



Zamknięcie procesu nagrywania oraz zwolnienie zasobów.

- [RE_API int32_t avunit_encoder_audio_channel_enabled \(EncoderControlHandle handle, uint32_t channel_index\)](#)
Pobranie informacji, czy dany kanał audio jest przetwarzany przez moduł nagrywający.
- [RE_API int32_t avunit_encoder_video_frame_enabled \(EncoderControlHandle handle, uint32_t frame_index\)](#)
Pobranie informacji, czy dana klatka wideo jest przetwarzana przez moduł nagrywający.
- [RE_API int32_t avunit_encoder_control_deserialize_state \(EncoderControlHandle handle, const char *file_name\)](#)
Otwarcie procesu nagrywania wykorzystując do tego zapisany wcześniej stan serializacji.
- [RE_API int32_t avunit_encoder_control_serialize_state \(EncoderControlHandle handle\)](#)
Zapis stanu nagrywania, umożliwiając późniejsze wznowienie nagrywania z tymi samymi parametrami oraz stanem.
- [RE_API int32_t avunit_encoder_control_serialization_supported \(EncoderControlHandle handle\)](#)
Sprawdzenie, czy serializacja jest obsługiwana dla parametrów podanych przy otwarciu pliku do nagrywania.
- [RE_API int32_t avunit_encoder_control_start \(EncoderControlHandle handle\)](#)
Wystartowanie procesu nagrywania ze stanu otwartego pliku lub pauzy.
- [RE_API int32_t avunit_encoder_control_pause \(EncoderControlHandle handle\)](#)
Tymczasowe zatrzymanie procesu nagrywania.
- [RE_API int64_t avunit_encoder_control_recording_time \(EncoderControlHandle handle\)](#)
Pobranie aktualnego czasu nagrywania w milisekundach.

1.6.2 Opis szczegółowy

Funkcje służące do zarządzania procesem zapisu nagrania do pliku.

Moduł ten odpowiada za zakodowanie i zapis do pliku zarejestrowanego dźwięku i obrazu z wykorzystaniem dostępnych kodeków. Obsługuje także mechanizm przerywania i wznowienia nagrywania. Dostępne funkcje pozwalają monitorować stan nagrywania, uzyskać informacje o tym czy dany kanał audio lub dana klatka wideo są przetwarzane przez moduł, a także pobrać czas nagrania.

1.6.3 Dokumentacja funkcji

1.6.3.1 [RE_API int32_t avunit_encoder_control_set_state_changed_callback \(EncoderControlHandle handle, AVUnitCallbackStateHandle callback_state, StateChangedCallback callback\)](#)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu nagrywania.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
----	---------------	-----------------------------



in	<code>callback_state</code>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<code>callback</code>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Wartością przekazaną do funkcji jest wartość wyliczenia [AVUnitEncoderState](#).

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana przed wywołaniem funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.6.3.2 [RE_API](#) `int32_t` [avunit_encoder_control_remove_state_changed_callback](#)
([EncoderControlHandle](#) *handle*, [AVUnitCallbackStateHandle](#) *callback_state*)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_encoder_control_set_state_changed_callback](#).

Parametry:

in	<code>handle</code>	Uchwyt do kontroli enkodera
in	<code>callback_state</code>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Zobacz również:

avunit_encoder_control_set_state_changed_callback

1.6.3.3 [RE_API](#) int32_t avunit_encoder_control_open ([EncoderControlHandle](#) handle, const char * file_name, const char * audio_codec, const char * video_codec, float audio_quality, float video_quality, int32_t video_enabled)

Otwarcie procesu nagrywania z podanymi parametrami.

Funkcja ta jedynie otwiera plik oraz inicjalizuje cały proces nagrywający, lecz nie startuje go. Aby wystartować proces nagrywający wymagane jest zawołanie funkcji [avunit_encoder_control_start](#).

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
in	<i>file_name</i>	Pełna ścieżka do pliku, do którego zostaną zapisane dane audio/video
in	<i>audio_codec</i>	Nazwa kodeka, który zostanie użyty do zapisu dźwięku. Dostępne wartości:

- "speex"
- "vorbis"
- "opus"

in	<i>video_codec</i>	Nazwa kodeka, który zostanie u Dostępne wartości:
----	--------------------	--

- "theora"
- "dirac"

in	<i>audio_quality</i>	Jakość zapisywanego dźwięku jak [0.0, 1.0]
in	<i>video_quality</i>	Jakość zapisywanego obrazu jak [0.0, 1.0]
in	<i>video_enabled</i>	Wartość logiczna określająca, c obraz i dźwięk (wartość 1) czy tyl 0).

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik do zapisu

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka do pliku ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek audio ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek wideo ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder nie znajduje się w stanie umożliwiającym otwarcie nagrywania

[AVUNIT_ERROR_INVALID_AUDIO_CODEC](#) - przekazany kodek audio jest nieobsługiwany

[AVUNIT_ERROR_INVALID_VIDEO_CODEC](#) - przekazany kodek wideo jest nieobsługiwany

[AVUNIT_ERROR_FAILED_TO_OPEN_FILE](#) - otwarcie podanego pliku do zapisu nie powiodło się

Zobacz również:

avunit_encoder_control_deserialize_state

1.6.3.4 [RE API](#) int32_t avunit_encoder_control_finalize_and_close ([EncoderControlHandle](#)
handle)

Finalizacja pliku, zamknięcie procesu nagrywania oraz zwolnienie zasobów.

Funkcja ta kończy proces nagrywania, oraz zamyka plik, zapisując jednocześnie wszystkie potrzebne struktury, które są wymagane przez kontener multimedialny. Funkcja ta powoduje także zapis wszystkich danych, które zostały zbuforowane i które nie zostały wcześniej zapisane.

Parametry:

In	<i>handle</i>	Uchwyt do kontroli enkodera
----	---------------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zamknięto otwarty plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder został już wcześniej zamknięty

[AVUNIT_ERROR_OPERATION_FAILED](#) - proces finalizacji pliku nie powiódł się; enkoder zostaje jednak zamknięty

Spostrzeżenia:

Po użyciu tej funkcji kontynuacja nagrywania do tego samego pliku nie jest już możliwa. W celu zamknięcia nagrania z umożliwieniem późniejszej kontynuacji (po wcześniejszym zapisie stanu nagrywania) należy użyć funkcji [avunit_encoder_control_close\(\)](#).

1.6.3.5 [RE_API int32_t avunit_encoder_control_close](#) ([EncoderControlHandle handle](#))

Zamknięcie procesu nagrywania oraz zwolnienie zasobów.

Funkcja ta w przeciwieństwie do funkcji [avunit_encoder_control_finalize_and_close\(\)](#) nie finalizuje pliku, a jedynie zamyka proces rejestracji. Funkcjonalność taka jest przydatna, jeśli wcześniej zapisaliśmy stan nagrywania za pomocą funkcji [avunit_encoder_control_serialize_state\(\)](#). Samo zamknięcie procesu nagrywania nie modyfikuje pliku, dzięki czemu możliwe jest wznowienie nagrywania od momentu serializacji.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
----	---------------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zamknięto otwarty plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder został już wcześniej zamknięty

Spostrzeżenia:

W celu zakończenia procesu nagrywania, w którym chcemy uzyskać finalny plik, należy użyć funkcji [avunit_encoder_control_finalize_and_close\(\)](#).

1.6.3.6 [RE_API int32_t avunit_encoder_audio_channel_enabled](#) ([EncoderControlHandle handle, uint32_t channel_index](#))

Pobranie informacji, czy dany kanał audio jest przetwarzany przez moduł nagrywający.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
in	<i>channel_index</i>	Poprawny indeks kanału audio

Zwraca:

Wartość 1 oznacza, że dany kanał audio jest przetwarzany przez moduł kodujący.

Wartość 0 oznacza, że dany kanał audio nie jest przetwarzany przez moduł kodujący.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) - przekazany indeks kanału audio jest niepoprawny

1.6.3.7 [RE_API int32_t avunit_encoder_video_frame_enabled](#) ([EncoderControlHandle handle, uint32_t frame_index](#))

Pobranie informacji, czy dana klatka wideo jest przetwarzana przez moduł nagrywający.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
in	<i>frame_index</i>	Poprawny indeks klatki wideo

Zwraca:

Wartość 1 oznacza, że dana klatka wideo jest przetwarzana przez moduł kodujący.

Wartość 0 oznacza, że dana klatka wideo nie jest przetwarzana przez moduł kodujący.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) - przekazany indeks klatki wideo jest niepoprawny

1.6.3.8 [RE_API](#) `int32_t avunit_encoder_control_deserialize_state (EncoderControlHandle handle, const char * file_name)`

Otwarcie procesu nagrywania wykorzystując do tego zapisany wcześniej stan serializacji.

Funkcja ta umożliwia wznowienie nagrywania z wykorzystaniem takich samych parametrów, jak ustawione wcześniej. W trakcie nagrywania (podczas pauzy) można wywołać funkcję [avunit_encoder_control_serialize_state](#), która zapisze aktualny stan całego procesu nagrywania w plikach na dysku. Zapis stanu nagrywania odbywa się również automatycznie w określonych przedziałach czasowych, co umożliwia szybkie wznowienie nagrywania po awarii sprzętu lub aplikacji.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli enkodera
in	<i>file_name</i>	Pełna ścieżka do pliku, do którego zostaną zapisane dane audio/wideo. Plik powinien już istnieć oraz powinny być do niego zapisane dane z wcześniejszej sesji nagrywania. Aplikacja powinna znaleźć dane umożliwiające wczytanie stanu nagrywania dla przekazanego pliku.

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik do zapisu

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder nie znajduje się w stanie umożliwiającym otwarcie nagrywania

[AVUNIT_ERROR_DESERIALIZATION_FAILED](#) - proces wznowienia nagrywania dla określonego pliku nie powiódł się

Zobacz również:

avunit_encoder_control_serialize_state

1.6.3.9 [RE_API](#) int32_t avunit_encoder_control_serialize_state ([EncoderControlHandle](#) handle)

Zapis stanu nagrywania, umożliwiając późniejsze wznowienie nagrywania z tymi samymi parametrami oraz stanem.

Parametry:

in	handle	Uchwyt do kontroli enkodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano aktualny stan nagrywania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder nie znajduje się w stanie pauzy

[AVUNIT_ERROR_SERIALIZATION_FAILED](#) - proces zapisu stanu nagrywania nie powiódł się

Spostrzeżenia:

Proces nagrywający musi znajdować się w stanie pauzy, aby móc użyć tej funkcji.

Zobacz również:

avunit_encoder_control_deserialize_state

1.6.3.10 [RE_API](#) int32_t avunit_encoder_control_serialization_supported
([EncoderControlHandle](#) handle)

Sprawdzenie, czy serializacja jest obsługiwana dla parametrów podanych przy otwarciu pliku do nagrywania.

Funkcja jest dostępna tylko wtedy, gdy otwarty jest plik do nagrywania (włączając w to stan nagrywania i pauzy). Jeśli dana kombinacja kodeków nie obsługuje serializacji, funkcja zwróci stosowną informację.

Parametry:

in	handle	Uchwyt do kontroli enkodera
----	--------	-----------------------------

Zwraca:

Wartość 1 oznacza, że serializacja jest obsługiwana.

Wartość 0 oznacza, że serializacja nie jest obsługiwana.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder nie posiada otwartego pliku

Zobacz również:

avunit_encoder_control_deserialize_state

1.6.3.11 [RE_API](#) int32_t avunit_encoder_control_start ([EncoderControlHandle](#) handle)

Wystartowanie procesu nagrywania ze stanu otwartego pliku lub pauzy.

Parametry:

in	handle	Uchwyt do kontroli enkodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wystartowano nagrywanie

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - enkoder nie znajduje się w stanie umożliwiającym wystartowanie nagrywania

1.6.3.12 [RE_API](#) int32_t avunit_encoder_control_pause ([EncoderControlHandle](#) handle)

Tymczasowe zatrzymanie procesu nagrywania.

Po pomyślnym zatrzymaniu procesu nagrywania można go ponownie wystartować używając do tego celu funkcji [avunit_encoder_control_start](#).

Parametry:

in	handle	Uchwyt do kontroli enkodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zatrzymano nagrywanie

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - encoder nie znajduje się w stanie umożliwiającym zatrzymanie nagrywania

1.6.3.13 [RE_API](#) int64_t avunit_encoder_control_recording_time ([EncoderControlHandle](#) handle)

Pobranie aktualnego czasu nagrywania w milisekundach.

Parametry:

in	handle	Uchwyt do kontroli enkodera
----	--------	-----------------------------

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest czasem nagrania w milisekundach. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny



1.7 Funkcje modułu dekodującego

Funkcje służące do zarządzania procesem odczytu nagrań z plików.

1.7.1 Funkcje

- [RE_API int32_t avunit_decoder_control_set_state_changed_callback](#) ([DecoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [StateChangedCallback](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu odtwarzania.
- [RE_API int32_t avunit_decoder_control_remove_state_changed_callback](#) ([DecoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_decoder_control_set_state_changed_callback](#).
- [RE_API int32_t avunit_decoder_control_open](#) ([DecoderControlHandle](#) handle, const char *file_name)
Otwarcie pliku do odtwarzania.
- [RE_API int32_t avunit_decoder_control_close](#) ([DecoderControlHandle](#) handle)
Zamknięcie odtwarzanego pliku.
- [RE_API int64_t avunit_decoder_control_beginning](#) ([DecoderControlHandle](#) handle)
Pobranie początkowej pozycji nagrania wyrażonej w milisekundach.
- [RE_API int64_t avunit_decoder_control_duration](#) ([DecoderControlHandle](#) handle)
Pobranie końcowej pozycji nagrania wyrażonej w milisekundach.
- [RE_API int32_t avunit_decoder_control_start](#) ([DecoderControlHandle](#) handle)
Rozpoczęcie odtwarzania wcześniej otwartego pliku lub wznowienie odtwarzania po jego zatrzymaniu.
- [RE_API int32_t avunit_decoder_control_pause](#) ([DecoderControlHandle](#) handle)
Zatrzymanie odtwarzania wcześniej otwartego pliku.
- [RE_API int32_t avunit_decoder_control_seek](#) ([DecoderControlHandle](#) handle, uint64_t position)
Przeskok do wybranej pozycji we wcześniej otwartym pliku.
- [RE_API int32_t avunit_decoder_control_start_recording](#) ([DecoderControlHandle](#) handle)
Rozpoczęcie rejestracji odtwarzanego nagrania w nagrywanym pliku.
- [RE_API int32_t avunit_decoder_control_stop_recording](#) ([DecoderControlHandle](#) handle)
Zakończenie rejestracji odtwarzanego nagrania w nagrywanym pliku.
- [RE_API int64_t avunit_decoder_control_position](#) ([DecoderControlHandle](#) handle)
Pobranie aktualnej pozycji w odtwarzanym pliku wyrażonej w milisekundach.

1.7.2 Opis szczegółowy

Funkcje służące do zarządzania procesem odczytu nagrań z plików.

Ten moduł odpowiada za odczyt nagrania z pliku, jego dekodowanie i odtwarzanie. Pozwala również na pobranie informacji dotyczących czasu nagrania oraz na kontrolę odtwarzania, w tym kontrolę nagrywania odczytywanego pliku do wynikowego nagrania.

1.7.3 Dokumentacja funkcji

1.7.3.1 [RE_API](#) `int32_t` `avunit_decoder_control_set_state_changed_callback`
([DecoderControlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`,
[StateChangedCallback](#) `callback`)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu odtwarzania.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli dekodera
in	<code>callback_state</code>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<code>callback</code>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Wartością przekazaną do funkcji jest wartość wyliczenia [AVUnitDecoderState](#).

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana przed wywołaniem funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.7.3.2 [RE_API](#) `int32_t` `avunit_decoder_control_remove_state_changed_callback`
([DecoderControlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_decoder_control_set_state_changed_callback](#).



Parametry:

in	<i>handle</i>	Uchwyt do kontroli dekodera
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Zobacz również:

avunit_decoder_control_set_state_changed_callback

1.7.3.3 [RE_API](#) int32_t avunit_decoder_control_open ([DecoderControlHandle](#) *handle*, const char * *file_name*)

Otwarcie pliku do odtwarzania.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli dekodera
in	<i>file_name</i>	Pełna ścieżka do pliku, który zostanie otwarty

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka do pliku ma wartość NULL

[AVUNIT_ERROR_FAILED_TO_OPEN_FILE](#) - nie udało się otworzyć pliku do odtwarzania

[AVUNIT_ERROR_INVALID_STATE](#) - dekodер znajduje się w stanie, który nie umożliwia otwarcia pliku

1.7.3.4 [RE_API](#) int32_t avunit_decoder_control_close ([DecoderControlHandle](#) *handle*)

Zamknięcie odtwarzanego pliku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli dekodera
----	---------------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny



1.7.3.5 [RE_API int64_t avunit_decoder_control_beginning \(DecoderControlHandle handle\)](#)

Pobranie początkowej pozycji nagrania wyrażonej w milisekundach.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

Spostrzeżenia:

Jeśli otwierane nagranie jest fragmentem, wartość ta będzie większa od 0.

1.7.3.6 [RE_API int64_t avunit_decoder_control_duration \(DecoderControlHandle handle\)](#)

Pobranie końcowej pozycji nagrania wyrażonej w milisekundach.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

Spostrzeżenia:

Jeśli otwierane nagranie jest fragmentem, wartość ta będzie odzwierciedlać długość oryginalnego nagrania (nie fragmentu).

1.7.3.7 [RE_API int32_t avunit_decoder_control_start \(DecoderControlHandle handle\)](#)

Rozpoczęcie odtwarzania wcześniej otwartego pliku lub wznowienie odtwarzania po jego zatrzymaniu.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

1.7.3.8 [RE_API](#) int32_t avunit_decoder_control_pause ([DecoderControlHandle](#) handle)

Zatrzymanie odtwarzania wcześniej otwartego pliku.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

1.7.3.9 [RE_API](#) int32_t avunit_decoder_control_seek ([DecoderControlHandle](#) handle, uint64_t position)

Przeskok do wybranej pozycji we wcześniej otwartym pliku.

Parametry:

in	handle	Uchwyt do kontroli dekodera
in	position	Pozycja w milisekundach, do której chcemy przeskoczyć

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana pozycja znajduje się za końcem nagrania

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

[AVUNIT_ERROR_OPERATION_FAILED](#) - przeskok do wybranej pozycji nie powiódł się

1.7.3.10 [RE_API](#) int32_t avunit_decoder_control_start_recording ([DecoderControlHandle](#) handle)

Rozpoczęcie rejestracji odtwarzanego nagrania w nagrywanym pliku.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie rozpoczęto rejestrację

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

Spostrzeżenia:

Funkcja włącza rejestrację odtwarzanego materiału w wynikowym obrazie oraz w wynikowym pliku dźwiękowym. Do momentu zawołania tej funkcji ani obraz ani dźwięk z dekodera nie są rejestrowane w wynikowym pliku.

1.7.3.11 [RE_API](#) int32_t avunit_decoder_control_stop_recording ([DecoderControlHandle handle](#))

Zakończenie rejestracji odtwarzanego nagrania w nagrywanym pliku.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zakończono rejestrację

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak otwartego pliku

Zobacz również:

avunit_decoder_control_start_recording

1.7.3.12 [RE_API](#) int64_t avunit_decoder_control_position ([DecoderControlHandle handle](#))

Pobranie aktualnej pozycji w odtwarzanym pliku wyrażonej w milisekundach.

Parametry:

in	handle	Uchwyt do kontroli dekodera
----	--------	-----------------------------

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest pozycją w odtwarzanym pliku w milisekundach. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.8 Funcje do przetwarzania plików multimedialnych

Funcje służące do obsługi wszystkich procesów przetwarzania plików multimedialnych.

1.8.1 Moduły

- [Funkcje ekstrakcji](#)
- *Funkcje służące do ekstrakcji strumieni z plików nagrań.* [Funkcje do przekonwertowania nagrania](#)

1.8.2 Funkcje służące do przekonwertowania nagrania na inny format.

Funcje:

- [RE_API](#) int32_t avunit_media_processing_open ([MediaProcessingControlHandle handle](#), const char *file_name)



Otwarcie pliku źródłowego w obiekcie przetwarzania multimediów.

- [RE_API int32_t avunit_media_processing_close \(MediaProcessingControlHandle handle\)](#)
Zamknięcie pliku źródłowego w obiekcie przetwarzania multimediów strumieni oraz zwolnienie wszystkich związanych z nim zasobów.
- [RE_API int32_t avunit_media_processing_stream_count \(MediaProcessingControlHandle handle\)](#)
Pobranie ilości strumieni w pliku źródłowym.
- [RE_API int32_t avunit_media_processing_stream_codec \(MediaProcessingControlHandle handle, uint32_t stream_index, const char **codec\)](#)
Pobranie kodeka użytego do zakodowania ścieżki o podanym indeksie.
- [RE_API int32_t avunit_media_processing_stream_internal_index \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie wewnętrznego indeksu strumienia dla ścieżki o podanym indeksie.
- [RE_API int32_t avunit_media_processing_stream_is_audio \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie typu strumienia.
- [RE_API int32_t avunit_media_processing_stream_audio_sampling_rate \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie częstotliwości próbkowania strumienia audio.
- [RE_API int32_t avunit_media_processing_stream_audio_num_channels \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie ilości kanałów w strumieniu audio.
- [RE_API int32_t avunit_media_processing_stream_video_width \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie szerokości klatki w ścieżce wideo.
- [RE_API int32_t avunit_media_processing_stream_video_height \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie wysokości klatki w ścieżce wideo.
- [RE_API int32_t avunit_media_processing_stream_video_frame_rate_numerator \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie licznika ułamka określającego ilość klatek na sekundę.
- [RE_API int32_t avunit_media_processing_stream_video_frame_rate_denominator \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Pobranie mianownika ułamka określającego ilość klatek na sekundę.
- [RE_API int32_t avunit_media_processing_enable_stream \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Włączenie wypakowania strumienia do wyjściowego pliku.
- [RE_API int32_t avunit_media_processing_disable_stream \(MediaProcessingControlHandle handle, uint32_t stream_index\)](#)
Wyłączenie wypakowania strumienia do wyjściowego pliku.

1.8.3 Opis szczegółowy

Funkcje służące do obsługi wszystkich procesów przetwarzania plików multimedialnych.

Zadaniem modułu przetwarzania plików multimedialnych jest obsługa pobierania informacji dotyczących strumieni audio i wideo z plików z nagraniami oraz dokonanie wyboru konkretnych strumieni w celu ich przetworzenia. Informacje udostępniane dotyczą ilości i typu strumieni a także ich parametrów.

1.8.4 Dokumentacja funkcji

1.8.4.1 [RE_API](#) `int32_t avunit_media_processing_open` ([MediaProcessingControlHandle handle](#), `const char * file_name`)

Otwarcie pliku źródłowego w obiekcie przetwarzania multimediiów.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>file_name</i>	Pełna ścieżka do źródłowego pliku, który ma zostać otwarty

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie otwarto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka do pliku ma wartość NULL

[AVUNIT_ERROR_FAILED_TO_OPEN_FILE](#) - nie udało się otworzyć pliku w ekstraktorze. -

1.8.4.2 [RE_API](#) `int32_t avunit_media_processing_close` ([MediaProcessingControlHandle handle](#))

Zamknięcie pliku źródłowego w obiekcie przetwarzania multimediiów strumieni oraz zwolnienie wszystkich związanych z nim zasobów.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie zamknięto plik

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.8.4.3 [RE_API](#) `int32_t avunit_media_processing_stream_count` ([MediaProcessingControlHandle handle](#))

Pobranie ilości strumieni w pliku źródłowym.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest ilością strumieni w pliku źródłowym. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

1.8.4.4 [RE_API](#) `int32_t` `avunit_media_processing_stream_codec`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*, `const char **` *codec*)

Pobranie kodeka użytego do zakodowania ścieżki o podanym indeksie.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację
out	<i>codec</i>	Wskaźnik, do którego zostanie zapisany kodek

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie odczytano kodek strumienia

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kodek ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

Spostrzeżenia:

Pamięć, w której zapisany jest kod kodeka jest zarządzana przez obiekt i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_media_processing_close](#).

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

1.8.4.5 [RE_API](#) `int32_t` `avunit_media_processing_stream_internal_index`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Pobranie wewnętrznego indeksu strumienia dla ścieżki o podanym indeksie.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
----	---------------	---



in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację
----	---------------------	--

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest wewnętrznym indeksem strumienia. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

1.8.4.6 [RE API](#) `int32_t` `avunit_media_processing_stream_is_audio`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Pobranie typu strumienia.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediów
in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Wartość 1 oznacza, że strumień zawiera ścieżkę dźwiękową.

Wartość 0 oznacza, że strumień zawiera obraz.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

1.8.4.7 [RE API](#) `int32_t` `avunit_media_processing_stream_audio_sampling_rate`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Pobranie częstotliwości próbkowania strumienia audio.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediów
----	---------------	--



in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację
----	---------------------	--

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest częstotliwością próbkowania ścieżki audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych audio

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera ścieżkę dźwiękową.

1.8.4.8 [RE_API](#) `int32_t` `avunit_media_processing_stream_audio_num_channels`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Pobranie ilości kanałów w strumieniu audio.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest ilością kanałów w ścieżce audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych audio

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera ścieżkę dźwiękową.



1.8.4.9 [RE_API](#) `int32_t` `avunit_media_processing_stream_video_width`
([MediaProcessingControlHandle](#) `handle`, `uint32_t` `stream_index`)

Pobranie szerokości klatki w ścieżce wideo.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli przetwarzania multimediiów
in	<code>stream_index</code>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest szerokością klatki ścieżki wideo. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych wideo

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera obraz.

1.8.4.10 [RE_API](#) `int32_t` `avunit_media_processing_stream_video_height`
([MediaProcessingControlHandle](#) `handle`, `uint32_t` `stream_index`)

Pobranie wysokości klatki w ścieżce wideo.

Parametry:

in	<code>handle</code>	Uchwyt do kontroli przetwarzania multimediiów
in	<code>stream_index</code>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest wysokością klatki ścieżki wideo. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych wideo



Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera obraz.

1.8.4.11 [RE API](#) `int32_t avunit_media_processing_stream_video_frame_rate_numerator`
([MediaProcessingControlHandle](#) *handle*, `uint32_t stream_index`)

Pobranie licznika ułamka określającego ilość klatek na sekundę.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest licznikiem ułamka określającego ilość klatek na sekundę. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych wideo

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera obraz.

Ilość klatek na sekundę w strumieniu określona jest wzorem (numerator/denominator).

1.8.4.12 [RE API](#) `int32_t avunit_media_processing_stream_video_frame_rate_denominator`
([MediaProcessingControlHandle](#) *handle*, `uint32_t stream_index`)

Pobranie mianownika ułamka określającego ilość klatek na sekundę.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks, dla którego chcemy pobrać informację

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest mianownikiem ułamka określającego ilość klatek na sekundę. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:



[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

[AVUNIT_ERROR_INVALID_STREAM_TYPE](#) - ścieżka nie zawiera danych wideo

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

Funkcja jest dostępna tylko wtedy, gdy strumień zawiera obraz.

Ilość klatek na sekundę w strumieniu określona jest wzorem (numerator/denominator).

1.8.4.13 [RE_API](#) `int32_t` `avunit_media_processing_enable_stream`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Włączenie wypakowania strumienia do wyjściowego pliku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks strumienia, który chcemy wypakować

Zwraca:

[AVUNIT_SUCCESS](#) - włączono strumień do wypakowania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

1.8.4.14 [RE_API](#) `int32_t` `avunit_media_processing_disable_stream`
([MediaProcessingControlHandle](#) *handle*, `uint32_t` *stream_index*)

Wyłączenie wypakowania strumienia do wyjściowego pliku.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przetwarzania multimediiów
in	<i>stream_index</i>	Indeks strumienia, którego nie chcemy wypakować

Zwraca:

[AVUNIT_SUCCESS](#) - wyłączono strumień do wypakowania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) - ścieżka o podanym indeksie nie istnieje

Spostrzeżenia:

Indeksy strumieni są numerowane od 0 do wartości zwróconej przez funkcję [avunit_media_processing_stream_count\(\)](#) - 1.

1.9 Funkcje ekstrakcji

Funkcje służące do ekstrakcji strumieni z plików nagrań.

1.9.1 Funkcje

- [RE_API int32_t avunit_extractor_create \(MediaProcessingControlHandle *handle\)](#)
Utworzenie obiektu ekstraktora strumieni.
- [RE_API int32_t avunit_extractor_destroy \(MediaProcessingControlHandle handle\)](#)
Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).
- [RE_API int32_t avunit_extractor_extract_to \(MediaProcessingControlHandle handle, const char *output_file, ProgressCallback progress_callback\)](#)
Wypakowanie wybranych strumieni do pliku wyjściowego.
- [RE_API int32_t avunit_extractor_extract_cancel \(MediaProcessingControlHandle handle\)](#)
Przerwanie procesu ekstrakcji wykonywanego w innym wątku.

1.9.2 Opis szczegółowy

Funkcje służące do ekstrakcji strumieni z plików nagrań.

Zadaniem modułu ekstrakcji jest możliwość wyboru strumieni z pliku źródłowego oraz zgrania ich do nowego pliku.

1.9.3 Dokumentacja funkcji

1.9.3.1 [RE_API int32_t avunit_extractor_create \(MediaProcessingControlHandle * handle\)](#)

Utworzenie obiektu ekstraktora strumieni.

Parametry:

out	<i>handle</i>	Wskaźnik, do którego zostanie zapisany uchwyt do ekstraktora
-----	---------------	--

Zwraca:

[AVUNIT_SUCCESS](#) - utworzono nowy obiekt ekstraktora

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację

Spostrzeżenia:

W przypadku pomyślnego wykonania tej funkcji obiekt ekstraktora należy zniszczyć za pomocą funkcji [avunit_extractor_destroy](#).

Uchwyt utworzony za pomocą tej funkcji można używać w funkcjach z grup [Funkcje ekstrakcji](#) oraz [Funkcje do przetwarzania plików multimedialnych](#)

1.9.3.2 RE_API int32_t avunit_extractor_destroy ([MediaProcessingControlHandle handle](#))

Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).

Parametry:

in	<i>handle</i>	Uchwyt do kontroli ekstrakcji
----	---------------	-------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - zwolniono obiekt ekstraktora

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.9.3.3 RE_API int32_t avunit_extractor_extract_to ([MediaProcessingControlHandle handle](#), const char * *output_file*, [ProgressCallback progress_callback](#))

Wypakowanie wybranych strumieni do pliku wyjściowego.

Ekstraktor utworzy nowy plik o podanej nazwie, który będzie posiadał identyczny format, jak plik źródłowy. Plik będzie zawierał wybrane strumienie, które zostały włączone do ekstrakcji. W trakcie procesu wykonywana jest kopia danych strumienia - nie następuje przekodowanie jego danych od nowa. Dzięki takiemu zachowaniu jakość danych nie ulega zmianie.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli ekstrakcji
in	<i>output_file</i>	Pełna ścieżka do pliku, do którego zostaną zapisane wyekstraktowane ścieżki
in	<i>progress_callback</i>	Jeśli wartość nie jest NULL, to jest to poprawny callback, który będzie wywoływany w trakcie ekstrakcji, informując o postępie prac

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wypakowano wybrane strumienie do pliku wyjściowego

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_OPERATION_FAILED](#) - operacja nie powiodła się

[AVUNIT_ERROR_PROCESSING_CANCELED](#) - operacja została przerwana przez funkcję [avunit_extractor_extract_cancel\(\)](#)

Spostrzeżenia:

Aby móc wywołać tę funkcję, należy wcześniej włączyć przynajmniej jeden strumień za pomocą funkcji [avunit_media_processing_enable_stream\(\)](#).

1.9.3.4 [RE_API](#) int32_t [avunit_extractor_extract_cancel](#) ([MediaProcessingControlHandle](#) handle)

Przerwanie procesu ekstrakcji wykonywanego w innym wątku.

Funkcja ta umożliwia przerwanie długotrwałego procesu ekstrakcji strumieni, wykonywanego w innym wątku. Po wywołaniu tej funkcji zadanie ekstrakcji zostanie przerwane z kodem błędu [AVUNIT_ERROR_PROCESSING_CANCELED](#).

Parametry:

in	handle	Uchwyt do kontroli ekstrakcji
----	--------	-------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zlecono przerwanie procesu ekstrakcji

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

Spostrzeżenia:

Funkcja ta nie będzie miała żadnego efektu, jeśli aktualnie proces ekstrakcji nie jest aktywny.

1.10 Funkcje do przekonwertowania nagrania

Funkcje służące do przekonwertowania nagrania na inny format.

1.10.1 Funkcje

- [RE_API](#) int32_t [avunit_reencoder_create](#) ([MediaProcessingControlHandle](#) *handle)
Utworzenie obiektu ekstraktora strumieni.
- [RE_API](#) int32_t [avunit_reencoder_destroy](#) ([MediaProcessingControlHandle](#) handle)
Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).
- [RE_API](#) int32_t [avunit_reencoder_set_output_audio_parameters](#) ([MediaProcessingControlHandle](#) handle, const char *codec, float quality)
Ustawienie wybranych parametrów dla ścieżek audio.
- [RE_API](#) int32_t [avunit_reencoder_set_output_video_parameters](#) ([MediaProcessingControlHandle](#) handle, const char *codec, float quality)
Ustawienie wybranych parametrów dla ścieżek wideo.
- [RE_API](#) int32_t [avunit_reencoder_set_output_container](#) ([MediaProcessingControlHandle](#) handle, const char *container)
Ustawienie kontenera dla wyjściowego pliku.

- [RE_API int32_t avunit_reencoder_reencode_to](#) ([MediaProcessingControlHandle](#) handle, const char *output_file, [ProgressCallback](#) progress_callback)
Przekodowanie wybranych strumieni do pliku wyjściowego.
- [RE_API int32_t avunit_reencoder_reencode_cancel](#) ([MediaProcessingControlHandle](#) handle)
Przerwanie procesu przekodowania wykonywanego w innym wątku.

1.10.2 Opis szczegółowy

Funkcje służące do przekonwertowania nagrania na inny format.

Zadaniem modułu przekonwertowania nagrania jest możliwość wyboru strumieni źródłowych oraz przekodowanie ich do nowego kontenera multimedialnego.

1.10.3 Dokumentacja funkcji

1.10.3.1 [RE_API int32_t avunit_reencoder_create](#) ([MediaProcessingControlHandle](#) * handle)

Utworzenie obiektu ekstraktora strumieni.

Parametry:

out	handle	Wskaźnik, do którego zostanie zapisany uchwyt do kontroli przekodowywania
-----	--------	---

Zwraca:

[AVUNIT_SUCCESS](#) - utworzono nowy obiekt przekodowywania

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację

Spostrzeżenia:

W przypadku pomyślnego wykonania tej funkcji obiekt przekodowywania należy zniszczyć za pomocą funkcji [avunit_reencoder_destroy](#).

Uchwyt utworzony za pomocą tej funkcji można używać w funkcjach z grup [Funkcje do przekonwertowania nagrania](#) oraz [Funkcje do przetwarzania plików multimedialnych](#)

1.10.3.2 [RE_API int32_t avunit_reencoder_destroy](#) ([MediaProcessingControlHandle](#) handle)

Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).

Parametry:

in	handle	Uchwyt do kontroli przekodowywania
----	--------	------------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - zwolniono obiekt przekodowywania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.10.3.3 [RE_API](#) `int32_t` `avunit_reencoder_set_output_audio_parameters` ([MediaProcessingControlHandle](#) *handle*, `const char *` *codec*, `float` *quality*)

Ustawienie wybranych parametrów dla ścieżek audio.

Funkcja pozwala wybrać kodek oraz jakość dla wszystkich wyjściowych ścieżek audio. Jeśli funkcja ta nie zostanie zawołana, to zostanie użyty taki sam kodek jak w pliku źródłowym, a jakość zostanie ustawiona na 0.8.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przekodowywania	
in	<i>codec</i>	Nazwa kodeka, który zostanie użyty do zapisu dźwięku. Dostępne wartości:	
		<ul style="list-style-type: none"> • "speex" • "vorbis" • "opus" 	
in		<i>quality</i>	Jakość zapisywanego dźwięku ja [0.0, 1.0]

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono wybrane parametry audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CODEC](#) - przekazany kodek audio jest nieobsługiwany

1.10.3.4 [RE_API](#) `int32_t` `avunit_reencoder_set_output_video_parameters` ([MediaProcessingControlHandle](#) *handle*, `const char *` *codec*, `float` *quality*)

Ustawienie wybranych parametrów dla ścieżek wideo.

Funkcja pozwala wybrać kodek oraz jakość dla wszystkich wyjściowych ścieżek wideo. Jeśli funkcja ta nie zostanie zawołana, to zostanie użyty taki sam kodek jak w pliku źródłowym, a jakość zostanie ustawiona na 0.4.

Parametry:

in	<i>handle</i>	Uchwyt do kontroli przekodowywania	
in	<i>codec</i>	Nazwa kodeka, który zostanie użyty do zapisu obrazu. Dostępne wartości:	
		<ul style="list-style-type: none"> • "theora" • "dirac" • "vp8" 	

in	quality	Jakość zapisywanego obrazu jak [0.0, 1.0]

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono wybrane parametry wideo

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek ma wartość NULL

[AVUNIT_ERROR_INVALID_VIDEO_CODEC](#) - przekazany kodek wideo jest nieobsługiwany

1.10.3.5 [RE_API](#) int32_t avunit_reencoder_set_output_container
([MediaProcessingControlHandle](#) handle, const char * container)

Ustawienie kontenera dla wyjściowego pliku.

Funkcja ta pozwala wybrać kontener, w jakim zostaną zapisane przekodowane ścieżki. Jeśli funkcja ta nie zostanie zawołana, to zostanie użyty taki sam kontener jak w pliku źródłowym.

Parametry:

in	handle	Uchwyt do kontroli przekodowywania
in	container	Nazwa kontenera, w którym zostaną zapisane przekodowane ścieżki. Dostępne wartości:

- "ogg"
- "webm"

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono wybrany kontener

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kontener ma wartość NULL

[AVUNIT_ERROR_INVALID_CONTAINER](#) - przekazany kontener jest nieobsługiwany

1.10.3.6 [RE_API](#) int32_t avunit_reencoder_reencode_to ([MediaProcessingControlHandle](#) handle, const char * output_file, [ProgressCallback](#) progress_callback)

Przekodowanie wybranych strumieni do pliku wyjściowego.

Reencoder utworzy nowy plik o podanej nazwie, który będzie posiadał format przekazany wcześniej. Jeśli parametry formatu nie zostaną przekazane, zostaną użyte takie same wartości (kontener, kodeki) jak w pliku źródłowym. Plik będzie zawierał wybrane strumienie, które zostały włączone do procesu przetwarzania.

Parametry:

In	<i>handle</i>	Uchwyt do kontroli przekodowywania
In	<i>output_file</i>	Pełna ścieżka do pliku wyjściowego
In	<i>progress_callback</i>	Jeśli wartość nie jest NULL, to jest to poprawny callback, który będzie wywoływany w trakcie ekstrakcji, informując o postępie prac

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie przekodowano wybrane strumienie do pliku wyjściowego

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

[AVUNIT_ERROR_OPERATION_FAILED](#) - operacja nie powiodła się

[AVUNIT_ERROR_PROCESSING_CANCELED](#) - operacja została przerwana przez funkcję [avunit_reencoder_reencode_cancel\(\)](#)

Spostrzeżenia:

Aby móc wywołać tę funkcję, należy wcześniej włączyć przynajmniej jeden strumień za pomocą funkcji [avunit_media_processing_enable_stream\(\)](#).

1.10.3.7 [RE_API](#) `int32_t avunit_reencoder_reencode_cancel` ([MediaProcessingControlHandle handle](#))

Przerwanie procesu przekodowania wykonywanego w innym wątku.

Funkcja ta umożliwi przerwanie długotrwałego procesu przekodowania nagrania, wykonywanego w innym wątku. Po wywołaniu tej funkcji zadanie przekodowania zostanie przerwane z kodem błędu [AVUNIT_ERROR_PROCESSING_CANCELED](#).

Parametry:

In	<i>handle</i>	Uchwyt do kontroli przekodowywania
----	---------------	------------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zlecono przerwanie procesu przekodowania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - plik źródłowy nie jest otwarty

Spostrzeżenia:

Funkcja ta nie będzie miała żadnego efektu, jeśli aktualnie proces przekodowania nie jest aktywny.



1.11 Obsługa urządzeń zewnętrznych

Funkcje służące do obsługi zewnętrznych urządzeń, asystujących w procesie rejestracji.

1.11.1 Moduły

- [Sterowanie kamerami poprzez interfejs PTZ](#)
Funkcje służące do sterowania kamerami poprzez interfejs PTZ.
- [Komunikacja z kartami akwizycji dźwięku](#)
Funkcje służące do komunikacji z kartami akwizycji dźwięku.
- [Komunikacja ze wzmacniaczem](#)
Funkcje służące do komunikacji ze wzmacniaczem mocy dźwięku.
- [Komunikacja z zasilaczem awaryjnym UPS](#)
Funkcje służące do komunikacji z zasilaczem awaryjnym UPS.

1.11.2 Funkcje

- [RE_API int32_t avunit_external_devices_get_ptz_control \(ExternalDevicesHandle handle, PTZControlHandle *ptz_handle\)](#)
Pobranie uchwytu do sterowania ustawieniami kamer poprzez interfejs PTZ.
- [RE_API int32_t avunit_external_devices_get_carec mic control \(ExternalDevicesHandle handle, CarecMicControlHandle *carec_handle\)](#)
Pobranie uchwytu do pobierania informacji z karty akwizycji danych audio.
- [RE_API int32_t avunit_external_devices_get_amplifier_control \(ExternalDevicesHandle handle, AmplifierControlHandle *amplifier_handle\)](#)
Pobranie uchwytu do sterowania wzmacniaczem dźwięku.
- [RE_API int32_t avunit_external_devices_get_ups_control \(ExternalDevicesHandle handle, UPSControlHandle *ups_handle\)](#)
Pobranie uchwytu do sterowania zasilaczem awaryjnym UPS.

1.11.3 Opis szczegółowy

Funkcje służące do obsługi zewnętrznych urządzeń, asystujących w procesie rejestracji.

1.11.4 Dokumentacja funkcji

1.11.4.1 [RE_API int32_t avunit_external_devices_get_ptz_control \(ExternalDevicesHandle handle, PTZControlHandle * ptz_handle\)](#)

Pobranie uchwytu do sterowania ustawieniami kamer poprzez interfejs PTZ.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi
out	<i>ptz_handle</i>	Wskaźnik, do którego zostanie zapisany uchwyt do sterowania ustawieniami kamer

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano uchwyt do sterowania ustawieniami kamer

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL

[AVUNIT_WARNING_NO_SUCH_CONTROL](#) - kontrola sterowania kamerami poprzez interfejs PTZ jest niedostępna

Spostrzeżenia:

Niedostępność kontrolki jest spowodowana brakiem konfiguracji i/lub sprzętu dla danego elementu.

Funkcja może zostać zawołana po wywołaniu funkcji [avunit_open\(\)](#).

1.11.4.2 [RE_API](#) int32_t avunit_external_devices_get_carec_mic_control
([ExternalDevicesHandle](#) handle, [CarecMicControlHandle](#) * carec_handle)

Pobranie uchwytu do pobierania informacji z karty akwizycji danych audio.

Parametry:

in	handle	Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi
out	carec_handle	Wskaźnik, do którego zostanie zapisany uchwyt do komunikacji z kartami audio

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano uchwyt do komunikacji z kartami akwizycji dźwięku

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL

[AVUNIT_WARNING_NO_SUCH_CONTROL](#) - komunikacja z kartami audio jest niedostępna

Spostrzeżenia:

Niedostępność kontrolki jest spowodowana brakiem konfiguracji i/lub sprzętu dla danego elementu.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

1.11.4.3 [RE_API](#) int32_t avunit_external_devices_get_amplifier_control ([ExternalDevicesHandle](#) handle, [AmplifierControlHandle](#) * amplifier_handle)

Pobranie uchwytu do sterowania wzmacniaczem dźwięku.

Parametry:

in	handle	Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi
----	--------	--



out	<i>amplifier_handle</i>	Wskaźnik, do którego zostanie zapisany uchwyt do sterowania wzmacniaczem.
-----	-------------------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano uchwyt do sterowania wzmacniaczem dźwięku

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL

[AVUNIT_WARNING_NO_SUCH_CONTROL](#) - sterowanie wzmacniaczem jest niedostępne

Spostrzeżenia:

Niedostępność kontrolki jest spowodowana brakiem konfiguracji i/lub sprzętu dla danego elementu.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

1.11.4.4 [RE_API](#) int32_t avunit_external_devices_get_ups_control ([ExternalDevicesHandle handle](#), [UPSControlHandle](#) * ups_handle)

Pobranie uchwytu do sterowania zasilaczem awaryjnym UPS.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi
out	<i>ups_handle</i>	Wskaźnik, do którego zostanie zapisany uchwyt do sterowania zasilaczem awaryjnym.

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano uchwyt do sterowania zasilaczem awaryjnym

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL

[AVUNIT_WARNING_NO_SUCH_CONTROL](#) - sterowanie wzmacniaczem jest niedostępne

Spostrzeżenia:

Niedostępność kontrolki jest spowodowana brakiem konfiguracji i/lub sprzętu dla danego elementu.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

1.12 Sterowanie kamerami poprzez interfejs PTZ

Funkcje służące do sterowania kamerami poprzez interfejs PTZ.

1.12.1 Funkcje

- [RE_API int32_t avunit_ptz_control_count \(PTZControlHandle handle\)](#)
Pobranie ilości kamer skonfigurowanych do obsługi poprzez interfejs PTZ.
- [RE_API int32_t avunit_ptz_control_zoom_in \(PTZControlHandle handle, uint32_t id\)](#)
Wysłanie do kamery żądania przybliżenia obrazu.
- [RE_API int32_t avunit_ptz_control_zoom_out \(PTZControlHandle handle, uint32_t id\)](#)
Wysłanie do kamery żądania oddalenia obrazu.
- [RE_API int32_t avunit_ptz_control_focus_near \(PTZControlHandle handle, uint32_t id\)](#)
Wysłanie do kamery żądania ustawienia ostrości na bliższe obiekty.
- [RE_API int32_t avunit_ptz_control_focus_far \(PTZControlHandle handle, uint32_t id\)](#)
Wysłanie do kamery żądania ustawienia ostrości na dalsze obiekty.
- [RE_API int32_t avunit_ptz_control_stop \(PTZControlHandle handle, uint32_t id\)](#)
Wysłanie do kamery żądania zaprzestania zmian ostrości oraz przybliżenia.

1.12.2 Opis szczegółowy

Funkcje służące do sterowania kamerami poprzez interfejs PTZ.

Ta grupa funkcji obejmuje pobieranie informacji dotyczących liczby dostępnych kamer PTZ oraz kontrolę przybliżania obrazu i ustawień ostrości.

1.12.3 Dokumentacja funkcji

1.12.3.1 [RE_API int32_t avunit_ptz_control_count \(PTZControlHandle handle\)](#)

Pobranie ilości kamer skonfigurowanych do obsługi poprzez interfejs PTZ.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
----	---------------	--

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest ilością kamer skonfigurowanych do obsługi poprzez interfejs PTZ. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.12.3.2 [RE_API int32_t avunit_ptz_control_zoom_in \(PTZControlHandle handle, uint32_t id\)](#)

Wysłanie do kamery żądania przybliżenia obrazu.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
in	<i>id</i>	Indeks kamery, do której chcemy wysłać komendę

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę do kamery

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kamery jest niepoprawny

Spostrzeżenia:

Indeksy kamer są numerowane od 0 do wartości zwróconej przez funkcję [avunit_ptz_control_count\(\)](#) - 1.

Kamera będzie zmieniać przybliżenie do momentu zawołania funkcji [avunit_ptz_control_stop](#).

1.12.3.3 [RE_API](#) int32_t avunit_ptz_control_zoom_out ([PTZControlHandle](#) handle, uint32_t id)

Wysłanie do kamery żądania oddalenia obrazu.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
in	<i>id</i>	Indeks kamery, do której chcemy wysłać komendę

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę do kamery

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kamery jest niepoprawny

Spostrzeżenia:

Indeksy kamer są numerowane od 0 do wartości zwróconej przez funkcję [avunit_ptz_control_count\(\)](#) - 1.

Kamera będzie zmieniać przybliżenie do momentu zawołania funkcji [avunit_ptz_control_stop](#).

1.12.3.4 [RE_API](#) int32_t avunit_ptz_control_focus_near ([PTZControlHandle](#) handle, uint32_t id)

Wysłanie do kamery żądania ustawienia ostrości na bliższe obiekty.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
in	<i>id</i>	Indeks kamery, do której chcemy wysłać komendę

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę do kamery

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kamery jest niepoprawny

Spostrzeżenia:

Indeksy kamer są numerowane od 0 do wartości zwróconej przez funkcję [avunit_ptz_control_count\(\)](#) - 1.

Kamera będzie zmieniać ostrość do momentu zwołania funkcji [avunit_ptz_control_stop](#).

1.12.3.5 [RE_API int32_t avunit_ptz_control_focus_far \(PTZControlHandle handle, uint32_t id\)](#)

Wysłanie do kamery żądania ustawienia ostrości na dalsze obiekty.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
in	<i>id</i>	Indeks kamery, do której chcemy wysłać komendę

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę do kamery

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kamery jest niepoprawny

Spostrzeżenia:

Indeksy kamer są numerowane od 0 do wartości zwróconej przez funkcję [avunit_ptz_control_count\(\)](#) - 1.

Kamera będzie zmieniać ostrość do momentu zwołania funkcji [avunit_ptz_control_stop](#).

1.12.3.6 [RE_API int32_t avunit_ptz_control_stop \(PTZControlHandle handle, uint32_t id\)](#)

Wysłanie do kamery żądania zaprzestania zmian ostrości oraz przybliżenia.

Parametry:

in	<i>handle</i>	Uchwyt umożliwiający sterowanie kamerami
in	<i>id</i>	Indeks kamery, do której chcemy wysłać komendę

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę do kamery

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kamery jest niepoprawny

Spostrzeżenia:

Indeksy kamer są numerowane od 0 do wartości zwróconej przez funkcję [avunit_ptz_control_count\(\)](#) - 1.

Kamera będzie zmieniać ostrość/przybliżenie do momentu zwołania tej funkcji.



1.13 Komunikacja z kartami akwizycji dźwięku

Funkcje służące do komunikacji z kartami akwizycji dźwięku.

1.13.1 Funkcje

- [RE_API int32_t avunit_carec_mic_control_set_state_changed_callback](#) ([CarecMicControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [CarecMicrophoneStateChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia mikrofonu do karty audio.
- [RE_API int32_t avunit_carec_mic_control_remove_state_changed_callback](#) ([CarecMicControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_carec_mic_control_set_state_changed_callback](#).
- [RE_API int32_t avunit_carec_mic_control_count](#) ([CarecMicControlHandle](#) handle)
Pobranie ilości kanałów audio, które są skonfigurowane jako pochodzące z karty audio.
- [RE_API int32_t avunit_carec_mic_control_has](#) ([CarecMicControlHandle](#) handle, const char *channel_code)
Pobranie informacji, czy kanał audio z przekazanym kodem pochodzi z karty akwizycji audio.
- [RE_API int32_t avunit_carec_mic_control_state](#) ([CarecMicControlHandle](#) handle, const char *channel_code)
Pobranie informacji, czy do kanału audio z przekazanym kodem jest podłączony mikrofon.
- [RE_API int32_t avunit_carec_mic_control_parameter_count](#) ([CarecMicControlHandle](#) handle)
Pobranie ilości parametrów obsługiwanych przez kartę akwizycji audio dla każdego z kanałów.
- [RE_API int32_t avunit_carec_mic_control_parameter_type](#) ([CarecMicControlHandle](#) handle, uint32_t index)
Pobranie typu parametru pod wskazanym indeksem.
- [RE_API int32_t avunit_carec_mic_control_parameter_name](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char **name)
Pobranie nazwy parametru pod wskazanym indeksem.
- [RE_API int32_t avunit_carec_mic_control_parameter_integer](#) ([CarecMicControlHandle](#) handle, uint32_t index, [AVUnitParameterInteger](#) *parameter)
Pobranie dodatkowych informacji o parametrze typu liczbowego.
- [RE_API int32_t avunit_carec_mic_control_parameter_list](#) ([CarecMicControlHandle](#) handle, uint32_t index, [AVUnitParameterList](#) *parameter)
Pobranie dodatkowych informacji o parametrze typu lista.
- [RE_API int32_t avunit_carec_mic_control_parameter_get_value](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char *channel_code, int32_t *value)
Pobranie wartości parametru dla poszczególnego kanału audio.
- [RE_API int32_t avunit_carec_mic_control_parameter_set_value](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char *channel_code, int32_t value)
Ustawienie wartości parametru dla poszczególnego kanału audio.



- [RE_API int32_t avunit_carec_mic_control_parameters_save \(CarecMicControlHandle handle\)](#)
Zapisanie obecnie ustawionych parametrów na karcie akwizycji audio.
- [RE_API int32_t avunit_carec_mic_control_parameters_restore \(CarecMicControlHandle handle\)](#)
Odrzucenie obecnie ustawionych parametrów na karcie akwizycji audio i przywrócenie ustawień na niej zapisanych.

1.13.2 Opis szczegółowy

Funkcje służące do komunikacji z kartami akwizycji dźwięku.

Obsługiwane karty audio posiadają możliwość wykrycia, czy jest do nich podłączony mikrofon. Podczas startu aplikacji należy odczytać, czy do danego kanału jest on podłączony, czy też nie. W trakcie pracy aplikacji po wykryciu odłączenia/podłączenia mikrofonu do jednego z kanałów moduł AVUnit zawoła przekazany przez aplikację callback, przekazując informację o tym zdarzeniu.

1.13.3 Dokumentacja funkcji

1.13.3.1 [RE_API int32_t avunit_carec_mic_control_set_state_changed_callback \(CarecMicControlHandle handle, AVUnitCallbackStateHandle callback_state, CarecMicrophoneStateChanged callback\)](#)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia mikrofonu do karty audio.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji dźwięku
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<i>callback</i>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy



AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.13.3.2 [RE_API](#) int32_t avunit_carec_mic_control_remove_state_changed_callback
([CarecMicControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_carec_mic_control_set_state_changed_callback](#).

Parametry:

in	handle	Uchwyt do komunikacji z kartą akwizycji danych audio
in	callback_state	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

avunit_carec_mic_control_set_state_changed_callback

1.13.3.3 [RE_API](#) int32_t avunit_carec_mic_control_count ([CarecMicControlHandle](#) handle)

Pobranie ilości kanałów audio, które są skonfigurowane jako pochodzące z karty audio.

Parametry:

in	handle	Uchwyt do komunikacji z kartą akwizycji dźwięku
----	--------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest ilością kanałów audio pochodzących z karty akwizycji danych audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.13.3.4 [RE_API](#) int32_t avunit_carec_mic_control_has ([CarecMicControlHandle](#) handle, const char * channel_code)

Pobranie informacji, czy kanał audio z przekazanym kodem pochodzi z karty akwizycji audio.

Parametry:

in	handle	Uchwyt do komunikacji z kartą audio
in	channel_code	Kod kanału audio, dla którego chcemy sprawdzić, czy pochodzi z karty akwizycji dźwięku

Zwraca:

Wartość 1 oznacza, że kanał pochodzi z karty audio.

Wartość 0 oznacza, że kanał nie pochodzi z karty audio.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału audio ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - przekazany kod kanału audio jest niepoprawny

1.13.3.5 [RE_API](#) int32_t avunit_carec_mic_control_state ([CarecMicControlHandle](#) handle, const char * channel_code)

Pobranie informacji, czy do kanału audio z przekazanym kodem jest podłączony mikrofon.

Parametry:

in	handle	Uchwyt do komunikacji z kartą akwizycji audio
in	channel_code	Kod kanału audio, dla którego chcemy sprawdzić podłączenie mikrofonu

Zwraca:

Wartość 1 oznacza, że kanał posiada podłączony mikrofon.

Wartość 0 oznacza, że kanał nie posiada podłączonego mikrofonu.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kod kanału audio ma wartość NULL

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - kanał audio o przekazanym kodzie nie istnieje

[AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) - kanał audio o przekazanym kodzie nie jest podłączony do karty akwizycji dźwięku

1.13.3.6 [RE_API int32_t avunit_carec_mic_control_parameter_count \(CarecMicControlHandle handle\)](#)

Pobranie ilości parametrów obsługiwanych przez kartę akwizycji audio dla każdego z kanałów.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
----	---------------	---

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to otrzymana wartość jest ilością obsługiwanych parametrów. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

Spostrzeżenia:

Jeśli funkcja ta zwróci 0, to parametry nie są obsługiwane.

1.13.3.7 [RE_API int32_t avunit_carec_mic_control_parameter_type \(CarecMicControlHandle handle, uint32_t index\)](#)

Pobranie typu parametru pod wskazanym indeksem.

Typ parametru służy do wyświetlenia poprawnej kontrolki do zarządzania tym parametrem w GUI dla użytkownika.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to należy ją zinterpretować jako wartość wyliczenia [AVUnitParameterType](#).

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

1.13.3.8 [RE API](#) `int32_t avunit_carec_mic_control_parameter_name` ([CarecMicControlHandle handle, uint32_t index, const char ** name](#))

Pobranie nazwy parametru pod wskazanym indeksem.

Nazwa parametru służy do prezentacji jej w GUI dla użytkownika.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru
out	<i>name</i>	Wskaźnik, do którego zostanie zapisana nazwa parametru

Zwraca:

[AVUNIT SUCCESS](#) - pomyślnie pobrano nazwę parametru

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę parametru ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

Pamięć, w której zapisana jest nazwa parametru jest zarządzana przez AVUnit i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [avunit_close\(\)](#).

1.13.3.9 [RE API](#) `int32_t avunit_carec_mic_control_parameter_integer` ([CarecMicControlHandle handle, uint32_t index, AVUnitParameterInteger * parameter](#))

Pobranie dodatkowych informacji o parametrze typu liczbowego.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru
out	<i>parameter</i>	Wskaźnik na strukturę, do której zostaną zapisane informacje o parametrze

Zwraca:

[AVUNIT SUCCESS](#) - pomyślnie pobrano informacje o parametrze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na strukturę ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

[AVUNIT_ERROR_INVALID_TYPE](#) - parametr o podanym indeksie nie jest typu liczbowego

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

1.13.3.10 [RE API](#) `int32_t` `avunit_carec_mic_control_parameter_list`
([CarecMicControlHandle](#) *handle*, `uint32_t` *index*, [AVUnitParameterList](#) * *parameter*)

Pobranie dodatkowych informacji o parametrze typu lista.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru
out	<i>parameter</i>	Wskaźnik na strukturę, do której zostaną zapisane informacje o parametrze

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano informacje o parametrze

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na strukturę ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

[AVUNIT_ERROR_INVALID_TYPE](#) - parametr o podanym indeksie nie jest listą

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

1.13.3.11 [RE API](#) `int32_t` `avunit_carec_mic_control_parameter_get_value`
([CarecMicControlHandle](#) *handle*, `uint32_t` *index*, `const char *` *channel_code*, `int32_t *` *value*)

Pobranie wartości parametru dla poszczególnego kanału audio.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru
in	<i>channel_code</i>	Kod kanału audio, dla którego chcemy pobrać wartość parametru
out	<i>value</i>	Wskaźnik, do którego zostanie zapisana wartość parametru

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano wartość parametru

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na parametr ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

1.13.3.12 [RE_API](#) int32_t avunit_carec_mic_control_parameter_set_value
([CarecMicControlHandle](#) handle, uint32_t index, const char * channel_code, int32_t value)

Ustawienie wartości parametru dla poszczególnego kanału audio.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
in	<i>index</i>	Indeks parametru
in	<i>channel_code</i>	Kod kanału audio, dla którego chcemy ustawić wartość parametru
in	<i>value</i>	Wartość parametru do ustawienia

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano wartość parametru

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks jest spoza zakresu

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana wartość jest poza zakresem dla wskazanego parametru

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

Spostrzeżenia:

Indeksy parametrów są numerowane od 0 do wartości zwróconej przez funkcję [avunit_carec_mic_control_parameter_count\(\)](#) - 1.

Wartość parametru będzie zapisana na karcie akwizycji audio i użyta przy następnym uruchomieniu tylko w przypadku wywołania funkcji [avunit_carec_mic_control_parameters_save\(\)](#). W przypadku wywołania funkcji [avunit_carec_mic_control_parameters_restore\(\)](#) przywrócona zostanie poprzednio zapisana wartość.

Wartości przyjmowane przez funkcje należy zinterpretować następująco:

- dla parametrów typu logicznego (bool) należy podać wartość 0 (fałsz) lub 1 (prawda).
- dla parametrów typu liczbowego (integer) należy podać wartość z obsługiwanego przedziału.
- dla parametrów typu lista należy podać indeks wybranej wartości.

1.13.3.13 [RE API](#) `int32_t` [avunit_carec_mic_control_parameters_save](#)
([CarecMicControlHandle](#) *handle*)

Zapisanie obecnie ustawionych parametrów na karcie akwizycji audio.

Ustawienia po zapisie będą automatycznie ładowane i przywracane po ponownym uruchomieniu.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano ustawienia na karcie akwizycji audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

1.13.3.14 [RE API](#) `int32_t` [avunit_carec_mic_control_parameters_restore](#)
([CarecMicControlHandle](#) *handle*)

Odrzucenie obecnie ustawionych parametrów na karcie akwizycji audio i przywrócenie ustawień na niej zapisanych.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z kartą akwizycji audio
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zapisano ustawienia na karcie akwizycji audio

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - parametry nie są obsługiwane

1.14 Komunikacja ze wzmacniaczem

Funkcje służące do komunikacji ze wzmacniaczem mocy dźwięku.

1.14.1 Funkcje

- [RE_API int32_t avunit_amplifier_control_set_connected_changed_callback](#) ([AmplifierControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [DeviceConnectedChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do wzmacniacza dźwięku.
- [RE_API int32_t avunit_amplifier_control_remove_connected_changed_callback](#) ([AmplifierControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_amplifier_control_set_connected_changed_callback](#).
- [RE_API int32_t avunit_amplifier_control_is_connected](#) ([AmplifierControlHandle](#) handle)
Pobranie informacji, czy połączenie ze wzmacniaczem dźwięku jest aktywne.
- [RE_API int32_t avunit_amplifier_control_get_enabled](#) ([AmplifierControlHandle](#) handle, uint32_t channel)
Pobranie informacji, czy dany kanał wzmacniacza dźwięku jest włączony.
- [RE_API int32_t avunit_amplifier_control_get_source](#) ([AmplifierControlHandle](#) handle, uint32_t channel)
Pobranie źródła sygnału na danym kanale wzmacniacza dźwięku.
- [RE_API int32_t avunit_amplifier_control_get_volume](#) ([AmplifierControlHandle](#) handle, uint32_t channel, float *volume)
Pobranie ustawienia głośności sygnału na danym kanale wzmacniacza dźwięku.
- [RE_API int32_t avunit_amplifier_control_set_enabled](#) ([AmplifierControlHandle](#) handle, uint32_t channel, int32_t enabled)
Włączenie lub wyłączenie kanału wzmacniacza dźwięku.
- [RE_API int32_t avunit_amplifier_control_set_source](#) ([AmplifierControlHandle](#) handle, uint32_t channel, uint32_t source)
Ustawienie źródła sygnału dla danego kanału wzmacniacza dźwięku.
- [RE_API int32_t avunit_amplifier_control_set_volume](#) ([AmplifierControlHandle](#) handle, uint32_t channel, float volume)
Ustawienie głośności sygnału na danym kanale wzmacniacza dźwięku.

1.14.2 Opis szczegółowy

Funkcje służące do komunikacji ze wzmacniaczem mocy dźwięku.

Obsługiwany wzmacniacz dźwięku posiada 6 kanałów wyjściowych, którymi można sterować niezależnie. Dla każdego z wyjść możemy ustawić jego głośność oraz źródło sygnału - wspólne lub dedykowane. Każde z wyjść można także włączać i wyłączać niezależnie od pozostałych.

1.14.3 Dokumentacja funkcji

1.14.3.1 [RE_API](#) `int32_t` `avunit_amplifier_control_set_connected_changed_callback`
([AmplifierControlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`,
[DeviceConnectedChanged](#) `callback`)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do wzmacniacza dźwięku.

Parametry:

in	<code>handle</code>	Uchwyt do komunikacji ze wzmacniaczem
in	<code>callback_state</code>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<code>callback</code>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.14.3.2 [RE_API](#) `int32_t` `avunit_amplifier_control_remove_connected_changed_callback`
([AmplifierControlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_amplifier_control_set_connected_changed_callback](#).

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem dźwięku
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

[avunit_amplifier_control_set_connected_changed_callback](#)

1.14.3.3 [RE_API](#) `int32_t avunit_amplifier_control_is_connected` ([AmplifierControlHandle handle](#))

Pobranie informacji, czy połączenie ze wzmacniaczem dźwięku jest aktywne.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
----	---------------	---------------------------------------

Zwraca:

Wartość 1 oznacza, że połączenie do wzmacniacza jest aktywne.

Wartość 0 oznacza, że aktualnie brak połączenia ze wzmacniaczem.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.14.3.4 [RE_API](#) `int32_t avunit_amplifier_control_get_enabled` ([AmplifierControlHandle handle](#), `uint32_t channel`)

Pobranie informacji, czy dany kanał wzmacniacza dźwięku jest włączony.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
in	<i>channel</i>	Indeks kanału wzmacniacza

Zwraca:

Wartość 1 oznacza, że kanał wzmacniacza jest włączony.

Wartość 0 oznacza, że kanał wzmacniacza jest wyłączony.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.14.3.5 [RE API](#) `int32_t avunit_amplifier_control_get_source (AmplifierControlHandle handle, uint32_t channel)`

Pobranie źródła sygnału na danym kanale wzmacniacza dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
in	<i>channel</i>	Indeks kanału wzmacniacza

Zwraca:

Jeżeli wartość jest nieujemna, to należy ją zinterpretować jako wartość wyliczenia [AVUnitAmplifierSource](#).

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.14.3.6 [RE API](#) `int32_t avunit_amplifier_control_get_volume (AmplifierControlHandle handle, uint32_t channel, float * volume)`

Pobranie ustawienia głośności sygnału na danym kanale wzmacniacza dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
in	<i>channel</i>	Indeks kanału wzmacniacza
out	<i>volume</i>	Wskaźnik, pod który zostanie zapisana aktualna głośność kanału

Zwraca:

- [AVUNIT_SUCCESS](#) - pomyślnie odczytano głośność sygnału
- [AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na głośność ma wartość NULL
- [AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny
- [AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.14.3.7 [RE_API](#) `int32_t avunit_amplifier_control_set_enabled (AmplifierControlHandle handle, uint32_t channel, int32_t enabled)`

Włączenie lub wyłączenie kanału wzmacniacza dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
in	<i>channel</i>	Indeks kanału wzmacniacza
in	<i>enabled</i>	Wartość logiczna określająca, czy dany kanał chcemy włączyć (1) czy też wyłączyć (0)

Zwraca:

- [AVUNIT_SUCCESS](#) - pomyślnie odczytano głośność sygnału
- [AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny
- [AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem
- [AVUNIT_ERROR_OPERATION_FAILED](#) - nie udało się włączyć/wyłączyć kanału

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.14.3.8 [RE_API](#) `int32_t avunit_amplifier_control_set_source (AmplifierControlHandle handle, uint32_t channel, uint32_t source)`

Ustawienie źródła sygnału dla danego kanału wzmacniacza dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
----	---------------	---------------------------------------



in	<i>channel</i>	Indeks kanału wzmacniacza
in	<i>source</i>	Źródło sygnału jako wartość wyliczenia AVUnitAmplifierSource

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie odczytano głośność sygnału

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana wartość dla źródła sygnału jest niepoprawna

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem

[AVUNIT_ERROR_OPERATION_FAILED](#) - nie udało się ustawić źródła sygnału

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.14.3.9 [RE_API int32_t avunit_amplifier_control_set_volume](#) ([AmplifierControlHandle handle](#), [uint32_t channel](#), [float volume](#))

Ustawienie głośności sygnału na danym kanale wzmacniacza dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji ze wzmacniaczem
in	<i>channel</i>	Indeks kanału wzmacniacza
in	<i>volume</i>	Głośność sygnału, która ma zostać ustawiona jako wartość z przedziału [0.0, 1.0]

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono głośność sygnału

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia ze wzmacniaczem

[AVUNIT_ERROR_OPERATION_FAILED](#) - nie udało się ustawić głośności

Spostrzeżenia:

W związku z tym, że wzmacniacz posiada 6 kanałów indeksy kanałów są numerowane od 0 do 5.

1.15 Komunikacja z zasilaczem awaryjnym UPS

Funkcje służące do komunikacji z zasilaczem awaryjnym UPS.

1.15.1 Funkcje

- [RE_API int32_t avunit ups control set connected changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [DeviceConnectedChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do zasilacza awaryjnego UPS.
- [RE_API int32_t avunit ups control remove connected changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit ups control set connected changed callback](#).
- [RE_API int32_t avunit ups control set power source changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [StateChangedCallback](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany źródła zasilania zasilacza awaryjnego.
- [RE_API int32_t avunit ups control remove power source changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit ups control set power source changed callback](#).
- [RE_API int32_t avunit ups control is connected](#) ([UPSControlHandle](#) handle)
Pobranie informacji, czy połączenie z zasilaczem awaryjnym UPS jest aktywne.
- [RE_API int32_t avunit ups control power source](#) ([UPSControlHandle](#) handle)
Pobranie aktualnego źródła zasilania zasilacza awaryjnego UPS.
- [RE_API int32_t avunit ups control beeper enabled](#) ([UPSControlHandle](#) handle)
Pobranie informacji, czy w zasilaczu awaryjnym UPS jest włączony sygnał dźwiękowy informujący o przejściu na zasilanie awaryjne.
- [RE_API int32_t avunit ups control set beeper enabled](#) ([UPSControlHandle](#) handle, int32_t enabled)
Włączenie lub wyłączenie sygnału dźwiękowego informującego o przejściu zasilacza awaryjnego UPS na zasilanie awaryjne.
- [RE_API int32_t avunit ups control get battery charge](#) ([UPSControlHandle](#) handle, float *charge)
Pobranie aktualnego stanu naładowania baterii.
- [RE_API int32_t avunit ups control get run time estimate](#) ([UPSControlHandle](#) handle)
Pobranie aktualnej estymacji pozostałego czasu pracy na zasilaniu akumulatorowym.

1.15.2 Opis szczegółowy

Funkcje służące do komunikacji z zasilaczem awaryjnym UPS.

Obsługiwane urządzenie UPS posiada możliwość informowania o zaniku zasilania sieciowego oraz o parametrach pracy. Ta grupa funkcji obsługuje funkcjonalność udostępnianą przez zasilacz awaryjny.

1.15.3 Dokumentacja funkcji

1.15.3.1 [RE_API](#) `int32_t` [avunit_ups_control_set_connected_changed_callback](#) ([UPSCtrlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`, [DeviceConnectedChanged](#) `callback`)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do zasilacza awaryjnego UPS.

Parametry:

in	<code>handle</code>	Uchwyt do komunikacji z zasilaczem awaryjnym
in	<code>callback_state</code>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<code>callback</code>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.15.3.2 [RE_API](#) `int32_t` [avunit_ups_control_remove_connected_changed_callback](#) ([UPSCtrlHandle](#) `handle`, [AVUnitCallbackStateHandle](#) `callback_state`)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_ups_control_set_connected_changed_callback](#).

Parametry:

in	<code>handle</code>	Uchwyt do komunikacji z zasilaczem awaryjnym
----	---------------------	--



in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
----	-----------------------	--

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

avunit_ups_control_set_connected_changed_callback

1.15.3.3 [RE_API](#) int32_t avunit_ups_control_set_power_source_changed_callback
([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state,
[StateChangedCallback](#) callback)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany źródła zasilania zasilacza awaryjnego.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<i>callback</i>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Wartością przekazaną do funkcji jest wartość wyliczenia [AVUnitUPSPowerSource](#).

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy

AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.15.3.4 [RE_API](#) int32_t avunit_ups_control_remove_power_source_changed_callback ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_ups_control_set_power_source_changed_callback](#).

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

avunit_ups_control_set_power_source_changed_callback

1.15.3.5 [RE_API](#) int32_t avunit_ups_control_is_connected ([UPSControlHandle](#) handle)

Pobranie informacji, czy połączenie z zasilaczem awaryjnym UPS jest aktywne.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
----	---------------	--

Zwraca:

Wartość 1 oznacza, że połączenie do zasilacza awaryjnego jest aktywne.

Wartość 0 oznacza, że aktualnie brak połączenia z zasilaczem awaryjnym.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.15.3.6 [RE_API int32_t avunit_ups_control_power_source \(UPSControlHandle handle\)](#)

Pobranie aktualnego źródła zasilania zasilacza awaryjnego UPS.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
----	---------------	--

Zwraca:

Jeżeli wartość jest nieujemna, to należy ją zinterpretować jako wartość wyliczenia [AVUnitUPSPowerSource](#).

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia z zasilaczem

1.15.3.7 [RE_API int32_t avunit_ups_control_beeper_enabled \(UPSControlHandle handle\)](#)

Pobranie informacji, czy w zasilaczu awaryjnym UPS jest włączony sygnał dźwiękowy informujący o przejściu na zasilanie awaryjne.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
----	---------------	--

Zwraca:

Wartość 1 oznacza, że sygnał dźwiękowy zasilacza awaryjnego jest włączony.

Wartość 0 oznacza, że sygnał dźwiękowy zasilacza awaryjnego jest wyłączony.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia z zasilaczem

1.15.3.8 [RE_API int32_t avunit_ups_control_set_beeper_enabled \(UPSControlHandle handle, int32_t enabled\)](#)

Włączenie lub wyłączenie sygnału dźwiękowego informującego o przejściu zasilacza awaryjnego UPS na zasilanie awaryjne.

Parametry:

in	<i>handle</i>	Uchwyt do komunikacji z zasilaczem awaryjnym
in	<i>enabled</i>	Wartość logiczna określająca, czy sygnał dźwiękowy chcemy włączyć (1) czy też wyłączyć (0)

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie ustawiono stan sygnału dźwiękowego

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia z zasilaczem

1.15.3.9 [RE_API](#) int32_t avunit_ups_control_get_battery_charge ([UPSControlHandle](#) handle, float * charge)

Pobranie aktualnego stanu naładowania baterii.

Parametry:

in	handle	Uchwyt do komunikacji z zasilaczem awaryjnym
in	charge	Wskaźnik, do którego zostanie zapisany aktualny stan naładowania baterii jako wartość z przedziału [0.0, 1.0]

Zwraca:

[AVUNIT_SUCCESS](#) - poprawnie ustawiono stan sygnału dźwiękowego

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na stan naładowania baterii ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia z zasilaczem

1.15.3.10 [RE_API](#) int32_t avunit_ups_control_get_run_time_estimate ([UPSControlHandle](#) handle)

Pobranie aktualnej estymacji pozostałego czasu pracy na zasilaniu akumulatorowym.

Parametry:

in	handle	Uchwyt do komunikacji z zasilaczem awaryjnym
----	--------	--

Zwraca:

Jeśli wartość ≥ 0 , to jest to aktualna estymacja pozostałego czasu pracy na zasilaniu akumulatorowym wyrażona w sekundach. Jeśli aktualnie zasilacz awaryjny pracuje na zasilaniu sieciowym, to wartością zwracaną jest 0. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na stan naładowania baterii ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia z zasilaczem

1.16 Obsługa odtwarzacza monitora wielkoformatowego

Funkcje umożliwiają zdalną obsługę wyświetlacza monitora wielkoformatowego.

1.16.1 Funkcje

- [RE API int32_t avunit tv control set connected changed callback \(TVControlHandle handle, AVUnitCallbackStateHandle callback_state, DeviceConnectedChanged callback\)](#)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control remove connected changed callback \(TVControlHandle handle, AVUnitCallbackStateHandle callback_state\)](#)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit tv control set connected changed callback](#).
- [RE API int32_t avunit tv control wake_on_lan \(TVControlHandle handle, const char *mac, const char *address, uint32_t port\)](#)
Wysłanie pakietu Wake-on-LAN do odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control connect \(TVControlHandle handle, const char *address, uint32_t port\)](#)
Podłączenie się do odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control disconnect \(TVControlHandle handle\)](#)
Rozłączenie wcześniej nawiązanego połączenia z odtwarzaczem monitora wielkoformatowego.
- [RE API int32_t avunit tv control connected \(TVControlHandle handle\)](#)
Zwrócenie statusu połączenia z odtwarzaczem monitora wielkoformatowego.
- [RE API int32_t avunit tv control version \(TVControlHandle handle, uint32_t *major, uint32_t *minor, uint32_t *patch\)](#)
Pobranie wersji odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control start streaming \(TVControlHandle handle\)](#)
Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modulem AVUnit wykonującym to żądanie.
- [RE API int32_t avunit tv control start streaming to \(TVControlHandle handle, const char *address, uint32_t port\)](#)
Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modulem AVUnit znajdującym się pod podanym adresem i portem.
- [RE API int32_t avunit tv control close \(TVControlHandle handle\)](#)
Wysłanie komendy, która ma za zadanie wyłączyć odtwarzacz telewizyjny.

1.16.2 Opis szczegółowy

Funkcje umożliwiają zdalną obsługę wyświetlacza monitora wielkoformatowego.

Obsługa monitora wielkoformatowego obejmuje funkcjonalność Wake-on-LAN oraz nawiązywanie i zrywanie połączenia z odtwarzaczem.

1.16.3 Dokumentacja funkcji

1.16.3.1 [RE API int32_t avunit tv control set connected changed callback \(TVControlHandle handle, AVUnitCallbackStateHandle callback_state, DeviceConnectedChanged callback\)](#)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do odtwarzacza monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state
in	<i>callback</i>	Poprawny callback, który zostanie zawołany w momencie zmiany stanu

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Callback zostanie wywołany zawsze wtedy, gdy stan ulegnie zmianie. Zmiana stanu może nastąpić na skutek wywołania odpowiedniej funkcji przez użytkownika lub przez wewnętrzne procesy AVUnit. Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Funkcja może zostać zawołana tylko po wywołaniu funkcji [avunit_open\(\)](#).

Zobacz również:

[avunit_allocate_callback_state](#)

1.16.3.2 [RE_API](#) `int32_t` [avunit_tv_control_remove_connected_changed_callback](#)
([TVControlHandle](#) *handle*, [AVUnitCallbackStateHandle](#) *callback_state*)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_tv_control_set_connected_changed_callback](#).

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
in	<i>callback_state</i>	Uchwyt do obiektu utworzonego za pomocą avunit_allocate_callback_state

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie usunięto callback

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany obiekt stanu wywołań zwrotnych ma wartość NULL

Spostrzeżenia:

Funkcja musi zostać zawołana przed wywołaniem funkcji [avunit_close\(\)](#).

Zobacz również:

avunit_tv_control_set_connected_changed_callback

1.16.3.3 [RE_API](#) int32_t avunit_tv_control_wake_on_lan ([TVControlHandle](#) handle, const char * mac, const char * address, uint32_t port)

Wysłanie pakietu Wake-on-LAN do odtwarzacza monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
in	<i>mac</i>	Adres MAC karty sieciowej odtwarzacza w formacie "00:00:00:00:00:00"
in	<i>address</i>	Adres IP, pod który zostanie wysłany pakiet
in	<i>port</i>	Port, pod który zostanie wysłany pakiet

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano pakiet

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres MAC ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres MAC jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany port jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - wysłanie pakietu nie powiodło się

Spostrzeżenia:

Aby sprzęt sieciowy poprawnie przekazał pakiet do wszystkich urządzeń, zalecane jest podanie IP broadcast sieci, do której podpięty jest odtwarzacz.

Obsługiwane są adresy IPv4 oraz adresy IPv6.

1.16.3.4 [RE_API](#) int32_t avunit_tv_control_connect ([TVControlHandle](#) handle, const char * address, uint32_t port)

Podłączenie się do odtwarzacza monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
in	<i>address</i>	Adres IP, pod którym odtwarzacz telewizyjny nasłuchuje połączeń
in	<i>port</i>	Port, pod którym odtwarzacz telewizyjny nasłuchuje połączeń

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie połączono do odtwarzacza

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany port jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - połączenie do odtwarzacza nie powiodło się

Spostrzeżenia:

Obsługiwane są adresy IPv4 oraz adresy IPv6.

Moduł po nawiązaniu połączenia stale je monitoruje i w wypadku jego zerwania automatycznie próbuje je wznowić. O zmianie stanu połączenia aplikacja jest informowana poprzez wywołanie zwrotne (callback).

1.16.3.5 [RE_API](#) int32_t avunit_tv_control_disconnect ([TVControlHandle](#) handle)

Rozłączenie wcześniej nawiązanego połączenia z odtwarzaczem monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie rozłączono z odtwarzaczem

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.16.3.6 [RE_API](#) int32_t avunit_tv_control_connected ([TVControlHandle](#) handle)

Zwrócenie statusu połączenia z odtwarzaczem monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
----	---------------	---

Zwraca:

Wartość 1 oznacza, że połączenie z odtwarzaczem monitora wielkoformatowego jest nawiązane.

Wartość 0 oznacza, że połączenie z odtwarzaczem monitora wielkoformatowego nie jest nawiązane.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.16.3.7 [RE_API](#) int32_t avunit_tv_control_version ([TVControlHandle](#) handle, uint32_t * major, uint32_t * minor, uint32_t * patch)

Pobranie wersji odtwarzacza monitora wielkoformatowego.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
out	<i>major</i>	Główny numer wersji
out	<i>minor</i>	Mniejszy numer wersji
out	<i>patch</i>	Numer poprawkowy

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie pobrano wersję odtwarzacza

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na numer wersji głównej ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na mniejszy numer wersji ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na poprawkowy numer wersji ma wartość NULL

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia do odtwarzacza

1.16.3.8 [RE_API int32_t avunit_tv_control_start_streaming \(TVControlHandle handle\)](#)

Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modułem AVUnit wykonującym to żądanie.

Po otrzymaniu tej komendy odtwarzacz na monitorze wielkoformatowym powinien zacząć wyświetlać na żywo podgląd obrazu i dźwięku rejestrowanego przez AVUnit.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - przestanie komendy nie powiodło się

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia do odtwarzacza

1.16.3.9 [RE_API int32_t avunit_tv_control_start_streaming_to \(TVControlHandle handle, const char * address, uint32_t port\)](#)

Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modułem AVUnit znajdującym się pod podanym adresem i portem.

Po otrzymaniu tej komendy odtwarzacz na monitorze wielkoformatowym powinien zacząć wyświetlać na żywo podgląd obrazu i dźwięku rejestrowanego przez AVUnit znajdujący się pod podanym adresem i portem.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
in	<i>address</i>	Adres IP komputera, pod którym znajduje się AVUnit, z którego chcemy wyświetlić podgląd
in	<i>port</i>	Port pod którym znajduje się AVUnit, z którego chcemy wyświetlić podgląd

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres IP jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany port jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - przesłanie komendy nie powiodło się

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia do odtwarzacza

Spostrzeżenia:

Obsługiwane są adresy IPv4 oraz adresy IPv6.

1.16.3.10 [RE_API int32_t avunit_tv_control_close \(TVControlHandle handle\)](#)

Wysłanie komendy, która ma za zadanie wyłączyć odtwarzacz telewizyjny.

Po otrzymaniu tej komendy odtwarzacz na monitorze wielkoformatowym powinien się wyłączyć. Połączenie zostaje rozłączone, podobnie jak przy użyciu funkcji [avunit_tv_control_disconnect](#).

Parametry:

in	handle	Uchwyt do zarządzania odtwarzaczem monitora wielkoformatowego
----	--------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie wysłano komendę

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - przesłanie komendy nie powiodło się

[AVUNIT_ERROR_INVALID_STATE](#) - brak połączenia do odtwarzacza

1.17 Obsługa odczytywania tekstu

Funkcje umożliwiają odczyt przekazanego tekstu.

1.17.1 Funkcje

- [RE_API int32_t avunit_speech_synthesis_control_read_text \(SpeechSynthesisControlHandle handle, const char *text\)](#)

Rozpoczęcie odczytywania tekstu przez syntezytor mowy.

- [RE_API int32_t avunit_speech_synthesis_control_stop_reading \(SpeechSynthesisControlHandle handle\)](#)

Zakończenie odczytywania tekstu przez syntezytor mowy.

- [RE_API int32_t avunit_speech_synthesis_control_gong_enabled \(SpeechSynthesisControlHandle handle\)](#)

Pobranie informacji, czy odtwarzanie dźwięku gongu jest włączone.

- [RE_API int32_t avunit_speech_synthesis_control_set_gong_enabled \(SpeechSynthesisControlHandle handle, int32_t enabled\)](#)

Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytanym tekstem.

1.17.2 Opis szczegółowy

Funkcje umożliwiają odczyt przekazanego tekstu.

Moduł syntezy mowy przyjmuje tekst w formacie UTF-8 i odczytuje jego treść. Moduł umożliwia także odtworzenie dźwięku gongu przed odczytem tekstu.

1.17.3 Dokumentacja funkcji

1.17.3.1 [RE_API](#) `int32_t` `avunit_speech_synthesis_control_read_text`
([SpeechSynthesisControlHandle](#) `handle`, `const char * text`)

Rozpoczęcie odczytywania tekstu przez syntezytor mowy.

Po wywołaniu tej funkcji syntezytor rozpocznie odczytywanie tekstu. Jeśli dźwięk gongu jest aktywny, zostanie on odegrany zaraz przed odczytanym tekstem.

Parametry:

in	<code>handle</code>	Uchwyt do zarządzania syntezą mowy
in	<code>text</code>	Tekst, który zostanie odczytany (w formacie UTF-8)

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie rozpoczęto odczyt tekstu

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany tekst ma wartość NULL

[AVUNIT_ERROR_OPERATION_FAILED](#) - rozpoczęcie odczytu nie powiodło się

Spostrzeżenia:

Jeśli aktualnie jest odczytywany inny tekst, jego odczyt zostanie zakończony i rozpocznie się odczyt przekazanego tekstu.

1.17.3.2 [RE_API](#) `int32_t` `avunit_speech_synthesis_control_stop_reading`
([SpeechSynthesisControlHandle](#) `handle`)

Zakończenie odczytywania tekstu przez syntezytor mowy.

Funkcja ta służy do przerywania odczytywania tekstu rozpoczętego za pomocą wcześniejszego wywołania funkcji [avunit_speech_synthesis_control_read_text\(\)](#).

Parametry:

in	<code>handle</code>	Uchwyt do zarządzania syntezą mowy
----	---------------------	------------------------------------

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie zakończono odczyt tekstu

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

Spostrzeżenia:

Jeśli aktualnie nie jest odczytywany żaden tekst, to funkcja ta nie zrobi nic.



1.17.3.3 [RE API](#) `int32_t` `avunit_speech_synthesis_control_gong_enabled` ([SpeechSynthesisControlHandle](#) *handle*)

Pobranie informacji, czy odtwarzanie dźwięku gongu jest włączone.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania syntezą mowy
----	---------------	------------------------------------

Zwraca:

Wartość 1 oznacza, że odtwarzanie dźwięku gongu jest włączone.

Wartość 0 oznacza, że odtwarzanie dźwięku gongu jest wyłączone.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.17.3.4 [RE API](#) `int32_t` `avunit_speech_synthesis_control_set_gong_enabled` ([SpeechSynthesisControlHandle](#) *handle*, `int32_t` *enabled*)

Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytanym tekstem.

Parametry:

in	<i>handle</i>	Uchwyt do zarządzania syntezą mowy
in	<i>enabled</i>	Wartość logiczna określająca, czy chcemy włączyć (1) czy też wyłączyć (0) odtwarzanie dźwięku gongu

Zwraca:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.18 Dostępne miejsce na dysku

Funkcje umożliwiają uzyskanie informacji o ilości pozostałego miejsca na dysku.

1.18.1 Funkcje

- [RE API](#) `int32_t` [avunit_free_space_control_set_recording_parameters](#)
([FreeSpaceControlHandle](#) *handle*, `const char *recording_dir_path`, `const char *audio_codec`, `const char *video_codec`, `float audio_quality`, `float video_quality`, `int32_t video_enabled`)
Ustawienie parametrów nagrywania, które będą miały wpływ na obliczenia pozostałego miejsca na dysku.
- [RE API](#) `int64_t` [avunit_free_space_control_total_bytes](#) ([FreeSpaceControlHandle](#) *handle*)
Pobranie całkowitego rozmiaru dysku w bajtach.
- [RE API](#) `int64_t` [avunit_free_space_control_total_minutes](#) ([FreeSpaceControlHandle](#) *handle*)
Pobranie całkowitego rozmiaru dysku w minutach.
- [RE API](#) `int64_t` [avunit_free_space_control_free_space_in_bytes](#) ([FreeSpaceControlHandle](#) *handle*)
Pobranie ilości pozostałego miejsca na dysku wyrażonego w bajtach.

- [RE_API int64_t avunit free space control free space in minutes \(FreeSpaceControlHandle handle\)](#)
Pobranie ilości pozostałego miejsca na dysku wyrażonego w minutach.
- [RE_API int64_t avunit free space control bytes per minute \(FreeSpaceControlHandle handle\)](#)
Pobranie ilości bajtów nagrania na każdą minutę.

1.18.2 Opis szczegółowy

Funkcje umożliwiają uzyskanie informacji o ilości pozostałego miejsca na dysku.

Informacja ta może zostać zwrócona zarówno w postaci ilości wolnych bajtów, jak i również w postaci ilości minut pozostałych na nagrania.

1.18.3 Dokumentacja funkcji

1.18.3.1 [RE_API int32_t avunit_free_space_control_set_recording_parameters \(FreeSpaceControlHandle handle, const char * recording_dir_path, const char * audio_codec, const char * video_codec, float audio_quality, float video_quality, int32_t video_enabled\)](#)

Ustawienie parametrów nagrywania, które będą miały wpływ na obliczenia pozostałego miejsca na dysku.

Parametry:

in	<i>handle</i>	Uchwyt do pobierania informacji o dostępnym miejscu
in	<i>recording_dir_path</i>	Pełna ścieżka do katalogu, w którym zapisane zostają nagrania; ścieżka użyta zostaje do odczytu dostępnego miejsca
in	<i>audio_codec</i>	Nazwa kodeka, który zostanie użyty do zapisu dźwięku.
in	<i>video_codec</i>	Nazwa kodeka, który zostanie użyty do zapisu obrazu.
in	<i>audio_quality</i>	Jakość zapisywanego dźwięku jako wartość z przedziału [0.0, 1.0]
in	<i>video_quality</i>	Jakość zapisywanego obrazu jako wartość z przedziału [0.0, 1.0]
in	<i>video_enabled</i>	Wartość logiczna określająca, czy zapisywać do pliku obraz i dźwięk (wartość 1) czy tylko sam dźwięk (wartość 0).

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie przekazano parametry nagrywania

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka do katalogu ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek audio ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany kodek wideo ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka do katalogu jest niepoprawna

[AVUNIT_ERROR_INVALID_AUDIO_CODEC](#) - przekazany kodek audio jest nieobsługiwany

[AVUNIT_ERROR_INVALID_VIDEO_CODEC](#) - przekazany kodek wideo jest nieobsługiwany

Spostrzeżenia:

Parametry nagrywania powinny być takie same jak te przekazywane do funkcji [avunit_encoder_control_open](#). Ścieżka do katalogu powinna wskazywać do nadrzędnego katalogu, w którym zostają utworzone pliki nagrań.

Funkcja ta musi zostać zawołana przed próbą wywołania innych funkcji pobierających informacje o dostępnym miejscu na dysku.

Zobacz również:

[avunit_encoder_control_open](#)

1.18.3.2 [RE API](#) int64_t avunit_free_space_control_total_bytes ([FreeSpaceControlHandle handle](#))

Pobranie całkowitego rozmiaru dysku w bajtach.

Parametry:

in	handle	Uchwyt do pobierania informacji o dostępnym miejscu
----	--------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to wielkość dysku w bajtach. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie przekazano poprawnych parametrów nagrywania

Zobacz również:

avunit_free_space_control_set_recording_parameters

1.18.3.3 [RE API](#) int64_t avunit_free_space_control_total_minutes ([FreeSpaceControlHandle handle](#))

Pobranie całkowitego rozmiaru dysku w minutach.

Parametry:

in	handle	Uchwyt do pobierania informacji o dostępnym miejscu
----	--------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość minut nagrania, na które pozwala cała pojemność dysku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie przekazano poprawnych parametrów nagrywania

Zobacz również:

avunit_free_space_control_set_recording_parameters

1.18.3.4 [RE_API](#) int64_t avunit_free_space_control_free_space_in_bytes
([FreeSpaceControlHandle](#) handle)

Pobranie ilości pozostałego miejsca na dysku wyrażonego w bajtach.

Parametry:

in	handle	Uchwyt do pobierania informacji o dostępnym miejscu
----	--------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość pozostałych bajtów na dysku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie przekazano poprawnych parametrów nagrywania

Zobacz również:

avunit_free_space_control_set_recording_parameters

1.18.3.5 [RE_API](#) int64_t avunit_free_space_control_free_space_in_minutes
([FreeSpaceControlHandle](#) handle)

Pobranie ilości pozostałego miejsca na dysku wyrażonego w minutach.

Parametry:

in	handle	Uchwyt do pobierania informacji o dostępnym miejscu
----	--------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość minut nagrania, na które pozwala dostępne miejsce. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie przekazano poprawnych parametrów nagrywania

Zobacz również:

avunit_free_space_control_set_recording_parameters

1.18.3.6 [RE API](#) `int64_t` `avunit_free_space_control_bytes_per_minute`
([FreeSpaceControlHandle](#) `handle`)

Pobranie ilości bajtów nagrania na każdą minutę.

Parametry:

in	<code>handle</code>	Uchwyt do pobierania informacji o dostępnym miejscu
----	---------------------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość bajtów, które zajęte będzie przez jedną minutę nagrania. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie przekazano poprawnych parametrów nagrywania

Zobacz również:

`avunit_free_space_control_set_recording_parameters`

1.19 Klient NTP

Funkcje umożliwiające synchronizację czasu z serwerem (serwerami) NTP.

1.19.1 Funkcje

- [RE API](#) `int32_t` [avunit_ntp_client_create](#) ([NTPClientHandle](#) *`handle`)
Utworzenie obiektu klienta NTP.
- [RE API](#) `int32_t` [avunit_ntp_client_destroy](#) ([NTPClientHandle](#) `handle`)
Zniszczenie obiektu klienta NTP stworzonego za pomocą funkcji [avunit_ntp_client_create](#).
- [RE API](#) `int32_t` [avunit_ntp_client_add_server](#) ([NTPClientHandle](#) `handle`, `const char` *`address`, `uint32_t` `port`)
Dodanie serwera NTP, z którym będzie przeprowadzana synchronizacja czasu.
- [RE API](#) `int32_t` [avunit_ntp_client_synchronize](#) ([NTPClientHandle](#) `handle`)
Synchronizacja zegara systemowego z podanymi wcześniej serwerami NTP.
- [RE API](#) `int32_t` [avunit_set_system_time](#) (`uint64_t` `seconds`, `uint32_t` `nanoseconds`)
Ustawienie czasu systemowego na przekazaną wartość.

1.19.2 Opis szczegółowy

Funkcje umożliwiające synchronizację czasu z serwerem (serwerami) NTP.

Funkcjonalność ta obejmuje stworzenie obiektu klienta, dodanie jednego lub kilku serwerów czasu (NTP) oraz synchronizację zegara systemowego. Poza wspomnianą funkcjonalnością udostępniona jest także możliwość ustawienia czasu systemowego na wskazaną wartość.

1.19.3 Dokumentacja funkcji

1.19.3.1 [RE API](#) `int32_t` [avunit_ntp_client_create](#) ([NTPClientHandle](#) * `handle`)

Utworzenie obiektu klienta NTP.

Parametry:

out	<i>handle</i>	Wskaźnik, do którego zostanie zapisany uchwyt klienta NTP
-----	---------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - utworzono nowy obiekt klienta NTP

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt jest niepoprawny

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację

Spostrzeżenia:

W przypadku pomyślnego wykonania tej funkcji obiekt klienta należy zniszczyć za pomocą funkcji [avunit_ntp_client_destroy](#).

1.19.3.2 [RE_API](#) `int32_t avunit_ntp_client_destroy (NTPClientHandle handle)`

Zniszczenie obiektu klienta NTP stworzonego za pomocą funkcji [avunit_ntp_client_create](#).

Parametry:

in	<i>handle</i>	Uchwyt do klienta NTP
----	---------------	-----------------------

Zwraca:

[AVUNIT_SUCCESS](#) - zwolniono obiekt klienta NTP

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.19.3.3 [RE_API](#) `int32_t avunit_ntp_client_add_server (NTPClientHandle handle, const char * address, uint32_t port)`

Dodanie serwera NTP, z którym będzie przeprowadzana synchronizacja czasu.

Parametry:

in	<i>handle</i>	Uchwyt do klienta NTP
in	<i>address</i>	Adres do serwera NTP
in	<i>port</i>	Port serwera, na którym działa usługa NTP (domyślnie 123)

Zwraca:

[AVUNIT_SUCCESS](#) - zwolniono obiekt klienta NTP

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany adres serwera ma wartość NULL

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany port jest niepoprawny

[AVUNIT_ERROR_OPERATION_FAILED](#) - nie udało się rozstrzygnąć adresu IP serwera na podstawie przekazanego adresu

Spostrzeżenia:

Możliwe jest dodanie kilku serwerów poprzez wielokrotne wywołanie tej funkcji z różnymi adresami.

Adres serwera może być podany jako IPv4, IPv6 lub nazwa hosta.

1.19.3.4 [RE_API int32_t avunit_ntp_client_synchronize \(NTPClientHandle handle\)](#)

Synchronizacja zegara systemowego z podanymi wcześniej serwerami NTP.

Parametry:

in	<i>handle</i>	Uchwyt do klienta NTP
----	---------------	-----------------------

Zwraca:

[AVUNIT_SUCCESS](#) - zwolniono obiekt klienta NTP

[AVUNIT_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[AVUNIT_ERROR_INVALID_STATE](#) - nie podano żadnego serwera NTP, z którym można przeprowadzić synchronizację

[AVUNIT_ERROR_OPERATION_FAILED](#) - synchronizacja nie powiodła się - żaden z dodanych serwerów nie odpowiedział w przewidzianym czasie

Spostrzeżenia:

Przed wywołaniem tej funkcji należy dodać co najmniej jeden serwer NTP do synchronizacji.

1.19.3.5 [RE_API int32_t avunit_set_system_time \(uint64_t seconds, uint32_t nanoseconds\)](#)

Ustawienie czasu systemowego na przekazaną wartość.

Funkcja umożliwia ustawienie czasu systemowego na podaną wartość. Przekazane parametry odpowiadają formatowi czasu uniksowego.

Parametry:

in	<i>seconds</i>	Ilość sekund od 1970 roku UTC (czas uniksowy)
in	<i>nanoseconds</i>	Ilość nanosekund; liczba z przedziału [0, 999999999]

Zwraca:

[AVUNIT_SUCCESS](#) - ustawiono czas systemowy na przekazaną wartość

[AVUNIT_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

[AVUNIT_ERROR_INVALID_PARAMETER](#) - ilość nanosekund jest większa lub równa 1000000000 (1s)



AVUNIT_ERROR_OPERATION_FAILED - ustawienie czasu systemowego nie powiodło się

Spostrzeżenia:

Wartości powinny wskazywać czas w UTC.

1.20 Obsługa błędów krytycznych

Funkcjonalność obsługująca błędy krytyczne.

1.20.1 Definicje typów

- typedef void(* [FatalErrorCallback](#)) (int32_t fatal_error_code)
Callback, który jest wywoływany w przypadku wystąpienia błędu krytycznego.

1.20.2 Wyliczenia

- enum [AVUnitFatalError](#) { [AVUNIT_FATAL_ERROR_AUDIO_ENCODER_OVERLOADED](#) = 0, [AVUNIT_FATAL_ERROR_VIDEO_ENCODER_OVERLOADED](#) = 1, [AVUNIT_FATAL_ERROR_AUDIO_MIXER_OVERLOADED](#) = 2, [AVUNIT_FATAL_ERROR_VIDEO_MIXER_OVERLOADED](#) = 3, [AVUNIT_FATAL_ERROR_AUDIO_NOT_RECORDING](#) = 4, [AVUNIT_FATAL_ERROR_VIDEO_NOT_RECORDING](#) = 5, [AVUNIT_FATAL_ERROR_VIDEO_MIXER_FAILED](#) = 6 } *Enumeracja możliwych wartości błędu krytycznego.*

1.20.3 Funkcje

- [RE_API](#) int32_t [avunit_fatal_error_set_callback](#) ([FatalErrorCallback](#) callback)
Ustawienie callbacka, który zostanie zawołany w przypadku wystąpienia błędu krytycznego.

1.20.4 Opis szczegółowy

Funkcjonalność obsługująca błędy krytyczne.

1.20.5 Dokumentacja definicji typów

1.20.5.1 typedef void(* [FatalErrorCallback](#)) (int32_t fatal_error_code)

Callback, który jest wywoływany w przypadku wystąpienia błędu krytycznego.

Parametry:

<code>fatal_error_code</code>	kod, który jest identyczny z wartością enumeracji <code>AVUnitFatalError</code>
-------------------------------	---

Zobacz również:

[AVUnitFatalError](#)

Definicja w linii [657](#) pliku [AVUnit.h](#).

1.20.6 Dokumentacja typów wyliczanych

1.20.6.1 enum [AVUnitFatalError](#)

Enumeracja możliwych wartości błędu krytycznego.

Wartości te przekazywane są do callbacka wywoływanego w razie wystąpienia błędu krytycznego.

Zobacz również:

[avunit_fatal_error_set_callback](#)

Wartości wyliczeń

AVUNIT_FATAL_ERROR_AUDIO_ENCODER_OVERLOADED Enkoder audio nie nadąża z procesowaniem danych.

AVUNIT_FATAL_ERROR_VIDEO_ENCODER_OVERLOADED Enkoder wideo nie nadąża z procesowaniem danych.

AVUNIT_FATAL_ERROR_AUDIO_MIXER_OVERLOADED Mikser audio nie nadąża z procesowaniem danych.

AVUNIT_FATAL_ERROR_VIDEO_MIXER_OVERLOADED Mikser wideo nie nadąża z procesowaniem danych.

AVUNIT_FATAL_ERROR_AUDIO_NOT_RECORDING Błąd pozyskiwania danych audio ze źródeł

AVUNIT_FATAL_ERROR_VIDEO_NOT_RECORDING Błąd pozyskiwania danych wideo ze źródeł

AVUNIT_FATAL_ERROR_VIDEO_MIXER_FAILED Błąd miksowania danych wideo.

Definicja w linii [256](#) pliku [AVUnit.h](#).

1.20.7 Dokumentacja funkcji

1.20.7.1 [RE_API](#) `int32_t avunit_fatal_error_set_callback` ([FatalErrorCallback](#) *callback*)

Ustawienie callbacka, który zostanie zawołany w przypadku wystąpienia błędu krytycznego.

Callback jest globalny dla wszystkich instancji AVUnit i można go utworzyć przed utworzeniem jakiegokolwiek obiektu.

Parametry:

<i>callback</i>	Poprawny callback, który zostanie zawołany w momencie wystąpienia błędu krytycznego
-----------------	---

Zwraca:

[AVUNIT_SUCCESS](#) - pomyślnie ustawiono callback

[AVUNIT_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL

Spostrzeżenia:

Funkcja obsługująca callback musi być przygotowana na jej wywołanie w dowolnym momencie. Wątek na którym zostaje wywołany callback nie może zostać blokowany przez funkcję obsługującą callback.

Zobacz również:

[FatalErrorCallback](#)



2 Dokumentacja struktur danych

2.1 Dokumentacja struktury AVUnitCrop

Struktura, która pozwala na ustawienie kadrowania klatki źródłowej w mikserze wideo.

```
#include <AVUnit.h>
```

2.1.1 Pola danych

- float [offset_x](#)
położenie względem osi X jako ułamek szerokości klatki źródłowej (wartość z przedziału [0.0, 1.0])
- float [offset_y](#)
położenie względem osi Y jako ułamek wysokości klatki źródłowej (wartość z przedziału [0.0, 1.0])
- float [zoom](#)
powiększenie określone jako ułamek wysokości i szerokości klatki źródłowej (wartość z przedziału [0.0, 1.0])

2.1.2 Opis szczegółowy

Struktura, która pozwala na ustawienie kadrowania klatki źródłowej w mikserze wideo.

Nota:

Punkt (0.0, 0.0) znajduje się w lewym górnym rogu ekranu. Offsety podane są jako wartości z przedziału [0.0, 1.0] i określają przesunięcie lewego górnego rogu widoku wewnątrz klatki źródłowej, jako ułamek jej wielkości. Powiększenie jest wartością z przedziału [0.0, 1.0] i określa ułamek wysokości i szerokości klatki źródłowej

Przykłady:

`offset_x = 0.0, offset_y = 0,0, zoom = 1.0` – brak kadrowania, wyświetlenie klatki w całości

`offset_x = 0.5, offset_y = 0.5, zoom = 0.5` – w kadrze znajdzie się prostokąt mający swój lewy górny róg dokładnie w środku obrazu źródłowego oraz wysokość i szerokość równe połowie jego rozmiaru (czyli prawą dolną ćwiartkę klatki źródłowej)

Definicja w linii [688](#) pliku [AVUnit.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [AVUnit.h](#)

2.2 Dokumentacja struktury AVUnitParameterInteger

Struktura, która opisuje parametr typu liczbowego.

```
#include <AVUnit.h>
```

2.2.1 Pola danych

- int32_t [min](#)
minimalna wartość przyjmowana przez parametr
- int32_t [max](#)

maksymalna wartość przyjmowana przez parametr

2.2.2 Opis szczegółowy

Struktura, która opisuje parametr typu liczbowego.

Definicja w linii [713](#) pliku [AVUnit.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [AVUnit.h](#)



2.3 Dokumentacja struktury AVUnitParameterList

Struktura, która opisuje parametr typu lista.

```
#include <AVUnit.h>
```

2.3.1 Pola danych

- `const char ** values`
tablica nazw wartości parametru do wyświetlenia w GUI; tablica posiada [AVUnitParameterList::value_count](#) elementów
- `uint32_t value_count`
ilość opcji do wyboru oraz ilość wpisów w tablicy [AVUnitParameterList::values](#)

2.3.2 Opis szczegółowy

Struktura, która opisuje parametr typu lista.

Spostrzeżenia:

Nie należy zwalniać pamięci zaalokowanej na pola tej struktury. Pamięć jest zarządzana przez AVUnit. Opcje są numerowane od 0 do wartości [AVUnitParameterList::value_count](#) - 1.

Definicja w linii [702](#) pliku [AVUnit.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [AVUnit.h](#)

2.4 Dokumentacja struktury AVUnitRect

Struktura, która pozwala na ustawienie położenia oraz rozmiaru klatek wideo w mikserze.

```
#include <AVUnit.h>
```

2.4.1 Pola danych

- `uint32_t x`
położenie względem osi X
- `uint32_t y`
położenie względem osi Y
- `uint32_t width`
szerokość klatki video
- `uint32_t height`
wysokość klatki video

2.4.2 Opis szczegółowy

Struktura, która pozwala na ustawienie położenia oraz rozmiaru klatek wideo w mikserze.

Nota:

Punkt (0, 0) znajduje się w lewym górnym rogu ekranu

Definicja w linii [663](#) pliku [AVUnit.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:



UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



- AVUnit.h

3 Dokumentacja plików

3.1 Dokumentacja pliku AVUnit.h

Plik zawierający publiczne API dla modułu AVUnit.

```
#include <stdint.h>
```

3.1.1 Struktury danych

- struct [AVUnitRect](#)
Struktura, która pozwala na ustawienie położenia oraz rozmiaru klatek wideo w mikserze.
- struct [AVUnitCrop](#)
Struktura, która pozwala na ustawienie kadrowania klatki źródłowej w mikserze wideo.
- struct [AVUnitParameterList](#)
Struktura, która opisuje parametr typu lista.
- struct [AVUnitParameterInteger](#)
Struktura, która opisuje parametr typu liczbowego.

3.1.2 Definicje

- #define [RE_API](#)
*Makro eksportujące funkcje API w bibliotece *.dll/*.so.*

3.1.3 Definicje typów

- typedef void * [AVUnitHandle](#)
Uchwyt do głównego obiektu AVUnit.
- typedef void * [AVUnitCallbackStateHandle](#)
Uchwyt do obiektu umożliwiającego ustawienie callbacków dla instancji AVUnit.
- typedef void * [EncoderControlHandle](#)
Uchwyt do kontroli procesu enkodowania.
- typedef void * [DecoderControlHandle](#)
Uchwyt do kontroli procesu dekodowania.
- typedef void * [MediaProcessingControlHandle](#)
Uchwyt do obiektu umożliwiającego przetwarzanie plików nagrań
- typedef void * [AudioDriverControlHandle](#)
Uchwyt do kontroli oraz odczytu stanu przetwarzania danych audio.
- typedef void * [VideoDriverControlHandle](#)
Uchwyt do kontroli oraz odczytu stanu przetwarzania danych wideo.
- typedef void * [FreeSpaceControlHandle](#)
Uchwyt do modułu umożliwiającego uzyskanie informacji o dostępnym miejscu na dysku.
- typedef void * [TVControlHandle](#)
Uchwyt umożliwiający sterowanie zewnętrznym odtwarzaczem.
- typedef void * [SpeechSynthesisControlHandle](#)
Uchwyt umożliwiający sterowanie syntezą mowy.
- typedef void * [ExternalDevicesHandle](#)
Uchwyt umożliwiający pobranie obiektów do sterowania urządzeniami zewnętrznymi.



- typedef void * [PTZControlHandle](#)
Uchwyt umożliwiający sterowanie kamerami poprzez interfejs PTZ.
- typedef void * [CarecMicControlHandle](#)
Uchwyt umożliwiający pobranie dodatkowych informacji z karty akwizycji danych audio.
- typedef void * [AmplifierControlHandle](#)
Uchwyt umożliwiający sterowanie wzmacniaczem dźwięku.
- typedef void * [UPSControlHandle](#)
Uchwyt umożliwiający sterowanie zasilaczem awaryjnym UPS.
- typedef void * [NTPClientHandle](#)
Uchwyt umożliwiający sterowanie klientem NTP.
- typedef void(* [StateChangedCallback](#)) (int32_t state)
Callback, który jest wołany podczas każdej zmiany stanu.
- typedef void(* [ProgressCallback](#)) (float progress)
Callback, który jest wykorzystywany do informowania o postępie procesu długotrwałego.
- typedef void(* [CarecMicrophoneStateChanged](#)) (const char *code, int32_t connected)
Callback, który jest wołany przy zmianie stanu mikrofonu podłączonego do karty akwizycji dźwięku.
- typedef void(* [VideoSourceStateChanged](#)) (const char *code, int32_t connected)
Callback, który jest wołany przy zmianie stanu źródła wideo.
- typedef void(* [DeviceConnectedChanged](#)) (int32_t connected)
Callback, który jest wołany przy zmianie stanu podłączenia do wzmacniacza dźwięku lub do zasilacza awaryjnego.
- typedef void(* [FatalErrorCallback](#)) (int32_t fatal_error_code)
Callback, który jest wywoływany w przypadku wystąpienia błędu krytycznego.

3.1.4 Wyliczenia

- enum [AVUnitFatalError](#) { [AVUNIT_FATAL_ERROR_AUDIO_ENCODER_OVERLOADED](#) = 0, [AVUNIT_FATAL_ERROR_VIDEO_ENCODER_OVERLOADED](#) = 1, [AVUNIT_FATAL_ERROR_AUDIO_MIXER_OVERLOADED](#) = 2, [AVUNIT_FATAL_ERROR_VIDEO_MIXER_OVERLOADED](#) = 3, [AVUNIT_FATAL_ERROR_AUDIO_NOT_RECORDING](#) = 4, [AVUNIT_FATAL_ERROR_VIDEO_NOT_RECORDING](#) = 5, [AVUNIT_FATAL_ERROR_VIDEO_MIXER_FAILED](#) = 6 } *Enumeracja możliwych wartości błędu krytycznego.*
- enum [AVUnitStatus](#) { [AVUNIT_SUCCESS](#) = 0, [AVUNIT_ERROR_INITIALIZATION_FAILED](#) = -1, [AVUNIT_ERROR_NOT_INITIALIZED](#) = -2, [AVUNIT_ERROR_INTERNAL](#) = -3, [AVUNIT_ERROR_OUT_OF_MEMORY](#) = -4, [AVUNIT_ERROR_INVALID_PARAMETER](#) = -5, [AVUNIT_ERROR_INVALID_HANDLE](#) = -6, [AVUNIT_ERROR_INVALID_STATE](#) = -7, [AVUNIT_ERROR_OPERATION_FAILED](#) = -8, [AVUNIT_ERROR_INVALID_TYPE](#) = -9, [AVUNIT_ERROR_FAILED_TO_OPEN](#) = -100, [AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_INDEX](#) = -200, [AVUNIT_ERROR_INVALID_AUDIO_CHANNEL_CODE](#) = -201, [AVUNIT_ERROR_INVALID_AUDIO_MIXER_INDEX](#) = -202, [AVUNIT_ERROR_INVALID_AUDIO_MIXER_CODE](#) = -203,



[AVUNIT_ERROR_INVALID_VIDEO_FRAME_INDEX](#) = -300,
[AVUNIT_ERROR_INVALID_VIDEO_FRAME_CODE](#) = -301,
[AVUNIT_ERROR_INVALID_VIDEO_MIXER_INDEX](#) = -302,
[AVUNIT_ERROR_INVALID_VIDEO_MIXER_CODE](#) = -303,
[AVUNIT_ERROR_FAILED_TO_OPEN_FILE](#) = -400, [AVUNIT_ERROR_INVALID_AUDIO_CODEC](#) =
-401, [AVUNIT_ERROR_INVALID_VIDEO_CODEC](#) = -402,
[AVUNIT_ERROR_SERIALIZATION_FAILED](#) = -403, [AVUNIT_ERROR_DESERIALIZATION_FAILED](#)
= -404, [AVUNIT_ERROR_INVALID_CONTAINER](#) = -405,
[AVUNIT_ERROR_INVALID_STREAM_INDEX](#) = -600, [AVUNIT_ERROR_INVALID_STREAM_TYPE](#)
= -601, [AVUNIT_ERROR_PROCESSING_CANCELED](#) = -602, [AVUNIT_ERROR_INVALID_LICENSE](#)
= -1000, [AVUNIT_WARNING_NO_SUCH_CONTROL](#) = -10000 }Enumeracja możliwych
wartości zwracanych z funkcji API.

- enum [AVUnitAmplifierSource](#) { [AVUNIT_AMPLIFIER_SOURCE_COMMON](#) = 0,
[AVUNIT_AMPLIFIER_SOURCE_DEDICATED](#) = 1 }Wartości dla źródła dźwięku we wzmacniaczu
audio.
- enum [AVUnitUPSPowerSource](#) { [AVUNIT_UPS_POWER_SOURCE_MAINS](#) = 0,
[AVUNIT_UPS_POWER_SOURCE_BATTERY](#) = 1,
[AVUNIT_UPS_POWER_SOURCE_BATTERY_LOW](#) = 2 }Wartości dla źródła zasilania w zasilaczu
awaryjnym UPS.
- enum [AVUnitParameterType](#) { [AVUNIT_PARAMETER_TYPE_BOOL](#) = 0,
[AVUNIT_PARAMETER_TYPE_INTEGER](#) = 1, [AVUNIT_PARAMETER_TYPE_LIST](#) = 2 }Wartości dla
typu parametrów obsługiwanych przez urządzenia.
- enum [AVUnitEncoderState](#) { [AVUNIT_ENCODER_STATE_CLOSED](#) = 0,
[AVUNIT_ENCODER_STATE_OPENED](#) = 1, [AVUNIT_ENCODER_STATE_RECORDING](#) = 2,
[AVUNIT_ENCODER_STATE_PAUSED](#) = 3 }Stan enkodera.
- enum [AVUnitDecoderState](#) { [AVUNIT_DECODER_STATE_CLOSED](#) = 0,
[AVUNIT_DECODER_STATE_PLAYING](#) = 1, [AVUNIT_DECODER_STATE_RECORDING](#) = 2,
[AVUNIT_DECODER_STATE_PAUSED](#) = 3 }Stan dekodera.

3.1.5 Funkcje

- [RE_API](#) void [avunit_version](#) (uint32_t *major, uint32_t *minor, uint32_t *patch, uint32_t *build)
Zwrócenie wersji AVUnit.
- [RE_API](#) uint32_t [avunit_api_version](#) ()
Zwrócenie aktualnej wersji API, używanej przez modul.
- [RE_API](#) int32_t [avunit_module_initialize](#) (const char *data_dir)
Zainicjalizowanie biblioteki.
- [RE_API](#) int32_t [avunit_module_shutdown](#) ()
Zamknięcie modułu AVUnit.
- [RE_API](#) int32_t [avunit_machine_serial_number](#) (const char **serial_number)
Pobranie numeru seryjnego maszyny, na którym działa AVUnit.
- [RE_API](#) int32_t [avunit_create](#) ([AVUnitHandle](#) *handle)



Utworzenie głównego obiektu AVUnit.

- [RE_API int32_t avunit_destroy \(AVUnitHandle handle\)](#)
Zniszczenie obiektu utworzonego przez funkcję [avunit_create\(\)](#).
- [RE_API int32_t avunit_allocate_callback_state \(AVUnitCallbackStateHandle *handle\)](#)
Utworzenie obiektu ze stanem wywołań zwrotnych.
- [RE_API int32_t avunit_free_callback_state \(AVUnitCallbackStateHandle handle\)](#)
Zniszczenie obiektu ze stanem wywołań zwrotnych.
- [RE_API int32_t avunit_open \(AVUnitHandle handle\)](#)
Otwarcie oraz inicjalizacja obiektu AVUnit.
- [RE_API int32_t avunit_close \(AVUnitHandle handle\)](#)
Zamknięcie obiektu AVUnit.
- [RE_API int32_t avunit_get_encoder_control \(AVUnitHandle handle, EncoderControlHandle *encoder_handle\)](#)
Pobranie uchwytu do kontroli procesu enkodowania.
- [RE_API int32_t avunit_get_decoder_control \(AVUnitHandle handle, DecoderControlHandle *decoder_handle\)](#)
Pobranie uchwytu do kontroli procesu dekodowania.
- [RE_API int32_t avunit_get_audio_driver_control \(AVUnitHandle handle, AudioDriverControlHandle *audio_driver_handle\)](#)
Pobranie uchwytu do kontroli przetwarzania danych audio.
- [RE_API int32_t avunit_get_video_driver_control \(AVUnitHandle handle, VideoDriverControlHandle *video_driver_handle\)](#)
Pobranie uchwytu do kontroli przetwarzania danych wideo.
- [RE_API int32_t avunit_get_free_space_control \(AVUnitHandle handle, FreeSpaceControlHandle *free_space_handle\)](#)
Pobranie uchwytu do pobierania informacji o dostępności miejsca na dysku.
- [RE_API int32_t avunit_get_tv_control \(AVUnitHandle handle, TVControlHandle *tv_handle\)](#)
Pobranie uchwytu do kontroli odtwarzacza znajdującego się przy monitorze wielkoformatowym.
- [RE_API int32_t avunit_get_speech_synthesis_control \(AVUnitHandle handle, SpeechSynthesisControlHandle *speech_synthesis_handle\)](#)
Pobranie uchwytu do kontroli syntezy mowy.
- [RE_API int32_t avunit_get_external_devices_control \(AVUnitHandle handle, ExternalDevicesHandle *external_devices_handle\)](#)
Pobranie uchwytu do kontroli urządzeń zewnętrznych.
- [RE_API int32_t avunit_audio_driver_control_get_channel_count \(AudioDriverControlHandle handle\)](#)
Pobranie ilości wszystkich obsługiwanych kanałów audio.
- [RE_API int32_t avunit_audio_driver_control_get_mixer_count \(AudioDriverControlHandle handle\)](#)
Pobranie ilości wszystkich obsługiwanych mikserów audio.



- [RE API int32_t avunit audio driver control get channel index](#) ([AudioDriverControlHandle](#) handle, const char *channel_code)
Pobranie indeksu kanału audio na podstawie jego kodu.
- [RE API int32_t avunit audio driver control get channel code](#) ([AudioDriverControlHandle](#) handle, uint32_t channel_index, const char **channel_code)
Pobranie kodu kanału audio na podstawie jego indeksu.
- [RE API int32_t avunit audio driver control get mixer code](#) ([AudioDriverControlHandle](#) handle, uint32_t mixer_index, const char **mixer_code)
Pobranie kodu miksera audio na podstawie jego indeksu.
- [RE API int32_t avunit audio driver control get channel name by code](#) ([AudioDriverControlHandle](#) handle, const char *channel_code, const char **channel_name)
Pobranie opisowej nazwy kanału audio na podstawie jego kodu.
- [RE API int32_t avunit audio driver control get channel name by index](#) ([AudioDriverControlHandle](#) handle, uint32_t channel_index, const char **channel_name)
Pobranie opisowej nazwy kanału audio na podstawie jego indeksu.
- [RE API int32_t avunit audio driver control get mixer channel count](#) ([AudioDriverControlHandle](#) handle, const char *mixer_code)
Pobranie ilości kanałów audio miksowanych w danym mikserze.
- [RE API int32_t avunit audio driver control does mixer contain channel](#) ([AudioDriverControlHandle](#) handle, const char *mixer_code, uint32_t channel_index)
Sprawdzenie, czy dany kanał audio jest miksowany w danym mikserze.
- [RE API int32_t avunit audio driver control get master volume](#) ([AudioDriverControlHandle](#) handle, float *volume)
Pobranie wartości głośności dźwięku.
- [RE API int32_t avunit audio driver control get channel volume](#) ([AudioDriverControlHandle](#) handle, const char *channel_code, float *volume)
Pobranie wartości głośności dźwięku dla danego kanału audio.
- [RE API int32_t avunit audio driver control set master volume](#) ([AudioDriverControlHandle](#) handle, float volume)
Ustawienie wartości głośności dźwięku.
- [RE API int32_t avunit audio driver control set channel volume](#) ([AudioDriverControlHandle](#) handle, const char *channel_code, float volume)
Ustawienie wartości głośności dźwięku dla danego kanału audio.
- [RE API int32_t avunit audio driver control set channels volume](#) ([AudioDriverControlHandle](#) handle, float *volume, uint32_t volume_count)
Ustawienie wartości głośności dźwięku dla wszystkich kanałów audio jednocześnie.
- [RE API int32_t avunit video driver control set state changed callback](#) ([VideoDriverControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [VideoSourceStateChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu źródła wideo.



- [RE API int32_t avunit video driver control remove state changed callback](#) ([VideoDriverControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit video driver control set state changed callback](#).
- [RE API int32_t avunit video driver control get frame count](#) ([VideoDriverControlHandle](#) handle)
Pobranie ilości wszystkich obsługiwanych klatek wideo.
- [RE API int32_t avunit video driver control get mixer count](#) ([VideoDriverControlHandle](#) handle)
Pobranie ilości wszystkich obsługiwanych mikserów wideo.
- [RE API int32_t avunit video driver control get frame index](#) ([VideoDriverControlHandle](#) handle, const char *frame_code)
Pobranie indeksu klatki wideo na podstawie jej kodu.
- [RE API int32_t avunit video driver control get frame code](#) ([VideoDriverControlHandle](#) handle, uint32_t frame_index, const char **frame_code)
Pobranie kodu klatki wideo na podstawie jej indeksu.
- [RE API int32_t avunit video driver control get frame name by code](#) ([VideoDriverControlHandle](#) handle, const char *frame_code, const char **frame_name)
Pobranie opisowej nazwy klatki wideo na podstawie jej kodu.
- [RE API int32_t avunit video driver control get frame name by index](#) ([VideoDriverControlHandle](#) handle, uint32_t frame_index, const char **frame_name)
Pobranie opisowej nazwy klatki wideo na podstawie jej indeksu.
- [RE API int32_t avunit video driver control state](#) ([VideoDriverControlHandle](#) handle, const char *frame_code)
Pobranie stanu podłączenia źródła wideo.
- [RE API int32_t avunit video driver control get mixer code](#) ([VideoDriverControlHandle](#) handle, uint32_t mixer_index, const char **mixer_code)
Pobranie kodu miksera wideo na podstawie jego indeksu.
- [RE API int32_t avunit video driver control get mixer frame count](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code)
Pobranie ilości klatek wideo miksowanych w danym mikserze.
- [RE API int32_t avunit video driver control get mixer supported size](#) ([VideoDriverControlHandle](#) handle, uint32_t *min_width, uint32_t *min_height, uint32_t *max_width, uint32_t *max_height)
Pobranie zakresu rozdzielczości wyjściowych obsługiwanych przez mikser wideo.
- [RE API int32_t avunit video driver control get mixer output size](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *width, uint32_t *height)
Pobranie rozmiaru klatki wideo produkowanej przez dany mikser.



- [RE_API int32_t avunit video driver control set mixer output size](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t width, uint32_t height)
Ustawienie rozmiaru klatki video produkowanej przez dany mikser.
- [RE_API int32_t avunit video driver control does mixer contain frame](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t frame_index, uint32_t *internal_index)
Sprawdzenie, czy dana klatka video jest miksowana w danym mikserze.
- [RE_API int32_t avunit video driver control set mixer frame order](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *order, uint32_t order_count)
Ustawienie kolejności klatek (z-indeks) w mikserze.
- [RE_API int32_t avunit video driver control get mixer frame order](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, uint32_t *order, uint32_t order_count)
Pobranie kolejności klatek (z-indeks) w mikserze.
- [RE_API int32_t avunit video driver control set frame enabled in mixer](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, int32_t enabled)
Włączenie lub wyłączenie przetwarzania klatki w mikserze.
- [RE_API int32_t avunit video driver control enable frames in mixer](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const int8_t *enabled, uint32_t enabled_count)
Włączenie lub wyłączenie przetwarzania dla wszystkich klatek w mikserze.
- [RE_API int32_t avunit video driver control set mixer frame area](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitRect](#) *area)
Ustawienie obszaru klatki w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control set mixer frame areas](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitRect](#) *areas, uint32_t areas_count)
Ustawienie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control get mixer frame area](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitRect](#) *area)
Pobranie obszaru klatki w wynikowym obrazie produkowanym przez mikser.
- [RE_API int32_t avunit video driver control get mixer frame areas](#)
([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitRect](#) *areas, uint32_t areas_count)
Pobranie obszarów wszystkich klatek w wynikowym obrazie produkowanym przez mikser.



- [RE API int32_t avunit video driver control set mixer frame crop](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, const [AVUnitCrop](#) *crop)
Ustawienie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control set mixer frame crops](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, const [AVUnitCrop](#) *crops, uint32_t crop_count)
Ustawienie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control get mixer frame crop](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, const char *frame_code, [AVUnitCrop](#) *crop)
Pobranie parametrów kadrowania klatki w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit video driver control get mixer frame crops](#) ([VideoDriverControlHandle](#) handle, const char *mixer_code, [AVUnitCrop](#) *crops, uint32_t crop_count)
Pobranie parametrów kadrowania wszystkich klatek w wynikowym obrazie produkowanym przez mikser.
- [RE API int32_t avunit encoder control set state changed callback](#) ([EncoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [StateChangedCallback](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu nagrywania.
- [RE API int32_t avunit encoder control remove state changed callback](#) ([EncoderControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_encoder_control_set_state_changed_callback](#).
- [RE API int32_t avunit encoder control open](#) ([EncoderControlHandle](#) handle, const char *file_name, const char *audio_codec, const char *video_codec, float audio_quality, float video_quality, int32_t video_enabled)
Otwarcie procesu nagrywania z podanymi parametrami.
- [RE API int32_t avunit encoder control finalize and close](#) ([EncoderControlHandle](#) handle)
Finalizacja pliku, zamknięcie procesu nagrywania oraz zwolnienie zasobów.
- [RE API int32_t avunit encoder control close](#) ([EncoderControlHandle](#) handle)
Zamknięcie procesu nagrywania oraz zwolnienie zasobów.
- [RE API int32_t avunit encoder audio channel enabled](#) ([EncoderControlHandle](#) handle, uint32_t channel_index)
Pobranie informacji, czy dany kanał audio jest przetwarzany przez moduł nagrywający.
- [RE API int32_t avunit encoder video frame enabled](#) ([EncoderControlHandle](#) handle, uint32_t frame_index)
Pobranie informacji, czy dana klatka wideo jest przetwarzana przez moduł nagrywający.
- [RE API int32_t avunit encoder control deserialize state](#) ([EncoderControlHandle](#) handle, const char *file_name)
Otwarcie procesu nagrywania wykorzystując do tego zapisany wcześniej stan serializacji.



- [RE_API int32_t avunit_encoder_control_serialize_state \(EncoderControlHandle handle\)](#)
Zapis stanu nagrywania, umożliwiający późniejsze wznowienie nagrywania z tymi samymi parametrami oraz stanem.
- [RE_API int32_t avunit_encoder_control_serialization_supported \(EncoderControlHandle handle\)](#)
Sprawdzenie, czy serializacja jest obsługiwana dla parametrów podanych przy otwarciu pliku do nagrywania.
- [RE_API int32_t avunit_encoder_control_start \(EncoderControlHandle handle\)](#)
Wystartowanie procesu nagrywania ze stanu otwartego pliku lub pauzy.
- [RE_API int32_t avunit_encoder_control_pause \(EncoderControlHandle handle\)](#)
Tymczasowe zatrzymanie procesu nagrywania.
- [RE_API int64_t avunit_encoder_control_recording_time \(EncoderControlHandle handle\)](#)
Pobranie aktualnego czasu nagrywania w milisekundach.
- [RE_API int32_t avunit_decoder_control_set_state_changed_callback \(DecoderControlHandle handle, AVUnitCallbackStateHandle callback_state, StateChangedCallback callback\)](#)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu odtwarzania.
- [RE_API int32_t avunit_decoder_control_remove_state_changed_callback \(DecoderControlHandle handle, AVUnitCallbackStateHandle callback_state\)](#)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_decoder_control_set_state_changed_callback](#).
- [RE_API int32_t avunit_decoder_control_open \(DecoderControlHandle handle, const char *file_name\)](#)
Otwarcie pliku do odtwarzania.
- [RE_API int32_t avunit_decoder_control_close \(DecoderControlHandle handle\)](#)
Zamknięcie odtwarzanego pliku.
- [RE_API int64_t avunit_decoder_control_beginning \(DecoderControlHandle handle\)](#)
Pobranie początkowej pozycji nagrania wyrażonej w milisekundach.
- [RE_API int64_t avunit_decoder_control_duration \(DecoderControlHandle handle\)](#)
Pobranie końcowej pozycji nagrania wyrażonej w milisekundach.
- [RE_API int32_t avunit_decoder_control_start \(DecoderControlHandle handle\)](#)
Rozpoczęcie odtwarzania wcześniej otwartego pliku lub wznowienie odtwarzania po jego zatrzymaniu.
- [RE_API int32_t avunit_decoder_control_pause \(DecoderControlHandle handle\)](#)
Zatrzymanie odtwarzania wcześniej otwartego pliku.
- [RE_API int32_t avunit_decoder_control_seek \(DecoderControlHandle handle, uint64_t position\)](#)
Przeskok do wybranej pozycji we wcześniej otwartym pliku.
- [RE_API int32_t avunit_decoder_control_start_recording \(DecoderControlHandle handle\)](#)
Rozpoczęcie rejestracji odtwarzanego nagrania w nagrywanym pliku.
- [RE_API int32_t avunit_decoder_control_stop_recording \(DecoderControlHandle handle\)](#)
Zakończenie rejestracji odtwarzanego nagrania w nagrywanym pliku.
- [RE_API int64_t avunit_decoder_control_position \(DecoderControlHandle handle\)](#)



Pobranie aktualnej pozycji w odtwarzanym pliku wyrażonej w milisekundach.

- [RE_API int32_t avunit_extractor_create](#) ([MediaProcessingControlHandle](#) *handle)
Utworzenie obiektu ekstraktora strumieni.
- [RE_API int32_t avunit_extractor_destroy](#) ([MediaProcessingControlHandle](#) handle)
Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).
- [RE_API int32_t avunit_extractor_extract_to](#) ([MediaProcessingControlHandle](#) handle, const char *output_file, [ProgressCallback](#) progress_callback)
Wypakowanie wybranych strumieni do pliku wyjściowego.
- [RE_API int32_t avunit_extractor_extract_cancel](#) ([MediaProcessingControlHandle](#) handle)
Przerwanie procesu ekstrakcji wykonywanego w innym wątku.
- [RE_API int32_t avunit_reencoder_create](#) ([MediaProcessingControlHandle](#) *handle)
Utworzenie obiektu ekstraktora strumieni.
- [RE_API int32_t avunit_reencoder_destroy](#) ([MediaProcessingControlHandle](#) handle)
Zniszczenie obiektu ekstraktora strumieni stworzonego za pomocą funkcji [avunit_extractor_create](#).
- [RE_API int32_t avunit_reencoder_set_output_audio_parameters](#) ([MediaProcessingControlHandle](#) handle, const char *codec, float quality)
Ustawienie wybranych parametrów dla ścieżek audio.
- [RE_API int32_t avunit_reencoder_set_output_video_parameters](#) ([MediaProcessingControlHandle](#) handle, const char *codec, float quality)
Ustawienie wybranych parametrów dla ścieżek video.
- [RE_API int32_t avunit_reencoder_set_output_container](#) ([MediaProcessingControlHandle](#) handle, const char *container)
Ustawienie kontenera dla wyjściowego pliku.
- [RE_API int32_t avunit_reencoder_reencode_to](#) ([MediaProcessingControlHandle](#) handle, const char *output_file, [ProgressCallback](#) progress_callback)
Przekodowanie wybranych strumieni do pliku wyjściowego.
- [RE_API int32_t avunit_reencoder_reencode_cancel](#) ([MediaProcessingControlHandle](#) handle)
Przerwanie procesu przekodowania wykonywanego w innym wątku.
- [RE_API int32_t avunit_media_processing_open](#) ([MediaProcessingControlHandle](#) handle, const char *file_name)
Otwarcie pliku źródłowego w obiekcie przetwarzania multimedialnych.
- [RE_API int32_t avunit_media_processing_close](#) ([MediaProcessingControlHandle](#) handle)
Zamknięcie pliku źródłowego w obiekcie przetwarzania multimedialnych strumieni oraz zwolnienie wszystkich związanych z nim zasobów.
- [RE_API int32_t avunit_media_processing_stream_count](#) ([MediaProcessingControlHandle](#) handle)
Pobranie ilości strumieni w pliku źródłowym.
- [RE_API int32_t avunit_media_processing_stream_codec](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index, const char **codec)
Pobranie kodeka użytego do zakodowania ścieżki o podanym indeksie.



- [RE_API int32_t avunit media processing stream internal index](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie wewnętrznego indeksu strumienia dla ścieżki o podanym indeksie.
- [RE_API int32_t avunit media processing stream is audio](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie typu strumienia.
- [RE_API int32_t avunit media processing stream audio sampling rate](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie częstotliwości próbkowania strumienia audio.
- [RE_API int32_t avunit media processing stream audio num channels](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie ilości kanałów w strumieniu audio.
- [RE_API int32_t avunit media processing stream video width](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie szerokości klatki w ścieżce wideo.
- [RE_API int32_t avunit media processing stream video height](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie wysokości klatki w ścieżce wideo.
- [RE_API int32_t avunit media processing stream video frame rate numerator](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie licznika ułamka określającego ilość klatek na sekundę.
- [RE_API int32_t avunit media processing stream video frame rate denominator](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Pobranie mianownika ułamka określającego ilość klatek na sekundę.
- [RE_API int32_t avunit media processing enable stream](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Włączenie wypakowania strumienia do wyjściowego pliku.
- [RE_API int32_t avunit media processing disable stream](#) ([MediaProcessingControlHandle](#) handle, uint32_t stream_index)
Wyłączenie wypakowania strumienia do wyjściowego pliku.
- [RE_API int32_t avunit external devices get ptz control](#) ([ExternalDevicesHandle](#) handle, [PTZControlHandle](#) *ptz_handle)
Pobranie uchwytu do sterowania ustawieniami kamer poprzez interfejs PTZ.
- [RE_API int32_t avunit external devices get carec mic control](#) ([ExternalDevicesHandle](#) handle, [CarecMicControlHandle](#) *carec_handle)
Pobranie uchwytu do pobierania informacji z karty akwizycji danych audio.
- [RE_API int32_t avunit external devices get amplifier control](#) ([ExternalDevicesHandle](#) handle, [AmplifierControlHandle](#) *amplifier_handle)
Pobranie uchwytu do sterowania wzmacniaczem dźwięku.
- [RE_API int32_t avunit external devices get ups control](#) ([ExternalDevicesHandle](#) handle, [UPSControlHandle](#) *ups_handle)



Pobranie uchwytu do sterowania zasilaczem awaryjnym UPS.

- [RE_API int32_t avunit_ptz_control_count](#) ([PTZControlHandle](#) handle)
Pobranie ilości kamer skonfigurowanych do obsługi poprzez interfejs PTZ.
- [RE_API int32_t avunit_ptz_control_zoom_in](#) ([PTZControlHandle](#) handle, uint32_t id)
Wysłanie do kamery żądania przybliżenia obrazu.
- [RE_API int32_t avunit_ptz_control_zoom_out](#) ([PTZControlHandle](#) handle, uint32_t id)
Wysłanie do kamery żądania oddalenia obrazu.
- [RE_API int32_t avunit_ptz_control_focus_near](#) ([PTZControlHandle](#) handle, uint32_t id)
Wysłanie do kamery żądania ustawienia ostrości na bliższe obiekty.
- [RE_API int32_t avunit_ptz_control_focus_far](#) ([PTZControlHandle](#) handle, uint32_t id)
Wysłanie do kamery żądania ustawienia ostrości na dalsze obiekty.
- [RE_API int32_t avunit_ptz_control_stop](#) ([PTZControlHandle](#) handle, uint32_t id)
Wysłanie do kamery żądania zaprzestania zmian ostrości oraz przybliżenia.
- [RE_API int32_t avunit_carec_mic_control_set_state_changed_callback](#) ([CarecMicControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [CarecMicrophoneStateChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia mikrofonu do karty audio.
- [RE_API int32_t avunit_carec_mic_control_remove_state_changed_callback](#) ([CarecMicControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_carec_mic_control_set_state_changed_callback](#).
- [RE_API int32_t avunit_carec_mic_control_count](#) ([CarecMicControlHandle](#) handle)
Pobranie ilości kanałów audio, które są skonfigurowane jako pochodzące z karty audio.
- [RE_API int32_t avunit_carec_mic_control_has](#) ([CarecMicControlHandle](#) handle, const char *channel_code)
Pobranie informacji, czy kanał audio z przekazanym kodem pochodzi z karty akwizycji audio.
- [RE_API int32_t avunit_carec_mic_control_state](#) ([CarecMicControlHandle](#) handle, const char *channel_code)
Pobranie informacji, czy do kanału audio z przekazanym kodem jest podłączony mikrofon.
- [RE_API int32_t avunit_carec_mic_control_parameter_count](#) ([CarecMicControlHandle](#) handle)
Pobranie ilości parametrów obsługiwanych przez kartę akwizycji audio dla każdego z kanałów.
- [RE_API int32_t avunit_carec_mic_control_parameter_type](#) ([CarecMicControlHandle](#) handle, uint32_t index)
Pobranie typu parametru pod wskazanym indeksem.
- [RE_API int32_t avunit_carec_mic_control_parameter_name](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char **name)
Pobranie nazwy parametru pod wskazanym indeksem.
- [RE_API int32_t avunit_carec_mic_control_parameter_integer](#) ([CarecMicControlHandle](#) handle, uint32_t index, [AVUnitParameterInteger](#) *parameter)



Pobranie dodatkowych informacji o parametrze typu liczbowego.

- [RE_API int32_t avunit_carec_mic_control_parameter_list](#) ([CarecMicControlHandle](#) handle, uint32_t index, [AVUnitParameterList](#) *parameter)

Pobranie dodatkowych informacji o parametrze typu lista.

- [RE_API int32_t avunit_carec_mic_control_parameter_get_value](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char *channel_code, int32_t *value)

Pobranie wartości parametru dla poszczególnego kanału audio.

- [RE_API int32_t avunit_carec_mic_control_parameter_set_value](#) ([CarecMicControlHandle](#) handle, uint32_t index, const char *channel_code, int32_t value)

Ustawienie wartości parametru dla poszczególnego kanału audio.

- [RE_API int32_t avunit_carec_mic_control_parameters_save](#) ([CarecMicControlHandle](#) handle)

Zapisanie obecnie ustawionych parametrów na karcie akwizycji audio.

- [RE_API int32_t avunit_carec_mic_control_parameters_restore](#) ([CarecMicControlHandle](#) handle)

Odrzucenie obecnie ustawionych parametrów na karcie akwizycji audio i przywrócenie ustawień na niej zapisanych.

- [RE_API int32_t avunit_amplifier_control_set_connected_changed_callback](#) ([AmplifierControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [DeviceConnectedChanged](#) callback)

Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do wzmacniacza dźwięku.

- [RE_API int32_t avunit_amplifier_control_remove_connected_changed_callback](#) ([AmplifierControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)

Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit_amplifier_control_set_connected_changed_callback](#).

- [RE_API int32_t avunit_amplifier_control_is_connected](#) ([AmplifierControlHandle](#) handle)

Pobranie informacji, czy połączenie ze wzmacniaczem dźwięku jest aktywne.

- [RE_API int32_t avunit_amplifier_control_get_enabled](#) ([AmplifierControlHandle](#) handle, uint32_t channel)

Pobranie informacji, czy dany kanał wzmacniacza dźwięku jest włączony.

- [RE_API int32_t avunit_amplifier_control_get_source](#) ([AmplifierControlHandle](#) handle, uint32_t channel)

Pobranie źródła sygnału na danym kanale wzmacniacza dźwięku.

- [RE_API int32_t avunit_amplifier_control_get_volume](#) ([AmplifierControlHandle](#) handle, uint32_t channel, float *volume)

Pobranie ustawienia głośności sygnału na danym kanale wzmacniacza dźwięku.

- [RE_API int32_t avunit_amplifier_control_set_enabled](#) ([AmplifierControlHandle](#) handle, uint32_t channel, int32_t enabled)

Włączenie lub wyłączenie kanału wzmacniacza dźwięku.



- [RE API int32_t avunit amplifier control set source](#) ([AmplifierControlHandle](#) handle, uint32_t channel, uint32_t source)
Ustawienie źródła sygnału dla danego kanału wzmacniacza dźwięku.
- [RE API int32_t avunit amplifier control set volume](#) ([AmplifierControlHandle](#) handle, uint32_t channel, float volume)
Ustawienie głośności sygnału na danym kanale wzmacniacza dźwięku.
- [RE API int32_t avunit ups control set connected changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [DeviceConnectedChanged](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do zasilacza awaryjnego UPS.
- [RE API int32_t avunit ups control remove connected changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit ups control set connected changed callback](#).
- [RE API int32_t avunit ups control set power source changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [StateChangedCallback](#) callback)
Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany źródła zasilania zasilacza awaryjnego.
- [RE API int32_t avunit ups control remove power source changed callback](#) ([UPSControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit ups control set power source changed callback](#).
- [RE API int32_t avunit ups control is connected](#) ([UPSControlHandle](#) handle)
Pobranie informacji, czy połączenie z zasilaczem awaryjnym UPS jest aktywne.
- [RE API int32_t avunit ups control power source](#) ([UPSControlHandle](#) handle)
Pobranie aktualnego źródła zasilania zasilacza awaryjnego UPS.
- [RE API int32_t avunit ups control beeper enabled](#) ([UPSControlHandle](#) handle)
Pobranie informacji, czy w zasilaczu awaryjnym UPS jest włączony sygnał dźwiękowy informujący o przejściu na zasilanie awaryjne.
- [RE API int32_t avunit ups control set beeper enabled](#) ([UPSControlHandle](#) handle, int32_t enabled)
Włączenie lub wyłączenie sygnału dźwiękowego informującego o przejściu zasilacza awaryjnego UPS na zasilanie awaryjne.
- [RE API int32_t avunit ups control get battery charge](#) ([UPSControlHandle](#) handle, float *charge)
Pobranie aktualnego stanu naładowania baterii.
- [RE API int32_t avunit ups control get run time estimate](#) ([UPSControlHandle](#) handle)
Pobranie aktualnej estymacji pozostałego czasu pracy na zasilaniu akumulatorowym.
- [RE API int32_t avunit tv control set connected changed callback](#) ([TVControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state, [DeviceConnectedChanged](#) callback)



Ustawienie callbacka, który zostanie wywołany przez aplikację w momencie zmiany stanu podłączenia do odtwarzacza monitora wielkoformatowego.

- [RE API int32_t avunit tv control remove connected changed callback](#) ([TVControlHandle](#) handle, [AVUnitCallbackStateHandle](#) callback_state)
Usunięcie callbacka ustawionego wcześniej za pomocą funkcji [avunit tv control set connected changed callback](#).
- [RE API int32_t avunit tv control wake on lan](#) ([TVControlHandle](#) handle, const char *mac, const char *address, uint32_t port)
Wysłanie pakietu Wake-on-LAN do odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control connect](#) ([TVControlHandle](#) handle, const char *address, uint32_t port)
Podłączenie się do odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control disconnect](#) ([TVControlHandle](#) handle)
Rozłączenie wcześniej nawiązanego połączenia z odtwarzaczem monitora wielkoformatowego.
- [RE API int32_t avunit tv control connected](#) ([TVControlHandle](#) handle)
Zwrócenie statusu połączenia z odtwarzaczem monitora wielkoformatowego.
- [RE API int32_t avunit tv control version](#) ([TVControlHandle](#) handle, uint32_t *major, uint32_t *minor, uint32_t *patch)
Pobranie wersji odtwarzacza monitora wielkoformatowego.
- [RE API int32_t avunit tv control start streaming](#) ([TVControlHandle](#) handle)
Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modulem AVUnit wykonującym to żądanie.
- [RE API int32_t avunit tv control start streaming to](#) ([TVControlHandle](#) handle, const char *address, uint32_t port)
Wysłanie komendy do odtwarzacza monitora wielkoformatowego, aby połączył się z modulem AVUnit znajdującym się pod podanym adresem i portem.
- [RE API int32_t avunit tv control close](#) ([TVControlHandle](#) handle)
Wysłanie komendy, która ma za zadanie wyłączyć odtwarzacz telewizyjny.
- [RE API int32_t avunit speech synthesis control read text](#) ([SpeechSynthesisControlHandle](#) handle, const char *text)
Rozpoczęcie odczytywania tekstu przez syntezytor mowy.
- [RE API int32_t avunit speech synthesis control stop reading](#) ([SpeechSynthesisControlHandle](#) handle)
Zakończenie odczytywania tekstu przez syntezytor mowy.
- [RE API int32_t avunit speech synthesis control gong enabled](#) ([SpeechSynthesisControlHandle](#) handle)
Pobranie informacji, czy odtwarzanie dźwięku gongu jest włączone.
- [RE API int32_t avunit speech synthesis control set gong enabled](#) ([SpeechSynthesisControlHandle](#) handle, int32_t enabled)
Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytanym tekstem.



- [RE_API int32_t avunit free space control set recording parameters](#) ([FreeSpaceControlHandle](#) handle, const char *recording_dir_path, const char *audio_codec, const char *video_codec, float audio_quality, float video_quality, int32_t video_enabled)
Ustawienie parametrów nagrywania, które będą miały wpływ na obliczenia pozostałego miejsca na dysku.
- [RE_API int64_t avunit free space control total bytes](#) ([FreeSpaceControlHandle](#) handle)
Pobranie całkowitego rozmiaru dysku w bajtach.
- [RE_API int64_t avunit free space control total minutes](#) ([FreeSpaceControlHandle](#) handle)
Pobranie całkowitego rozmiaru dysku w minutach.
- [RE_API int64_t avunit free space control free space in bytes](#) ([FreeSpaceControlHandle](#) handle)
Pobranie ilości pozostałego miejsca na dysku wyrażonego w bajtach.
- [RE_API int64_t avunit free space control free space in minutes](#) ([FreeSpaceControlHandle](#) handle)
Pobranie ilości pozostałego miejsca na dysku wyrażonego w minutach.
- [RE_API int64_t avunit free space control bytes per minute](#) ([FreeSpaceControlHandle](#) handle)
Pobranie ilości bajtów nagrania na każdą minutę.
- [RE_API int32_t avunit ntp client create](#) ([NTPClientHandle](#) *handle)
Utworzenie obiektu klienta NTP.
- [RE_API int32_t avunit ntp client destroy](#) ([NTPClientHandle](#) handle)
Zniszczenie obiektu klienta NTP stworzonego za pomocą funkcji [avunit ntp client create](#).
- [RE_API int32_t avunit ntp client add server](#) ([NTPClientHandle](#) handle, const char *address, uint32_t port)
Dodanie serwera NTP, z którym będzie przeprowadzana synchronizacja czasu.
- [RE_API int32_t avunit ntp client synchronize](#) ([NTPClientHandle](#) handle)
Synchronizacja zegara systemowego z podanymi wcześniej serwerami NTP.
- [RE_API int32_t avunit set system time](#) (uint64_t seconds, uint32_t nanoseconds)
Ustawienie czasu systemowego na przekazaną wartość.
- [RE_API int32_t avunit fatal error set callback](#) ([FatalErrorCallback](#) callback)
Ustawienie callbacka, który zostanie zawołany w przypadku wystąpienia błędu krytycznego.

3.1.6 Zmienne

- const uint32_t [AVUNIT_API_VERSION](#) = 11
Zmienna określająca wersję API.

3.1.7 Opis szczegółowy

Plik zawierający publiczne API dla modułu AVUnit.

Definicja w pliku [AVUnit.h](#).

3.2 AVUnit.h

```
1 #pragma once
2
3 #include <stdint.h>
```



```
4
188 #if defined(_WIN32) || defined(WIN32)
189     #if defined(AVUNIT_BUILD)
190         #define RE_API __declspec(dllexport)
191     #else /* AVUNIT_BUILD */
192         #define RE_API __declspec(dllimport)
193     #endif /* AVUNIT_BUILD */
194 #elif defined(__linux__)
195     #if defined(AVUNIT_BUILD)
196         #define RE_API __attribute__((visibility("default")))
197     #else /* AVUNIT_BUILD */
198         #define RE_API
199     #endif /* AVUNIT_BUILD */
200 #else
201     #define RE_API
202     #error "Unsupported platform!"
203 #endif
204
205 #if defined(__cplusplus)
206 extern "C" {
207 #endif /* __cplusplus */
208
209 const uint32_t AVUNIT_API_VERSION = 11;
210
211 enum AVUnitFatalError {
212     AVUNIT_FATAL_ERROR_AUDIO_ENCODER_OVERLOADED = 0,
213     AVUNIT_FATAL_ERROR_VIDEO_ENCODER_OVERLOADED = 1,
214     AVUNIT_FATAL_ERROR_AUDIO_MIXER_OVERLOADED = 2,
215     AVUNIT_FATAL_ERROR_VIDEO_MIXER_OVERLOADED = 3,
216     AVUNIT_FATAL_ERROR_AUDIO_NOT_RECORDING = 4,
217     AVUNIT_FATAL_ERROR_VIDEO_NOT_RECORDING = 5,
218     AVUNIT_FATAL_ERROR_VIDEO_MIXER_FAILED = 6
219 };
220
221 enum AVUnitStatus {
222     AVUNIT_SUCCESS = 0,
223
224     AVUNIT_ERROR_INITIALIZATION_FAILED = -1,
225     AVUNIT_ERROR_NOT_INITIALIZED = -2,
226
227     AVUNIT_ERROR_INTERNAL = -3,
228     AVUNIT_ERROR_OUT_OF_MEMORY = -4,
229     AVUNIT_ERROR_INVALID_PARAMETER = -5,
230     AVUNIT_ERROR_INVALID_HANDLE = -6,
231     AVUNIT_ERROR_INVALID_STATE = -7,
232     AVUNIT_ERROR_OPERATION_FAILED = -8,
```



```
400 AVUNIT\_ERROR\_INVALID\_TYPE = -9,  
401  
402 AVUNIT\_ERROR\_FAILED\_TO\_OPEN = -100,  
403  
404 AVUNIT\_ERROR\_INVALID\_AUDIO\_CHANNEL\_INDEX = -200,  
405 AVUNIT\_ERROR\_INVALID\_AUDIO\_CHANNEL\_CODE = -201,  
406 AVUNIT\_ERROR\_INVALID\_AUDIO\_MIXER\_INDEX = -202,  
407 AVUNIT\_ERROR\_INVALID\_AUDIO\_MIXER\_CODE = -203,  
408  
409 AVUNIT\_ERROR\_INVALID\_VIDEO\_FRAME\_INDEX = -300,  
410 AVUNIT\_ERROR\_INVALID\_VIDEO\_FRAME\_CODE = -301,  
411 AVUNIT\_ERROR\_INVALID\_VIDEO\_MIXER\_INDEX = -302,  
412 AVUNIT\_ERROR\_INVALID\_VIDEO\_MIXER\_CODE = -303,  
413  
414 AVUNIT\_ERROR\_FAILED\_TO\_OPEN\_FILE = -400,  
415 AVUNIT\_ERROR\_INVALID\_AUDIO\_CODEC = -401,  
416 AVUNIT\_ERROR\_INVALID\_VIDEO\_CODEC = -402,  
417 AVUNIT\_ERROR\_SERIALIZATION\_FAILED = -403,  
418 AVUNIT\_ERROR\_DESERIALIZATION\_FAILED = -404,  
419 AVUNIT\_ERROR\_INVALID\_CONTAINER = -405,  
420  
421 AVUNIT\_ERROR\_INVALID\_STREAM\_INDEX = -600,  
422 AVUNIT\_ERROR\_INVALID\_STREAM\_TYPE = -601,  
423 AVUNIT\_ERROR\_PROCESSING\_CANCELED = -602,  
424  
425 AVUNIT\_ERROR\_INVALID\_LICENSE = -1000,  
426  
427 AVUNIT\_WARNING\_NO\_SUCH\_CONTROL = -10000  
428 };  
429  
444 enum AVUnitAmplifierSource {  
445 AVUNIT\_AMPLIFIER\_SOURCE\_COMMON = 0,  
446 AVUNIT\_AMPLIFIER\_SOURCE\_DEDICATED = 1,  
447 };  
448  
462 enum AVUnitUPSPowerSource {  
463 AVUNIT\_UPS\_POWER\_SOURCE\_MAINS = 0,  
464 AVUNIT\_UPS\_POWER\_SOURCE\_BATTERY = 1,  
465 AVUNIT\_UPS\_POWER\_SOURCE\_BATTERY\_LOW = 2,  
466 };  
467  
481 enum AVUnitParameterType {  
482 AVUNIT\_PARAMETER\_TYPE\_BOOL = 0,  
483 AVUNIT\_PARAMETER\_TYPE\_INTEGER = 1,  
484 AVUNIT\_PARAMETER\_TYPE\_LIST = 2  
485 };
```



```
486
503 enum AVUnitEncoderState {
504     AVUNIT\_ENCODER\_STATE\_CLOSED = 0,
505     AVUNIT\_ENCODER\_STATE\_OPENED = 1,
506     AVUNIT\_ENCODER\_STATE\_RECORDING = 2,
507     AVUNIT\_ENCODER\_STATE\_PAUSED = 3
508 };
509
526 enum AVUnitDecoderState {
527     AVUNIT\_DECODER\_STATE\_CLOSED = 0,
528     AVUNIT\_DECODER\_STATE\_PLAYING = 1,
529     AVUNIT\_DECODER\_STATE\_RECORDING = 2,
530     AVUNIT\_DECODER\_STATE\_PAUSED = 3
531 };
532
536 typedef void* AVUnitHandle;
537
541 typedef void* AVUnitCallbackStateHandle;
542
546 typedef void* EncoderControlHandle;
547
551 typedef void* DecoderControlHandle;
552
556 typedef void* MediaProcessingControlHandle;
557
561 typedef void* AudioDriverControlHandle;
562
566 typedef void* VideoDriverControlHandle;
567
571 typedef void* FreeSpaceControlHandle;
572
579 typedef void* TVControlHandle;
580
584 typedef void* SpeechSynthesisControlHandle;
585
589 typedef void* ExternalDevicesHandle;
590
594 typedef void* PTZControlHandle;
595
599 typedef void* CarecMicControlHandle;
600
604 typedef void* AmplifierControlHandle;
605
609 typedef void* UPSControlHandle;
610
614 typedef void* NTPClientHandle;
```



```
615
622 typedef void (\*StateChangedCallback)(int32_t state);
623
628 typedef void (\*ProgressCallback)(float progress);
629
635 typedef void (\*CarecMicrophoneStateChanged)(const char* code, int32_t connected);
636
642 typedef void (\*VideoSourceStateChanged)(const char* code, int32_t connected);
643
649 typedef void (\*DeviceConnectedChanged)(int32_t connected);
650
657 typedef void (\*FatalErrorCallback)(int32_t fatal_error_code);
658
663 typedef struct {
665     uint32_t x;
667     uint32_t y;
669     uint32_t width;
671     uint32_t height;
672 } AVUnitRect;
673
688 typedef struct {
690     float offset\_x;
692     float offset\_y;
694     float zoom;
695 } AVUnitCrop;
696
702 typedef struct {
705     const char** values;
707     uint32_t value\_count;
708 } AVUnitParameterList;
709
713 typedef struct {
715     int32_t min;
717     int32_t max;
718 } AVUnitParameterInteger;
719
724 /* -----
725 * | Main AVUnit API
726 * -----
727 */
728
738 RE API void avunit\_version(uint32_t* major, uint32_t* minor, uint32_t* patch, uint32_t* build);
739
750 RE API uint32_t avunit\_api\_version();
751
771 RE API int32_t avunit\_module\_initialize(const char* data_dir);
```




```

772
785 RE API int32_t avunit\_module\_shutdown\(\);
786
800 RE API int32_t avunit\_machine\_serial\_number(const char** serial_number);
801
819 RE API int32_t avunit\_create(AVUnitHandle* handle);
820
832 RE API int32_t avunit\_destroy(AVUnitHandle handle);
833
851 RE API int32_t avunit\_allocate\_callback\_state(AVUnitCallbackStateHandle* handle);
852
865 RE API int32_t avunit\_free\_callback\_state(AVUnitCallbackStateHandle handle);
866
877 RE API int32_t avunit\_open(AVUnitHandle handle);
878
892 RE API int32_t avunit\_close(AVUnitHandle handle);
893
909 RE API int32_t avunit\_get\_encoder\_control(AVUnitHandle handle, EncoderControlHandle*
encoder_handle);
910
926 RE API int32_t avunit\_get\_decoder\_control(AVUnitHandle handle, DecoderControlHandle*
decoder_handle);
927
943 RE API int32_t avunit\_get\_audio\_driver\_control(AVUnitHandle handle, AudioDriverControlHandle*
audio_driver_handle);
944
960 RE API int32_t avunit\_get\_video\_driver\_control(AVUnitHandle handle, VideoDriverControlHandle*
video_driver_handle);
961
977 RE API int32_t avunit\_get\_free\_space\_control(AVUnitHandle handle, FreeSpaceControlHandle*
free_space_handle);
978
994 RE API int32_t avunit\_get\_tv\_control(AVUnitHandle handle, TVControlHandle* tv_handle);
995
1012 RE API int32_t avunit\_get\_speech\_synthesis\_control(AVUnitHandle handle,
SpeechSynthesisControlHandle* speech_synthesis_handle);
1013
1029 RE API int32_t avunit\_get\_external\_devices\_control(AVUnitHandle handle,
ExternalDevicesHandle* external_devices_handle);
1030
1031
1032 /* -----
1033 * | Audio driver API
1034 * -----
1035 */
1036

```




```
1047 RE API int32_t avunit\_audio\_driver\_control\_get\_channel\_count(AudioDriverControlHandle
handle);
1048
1059 RE API int32_t avunit\_audio\_driver\_control\_get\_mixer\_count(AudioDriverControlHandle handle);
1060
1074 RE API int32_t avunit\_audio\_driver\_control\_get\_channel\_index(AudioDriverControlHandle
handle, const char* channel_code);
1075
1096 RE API int32_t avunit\_audio\_driver\_control\_get\_channel\_code(AudioDriverControlHandle handle,
uint32_t channel_index, const char** channel_code);
1097
1118 RE API int32_t avunit\_audio\_driver\_control\_get\_mixer\_code(AudioDriverControlHandle handle,
uint32_t mixer_index, const char** mixer_code);
1119
1139 RE API int32_t
avunit\_audio\_driver\_control\_get\_channel\_name\_by\_code(AudioDriverControlHandle handle, const
char* channel_code, const char** channel_name);
1140
1163 RE API int32_t
avunit\_audio\_driver\_control\_get\_channel\_name\_by\_index(AudioDriverControlHandle handle, uint32_t
channel_index, const char** channel_name);
1164
1178 RE API int32_t avunit\_audio\_driver\_control\_get\_mixer\_channel\_count(AudioDriverControlHandle
handle, const char* mixer_code);
1179
1200 RE API int32_t
avunit\_audio\_driver\_control\_does\_mixer\_contain\_channel(AudioDriverControlHandle handle, const
char* mixer_code, uint32_t channel_index);
1201
1216 RE API int32_t avunit\_audio\_driver\_control\_get\_master\_volume(AudioDriverControlHandle
handle, float* volume);
1217
1235 RE API int32_t avunit\_audio\_driver\_control\_get\_channel\_volume(AudioDriverControlHandle
handle, const char* channel_code, float* volume);
1236
1252 RE API int32_t avunit\_audio\_driver\_control\_set\_master\_volume(AudioDriverControlHandle
handle, float volume);
1253
1271 RE API int32_t avunit\_audio\_driver\_control\_set\_channel\_volume(AudioDriverControlHandle
handle, const char* channel_code, float volume);
1272
1289 RE API int32_t avunit\_audio\_driver\_control\_set\_channels\_volume(AudioDriverControlHandle
handle, float* volume, uint32_t volume_count);
1290
1291
1292 /* -----
```



```
1293 * | Video driver API
1294 * -----
1295 */
1296
1319 RE API int32_t
avunit\_video\_driver\_control\_set\_state\_changed\_callback(VideoDriverControlHandle handle,
AVUnitCallbackStateHandle callback_state, VideoSourceStateChanged callback);
1320
1336 RE API int32_t
avunit\_video\_driver\_control\_remove\_state\_changed\_callback(VideoDriverControlHandle handle,
AVUnitCallbackStateHandle callback_state);
1337
1348 RE API int32_t avunit\_video\_driver\_control\_get\_frame\_count(VideoDriverControlHandle handle);
1349
1360 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_count(VideoDriverControlHandle handle);
1361
1375 RE API int32_t avunit\_video\_driver\_control\_get\_frame\_index(VideoDriverControlHandle handle,
const char* frame_code);
1376
1397 RE API int32_t avunit\_video\_driver\_control\_get\_frame\_code(VideoDriverControlHandle handle,
uint32_t frame_index, const char** frame_code);
1398
1418 RE API int32_t
avunit\_video\_driver\_control\_get\_frame\_name\_by\_code(VideoDriverControlHandle handle, const char*
frame_code, const char** frame_name);
1419
1440 RE API int32_t
avunit\_video\_driver\_control\_get\_frame\_name\_by\_index(VideoDriverControlHandle handle, uint32_t
frame_index, const char** frame_name);
1441
1459 RE API int32_t avunit\_video\_driver\_control\_state(VideoDriverControlHandle handle, const char*
frame_code);
1460
1481 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_code(VideoDriverControlHandle handle,
uint32_t mixer_index, const char** mixer_code);
1482
1496 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_count(VideoDriverControlHandle
handle, const char* mixer_code);
1497
1515 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_supported\_size(VideoDriverControlHandle
handle, uint32_t* min_width, uint32_t* min_height, uint32_t* max_width, uint32_t* max_height);
1516
1533 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_output\_size(VideoDriverControlHandle
handle, const char* mixer_code, uint32_t* width, uint32_t* height);
1534
```



```
1556 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_output\_size(VideoDriverControlHandle
handle, const char* mixer_code, uint32_t width, uint32_t height);
1557
1589 RE API int32_t
avunit\_video\_driver\_control\_does\_mixer\_contain\_frame(VideoDriverControlHandle handle, const char*
mixer_code, uint32_t frame_index, uint32_t* internal_index);
1590
1616 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_frame\_order(VideoDriverControlHandle
handle, const char* mixer_code, uint32_t* order, uint32_t order_count);
1617
1639 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_order(VideoDriverControlHandle
handle, const char* mixer_code, uint32_t* order, uint32_t order_count);
1640
1663 RE API int32_t
avunit\_video\_driver\_control\_set\_frame\_enabled\_in\_mixer(VideoDriverControlHandle handle, const
char* mixer_code, const char* frame_code, int32_t enabled);
1664
1687 RE API int32_t avunit\_video\_driver\_control\_enable\_frames\_in\_mixer(VideoDriverControlHandle
handle, const char* mixer_code, const int8_t* enabled, uint32_t enabled_count);
1688
1710 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_frame\_area(VideoDriverControlHandle
handle, const char* mixer_code, const char* frame_code, AVUnitRect* area);
1711
1734 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_frame\_areas(VideoDriverControlHandle
handle, const char* mixer_code, AVUnitRect* areas, uint32_t areas_count);
1735
1755 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_area(VideoDriverControlHandle
handle, const char* mixer_code, const char* frame_code, AVUnitRect* area);
1756
1778 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_areas(VideoDriverControlHandle
handle, const char* mixer_code, AVUnitRect* areas, uint32_t areas_count);
1779
1800 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_frame\_crop(VideoDriverControlHandle
handle, const char* mixer_code, const char* frame_code, const AVUnitCrop* crop);
1801
1824 RE API int32_t avunit\_video\_driver\_control\_set\_mixer\_frame\_crops(VideoDriverControlHandle
handle, const char* mixer_code, const AVUnitCrop* crops, uint32_t crop_count);
1825
1845 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_crop(VideoDriverControlHandle
handle, const char* mixer_code, const char* frame_code, AVUnitCrop* crop);
1846
1868 RE API int32_t avunit\_video\_driver\_control\_get\_mixer\_frame\_crops(VideoDriverControlHandle
handle, const char* mixer_code, AVUnitCrop* crops, uint32_t crop_count);
1869
1870
1871 /* -----
```



```
1872 * | Encoder API
1873 * -----
1874 */
1875
1899 RE API int32_t avunit_encoder_control_set_state_changed_callback(EncoderControlHandle
handle, AVUnitCallbackStateHandle callback_state, StateChangedCallback callback);
1900
1915 RE API int32_t avunit_encoder_control_remove_state_changed_callback(EncoderControlHandle
handle, AVUnitCallbackStateHandle callback_state);
1916
1953 RE API int32_t avunit_encoder_control_open(EncoderControlHandle handle, const char*
file_name, const char* audio_codec, const char* video_codec,
1954     float audio_quality, float video_quality, int32_t video_enabled);
1955
1975 RE API int32_t avunit_encoder_control_finalize_and_close(EncoderControlHandle handle);
1976
1995 RE API int32_t avunit_encoder_control_close(EncoderControlHandle handle);
1996
2010 RE API int32_t avunit_encoder_audio_channel_enabled(EncoderControlHandle handle, uint32_t
channel_index);
2011
2025 RE API int32_t avunit_encoder_video_frame_enabled(EncoderControlHandle handle, uint32_t
frame_index);
2026
2051 RE API int32_t avunit_encoder_control_deserialize_state(EncoderControlHandle handle, const
char* file_name);
2052
2068 RE API int32_t avunit_encoder_control_serialize_state(EncoderControlHandle handle);
2069
2089 RE API int32_t avunit_encoder_control_serialization_supported(EncoderControlHandle handle);
2090
2101 RE API int32_t avunit_encoder_control_start(EncoderControlHandle handle);
2102
2116 RE API int32_t avunit_encoder_control_pause(EncoderControlHandle handle);
2117
2128 RE API int64_t avunit_encoder_control_recording_time(EncoderControlHandle handle);
2129
2130
2131 /* -----
2132 * | Decoder API
2133 * -----
2134 */
2135
2159 RE API int32_t avunit_decoder_control_set_state_changed_callback(DecoderControlHandle
handle, AVUnitCallbackStateHandle callback_state, StateChangedCallback callback);
2160
```



```

2175 RE API int32_t avunit\_decoder\_control\_remove\_state\_changed\_callback(DecoderControlHandle
handle, AVUnitCallbackStateHandle callback_state);
2176
2190 RE API int32_t avunit\_decoder\_control\_open(DecoderControlHandle handle, const char*
file_name);
2191
2201 RE API int32_t avunit\_decoder\_control\_close(DecoderControlHandle handle);
2202
2215 RE API int64_t avunit\_decoder\_control\_beginning(DecoderControlHandle handle);
2216
2230 RE API int64_t avunit\_decoder\_control\_duration(DecoderControlHandle handle);
2231
2243 RE API int32_t avunit\_decoder\_control\_start(DecoderControlHandle handle);
2244
2255 RE API int32_t avunit\_decoder\_control\_pause(DecoderControlHandle handle);
2256
2270 RE API int32_t avunit\_decoder\_control\_seek(DecoderControlHandle handle, uint64_t position);
2271
2286 RE API int32_t avunit\_decoder\_control\_start\_recording(DecoderControlHandle handle);
2287
2300 RE API int32_t avunit\_decoder\_control\_stop\_recording(DecoderControlHandle handle);
2301
2313 RE API int64_t avunit\_decoder\_control\_position(DecoderControlHandle handle);
2314
2315
2316 /* -----
2317 * | Extractor API
2318 * -----
2319 */
2320
2337 RE API int32_t avunit\_extractor\_create(MediaProcessingControlHandle* handle);
2338
2349 RE API int32_t avunit\_extractor\_destroy(MediaProcessingControlHandle handle);
2350
2374 RE API int32_t avunit\_extractor\_extract\_to(MediaProcessingControlHandle handle, const char*
output_file, ProgressCallback progress_callback);
2375
2393 RE API int32_t avunit\_extractor\_extract\_cancel(MediaProcessingControlHandle handle);
2394
2395 /* -----
2396 * | Reencoder API
2397 * -----
2398 */
2399
2416 RE API int32_t avunit\_reencoder\_create(MediaProcessingControlHandle* handle);
2417

```



```
2428 RE API int32_t avunit\_reencoder\_destroy(MediaProcessingControlHandle handle);
2429
2451 RE API int32_t avunit\_reencoder\_set\_output\_audio\_parameters(MediaProcessingControlHandle
handle, const char* codec, float quality);
2452
2474 RE API int32_t avunit\_reencoder\_set\_output\_video\_parameters(MediaProcessingControlHandle
handle, const char* codec, float quality);
2475
2495 RE API int32_t avunit\_reencoder\_set\_output\_container(MediaProcessingControlHandle handle,
const char* container);
2496
2520 RE API int32_t avunit\_reencoder\_reencode\_to(MediaProcessingControlHandle handle, const
char* output_file, ProgressCallback progress_callback);
2521
2539 RE API int32_t avunit\_reencoder\_reencode\_cancel(MediaProcessingControlHandle handle);
2540
2541
2542 /* -----
2543 * | Media processing API
2544 * -----
2545 */
2546
2559 RE API int32_t avunit\_media\_processing\_open(MediaProcessingControlHandle handle, const
char* file_name);
2560
2571 RE API int32_t avunit\_media\_processing\_close(MediaProcessingControlHandle handle);
2572
2584 RE API int32_t avunit\_media\_processing\_stream\_count(MediaProcessingControlHandle handle);
2585
2605 RE API int32_t avunit\_media\_processing\_stream\_codec(MediaProcessingControlHandle handle,
uint32_t stream_index, const char** codec);
2606
2623 RE API int32_t avunit\_media\_processing\_stream\_internal\_index(MediaProcessingControlHandle
handle, uint32_t stream_index);
2624
2642 RE API int32_t avunit\_media\_processing\_stream\_is\_audio(MediaProcessingControlHandle
handle, uint32_t stream_index);
2643
2662 RE API int32_t
avunit\_media\_processing\_stream\_audio\_sampling\_rate(MediaProcessingControlHandle handle,
uint32_t stream_index);
2663
2682 RE API int32_t
avunit\_media\_processing\_stream\_audio\_num\_channels(MediaProcessingControlHandle handle,
uint32_t stream_index);
2683
```



```
2702 RE API int32_t avunit\_media\_processing\_stream\_video\_width(MediaProcessingControlHandle
handle, uint32_t stream_index);
2703
2722 RE API int32_t avunit\_media\_processing\_stream\_video\_height(MediaProcessingControlHandle
handle, uint32_t stream_index);
2723
2744 RE API int32_t
avunit\_media\_processing\_stream\_video\_frame\_rate\_numerator(MediaProcessingControlHandle
handle, uint32_t stream_index);
2745
2766 RE API int32_t
avunit\_media\_processing\_stream\_video\_frame\_rate\_denominator(MediaProcessingControlHandle
handle, uint32_t stream_index);
2767
2783 RE API int32_t avunit\_media\_processing\_enable\_stream(MediaProcessingControlHandle handle,
uint32_t stream_index);
2784
2800 RE API int32_t avunit\_media\_processing\_disable\_stream(MediaProcessingControlHandle handle,
uint32_t stream_index);
2801
2802
2803 /* -----
2804 * | External devices API
2805 * -----
2806 */
2807
2823 RE API int32_t avunit\_external\_devices\_get\_ptz\_control(ExternalDevicesHandle handle,
PTZControlHandle* ptz_handle);
2824
2840 RE API int32_t avunit\_external\_devices\_get\_carec\_mic\_control(ExternalDevicesHandle handle,
CarecMicControlHandle* carec_handle);
2841
2857 RE API int32_t avunit\_external\_devices\_get\_amplifier\_control(ExternalDevicesHandle handle,
AmplifierControlHandle* amplifier_handle);
2858
2874 RE API int32_t avunit\_external\_devices\_get\_ups\_control(ExternalDevicesHandle handle,
UPSControlHandle* ups_handle);
2875
2887 RE API int32_t avunit\_ptz\_control\_count(PTZControlHandle handle);
2888
2904 RE API int32_t avunit\_ptz\_control\_zoom\_in(PTZControlHandle handle, uint32_t id);
2905
2921 RE API int32_t avunit\_ptz\_control\_zoom\_out(PTZControlHandle handle, uint32_t id);
2922
2938 RE API int32_t avunit\_ptz\_control\_focus\_near(PTZControlHandle handle, uint32_t id);
2939
```




```
2955 RE API int32_t avunit\_ptz\_control\_focus\_far(PTZControlHandle handle, uint32_t id);
2956
2972 RE API int32_t avunit\_ptz\_control\_stop(PTZControlHandle handle, uint32_t id);
2973
2996 RE API int32_t avunit\_carec\_mic\_control\_set\_state\_changed\_callback(CarecMicControlHandle
handle, AVUnitCallbackStateHandle callback_state, CarecMicrophoneStateChanged callback);
2997
3013 RE API int32_t
avunit\_carec\_mic\_control\_remove\_state\_changed\_callback(CarecMicControlHandle handle,
AVUnitCallbackStateHandle callback_state);
3014
3026 RE API int32_t avunit\_carec\_mic\_control\_count(CarecMicControlHandle handle);
3027
3042 RE API int32_t avunit\_carec\_mic\_control\_has(CarecMicControlHandle handle, const char*
channel_code);
3043
3059 RE API int32_t avunit\_carec\_mic\_control\_state(CarecMicControlHandle handle, const char*
channel_code);
3060
3074 RE API int32_t avunit\_carec\_mic\_control\_parameter\_count(CarecMicControlHandle handle);
3075
3095 RE API int32_t avunit\_carec\_mic\_control\_parameter\_type(CarecMicControlHandle handle,
uint32_t index);
3096
3118 RE API int32_t avunit\_carec\_mic\_control\_parameter\_name(CarecMicControlHandle handle,
uint32_t index, const char** name);
3119
3138 RE API int32_t avunit\_carec\_mic\_control\_parameter\_integer(CarecMicControlHandle handle,
uint32_t index, AVUnitParameterInteger* parameter);
3139
3158 RE API int32_t avunit\_carec\_mic\_control\_parameter\_list(CarecMicControlHandle handle,
uint32_t index, AVUnitParameterList* parameter);
3159
3178 RE API int32_t avunit\_carec\_mic\_control\_parameter\_get\_value(CarecMicControlHandle handle,
uint32_t index, const char* channel_code, int32_t* value);
3179
3206 RE API int32_t avunit\_carec\_mic\_control\_parameter\_set\_value(CarecMicControlHandle handle,
uint32_t index, const char* channel_code, int32_t value);
3207
3220 RE API int32_t avunit\_carec\_mic\_control\_parameters\_save(CarecMicControlHandle handle);
3221
3233 RE API int32_t avunit\_carec\_mic\_control\_parameters\_restore(CarecMicControlHandle handle);
3234
3257 RE API int32_t
avunit\_amplifier\_control\_set\_connected\_changed\_callback(AmplifierControlHandle handle,
AVUnitCallbackStateHandle callback_state, DeviceConnectedChanged callback);
```




```
3258
3274 RE API int32_t
avunit\_amplifier\_control\_remove\_connected\_changed\_callback(AmplifierControlHandle handle,
AVUnitCallbackStateHandle callback_state);
3275
3287 RE API int32_t avunit\_amplifier\_control\_is\_connected(AmplifierControlHandle handle);
3288
3305 RE API int32_t avunit\_amplifier\_control\_get\_enabled(AmplifierControlHandle handle, uint32_t
channel);
3306
3323 RE API int32_t avunit\_amplifier\_control\_get\_source(AmplifierControlHandle handle, uint32_t
channel);
3324
3341 RE API int32_t avunit\_amplifier\_control\_get\_volume(AmplifierControlHandle handle, uint32_t
channel, float* volume);
3342
3359 RE API int32_t avunit\_amplifier\_control\_set\_enabled(AmplifierControlHandle handle, uint32_t
channel, int32_t enabled);
3360
3378 RE API int32_t avunit\_amplifier\_control\_set\_source(AmplifierControlHandle handle, uint32_t
channel, uint32_t source);
3379
3396 RE API int32_t avunit\_amplifier\_control\_set\_volume(AmplifierControlHandle handle, uint32_t
channel, float volume);
3397
3420 RE API int32_t avunit\_ups\_control\_set\_connected\_changed\_callback(UPSControlHandle handle,
AVUnitCallbackStateHandle callback_state, DeviceConnectedChanged callback);
3421
3437 RE API int32_t avunit\_ups\_control\_remove\_connected\_changed\_callback(UPSControlHandle
handle, AVUnitCallbackStateHandle callback_state);
3438
3462 RE API int32_t avunit\_ups\_control\_set\_power\_source\_changed\_callback(UPSControlHandle
handle, AVUnitCallbackStateHandle callback_state, StateChangedCallback callback);
3463
3479 RE API int32_t avunit\_ups\_control\_remove\_power\_source\_changed\_callback(UPSControlHandle
handle, AVUnitCallbackStateHandle callback_state);
3480
3492 RE API int32_t avunit\_ups\_control\_is\_connected(UPSControlHandle handle);
3493
3506 RE API int32_t avunit\_ups\_control\_power\_source(UPSControlHandle handle);
3507
3521 RE API int32_t avunit\_ups\_control\_beeper\_enabled(UPSControlHandle handle);
3522
3535 RE API int32_t avunit\_ups\_control\_set\_beeper\_enabled(UPSControlHandle handle, int32_t
enabled);
3536
```



```
3549 RE API int32_t avunit\_ups\_control\_get\_battery\_charge(UPSControlHandle handle, float* charge);
3550
3565 RE API int32_t avunit\_ups\_control\_get\_run\_time\_estimate(UPSControlHandle handle);
3566
3567
3568 /* -----
3569 * | TV API
3570 * -----
3571 */
3572
3595 RE API int32_t avunit\_tv\_control\_set\_connected\_changed\_callback(TVControlHandle handle,
AVUnitCallbackStateHandle callback_state, DeviceConnectedChanged callback);
3596
3612 RE API int32_t avunit\_tv\_control\_remove\_connected\_changed\_callback(TVControlHandle handle,
AVUnitCallbackStateHandle callback_state);
3613
3636 RE API int32_t avunit\_tv\_control\_wake\_on\_lan(TVControlHandle handle, const char* mac, const
char* address, uint32_t port);
3637
3658 RE API int32_t avunit\_tv\_control\_connect(TVControlHandle handle, const char* address, uint32_t
port);
3659
3669 RE API int32_t avunit\_tv\_control\_disconnect(TVControlHandle handle);
3670
3682 RE API int32_t avunit\_tv\_control\_connected(TVControlHandle handle);
3683
3700 RE API int32_t avunit\_tv\_control\_version(TVControlHandle handle, uint32_t* major, uint32_t*
minor, uint32_t* patch);
3701
3717 RE API int32_t avunit\_tv\_control\_start\_streaming(TVControlHandle handle);
3718
3742 RE API int32_t avunit\_tv\_control\_start\_streaming\_to(TVControlHandle handle, const char*
address, uint32_t port);
3743
3758 RE API int32_t avunit\_tv\_control\_close(TVControlHandle handle);
3759
3760
3761 /* -----
3762 * | Speech synthesis API
3763 * -----
3764 */
3765
3785 RE API int32_t avunit\_speech\_synthesis\_control\_read\_text(SpeechSynthesisControlHandle
handle, const char* text);
3786
```



```
3802 RE API int32_t avunit_speech_synthesis_control_stop_reading(SpeechSynthesisControlHandle
handle);
3803
3816 RE API int32_t avunit_speech_synthesis_control_gong_enabled(SpeechSynthesisControlHandle
handle);
3817
3828 RE API int32_t
avunit_speech_synthesis_control_set_gong_enabled(SpeechSynthesisControlHandle handle, int32_t
enabled);
3829
3830
3831 /* -----
3832 * | Free space API
3833 * -----
3834 */
3835
3867 RE API int32_t avunit_free_space_control_set_recording_parameters(FreeSpaceControlHandle
handle, const char* recording_dir_path,
3868     const char* audio_codec, const char* video_codec, float audio_quality, float video_quality,
int32_t video_enabled);
3869
3883 RE API int64_t avunit_free_space_control_total_bytes(FreeSpaceControlHandle handle);
3884
3898 RE API int64_t avunit_free_space_control_total_minutes(FreeSpaceControlHandle handle);
3899
3913 RE API int64_t avunit_free_space_control_free_space_in_bytes(FreeSpaceControlHandle handle);
3914
3928 RE API int64_t avunit_free_space_control_free_space_in_minutes(FreeSpaceControlHandle
handle);
3929
3944 RE API int64_t avunit_free_space_control_bytes_per_minute(FreeSpaceControlHandle handle);
3945
3946 /* -----
3947 * | NTP API
3948 * -----
3949 */
3950
3965 RE API int32_t avunit_ntp_client_create(NTPClientHandle* handle);
3966
3977 RE API int32_t avunit_ntp_client_destroy(NTPClientHandle handle);
3978
3997 RE API int32_t avunit_ntp_client_add_server(NTPClientHandle handle, const char* address,
uint32_t port);
3998
4012 RE API int32_t avunit_ntp_client_synchronize(NTPClientHandle handle);
4013
```



```
4031 RE API int32_t avunit_set_system_time(uint64_t seconds, uint32_t nanoseconds);
4032
4033 /* -----
4034 * | Fatal error API
4035 * -----
4036 */
4037
4055 RE API int32_t avunit_fatal_error_set_callback(FatalErrorCallback callback);
4056
4057 #if defined(__cplusplus)
4058 } /* extern "C" */
4059 #endif /* __cplusplus */
```

Dokumentacja API dostarczanego przez bibliotekę casmb

Biblioteka służy do komunikacji po protokole SMB (CIFS) z przestrzeni użytkownika, bez polegania na implementacji w systemie operacyjnym. Biblioteka pozwala na jednoczesne połączenia do tego samego serwera używając różnych poświadczeń autoryzacyjnych, obchodząc ograniczenia systemu Windows.

Biblioteka udostępnia wspólne API, wykorzystywane zarówno na systemie Windows, jak i Linux.

Biblioteka jest wymagana przez projekt MediaServer. Wymaga on następujących plików:

- na systemie Windows wymagany jest 64 bitowy plik DLL o nazwie casmb.dll
- na systemie Linux wymagany jest 64 bitowy plik SO o nazwie libcasmb.so

1 Dokumentacja grup

1.1 Wartości globalne

Wartości globalne wykorzystywane przez bibliotekę.

1.1.1 Definicje

- #define [CASMB_EXPORT](#)
*Makro eksportujące funkcje API w bibliotece *.dll/*.so.*

1.1.2 Opis szczegółowy

Wartości globalne wykorzystywane przez bibliotekę.

1.2 Typy

Typy wykorzystywane przez bibliotekę.

1.2.1 Struktury danych

- struct [CASMBAttributes](#)
Struktura przechowująca atrybuty pliku.
- struct [CASMBDirectoryEntry](#)
Struktura opisuje jeden wpis w katalogu.

1.2.2 Definicje typów

- typedef struct [CASMBClient](#) [CASMBClient](#)
Struktura, przechowująca dane klienta biblioteki casmb.
- typedef struct [CASMBFile](#) [CASMBFile](#)
Struktura, przechowująca informacje o otwartym pliku (lub katalogu).
- typedef struct [CASMBSearch](#) [CASMBSearch](#)
Struktura, przechowująca informacje o aktualnie otwartym wyszukiwaniu (listowaniu) plików/katalogów znajdujących się w podanej ścieżce.
- typedef uint16_t [utf16_t](#)
Typ zmiennej przechowującej jeden znak w standardzie UTF16.
- typedef void *(* [CASMBAllocateFunction](#)) (size_t size, void *user_data)



Callback, który jest wołany przez bibliotekę w celu alokacji pamięci.

- typedef void(* [CASMBFreeFunction](#)) (void *ptr, void *user_data)
Callback, który jest wołany przez bibliotekę w celu zwolnienia wcześniej zaalokowanej przez nią pamięci.

1.2.3 Wyliczenia

- enum [CASMBResult](#) { [CASMB_OK](#) = 0, [CASMB_NO_MORE_ENTRIES](#) = 1, [CASMB_ERR_ALREADY_EXISTS](#) = -1, [CASMB_ERR_IS_DIRECTORY](#) = -2, [CASMB_ERR_IS_FILE](#) = -3, [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) = -6, [CASMB_ERR_FILE_NOT_FOUND](#) = -7, [CASMB_ERR_ACCESS_DENIED](#) = -8, [CASMB_ERR_DISK_FULL](#) = -9, [CASMB_ERR_OUT_OF_MEMORY](#) = -100, [CASMB_ERR_INVALID_ARGUMENT](#) = -101, [CASMB_ERR_INVALID_CLIENT](#) = -102, [CASMB_ERR_NOT_CONNECTED](#) = -103, [CASMB_ERR_ALREADY_CONNECTED](#) = -104, [CASMB_ERR_NO_SERVER](#) = -105, [CASMB_ERR_NO_SHARE](#) = -106, [CASMB_ERR_NETWORK_ERROR](#) = -107, [CASMB_ERR_NETWORK_TIMEOUT](#) = -108, [CASMB_ERR_INVALID_RESPONSE](#) = -109, [CASMB_ERR_INVALID_CREDENTIALS](#) = -110, [CASMB_ERR_CONNECTION_CLOSED](#) = -111, [CASMB_ERR_PLAINTEXT_LOGIN_NOT_SUPPORTED](#) = -112, [CASMB_ERR_ASCII_NOT_SUPPORTED](#) = -113, [CASMB_ERR_NOT_IMPLEMENTED](#) = -1000, [CASMB_ERR_INTERNAL_ERROR](#) = -1001 } *Enumeracja możliwych wartości zwracanych z funkcji API.*
- enum [CASMBDirEntType](#) { [CASMB_ENTRY_TYPE_DIRECTORY](#) = 1, [CASMB_ENTRY_TYPE_FILE](#) = 2 } *Flagi bitowe, oznaczające typ danych do wyszukania w podanym katalogu.*
- enum [CASMBDesiredAccess](#) { [CASMB_FILE_READ_DATA](#) = 0x00000001, [CASMB_FILE_WRITE_DATA](#) = 0x00000002, [CASMB_FILE_APPEND_DATA](#) = 0x00000004, [CASMB_FILE_READ_EA](#) = 0x00000008, [CASMB_FILE_WRITE_EA](#) = 0x00000010, [CASMB_FILE_EXECUTE](#) = 0x00000020, [CASMB_FILE_READ_ATTRIBUTES](#) = 0x00000080, [CASMB_FILE_WRITE_ATTRIBUTES](#) = 0x00000100, [CASMB_FILE_DELETE](#) = 0x00010000, [CASMB_FILE_MAXIMUM_ALLOWED](#) = 0x02000000, [CASMB_FILE_GENERIC_ALL](#) = 0x10000000, [CASMB_FILE_GENERIC_EXECUTE](#) = 0x20000000, [CASMB_FILE_GENERIC_WRITE](#) = 0x40000000, [CASMB_FILE_GENERIC_READ](#) = 0x80000000 } *Flagi bitowe, określające dostęp do poszczególnych operacji na pliku.*
- enum [CASMBFileAttribute](#) { [CASMB_ATTR_READONLY](#) = 0x00000001, [CASMB_ATTR_HIDDEN](#) = 0x00000002, [CASMB_ATTR_SYSTEM](#) = 0x00000004, [CASMB_ATTR_DIRECTORY](#) = 0x00000010, [CASMB_ATTR_ARCHIVE](#) = 0x00000020, [CASMB_ATTR_NORMAL](#) = 0x00000080, [CASMB_ATTR_TEMPORARY](#) = 0x00000100, [CASMB_ATTR_COMPRESSED](#) = 0x00000800, [CASMB_ATTR_POSIX_SEMANTICS](#) = 0x01000000, [CASMB_ATTR_BACKUP_SEMANTICS](#) = 0x02000000, [CASMB_ATTR_DELETE_ON_CLOSE](#) = 0x04000000, [CASMB_ATTR_SEQUENTIAL_SCAN](#) = 0x08000000, [CASMB_ATTR_RANDOM_ACCESS](#) = 0x10000000, [CASMB_ATTR_NO_BUFFERING](#) = 0x20000000, [CASMB_ATTR_WRITE_THROUGH](#) = 0x80000000 } *Flagi bitowe, określające atrybuty plików.*



- enum [CASMBShareAccess](#) { [CASMB_FILE_SHARE_NONE](#) = 0x00000000, [CASMB_FILE_SHARE_READ](#) = 0x00000001, [CASMB_FILE_SHARE_WRITE](#) = 0x00000002, [CASMB_FILE_SHARE_DELETE](#) = 0x00000004 } *Flagi bitowe, określające sposób współdzielenia otwartego pliku.*
- enum [CASMBCreateDisposition](#) { [CASMB_FILE_SUPERSEDE](#) = 0x00000000, [CASMB_FILE_OPEN](#) = 0x00000001, [CASMB_FILE_CREATE](#) = 0x00000002, [CASMB_FILE_OPEN_IF](#) = 0x00000003, [CASMB_FILE_OVERWRITE](#) = 0x00000004, [CASMB_FILE_OVERWRITE_IF](#) = 0x00000005 } *Akcja, jaką należy podjąć w przypadku pliku lub katalogu który nie istnieje.*
- enum [CASMBSeekDirection](#) { [CASMB_SEEK_BEGINNING](#) = 0, [CASMB_SEEK_SKIP](#) = 1, [CASMB_SEEK_END](#) = 2 } *Sposób przesunięcia (zmiany pozycji kursora) w pliku.*

1.2.4 Zmienne

- uint64_t [CASMBAttributes::creation_time](#)
Czas utworzenia pliku.
- uint64_t [CASMBAttributes::last_access_time](#)
Czas ostatniego dostępu do pliku.
- uint64_t [CASMBAttributes::last_write_time](#)
Czas ostatniego zapisu do pliku.
- uint64_t [CASMBAttributes::change_time](#)
Czas ostatniej zmiany w pliku.
- uint64_t [CASMBAttributes::file_size](#)
Rozmiar pliku w bajtach.
- uint32_t [CASMBAttributes::attributes](#)
Atrybuty pliku zapisane jako suma bitowa wartości [CASMBFileAttribute](#).
- [CASMBAttributes](#) [CASMBDirectoryEntry::attributes](#)
Atrybuty pliku.
- uint32_t [CASMBDirectoryEntry::name_length](#)
Długość nazwy pliku w znakach UTF-16.
- const [utf16_t](#) * [CASMBDirectoryEntry::name](#)
Nazwa pliku lub katalogu.

1.2.5 Opis szczegółowy

Typy wykorzystywane przez bibliotekę.

1.2.6 Dokumentacja definicji typów

1.2.6.1 typedef struct [CASMBClient](#) [CASMBClient](#)

Struktura, przechowująca dane klienta biblioteki casmb.

Spostrzeżenia:

Zawartość struktury nie jest podana jawnie i jest szczegółem implementacyjnym.

Definicja w linii [482](#) pliku [casmb.h](#).



1.2.6.2 typedef struct [CASMBFile](#) [CASMBFile](#)

Struktura, przechowująca informacje o otwartym pliku (lub katalogu).

Spostrzeżenia:

Zawartość struktury nie jest podana jawnie i jest szczegółem implementacyjnym.

Definicja w linii [489](#) pliku [casmb.h](#).

1.2.6.3 typedef struct [CASMBSearch](#) [CASMBSearch](#)

Struktura, przechowująca informacje o aktualnie otwartym wyszukiwaniu (listowaniu) plików/katalogów znajdujących się w podanej ścieżce.

Spostrzeżenia:

Zawartość struktury nie jest podana jawnie i jest szczegółem implementacyjnym.

Definicja w linii [497](#) pliku [casmb.h](#).

1.2.6.4 typedef void>(* CASMBAllocateFunction) (size_t size, void *user_data)

Callback, który jest wołany przez bibliotekę w celu alokacji pamięci.

Callback jest odpowiednikiem funkcji malloc() i umożliwia zdefiniowanie własnych metod na alokację pamięci.

Parametry:

in	<i>size</i>	Rozmiar alokacji w bajtach
in	<i>user_data</i>	Wskaźnik na dane przekazane przez użytkownika biblioteki

Zwraca:

Wskaźnik na nowo zaalokowany obszar pamięci lub NULL w przypadku niepowodzenia

Definicja w linii [581](#) pliku [casmb.h](#).

1.2.6.5 typedef void(* CASMBFreeFunction) (void *ptr, void *user_data)

Callback, który jest wołany przez bibliotekę w celu zwolnienia wcześniej zaalokowanej przez nią pamięci.

Callback jest odpowiednikiem funkcji free() i umożliwia zdefiniowanie własnych metod na alokację pamięci.

Parametry:

in	<i>ptr</i>	Wskaźnik na pamięć zaalokowaną wcześniej przez bibliotekę
----	------------	---



in	<i>user_data</i>	Wskaźnik na dane przekazane przez użytkownika biblioteki
----	------------------	--

Definicja w linii [593](#) pliku [casmb.h](#).

1.2.7 Dokumentacja typów wyliczanych

1.2.7.1 enum [CASMBResult](#)

Enumeracja możliwych wartości zwracanych z funkcji API.

Wartości błędów zawsze są mniejsze od 0, dzięki czemu możliwe jest odróżnienie wystąpienia błędu od poprawnej wartości. W opisie każdej z funkcji jest dostępna lista błędów możliwych do wystąpienia.

Wartości wyliczeń

- **CASMB_OK** Funkcja zakończyła się poprawnie.
Wartość zwracana jest, jeśli funkcja nie zwraca innej wartości dodatniej
- **CASMB_NO_MORE_ENTRIES** Wszystkie elementy danej zostały już zwrócone.
Wartość zwracana jest, jeśli odczytano już wszystkie wartości, które zostały zwrócone przez żądanie. Nie jest to błąd, ale informacja dla aplikacji, żeby zakończyć operację.
- **CASMB_ERR_ALREADY_EXISTS** Plik o podanej nazwie już istnieje.
- **CASMB_ERR_IS_DIRECTORY** Żądano otwarcia pliku, ale element docelowy jest katalogiem.
- **CASMB_ERR_IS_FILE** Żądano otwarcia katalogu, ale element docelowy jest plikiem.
- **CASMB_ERR_NO_DIRECTORY_COMPONENT** Podana ścieżka jest nieprawidłowa - brak jednego z katalogów w ścieżce.
- **CASMB_ERR_FILE_NOT_FOUND** Podany plik nie istnieje.
- **CASMB_ERR_ACCESS_DENIED** Brak dostępu do elementu docelowego.
- **CASMB_ERR_DISK_FULL** Dysk na serwerze został zapełniony.
- **CASMB_ERR_OUT_OF_MEMORY** Nie można zaalokować wystarczającej ilości pamięci, aby wykonać operację
- **CASMB_ERR_INVALID_ARGUMENT** Jeden z przekazanych argumentów jest niepoprawny.
- **CASMB_ERR_INVALID_CLIENT** Przekazany uchwyt do obiektu klienta jest niepoprawny.
- **CASMB_ERR_NOT_CONNECTED** Klient nie jest połączony z udziałem.
- **CASMB_ERR_ALREADY_CONNECTED** Klient jest już połączony z udziałem.
- **CASMB_ERR_NO_SERVER** Nie udało się połączyć z podanym adresem serwera.
- **CASMB_ERR_NO_SHARE** Serwer nie ma podanego udziału.
- **CASMB_ERR_NETWORK_ERROR** Wystąpił błąd warstwy sieciowej.
- **CASMB_ERR_NETWORK_TIMEOUT** Minął czas przeznaczony na wykonanie operacji, ale serwer nie odpowiedział
- **CASMB_ERR_INVALID_RESPONSE** Serwer odpowiedział w nieobsługiwany sposób.
- **CASMB_ERR_INVALID_CREDENTIALS** Podane dane uwierzytelniające (login/hasło) są niepoprawne.



- **CASMB_ERR_CONNECTION_CLOSED** Serwer zamknął połączenie.
- **CASMB_ERR_PLAINTEXT_LOGIN_NOT_SUPPORTED** Serwer obsługuje tylko logowanie bez szyfrowania.
Ze względów bezpieczeństwa logowanie z jawnym podaniem danych uwierzytelniających nie jest obsługiwane.
- **CASMB_ERR_ASCII_NOT_SUPPORTED** Serwer wymaga kodowania ASCII.
Biblioteka obsługuje tylko kodowanie UNICODE, które umożliwia poprawne zapisanie nazw plików zawierających znaki narodowe.
- **CASMB_ERR_NOT_IMPLEMENTED** Serwer wymaga od nas funkcjonalności, która nie została jeszcze zaimplementowana.
- **CASMB_ERR_INTERNAL_ERROR** Wewnętrzny błąd biblioteki.

Definicja w linii [170](#) pliku [casmb.h](#).

1.2.7.2 enum [CASMBDirEntType](#)

Flagi bitowe, oznaczające typ danych do wyszukania w podanym katalogu.

Flagi są wymagane przez funkcję [casmb_search](#). Wartości mogą być ze sobą łączone za pomocą sumy bitowej, przykładowo aby wyszukać wpisy będące katalogiem lub plikiem:

```
1 uint32_t entry_type = CASMB_ENTRY_TYPE_DIRECTORY | CASMB_ENTRY_TYPE_FILE;
```

Wartości wyliczeń

- **CASMB_ENTRY_TYPE_DIRECTORY** Wyszukanie tylko katalogów.
- **CASMB_ENTRY_TYPE_FILE** Wyszukanie tylko plików.

Definicja w linii [220](#) pliku [casmb.h](#).

1.2.7.3 enum [CASMBDesiredAccess](#)

Flagi bitowe, określające dostęp do poszczególnych operacji na pliku.

Wartości wyliczeń

- **CASMB_FILE_READ_DATA** CASMB_FILE_READ_DATA.
Dostęp do odczytu z pliku.
- **CASMB_FILE_WRITE_DATA** CASMB_FILE_WRITE_DATA.
Dostęp do zapisu do pliku.
- **CASMB_FILE_APPEND_DATA** CASMB_FILE_APPEND_DATA.
Dostęp do zapisu tylko poza obecny koniec pliku.
- **CASMB_FILE_READ_EA** CASMB_FILE_READ_EA.
Dostęp do odczytu rozszerzonych atrybutów pliku.
- **CASMB_FILE_WRITE_EA** CASMB_FILE_WRITE_EA.
Dostęp do zapisu rozszerzonych atrybutów pliku.
- **CASMB_FILE_EXECUTE** CASMB_FILE_EXECUTE.
Dostęp do wykonywania pliku.
- **CASMB_FILE_READ_ATTRIBUTES** CASMB_FILE_READ_ATTRIBUTES.
Dostęp do odczytu atrybutów pliku.



- **CASMB_FILE_WRITE_ATTRIBUTES** CASMB_FILE_WRITE_ATTRIBUTES.
Dostęp do zapisu atrybutów pliku.
- **CASMB_FILE_DELETE** CASMB_FILE_DELETE.
Dostęp do usuwania pliku.
- **CASMB_FILE_MAXIMUM_ALLOWED** CASMB_FILE_MAXIMUM_ALLOWED.
Otwarcie pliku z najwyższym dozwolonym dostępem.
- **CASMB_FILE_GENERIC_ALL** CASMB_FILE_GENERIC_ALL.
Wartość jest tożsama z wyspecyfikowaniem wszystkich powyższych flag poza [CASMB_FILE_MAXIMUM_ALLOWED](#).
- **CASMB_FILE_GENERIC_EXECUTE** CASMB_FILE_GENERIC_EXECUTE.
Wartość jest tożsama z wyspecyfikowaniem flag [CASMB_FILE_READ_ATTRIBUTES](#) oraz [CASMB_FILE_EXECUTE](#).
- **CASMB_FILE_GENERIC_WRITE** CASMB_FILE_GENERIC_WRITE.
Wartość jest tożsama z wyspecyfikowaniem flag [CASMB_FILE_WRITE_DATA](#), [CASMB_FILE_APPEND_DATA](#), [CASMB_FILE_WRITE_ATTRIBUTES](#) oraz [CASMB_FILE_WRITE_EA](#).
- **CASMB_FILE_GENERIC_READ** CASMB_FILE_GENERIC_READ.
Wartość jest tożsama z wyspecyfikowaniem flag [CASMB_FILE_READ_DATA](#), [CASMB_FILE_READ_ATTRIBUTES](#) oraz [CASMB_FILE_READ_EA](#).

Definicja w linii [273](#) pliku [casmb.h](#).

1.2.7.4 enum [CASMBFileAttribute](#)

Flagi bitowe, określające atrybuty plików.

Wartości wyliczeń

- **CASMB_ATTR_READONLY** Plik tylko do odczytu. Aplikacje nie mogą zapisywać czy usunąć takiego pliku.
- **CASMB_ATTR_HIDDEN** Plik jest ukryty. Nie powinien on się pojawiać w zwykłym listowaniu zawartości katalogu.
- **CASMB_ATTR_SYSTEM** Plik jest częścią systemu operacyjnego lub jest stworzony dla jego wyłącznego użycia.
- **CASMB_ATTR_DIRECTORY** Plik jest katalogiem.
- **CASMB_ATTR_ARCHIVE** Plik nie został zarchiwizowany od ostatniej modyfikacji.
- **CASMB_ATTR_NORMAL** Plik nie posiada innych atrybutów. Wartość jest dopuszczalna tylko wtedy, gdy nie inne atrybuty nie są ustawione.
- **CASMB_ATTR_TEMPORARY** Plik jest plikiem tymczasowym.
- **CASMB_ATTR_COMPRESSED** Plik jest skompresowany na dysku. Nie zmienia to sposobu odczytu czy zapisu pliku.
- **CASMB_ATTR_POSIX_SEMANTICS** Dostęp do pliku powinien się odbywać zgodnie ze standardem POSIX.
- Między innymi oznacza to, że nazwy plików mogą się różnić tylko wielkością znaków.



- **CASMB_ATTR_BACKUP_SEMANTICS** Otwierany lub tworzony plik jest przeznaczony do tworzenia kopii zapasowej.
- **CASMB_ATTR_DELETE_ON_CLOSE** Plik zostanie skasowany po zamknięciu go przez wszystkie procesy.
- **CASMB_ATTR_SEQUENTIAL_SCAN** Plik powinien być odczytywany/zapisywany sekwencyjnie (od początku do końca).
- **CASMB_ATTR_RANDOM_ACCESS** Aplikacja będzie odczytywać plik przeskakując w losowe miejsca.
Flaga jest wykorzystywana przez serwer, aby zoptymalizować buforowanie.
- **CASMB_ATTR_NO_BUFFERING** Żądanie od serwera otwarcia pliku bez żadnego buforowania.
Serwer może zignorować tę flagę.
- **CASMB_ATTR_WRITE_THROUGH** Plik zostaje zapisany bezpośrednio na dysku, z pominięciem buforowania.
Po pomyślnym wykonaniu operacji zapisu mamy pewność, że dane znajdują się na fizycznym nośniku. Uwaga: operacje zapisu będą wielokrotnie wolniejsze po użyciu tej flagi.

Definicja w linii [350](#) pliku [casmb.h](#).

1.2.7.5 enum [CASMBShareAccess](#)

Flagi bitowe, określające sposób współdzielenia otwartego pliku.

Wartości mogą być ze sobą łączone za pomocą sumy bitowej. Przykładowo, aby współdzielić plik do odczytu i zapisu możemy napisać:

```
1 uint32_t share_access = CASMB_FILE_SHARE_READ | CASMB_FILE_SHARE_WRITE;
```

Wartości wyliczeń

- **CASMB_FILE_SHARE_NONE** Brak współdzielenia pliku po otwarciu.
Próba otwarcia tego pliku przez inne procesy nie będzie możliwa do czasu zamknięcia go.
- **CASMB_FILE_SHARE_READ** Współdzielenie tego pliku do odczytu.
Jeśli flaga nie zostanie podana, to inne procesy nie będą mogły otworzyć pliku, jeśli będą żądać dostępu do jego odczytu.
- **CASMB_FILE_SHARE_WRITE** Współdzielenie tego pliku do zapisu.
Współdzielenie możliwości usunięcia pliku.
Jeśli flaga nie zostanie podana, to inne procesy nie będą mogły otworzyć pliku, jeśli będą żądać dostępu do jego zapisu.
Jeśli flaga nie zostanie podana, to inne procesy nie będą mogły otworzyć pliku, jeśli będą żądać dostępu do usunięcia pliku.

Definicja w linii [402](#) pliku [casmb.h](#).

1.2.7.6 enum [CASMBCreateDisposition](#)

Akcja, jaką należy podjąć w przypadku pliku lub katalogu który nie istnieje.

Wartości wyliczeń

- **CASMB_FILE_SUPERSEDE** Utworzenie nowego pliku lub zastąpienie starego. Jeśli plik o podanej nazwie nie istnieje, to zostanie on utworzony. Jeśli taki plik istnieje, to zostanie on nadpisany przez nowy plik.
- **CASMB_FILE_OPEN** Otwarcie istniejącego pliku. Jeśli plik o podanej nazwie nie istnieje, to operacja otwarcia nie powiedzie się. Jeśli taki plik istnieje, to zostanie on otwarty.
- **CASMB_FILE_CREATE** Utworzenie nowego pliku. Jeśli plik o podanej nazwie nie istnieje, to zostanie on utworzony. Jeśli taki plik istnieje, to operacja utworzenia nie powiedzie się.
- **CASMB_FILE_OPEN_IF** Otwarcie istniejącego pliku lub utworzenie nowego. Jeśli plik o podanej nazwie nie istnieje, to zostanie on utworzony. Jeśli taki plik istnieje, to zostanie on otwarty.
- **CASMB_FILE_OVERWRITE** Zastąpienie istniejącego pliku. Jeśli plik o podanej nazwie nie istnieje, operacja zastąpienia nie powiedzie się. Jeśli taki plik istnieje, to zostanie on nadpisany przez nowy plik.
- **CASMB_FILE_OVERWRITE_IF** Zastąpienie istniejącego pliku lub utworzenie nowego. Jeśli plik o podanej nazwie nie istnieje, to zostanie on utworzony. Jeśli taki plik istnieje, to zostanie on nadpisany przez nowy plik.

Definicja w linii [449](#) pliku [casmb.h](#).

1.2.7.7 enum [CASMBSeekDirection](#)

Sposób przesunięcia (zmiany pozycji kursora) w pliku.

Wartości wyliczeń

- **CASMB_SEEK_BEGINNING** Przesunięcie jest ustawiane na podaną wartość
- **CASMB_SEEK_SKIP** Przesunięcie jest ustawiane na aktualną pozycję plus podany offset.
- **CASMB_SEEK_END** Przesunięcie jest ustawiane na rozmiar pliku plus podany offset.

Definicja w linii [471](#) pliku [casmb.h](#).

1.2.8 Dokumentacja zmiennych

1.2.8.1 [CASMBAttributes::creation_time](#)

Czas utworzenia pliku.

Czas zapisany jest jako wielokrotność wartości 100ns, gdzie wartość 0 oznacza początek roku 1601 (1 stycznia 1601) w kalendarzu Gregoriańskim. Wartość jest tożsama ze sposobem zapisu czasu przez system Windows w strukturze [FILETIME](#).

Definicja w linii [538](#) pliku [casmb.h](#).

1.2.8.2 [CASMBAttributes::last_access_time](#)

Czas ostatniego dostępu do pliku.

Czas zapisany jest identycznie jak [CASMBAttributes::creation_time](#)



Definicja w linii [539](#) pliku [casmb.h](#).

1.2.8.3 CASMBAttributes::last_write_time

Czas ostatniego zapisu do pliku.

Czas zapisany jest identycznie jak [CASMBAttributes::creation_time](#)

Definicja w linii [540](#) pliku [casmb.h](#).

1.2.8.4 CASMBAttributes::change_time

Czas ostatniej zmiany w pliku.

Czas zapisany jest identycznie jak [CASMBAttributes::creation_time](#)

Definicja w linii [541](#) pliku [casmb.h](#).

1.2.8.5 CASMBDirectoryEntry::name

Nazwa pliku lub katalogu.

Pamięć na nazwę alokowana jest przez bibliotekę casmb i nie należy jej zwalniać. Nie należy alokować tej wartości przy przekazywaniu tej struktury do biblioteki.

Definicja w linii [567](#) pliku [casmb.h](#).

1.3 Funkcje narzędziowe

Funkcje narzędziowe, które pomagają w pracy z biblioteką

1.3.1 Funkcje

- [CASMB_EXPORT](#) const char * [casmb_err_to_str](#) ([CASMBResult](#) result)
Zamiana wyniku zwracanego przez funkcje biblioteki na napis.

1.3.2 Opis szczegółowy

Funkcje narzędziowe, które pomagają w pracy z biblioteką

1.3.3 Dokumentacja funkcji

1.3.3.1 [CASMB_EXPORT](#) const char* [casmb_err_to_str](#) ([CASMBResult](#) result)

Zamiana wyniku zwracanego przez funkcje biblioteki na napis.

Parametry:

in	result	Wynik zwrócony przez jedną z funkcji biblioteki
----	--------	---

Zwraca:

Napis odpowiadający przekazanej wartości.

1.4 Funkcje inicjalizacyjne

Funkcje, które pozwalają stworzyć oraz zniszczyć obiekt klienta.

1.4.1 Funkcje

- [CASMB_EXPORT](#) void [casmb_network_initialize \(\)](#)
Inicjalizacja podsystemu sieci w systemie operacyjnym.
- [CASMB_EXPORT](#) void [casmb_network_shutdown \(\)](#)
Zamknięcie podsystemu sieci w systemie operacyjnym.
- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_open_client](#) ([CASMBClient](#) **client)
Utworzenie obiektu klienta, wykorzystując do tego celu domyślny alokator.
- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_open_client_with_allocator](#) ([CASMBClient](#) **client, [CASMBAllocateFunction](#) allocate, [CASMBFreeFunction](#) free, void *user_data)
Utworzenie obiektu klienta, wykorzystując do tego celu alokator przekazany przez użytkownika.
- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_close_client](#) ([CASMBClient](#) *client)
Zamknięcie klienta oraz zwolnienie zajmowanej przez niego pamięci.
- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_client_set_timeout](#) ([CASMBClient](#) *client, uint32_t timeout)
Ustawienie czasu, jaki klient będzie oczekiwał na odpowiedź od serwera.

1.4.2 Opis szczegółowy

Funkcje, które pozwalają stworzyć oraz zniszczyć obiekt klienta.

1.4.3 Dokumentacja funkcji

1.4.3.1 [CASMB_EXPORT](#) void [casmb_network_initialize \(\)](#)

Inicjalizacja podsystemu sieci w systemie operacyjnym.

Funkcja musi być wywołana przed użyciem innych funkcji biblioteki.

1.4.3.2 [CASMB_EXPORT](#) void [casmb_network_shutdown \(\)](#)

Zamknięcie podsystemu sieci w systemie operacyjnym.

Funkcja powinna być zawołana po zakończeniu pracy aplikacji z biblioteką.

1.4.3.3 [CASMB_EXPORT](#) [CASMBResult](#) [casmb_open_client](#) ([CASMBClient](#) ** client)

Utworzenie obiektu klienta, wykorzystując do tego celu domyślny alokator.

Parametry:

out	client	Wskaźnik, do którego zostanie zapisany utworzony obiekt klienta
-----	--------	---

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik ma wartość NULL



1.4.3.4 [CASMB_EXPORT CASMBResult casmb_open_client_with_allocator \(CASMBClient ** client, CASMBAllocateFunction allocate, CASMBFreeFunction free, void * user_data\)](#)

Utworzenie obiektu klienta, wykorzystując do tego celu alokator przekazany przez użytkownika.

Parametry:

out	<i>client</i>	Wskaźnik, do którego zostanie zapisany utworzony obiekt klienta
in	<i>allocate</i>	Callback, który będzie wołany do alokacji pamięci
in	<i>free</i>	Callback, który będzie wołany do zwalniania pamięci
in	<i>user_data</i>	Wskaźnik na dane użytkownika; wskaźnik ten będzie przekazywany do funkcji alokacji

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik ma wartość NULL

1.4.3.5 [CASMB_EXPORT CASMBResult casmb_close_client \(CASMBClient * client\)](#)

Zamknięcie klienta oraz zwolnienie zajmowanej przez niego pamięci.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
----	---------------	----------------------------

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny

1.4.3.6 [CASMB_EXPORT CASMBResult casmb_client_set_timeout \(CASMBClient * client, uint32_t timeout\)](#)

Ustawienie czasu, jaki klient będzie oczekiwał na odpowiedź od serwera.

Po upływie tego czasu klient będzie zwracał błąd związany z przekroczeniem czasu na odpowiedź.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>timeout</i>	Wartość w sekundach, jaką klient będzie czekał na odpowiedź od serwera

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny

1.5 Funkcje połączeniowe

Funkcje, które pozwalają na podłączenie się do udziału.

1.5.1 Funkcje

- [CASMB_EXPORT_CASMBResult_casmb_set_login_credentials](#) ([CASMBClient](#) *client, const [utf16_t](#) *domain, const [utf16_t](#) *username, const [utf16_t](#) *password)
Ustawienie danych autoryzacyjnych, które posłużą do połączenia z udziałem.
- [CASMB_EXPORT_CASMBResult_casmb_connect](#) ([CASMBClient](#) *client, const [utf16_t](#) *server, const [utf16_t](#) *share, uint32_t port)
Podłączenie się do udziału na serwerze.

1.5.2 Opis szczegółowy

Funkcje, które pozwalają na podłączenie się do udziału.

1.5.3 Dokumentacja funkcji

1.5.3.1 [CASMB_EXPORT_CASMBResult_casmb_set_login_credentials](#) ([CASMBClient](#) * client, const [utf16_t](#) * domain, const [utf16_t](#) * username, const [utf16_t](#) * password)

Ustawienie danych autoryzacyjnych, które posłużą do połączenia z udziałem.

Dane muszą być ustawione przed próbą połączenia. W przypadku braku danych autoryzacyjnych klient spróbuje się zalogować jako użytkownik anonimowy (gość).

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>domain</i>	Domena, do której należy serwer lub wartość NULL, jeśli brak domeny
in	<i>username</i>	Nazwa użytkownika, na którego konto chcemy się zalogować
in	<i>password</i>	Hasło użytkownika, na którego konto chcemy się zalogować

Zwraca:

- [CASMB_OK](#) - pomyślnie ustawiono dane autoryzacyjne
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana nazwa użytkownika ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazane hasło użytkownika ma wartość NULL
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację



1.5.3.2 [CASMB_EXPORT](#) [CASMBResult](#) [casmb_connect](#) ([CASMBClient](#) * *client*, const [utf16_t](#) * *server*, const [utf16_t](#) * *share*, [uint32_t](#) *port*)

Podłączenie się do udziału na serwerze.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>server</i>	Adres serwera, do którego chcemy się podłączyć
in	<i>share</i>	Nazwa udziału, do którego chcemy się podłączyć
in	<i>port</i>	Port serwera, na którym nasłuchuje serwer SMB (domyślnie 445)

Zwraca:

- [CASMB_OK](#) - pomyślnie połączono z udziałem na serwerze
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany adres serwera ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana nazwa udziału ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany port jest niepoprawny
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_ALREADY_CONNECTED](#) - połączenie do serwera jest już aktywne w kliencie
- [CASMB_ERR_NETWORK_ERROR](#) - podczas łączenia do serwera wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_NO_SERVER](#) - nie udało się rozstrzygnąć adresu serwera
- [CASMB_ERR_NO_SHARE](#) - serwer nie posiada wymaganego udziału
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_INVALID_CREDENTIALS](#) - autoryzacja przy pomocy podanych danych nie powiodła się
- [CASMB_ERR_PLAINTEXT_LOGIN_NOT_SUPPORTED](#) - serwer nie obsługuje szyfrowania danych autoryzacyjnych
- [CASMB_ERR_ASCII_NOT_SUPPORTED](#) - serwer nie obsługuje kodowania znaków UNICODE

Spostrzeżenia:

Adres serwera może być podany jako adres IPv4, IPv6 oraz jako nazwa domenowa.

1.6 Operacje na plikach

Otwieranie, tworzenie, odczyt oraz modyfikacja plików.

1.6.1 Funkcje

- [CASMB_EXPORT CASMBResult casmb_open](#) ([CASMBClient](#) *client, [CASMBFile](#) **file, const [utf16_t](#) *url, uint32_t desired_access, uint32_t create_mode, uint32_t share_mode, [CASMBAttributes](#) *attributes)
Otwarcie pliku lub katalogu zgodnie z przekazanymi flagami.
- [CASMB_EXPORT CASMBResult casmb_close](#) ([CASMBFile](#) *file)
Zamknięcie pliku na serwerze oraz zwolnienie obiektu pliku w pamięci.
- [CASMB_EXPORT CASMBResult casmb_set_size](#) ([CASMBFile](#) *file, uint64_t size)
Ustawienie nowego rozmiaru pliku.
- [CASMB_EXPORT](#) int64_t [casmb_size](#) ([CASMBFile](#) *file)
Pobranie aktualnego rozmiaru pliku w bajtach.
- [CASMB_EXPORT](#) int64_t [casmb_position](#) ([CASMBFile](#) *file)
Pobranie aktualnego przesunięcia (pozycji) w pliku licząc od jego początku.
- [CASMB_EXPORT](#) int64_t [casmb_seek](#) ([CASMBFile](#) *file, uint64_t offset, uint32_t whence)
Zmiana aktualnego przesunięcia (pozycji) w pliku.
- [CASMB_EXPORT](#) int32_t [casmb_read](#) ([CASMBFile](#) *file, void *data, uint32_t offset, uint32_t data_size)
Odczytanie danych z pliku.
- [CASMB_EXPORT](#) int32_t [casmb_write](#) ([CASMBFile](#) *file, const void *data, uint32_t offset, uint32_t data_size)
Zapisanie danych do pliku.

1.6.2 Opis szczegółowy

Otwieranie, tworzenie, odczyt oraz modyfikacja plików.

1.6.3 Dokumentacja funkcji

1.6.3.1 [CASMB_EXPORT CASMBResult casmb_open](#) ([CASMBClient](#) * client, [CASMBFile](#) ** file, const [utf16_t](#) * url, uint32_t desired_access, uint32_t create_mode, uint32_t share_mode, [CASMBAttributes](#) * attributes)

Otwarcie pliku lub katalogu zgodnie z przekazanymi flagami.

Funkcja umożliwia także utworzenie lub zastąpienie pliku. Tryb działania zależy od przekazanych argumentów.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>file</i>	Wskaźnik, do którego zostanie zapisany obiekt otwartego pliku
in	<i>url</i>	Pełna ścieżka do pliku (względem udziału)
in	<i>desired_access</i>	Suma bitowa flag CASMBDesiredAccess



in	<i>create_mode</i>	Jedna z wartości wyliczenia CASMBCreateDisposition
in	<i>share_mode</i>	Suma bitowa flag CASMBShareAccess
in	<i>attributes</i>	Wskaźnik, do którego zostaną zapisane atrybuty otwartego pliku lub NULL jeśli nie chcemy ich odczytywać; pamięć na strukturę musi być zaalokowana przez użytkownika

Zwraca:

- [CASMB_OK](#) - pomyślnie otwarto plik i zapisano obiekt do przekazanej struktury
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_DISK_FULL](#) - brak miejsca na dysku serwera
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony
- [CASMB_ERR_IS_DIRECTORY](#) - żądano otwarcia pliku, ale ścieżka wskazuje na katalog
- [CASMB_ERR_IS_FILE](#) - żądano otwarcia katalogu, ale ścieżka wskazuje na plik
- [CASMB_ERR_ALREADY_EXISTS](#) - plik o podanej nazwie już istnieje

Spostrzeżenia:

Biblioteka oczekuje, że podana ścieżka będzie pełną ścieżką względem udziału. Przykładowo, dla pliku "\\test-server\Test Share\Test Directory\File.txt ścieżka" powinna mieć postać "Test Directory\File.txt ścieżka"

1.6.3.2 [CASMB_EXPORT CASMBResult casmb_close \(CASMBFile * file\)](#)

Zamknięcie pliku na serwerze oraz zwolnienie obiektu pliku w pamięci.

Parametry:

in	<i>file</i>	Wskaźnik na obiekt pliku, stworzonego przez funkcję casmb_open
----	-------------	--

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

Spostrzeżenia:

Funkcja zwraca błędy w komunikacji z serwerem, ale zawsze poprawnie zwalnia obiekt pliku aby uniknąć wycieków pamięci.

1.6.3.3 [CASMB_EXPORT CASMBResult casmb_set_size \(CASMBFile * file, uint64_t size\)](#)

Ustawienie nowego rozmiaru pliku.

Parametry:

in	<i>file</i>	Wskaźnik na obiekt pliku
in	<i>size</i>	Nowy rozmiar pliku w bajtach

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_DISK_FULL](#) - brak miejsca na dysku serwera

1.6.3.4 [CASMB_EXPORT int64_t casmb_size \(CASMBFile * file\)](#)

Pobranie aktualnego rozmiaru pliku w bajtach.

Parametry:

in	<i>file</i>	Wskaźnik na obiekt pliku
----	-------------	--------------------------

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to wielkość pliku w bajtach. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:



- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

1.6.3.5 [CASMB_EXPORT](#) int64_t casmb_position ([CASMBFile](#) * file)

Pobranie aktualnego przesunięcia (pozycji) w pliku licząc od jego początku.

Parametry:

in	file	Wskaźnik na obiekt pliku
----	------	--------------------------

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to aktualna wartość przesunięcia w pliku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

1.6.3.6 [CASMB_EXPORT](#) int64_t casmb_seek ([CASMBFile](#) * file, uint64_t offset, uint32_t whence)

Zmiana aktualnego przesunięcia (pozycji) w pliku.

Parametry:

in	file	Wskaźnik na obiekt pliku
in	offset	Przesunięcie względem pozycji określonej przez argument whence
in	whence	Jedna z wartości wyliczenia CASMBSeekDirection

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to aktualna wartość przesunięcia w pliku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:



- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - podany argument 'whence' jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

1.6.3.7 [CASMB_EXPORT](#) int32_t casmb_read ([CASMBFile](#) * file, void * data, uint32_t offset, uint32_t data_size)

Odczytanie danych z pliku.

Parametry:

in	<i>file</i>	Wskaźnik na obiekt pliku
out	<i>data</i>	Wskaźnik na bufor, do którego zostaną zapisane odczytane dane
in	<i>offset</i>	Przesunięcie względem początku bufora, od którego zostaną zapisane odczytane dane
in	<i>data_size</i>	Maksymalna ilość bajtów, którą chcemy odczytać

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość bajtów odczytanych z pliku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na bufor ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - ilość danych do odczytu ma wartość 0
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu na wykonanie odczytu z pliku

1.6.3.8 [CASMB_EXPORT](#) int32_t casmb_write ([CASMBFile](#) * file, const void * data, uint32_t offset, uint32_t data_size)

Zapisanie danych do pliku.

Parametry:

in	<i>file</i>	Wskaźnik na obiekt pliku
in	<i>data</i>	Wskaźnik na bufor, z którego zostaną odczytane dane do zapisu
in	<i>offset</i>	Przesunięcie względem początku bufora, od którego zostaną odczytane dane
in	<i>data_size</i>	Maksymalna ilość bajtów, którą chcemy zapisać

Zwraca:

Jeśli wartość zwrócona ≥ 0 , to jest to ilość bajtów zapisanych do pliku. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na bufor ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - ilość danych do odczytu ma wartość 0
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_DISK_FULL](#) - brak miejsca na dysku serwera
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu na wykonanie zapisu do pliku

1.7 Listowanie zawartości katalogów

Wyszukiwanie oraz listowanie plików i katalogów w podanej ścieżce.

1.7.1 Funkcje

- [CASMB_EXPORT CASMBResult casmb_search](#) ([CASMBClient](#) *client, [CASMBSearch](#) **search, const [utf16_t](#) *pattern, uint32_t entry_type)
Otwarcie obiektu, służącego do listowania zawartości katalogu na podstawie przekazanego wzorca.
- [CASMB_EXPORT CASMBResult casmb_search_close](#) ([CASMBSearch](#) *search)
Zamknięcie przeszukiwania katalogów otwartego za pomocą funkcji [casmb_search](#).
- [CASMB_EXPORT CASMBResult casmb_search_next](#) ([CASMBSearch](#) *search, [CASMBDirectoryEntry](#) *entry)
Pobranie następnego elementu zwróconego przez wyszukiwanie (listowanie).

1.7.2 Opis szczegółowy

Wyszukiwanie oraz listowanie plików i katalogów w podanej ścieżce.



1.7.3 Dokumentacja funkcji

1.7.3.1 [CASMB_EXPORT CASMBResult casmb_search](#) ([CASMBClient](#) * *client*, [CASMBSearch](#) ** *search*, const [utf16_t](#) * *pattern*, [uint32_t](#) *entry_type*)

Otwarcie obiektu, służącego do listowania zawartości katalogu na podstawie przekazanego wzorca.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
out	<i>search</i>	Wskaźnik, do którego zostanie zapisany obiekt przeszukiwania katalogu
in	<i>pattern</i>	Wzorzec, według którego zostanie wylistowana zawartość. Jako wzorzec należy podać pełną ścieżkę do katalogu względem udziału, którego zawartość chcemy wylistować. Wzorzec obsługuje znaki specjalne '*' oraz '?'.
in	<i>entry_type</i>	Suma bitowa flag CASMBDirEntType lub 0, aby wylistować wszystkie typy

Zwraca:

- [CASMB_OK](#) - poprawnie otwarto obiekt przeszukiwania katalogu
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt przeszukiwania ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - podany wzorzec ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu na wykonanie zapisu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_IS_FILE](#) - żądano otwarcia katalogu, ale ścieżka wskazuje na plik

1.7.3.2 [CASMB_EXPORT CASMBResult casmb_search_close](#) ([CASMBSearch](#) * *search*)

Zamknięcie przeszukiwania katalogów otwartego za pomocą funkcji [casmb_search](#).

Parametry:

in	<i>search</i>	Wskaźnik na obiekt przeszukiwania katalogów
----	---------------	---

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

Spostrzeżenia:

Funkcja zwraca błędy w komunikacji z serwerem, ale zawsze poprawnie zwalnia obiekt przeszukiwania katalogów aby uniknąć wycieków pamięci.

1.7.3.3 [CASMB_EXPORT](#) [CASMBResult](#) [casmb_search_next](#) ([CASMBSearch](#) * [search](#), [CASMBDirectoryEntry](#) * [entry](#))

Pobranie następnego elementu zwróconego przez wyszukiwanie (listowanie).

Parametry:

in	search	Wskaźnik na obiekt przeszukiwania katalogów
out	entry	Wskaźnik na strukturę, do której zostaną zapisane dane o pliku/katalogu

Zwraca:

- [CASMB_OK](#) - operacja przebiegła pomyślnie
- [CASMB_NO_MORE_ENTRIES](#) - brak więcej wpisów w tym wyszukiwaniu (odczytano już wszystkie wpisy); należy zamknąć to przeszukiwanie
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazany wskaźnik na obiekt pliku ma wartość NULL
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób

1.8 Operacje na katalogach i atrybutach plików

Funkcje, które pozwalają na operacje na katalogach oraz odczyt atrybutów plików.

1.8.1 Funkcje

- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_exists](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, [int32_t](#) *is_directory)
- Sprawdzenie, czy dany plik w podanej ścieżce istnieje.*



- [CASMB_EXPORT_CASMBResult_casmb_attributes](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, [CASMBAttributes](#) *attributes)
Pobranie atrybutów pliku lub katalogu.
- [CASMB_EXPORT_CASMBResult_casmb_rename](#) ([CASMBClient](#) *client, const [utf16_t](#) *src, const [utf16_t](#) *dest)
Zmiana nazwy pliku wskazanego przez ścieżkę.
- [CASMB_EXPORT_CASMBResult_casmb_unlink](#) ([CASMBClient](#) *client, const [utf16_t](#) *url)
Usunięcie pliku z systemu plików.
- [CASMB_EXPORT_CASMBResult_casmb_rmdir](#) ([CASMBClient](#) *client, const [utf16_t](#) *url)
Usunięcie katalogu z systemu plików.
- [CASMB_EXPORT_CASMBResult_casmb_mkdir](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, [uint32_t](#) attributes)
Utworzenie nowego katalogu.

1.8.2 Opis szczegółowy

Funkcje, które pozwalają na operacje na katalogach oraz odczyt atrybutów plików.

1.8.3 Dokumentacja funkcji

1.8.3.1 [CASMB_EXPORT_CASMBResult_casmb_exists](#) ([CASMBClient](#) * client, const [utf16_t](#) * url, [int32_t](#) * is_directory)

Sprawdzenie, czy dany plik w podanej ścieżce istnieje.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>url</i>	Pełna ścieżka do pliku (względem udziału)
out	<i>is_directory</i>	Wskaźnik, do którego zostanie zapisana informacja czy pod podaną ścieżką znajduje się plik (0), czy też katalog (1). Jeśli podany wskaźnik ma wartość NULL, to informacja nie zostanie zapisana.

Zwraca:

- [CASMB_OK](#) - pod podaną ścieżką istnieje plik lub katalog
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób



- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony

1.8.3.2 [CASMB_EXPORT_CASMBResult](#) *casmb_attributes* ([CASMBClient](#) * *client*, const [utf16_t](#) * *url*, [CASMBAttributes](#) * *attributes*)

Pobranie atrybutów pliku lub katalogu.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>url</i>	Pełna ścieżka do pliku (względem udziału)
out	<i>attributes</i>	Struktura, do której zostaną zapisane atrybuty pliku

Zwraca:

- [CASMB_OK](#) - pomyślnie odczytano atrybuty pliku
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana wskaźnik na atrybuty pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony

1.8.3.3 [CASMB_EXPORT_CASMBResult](#) *casmb_rename* ([CASMBClient](#) * *client*, const [utf16_t](#) * *src*, const [utf16_t](#) * *dest*)

Zmiana nazwy pliku wskazanego przez ścieżkę.

Funkcja służy także do przenoszenia plików do innego katalogu.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>src</i>	Pełna ścieżka do pliku źródłowego (względem udziału)
in	<i>dest</i>	Pełna ścieżka do pliku docelowego (względem udziału)



Zwraca:

- [CASMB_OK](#) - pomyślnie zmieniono nazwę pliku
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony
- [CASMB_ERR_ALREADY_EXISTS](#) - plik o podanej nazwie już istnieje

1.8.3.4 [CASMB_EXPORT_CASMBResult casmb_unlink \(CASMBClient * client, const utf16_t * url\)](#)

Usunięcie pliku z systemu plików.

Za pomocą tej funkcji nie można kasować katalogów.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>url</i>	Pełna ścieżka do pliku (względem udziału)

Zwraca:

- [CASMB_OK](#) - pomyślnie usunięto plik
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do pliku jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony
- [CASMB_ERR_IS_DIRECTORY](#) - ścieżka wskazuje na katalog

1.8.3.5 [CASMB_EXPORT_CASMBResult](#) [casmb_rmdir](#) ([CASMBClient](#) * *client*, const [utf16_t](#) * *url*)

Usunięcie katalogu z systemu plików.

Funkcja ta nie usuwa plików wewnątrz katalogu. Usuwany katalog musi być pusty.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>url</i>	Pełna ścieżka do katalogu (względem udziału)

Zwraca:

- [CASMB_OK](#) - pomyślnie usunięto katalog
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do katalogu ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do katalogu jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_FILE_NOT_FOUND](#) - plik nie został znaleziony
- [CASMB_ERR_IS_FILE](#) - ścieżka wskazuje na plik

1.8.3.6 [CASMB_EXPORT_CASMBResult](#) [casmb_mkdir](#) ([CASMBClient](#) * *client*, const [utf16_t](#) * *url*, [uint32_t](#) *attributes*)

Utworzenie nowego katalogu.

Funkcja ta nie tworzy katalogów pośrednich - utworzenie ich jest zadaniem aplikacji.

Parametry:

in	<i>client</i>	Wskaźnik na obiekt klienta
in	<i>url</i>	Pełna ścieżka do katalogu (względem udziału)
in	<i>attributes</i>	Atrybuty tworzonego katalogu, suma bitowa wartości CASMBAttributes

Zwraca:

- [CASMB_OK](#) - pomyślnie utworzono katalog
- [CASMB_ERR_INVALID_CLIENT](#) - przekazany wskaźnik na obiekt klienta jest niepoprawny
- [CASMB_ERR_NOT_CONNECTED](#) - klient nie jest podłączony do serwera



- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do katalogu ma wartość NULL
- [CASMB_ERR_INVALID_ARGUMENT](#) - przekazana ścieżka do katalogu jest niepoprawna
- [CASMB_ERR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [CASMB_ERR_NETWORK_ERROR](#) - wystąpił błąd warstwy sieciowej
- [CASMB_ERR_NETWORK_TIMEOUT](#) - serwer nie odpowiedział w wymaganym czasie
- [CASMB_ERR_CONNECTION_CLOSED](#) - serwer zamknął połączenie
- [CASMB_ERR_INVALID_RESPONSE](#) - serwer odpowiedział w niepoprawny sposób
- [CASMB_ERR_ACCESS_DENIED](#) - brak dostępu do pliku
- [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) - brak jednego z katalogów pośrednich
- [CASMB_ERR_ALREADY_EXISTS](#) - plik o podanej nazwie już istnieje

2 Dokumentacja struktur danych

2.1 Dokumentacja struktury CASMBAttributes

Struktura przechowująca atrybuty pliku.

```
#include <casmb.h>
```

2.1.1 Pola danych

- uint64_t [creation_time](#)
Czas utworzenia pliku.
- uint64_t [last_access_time](#)
Czas ostatniego dostępu do pliku.
- uint64_t [last_write_time](#)
Czas ostatniego zapisu do pliku.
- uint64_t [change_time](#)
Czas ostatniej zmiany w pliku.
- uint64_t [file_size](#)
Rozmiar pliku w bajtach.
- uint32_t [attributes](#)
Atrybuty pliku zapisane jako suma bitowa wartości [CASMBFileAttribute](#).

2.1.2 Opis szczegółowy

Struktura przechowująca atrybuty pliku.

Definicja w linii [537](#) pliku [casmb.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [casmb.h](#)



2.2 Dokumentacja struktury CASMBDirectoryEntry

Struktura opisuje jeden wpis w katalogu.

```
#include <casmb.h>
```

2.2.1 Pola danych

- [CASMBAttributes attributes](#)
Atrybuty pliku.
- `uint32_t name length`
Długość nazwy pliku w znakach UTF-16.
- `const utf16_t * name`
Nazwa pliku lub katalogu.

2.2.2 Opis szczegółowy

Struktura opisuje jeden wpis w katalogu.

Wartość ta jest zwracana przez funkcję [casmb_search_next](#)

Definicja w linii [564](#) pliku [casmb.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [casmb.h](#)

3 Dokumentacja plików

3.1 Dokumentacja pliku casmb.h

Plik zawierający publiczne API dla biblioteki casmb.

```
#include <stddef.h>
```

```
#include <stdint.h>
```

3.1.1 Struktury danych

- `struct CASMBAttributes`
Struktura przechowująca atrybuty pliku.
- `struct CASMBDirectoryEntry`
Struktura opisuje jeden wpis w katalogu.

3.1.2 Definicje

- `#define CASMB_EXPORT`
*Makro eksportujące funkcje API w bibliotece *.dll/*.so.*

3.1.3 Definicje typów

- `typedef struct CASMBClient CASMBClient`
Struktura, przechowująca dane klienta biblioteki casmb.
- `typedef struct CASMBFile CASMBFile`
Struktura, przechowująca informacje o otwartym pliku (lub katalogu).



- typedef struct [CASMBSearch](#) [CASMBSearch](#)
Struktura, przechowująca informacje o aktualnie otwartym wyszukiwaniu (listowaniu) plików/katalogów znajdujących się w podanej ścieżce.
- typedef uint16_t [utf16_t](#)
Typ zmiennej przechowującej jeden znak w standardzie UTF16.
- typedef void (*[CASMBAllocateFunction](#)) (size_t size, void *user_data)
Callback, który jest wołany przez bibliotekę w celu alokacji pamięci.
- typedef void(*[CASMBFreeFunction](#)) (void *ptr, void *user_data)
Callback, który jest wołany przez bibliotekę w celu zwolnienia wcześniej zaalokowanej przez nią pamięci.

3.1.4 Wyliczenia

- enum [CASMBResult](#) { [CASMB_OK](#) = 0, [CASMB_NO_MORE_ENTRIES](#) = 1, [CASMB_ERR_ALREADY_EXISTS](#) = -1, [CASMB_ERR_IS_DIRECTORY](#) = -2, [CASMB_ERR_IS_FILE](#) = -3, [CASMB_ERR_NO_DIRECTORY_COMPONENT](#) = -6, [CASMB_ERR_FILE_NOT_FOUND](#) = -7, [CASMB_ERR_ACCESS_DENIED](#) = -8, [CASMB_ERR_DISK_FULL](#) = -9, [CASMB_ERR_OUT_OF_MEMORY](#) = -100, [CASMB_ERR_INVALID_ARGUMENT](#) = -101, [CASMB_ERR_INVALID_CLIENT](#) = -102, [CASMB_ERR_NOT_CONNECTED](#) = -103, [CASMB_ERR_ALREADY_CONNECTED](#) = -104, [CASMB_ERR_NO_SERVER](#) = -105, [CASMB_ERR_NO_SHARE](#) = -106, [CASMB_ERR_NETWORK_ERROR](#) = -107, [CASMB_ERR_NETWORK_TIMEOUT](#) = -108, [CASMB_ERR_INVALID_RESPONSE](#) = -109, [CASMB_ERR_INVALID_CREDENTIALS](#) = -110, [CASMB_ERR_CONNECTION_CLOSED](#) = -111, [CASMB_ERR_PLAINTEXT_LOGIN_NOT_SUPPORTED](#) = -112, [CASMB_ERR_ASCII_NOT_SUPPORTED](#) = -113, [CASMB_ERR_NOT_IMPLEMENTED](#) = -1000, [CASMB_ERR_INTERNAL_ERROR](#) = -1001 }Enumeracja możliwych wartości zwracanych z funkcji API.
- enum [CASMBDirEntType](#) { [CASMB_ENTRY_TYPE_DIRECTORY](#) = 1, [CASMB_ENTRY_TYPE_FILE](#) = 2 }Flagi bitowe, oznaczające typ danych do wyszukania w podanym katalogu.
- enum [CASMBDesiredAccess](#) { [CASMB_FILE_READ_DATA](#) = 0x00000001, [CASMB_FILE_WRITE_DATA](#) = 0x00000002, [CASMB_FILE_APPEND_DATA](#) = 0x00000004, [CASMB_FILE_READ_EA](#) = 0x00000008, [CASMB_FILE_WRITE_EA](#) = 0x00000010, [CASMB_FILE_EXECUTE](#) = 0x00000020, [CASMB_FILE_READ_ATTRIBUTES](#) = 0x00000080, [CASMB_FILE_WRITE_ATTRIBUTES](#) = 0x00000100, [CASMB_FILE_DELETE](#) = 0x00010000, [CASMB_FILE_MAXIMUM_ALLOWED](#) = 0x02000000, [CASMB_FILE_GENERIC_ALL](#) = 0x10000000, [CASMB_FILE_GENERIC_EXECUTE](#) = 0x20000000, [CASMB_FILE_GENERIC_WRITE](#) = 0x40000000, [CASMB_FILE_GENERIC_READ](#) = 0x80000000 }Flagi bitowe, określające dostęp do poszczególnych operacji na pliku.
- enum [CASMBFileAttribute](#) { [CASMB_ATTR_READONLY](#) = 0x00000001, [CASMB_ATTR_HIDDEN](#) = 0x00000002, [CASMB_ATTR_SYSTEM](#) = 0x00000004, [CASMB_ATTR_DIRECTORY](#) = 0x00000010, [CASMB_ATTR_ARCHIVE](#) = 0x00000020, [CASMB_ATTR_NORMAL](#) = 0x00000080, [CASMB_ATTR_TEMPORARY](#) = 0x00000100,



CASMB_ATTR_COMPRESSED = 0x00000800, CASMB_ATTR_POSIX_SEMANTICS = 0x01000000, CASMB_ATTR_BACKUP_SEMANTICS = 0x02000000, CASMB_ATTR_DELETE_ON_CLOSE = 0x04000000, CASMB_ATTR_SEQUENTIAL_SCAN = 0x08000000, CASMB_ATTR_RANDOM_ACCESS = 0x10000000, CASMB_ATTR_NO_BUFFERING = 0x20000000, CASMB_ATTR_WRITE_THROUGH = 0x80000000 }*Flagi bitowe, określające atrybuty plików.*

- enum CASMBShareAccess { CASMB_FILE_SHARE_NONE = 0x00000000, CASMB_FILE_SHARE_READ = 0x00000001, CASMB_FILE_SHARE_WRITE = 0x00000002, CASMB_FILE_SHARE_DELETE = 0x00000004 }*Flagi bitowe, określające sposób współdzielenia otwartego pliku.*
- enum CASMBCreateDisposition { CASMB_FILE_SUPERSEDE = 0x00000000, CASMB_FILE_OPEN = 0x00000001, CASMB_FILE_CREATE = 0x00000002, CASMB_FILE_OPEN_IF = 0x00000003, CASMB_FILE_OVERWRITE = 0x00000004, CASMB_FILE_OVERWRITE_IF = 0x00000005 }*Akcja, jaką należy podjąć w przypadku pliku lub katalogu który nie istnieje.*
- enum CASMBSeekDirection { CASMB_SEEK_BEGINNING = 0, CASMB_SEEK_SKIP = 1, CASMB_SEEK_END = 2 }*Sposób przesunięcia (zmiany pozycji kursora) w pliku.*

3.1.5 Funkcje

- CASMB_EXPORT void casmb_network_initialize ()
Inicjalizacja podsystemu sieci w systemie operacyjnym.
- CASMB_EXPORT void casmb_network_shutdown ()
Zamknięcie podsystemu sieci w systemie operacyjnym.
- CASMB_EXPORT const char * casmb_err_to_str (CASMBResult result)
Zamiana wyniku zwracanego przez funkcje biblioteki na napis.
- CASMB_EXPORT CASMBResult casmb_open_client (CASMBClient **client)
Utworzenie obiektu klienta, wykorzystując do tego celu domyślny alokator.
- CASMB_EXPORT CASMBResult casmb_open_client_with_allocator (CASMBClient **client, CASMBAllocateFunction allocate, CASMBFreeFunction free, void *user_data)
Utworzenie obiektu klienta, wykorzystując do tego celu alokator przekazany przez użytkownika.
- CASMB_EXPORT CASMBResult casmb_close_client (CASMBClient *client)
Zamknięcie klienta oraz zwolnienie zajmowanej przez niego pamięci.
- CASMB_EXPORT CASMBResult casmb_client_set_timeout (CASMBClient *client, uint32_t timeout)
Ustawienie czasu, jaki klient będzie oczekiwał na odpowiedź od serwera.
- CASMB_EXPORT CASMBResult casmb_set_login_credentials (CASMBClient *client, const utf16_t *domain, const utf16_t *username, const utf16_t *password)
Ustawienie danych autoryzacyjnych, które posłużą do połączenia z udziałem.
- CASMB_EXPORT CASMBResult casmb_connect (CASMBClient *client, const utf16_t *server, const utf16_t *share, uint32_t port)
Podłączenie się do udziału na serwerze.



- [CASMB_EXPORT CASMBResult casmb_open](#) ([CASMBClient](#) *client, [CASMBFile](#) **file, const [utf16_t](#) *url, uint32_t desired_access, uint32_t create_mode, uint32_t share_mode, [CASMBAttributes](#) *attributes)
Otwarcie pliku lub katalogu zgodnie z przekazanymi flagami.
- [CASMB_EXPORT CASMBResult casmb_close](#) ([CASMBFile](#) *file)
Zamknięcie pliku na serwerze oraz zwolnienie obiektu pliku w pamięci.
- [CASMB_EXPORT CASMBResult casmb_set_size](#) ([CASMBFile](#) *file, uint64_t size)
Ustawienie nowego rozmiaru pliku.
- [CASMB_EXPORT](#) int64_t [casmb_size](#) ([CASMBFile](#) *file)
Pobranie aktualnego rozmiaru pliku w bajtach.
- [CASMB_EXPORT](#) int64_t [casmb_position](#) ([CASMBFile](#) *file)
Pobranie aktualnego przesunięcia (pozycji) w pliku licząc od jego początku.
- [CASMB_EXPORT](#) int64_t [casmb_seek](#) ([CASMBFile](#) *file, uint64_t offset, uint32_t whence)
Zmiana aktualnego przesunięcia (pozycji) w pliku.
- [CASMB_EXPORT](#) int32_t [casmb_read](#) ([CASMBFile](#) *file, void *data, uint32_t offset, uint32_t data_size)
Odczytanie danych z pliku.
- [CASMB_EXPORT](#) int32_t [casmb_write](#) ([CASMBFile](#) *file, const void *data, uint32_t offset, uint32_t data_size)
Zapisanie danych do pliku.
- [CASMB_EXPORT CASMBResult casmb_search](#) ([CASMBClient](#) *client, [CASMBSearch](#) **search, const [utf16_t](#) *pattern, uint32_t entry_type)
Otwarcie obiektu, służącego do listowania zawartości katalogu na podstawie przekazanego wzorca.
- [CASMB_EXPORT CASMBResult casmb_search_close](#) ([CASMBSearch](#) *search)
Zamknięcie przeszukiwania katalogów otwartego za pomocą funkcji [casmb_search](#).
- [CASMB_EXPORT CASMBResult casmb_search_next](#) ([CASMBSearch](#) *search, [CASMBDirectoryEntry](#) *entry)
Pobranie następnego elementu zwróconego przez wyszukiwanie (listowanie).
- [CASMB_EXPORT CASMBResult casmb_exists](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, int32_t *is_directory)
Sprawdzenie, czy dany plik w podanej ścieżce istnieje.
- [CASMB_EXPORT CASMBResult casmb_attributes](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, [CASMBAttributes](#) *attributes)
Pobranie atrybutów pliku lub katalogu.
- [CASMB_EXPORT CASMBResult casmb_rename](#) ([CASMBClient](#) *client, const [utf16_t](#) *src, const [utf16_t](#) *dest)
Zmiana nazwy pliku wskazanego przez ścieżkę.
- [CASMB_EXPORT CASMBResult casmb_unlink](#) ([CASMBClient](#) *client, const [utf16_t](#) *url)
Usunięcie pliku z systemu plików.
- [CASMB_EXPORT CASMBResult casmb_rmdir](#) ([CASMBClient](#) *client, const [utf16_t](#) *url)
Usunięcie katalogu z systemu plików.



- [CASMB_EXPORT](#) [CASMBResult](#) [casmb_mkdir](#) ([CASMBClient](#) *client, const [utf16_t](#) *url, uint32_t attributes)
Utworzenie nowego katalogu.

3.1.6 Opis szczegółowy

Plik zawierający publiczne API dla biblioteki casmb.

Definicja w pliku [casmb.h](#).

3.2 casmb.h

```
1 #ifndef CASMB_H_
2 #define CASMB_H_
3
4 #include <stddef.h>
5 #include <stdint.h>
6
77 #if !defined(CASMB_EXPORT)
78     #if (defined(_WIN32) || defined(WIN32)) && defined(CASMB_BUILD)
79         #define CASMB_EXPORT __declspec(dllexport)
80     #elif defined(__GNUC__) && defined(CASMB_BUILD)
81         #define CASMB_EXPORT __attribute__((visibility("default")))
82     #else
83         #define CASMB_EXPORT
84     #endif
85 #endif /* CASMB_EXPORT */
86
87 #if defined(__cplusplus)
88 extern "C" {
89 #endif /* defined(__cplusplus) */
90
91 typedef enum {
92     171 CASMB\_OK = 0,
93     172 CASMB\_NO\_MORE\_ENTRIES = 1,
94     173
95     174 CASMB\_ERR\_ALREADY\_EXISTS = -1,
96     175 CASMB\_ERR\_IS\_DIRECTORY = -2,
97     176 CASMB\_ERR\_IS\_FILE = -3,
98     177
99     178 CASMB\_ERR\_NO\_DIRECTORY\_COMPONENT = -6,
100    179 CASMB\_ERR\_FILE\_NOT\_FOUND = -7,
101    180 CASMB\_ERR\_ACCESS\_DENIED = -8,
102    181 CASMB\_ERR\_DISK\_FULL = -9,
103    182
104    183 CASMB\_ERR\_OUT\_OF\_MEMORY = -100,
```



```
184 CASMB\_ERR\_INVALID\_ARGUMENT = -101,  
185 CASMB\_ERR\_INVALID\_CLIENT = -102,  
186 CASMB\_ERR\_NOT\_CONNECTED = -103,  
187 CASMB\_ERR\_ALREADY\_CONNECTED = -104,  
188 CASMB\_ERR\_NO\_SERVER = -105,  
189 CASMB\_ERR\_NO\_SHARE = -106,  
190 CASMB\_ERR\_NETWORK\_ERROR = -107,  
191 CASMB\_ERR\_NETWORK\_TIMEOUT = -108,  
192 CASMB\_ERR\_INVALID\_RESPONSE = -109,  
193 CASMB\_ERR\_INVALID\_CREDENTIALS = -110,  
194 CASMB\_ERR\_CONNECTION\_CLOSED = -111,  
195 CASMB\_ERR\_PLAINTEXT\_LOGIN\_NOT\_SUPPORTED = -112,  
196 CASMB\_ERR\_ASCII\_NOT\_SUPPORTED = -113,  
197  
198 CASMB\_ERR\_NOT\_IMPLEMENTED = -1000,  
199 CASMB\_ERR\_INTERNAL\_ERROR = -1001,  
200 } CASMBResult;  
201  
220 typedef enum {  
221 CASMB\_ENTRY\_TYPE\_DIRECTORY = 1,  
222 CASMB\_ENTRY\_TYPE\_FILE = 2  
223 } CASMBDirEntType;  
224  
273 typedef enum {  
274 CASMB\_FILE\_READ\_DATA = 0x00000001,  
275 CASMB\_FILE\_WRITE\_DATA = 0x00000002,  
276 CASMB\_FILE\_APPEND\_DATA = 0x00000004,  
277 CASMB\_FILE\_READ\_EA = 0x00000008,  
278 CASMB\_FILE\_WRITE\_EA = 0x00000010,  
279 CASMB\_FILE\_EXECUTE = 0x00000020,  
280 CASMB\_FILE\_READ\_ATTRIBUTES = 0x00000080,  
281 CASMB\_FILE\_WRITE\_ATTRIBUTES = 0x00000100,  
282 CASMB\_FILE\_DELETE = 0x00010000,  
283 CASMB\_FILE\_MAXIMUM\_ALLOWED = 0x02000000,  
284 CASMB\_FILE\_GENERIC\_ALL = 0x10000000,  
285 CASMB\_FILE\_GENERIC\_EXECUTE = 0x20000000,  
286 CASMB\_FILE\_GENERIC\_WRITE = 0x40000000,  
287 CASMB\_FILE\_GENERIC\_READ = 0x80000000  
288 } CASMBDesiredAccess;  
289  
350 typedef enum {  
351 CASMB\_ATTR\_READONLY = 0x00000001,
```



```
352 CASMB\_ATTR\_HIDDEN = 0x00000002,  
353 CASMB\_ATTR\_SYSTEM = 0x00000004,  
354 CASMB\_ATTR\_DIRECTORY = 0x00000010,  
355 CASMB\_ATTR\_ARCHIVE = 0x00000020,  
356 CASMB\_ATTR\_NORMAL = 0x00000080,  
357 CASMB\_ATTR\_TEMPORARY = 0x00000100,  
358 CASMB\_ATTR\_COMPRESSED = 0x00000800,  
359 CASMB\_ATTR\_POSIX\_SEMANTICS = 0x01000000,  
360 CASMB\_ATTR\_BACKUP\_SEMANTICS = 0x02000000,  
361 CASMB\_ATTR\_DELETE\_ON\_CLOSE = 0x04000000,  
362 CASMB\_ATTR\_SEQUENTIAL\_SCAN = 0x08000000,  
363 CASMB\_ATTR\_RANDOM\_ACCESS = 0x10000000,  
364 CASMB\_ATTR\_NO\_BUFFERING = 0x20000000,  
365 CASMB\_ATTR\_WRITE\_THROUGH = 0x80000000  
366 } CASMBFileAttribute;  
367  
402 typedef enum {  
403 CASMB\_FILE\_SHARE\_NONE = 0x00000000,  
404 CASMB\_FILE\_SHARE\_READ = 0x00000001,  
405 CASMB\_FILE\_SHARE\_WRITE = 0x00000002,  
406 CASMB_FILE_SHARE_DELETE = 0x00000004  
407 } CASMBShareAccess;  
408  
449 typedef enum {  
450 CASMB\_FILE\_SUPERSEDE = 0x00000000,  
451 CASMB\_FILE\_OPEN = 0x00000001,  
452 CASMB\_FILE\_CREATE = 0x00000002,  
453 CASMB\_FILE\_OPEN\_IF = 0x00000003,  
454 CASMB\_FILE\_OVERWRITE = 0x00000004,  
455 CASMB\_FILE\_OVERWRITE\_IF = 0x00000005  
456 } CASMBCreateDisposition;  
457  
471 typedef enum {  
472 CASMB\_SEEK\_BEGINNING = 0,  
473 CASMB\_SEEK\_SKIP = 1,  
474 CASMB\_SEEK\_END = 2  
475 } CASMBSeekDirection;  
476  
482 typedef struct CASMBClient CASMBClient;  
483  
489 typedef struct CASMBFile CASMBFile;  
490
```




```
497 typedef struct CASMBSearch CASMBSearch;  
498  
502 typedef uint16_t utf16\_t;  
503  
537 typedef struct {  
538     uint64_t creation\_time;  
539     uint64_t last\_access\_time;  
540     uint64_t last\_write\_time;  
541     uint64_t change\_time;  
542     uint64_t file\_size;  
543     uint32_t attributes;  
544 } CASMBAttributes;  
545  
564 typedef struct {  
565     CASMBAttributes attributes;  
566     uint32_t name\_length;  
567     const utf16_t* name;  
568 } CASMBDirectoryEntry;  
569  
581 typedef void* (*CASMBAllocateFunction)(size_t size, void* user_data);  
582  
593 typedef void (*CASMBFreeFunction)(void* ptr, void* user_data);  
594  
605 CASMB\_EXPORT void casmb\_network\_initialize();  
606  
613 CASMB\_EXPORT void casmb\_network\_shutdown();  
614  
623 CASMB\_EXPORT const char* casmb\_err\_to\_str(CASMBResult result);  
624  
635 CASMB\_EXPORT CASMBResult casmb\_open\_client(CASMBClient** client);  
636  
650 CASMB\_EXPORT CASMBResult casmb\_open\_client\_with\_allocator(CASMBClient** client,  
CASMBAllocateFunction allocate, CASMBFreeFunction free, void* user_data);  
651  
661 CASMB\_EXPORT CASMBResult casmb\_close\_client(CASMBClient* client);  
662  
676 CASMB\_EXPORT CASMBResult casmb\_client\_set\_timeout(CASMBClient* client, uint32_t  
timeout);  
677  
696 CASMB\_EXPORT CASMBResult casmb\_set\_login\_credentials(CASMBClient* client, const  
utf16_t* domain, const utf16_t* username, const utf16_t* password);  
697
```



```
726 CASMB\_EXPORT CASMBResult casmb\_connect(CASMBClient* client, const utf16_t*
server, const utf16_t* share, uint32_t port);
727
768 CASMB\_EXPORT CASMBResult casmb\_open(CASMBClient* client, CASMBFile** file, const
utf16_t* url, uint32_t desired_access, uint32_t create_mode, uint32_t share_mode,
CASMBAttributes* attributes);
769
787 CASMB\_EXPORT CASMBResult casmb\_close(CASMBFile* file);
788
806 CASMB\_EXPORT CASMBResult casmb\_set\_size(CASMBFile* file, uint64_t size);
807
824 CASMB\_EXPORT int64_t casmb\_size(CASMBFile* file);
825
842 CASMB\_EXPORT int64_t casmb\_position(CASMBFile* file);
843
863 CASMB\_EXPORT int64_t casmb\_seek(CASMBFile* file, uint64_t offset, uint32_t whence);
864
887 CASMB\_EXPORT int32_t casmb\_read(CASMBFile* file, void* data, uint32_t offset, uint32_t
data_size);
888
912 CASMB\_EXPORT int32_t casmb\_write(CASMBFile* file, const void* data, uint32_t offset,
uint32_t data_size);
913
939 CASMB\_EXPORT CASMBResult casmb\_search(CASMBClient* client, CASMBSearch**
search, const utf16_t* pattern, uint32_t entry_type);
940
958 CASMB\_EXPORT CASMBResult casmb\_search\_close(CASMBSearch* search);
959
977 CASMB\_EXPORT CASMBResult casmb\_search\_next(CASMBSearch* search,
CASMBDirectoryEntry* entry);
978
1003 CASMB\_EXPORT CASMBResult casmb\_exists(CASMBClient* client, const utf16_t* url,
int32_t* is_directory);
1004
1028 CASMB\_EXPORT CASMBResult casmb\_attributes(CASMBClient* client, const utf16_t* url,
CASMBAttributes* attributes);
1029
1055 CASMB\_EXPORT CASMBResult casmb\_rename(CASMBClient* client, const utf16_t* src,
const utf16_t* dest);
1056
1081 CASMB\_EXPORT CASMBResult casmb\_unlink(CASMBClient* client, const utf16_t* url);
1082
```




INNOWACYJNA
GOSPODARKA
NARODOWA STRATEGIA SPÓJNOŚCI



MINISTERSTWO
SPRAWIEDLIWOŚCI

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



```
1107 CASMB\_EXPORT\_CASMBResult\_casmb\_rmdir(CASMBClient* client, const utf16_t* url);  
1108  
1133 CASMB\_EXPORT\_CASMBResult\_casmb\_mkdir(CASMBClient* client, const utf16_t* url,  
uint32_t attributes);  
1134  
1135 #if defined(__cplusplus)  
1136 } /* extern "C" */  
1137 #endif /* defined(__cplusplus) */  
1138  
1139 #endif /* CASMB_H_ */
```

Dokumentacja API dostarczanego przez moduł RemoteView

Moduł umożliwia odtwarzanie na żywo podglądu obrazu i dźwięku wybranych źródeł udostępnianych przez serwer strumieniujący. Lista tych źródeł dostępna jest po podłączeniu do serwera. Dodatkowo, moduł udostępnia również na bieżąco aktualne poziomy dźwięku rejestrowanego przez wszystkie źródła. Aktualnie moduł RemoteView umożliwia podłączenie do modułu AVUnit oraz do wideoterminala Comarch (CVT).

Po pomyślnym pierwszym połączeniu do serwera strumieniującego moduł automatycznie wykrywa zerwanie połączenia. Moduł potrafi automatycznie wznowić połączenie oraz odtwarzanie obrazu i dźwięku w utworzonych wcześniej odtwarzaczach.

Aplikacja RECourt 2.0 wymaga 32 bitowego pliku DLL nazwanego RemoteView.dll, który udostępnia wszystkie funkcje opisane w niniejszym dokumencie. Aplikacja nie wymaga żadnego konkretnego protokołu do przesyłania danych; jest to wewnętrzny szczegół implementacyjny pomiędzy serwerem streamującym (AVUnit, CVT) a RemoteView.

1 Dokumentacja grup

1.1 Wartości globalne

Wartości globalne wykorzystywane w module.

1.1.1 Definicje

- #define [RE_API](#)
*Makro eksportujące funkcje API w bibliotece *.dll/*.so.*

1.1.2 Zmienne

- const uint32_t [REMOTEVIEW_API_VERSION](#) = 1
Zmienna określająca wersję API.

1.1.3 Opis szczegółowy

Wartości globalne wykorzystywane w module.

1.1.4 Dokumentacja zmiennych

1.1.4.1 [REMOTEVIEW_API_VERSION = 1](#)

Zmienna określająca wersję API.

Wersję tę należy porównać z wartością zwracaną przez funkcję [remoteview_api_version\(\)](#). Jeśli wartości te różnią się, nie należy używać tego modułu.

Zobacz również:

[remoteview_api_version](#)

Definicja w linii [73](#) pliku [RemoteView.h](#).

1.2 Typy

Typy wykorzystywane w module.

1.2.1 Definicje typów

- typedef void * [RemoteViewHandle](#)
Uchwyt do obiektu RemoteView.
- typedef void * [RemoteViewPlayerHandle](#)
Uchwyt do obiektu odtwarzacza, tworzonego przez obiekt RemoteView.
- typedef void(* [RemoteViewConnectedCallback](#)) (uint32_t connected)
Callback, który jest zostanie zawołany w momencie zmiany stanu połączenia z serwerem.

1.2.2 Wyliczenia

- enum [RemoteViewStatus](#) { [REMOTEVIEW_SUCCESS](#) = 0,
[REMOTEVIEW_ERROR_INITIALIZATION_FAILED](#) = -1,
[REMOTEVIEW_ERROR_NOT_INITIALIZED](#) = -2, [REMOTEVIEW_ERROR_INTERNAL](#) = -3,
[REMOTEVIEW_ERROR_OUT_OF_MEMORY](#) = -4,
[REMOTEVIEW_ERROR_INVALID_PARAMETER](#) = -5,
[REMOTEVIEW_ERROR_INVALID_HANDLE](#) = -6, [REMOTEVIEW_ERROR_NOT_CONNECTED](#)
= -7, [REMOTEVIEW_ERROR_OPERATION_FAILED](#) = -8,
[REMOTEVIEW_ERROR_INVALID_INDEX](#) = -9, [REMOTEVIEW_ERROR_INVALID_LICENSE](#) = -
1000 }*Enumeracja możliwych wartości zwracanych z funkcji API.*

1.2.3 Opis szczegółowy

Typy wykorzystywane w module.

1.2.4 Dokumentacja definicji typów

1.2.4.1 typedef void(* RemoteViewConnectedCallback) (uint32_t connected)

Callback, który jest zostanie zawołany w momencie zmiany stanu połączenia z serwerem.

Parametry:

<i>connected</i>	Wartość logiczna, która informuje o tym, czy właśnie nawiązano (1) czy też stracono (0) połączenie z serwerem.
------------------	--

Definicja w linii [155](#) pliku [RemoteView.h](#).

1.2.5 Dokumentacja typów wyliczanych

1.2.5.1 enum [RemoteViewStatus](#)

Enumeracja możliwych wartości zwracanych z funkcji API.

Wartości błędów zawsze są mniejsze od 0, dzięki czemu możliwe jest odróżnienie wystąpienia błędu od poprawnej wartości. W opisie każdej z funkcji jest dostępna lista błędów możliwych do wystąpienia.

Wartości wyliczeń

REMOTEVIEW_SUCCESS Funkcja zakończyła się poprawnie.

Wartość zwracana jest, jeśli funkcja nie zwraca innej wartości dodatniej.

REMOTEVIEW_ERROR_INITIALIZATION_FAILED Inicjalizacja biblioteki nie powiodła się

REMOTEVIEW_ERROR_NOT_INITIALIZED Próbowano użyć funkcji bez inicjalizacji biblioteki.

REMOTEVIEW_ERROR_INTERNAL Wewnętrzny błąd biblioteki.

REMOTEVIEW_ERROR_OUT_OF_MEMORY Nie można zaalokować wystarczającej ilości pamięci, aby wykonać operację

REMOTEVIEW_ERROR_INVALID_PARAMETER Przekazany parametr jest niepoprawny.

REMOTEVIEW_ERROR_INVALID_HANDLE Przekazany uchwyt do obiektu jest niepoprawny.

REMOTEVIEW_ERROR_NOT_CONNECTED Brak połączenia do serwera.

REMOTEVIEW_ERROR_OPERATION_FAILED Żądana operacja zakończyła się niepowodzeniem.

REMOTEVIEW_ERROR_INVALID_INDEX Przekazany indeks jest niepoprawny.

REMOTEVIEW_ERROR_INVALID_LICENSE Brak ważnej licencji na produkt.

Aplikacja nie może kontynuować.

Definicja w linii [123](#) pliku [RemoteView.h](#).

1.3 Funkcje inicjalizacyjne i połączeniowe

Funkcje te służą do inicjalizacji modułu oraz obiektu RemoteView.

1.3.1 Funkcje

- [RE API](#) `uint32_t remoteview_api_version ()`
Funkcja zwraca aktualną wersję API, używaną przez moduł.
- [RE API](#) `int32_t remoteview_initialize (const char *data_dir)`
Funkcja wywoływana aby zainicjalizować bibliotekę.
- [RE API](#) `int32_t remoteview_shutdown ()`
Zamknięcie modułu RemoteView.
- [RE API](#) `int32_t remoteview_create (RemoteViewHandle *handle, RemoteViewConnectedCallback connected_callback, const char *address, uint32_t port)`
Utworzenie nowego obiektu RemoteView, który od razu będzie próbował się łączyć do podanego adresu.
- [RE API](#) `int32_t remoteview_destroy (RemoteViewHandle handle)`
Zniszczenie obiektu RemoteView, utworzonego wcześniej za pomocą [remoteview_create](#).
- [RE API](#) `int32_t remoteview_wait_for_connection (RemoteViewHandle handle, float seconds)`
Oczekiwanie na nawiązanie połączenia z serwerem.
- [RE API](#) `int32_t remoteview_is_connected (RemoteViewHandle handle)`
Sprawdzenie, czy połączenie do serwera jest aktywne.
- [RE API](#) `int32_t remoteview_num_audio_sources (RemoteViewHandle handle)`
Pobranie ilości źródeł audio, przekazywanych do nas przez serwer.
- [RE API](#) `int32_t remoteview_num_video_sources (RemoteViewHandle handle)`



- Pobranie ilości źródeł video, przekazywanych do nas przez serwer.*
- [RE_API](#) int32_t [remoteview_num_audio_levels](#) ([RemoteViewHandle](#) handle)
Pobranie ilości źródeł audio, dla których posiadamy informację o aktualnym poziomie głośności.
 - [RE_API](#) int32_t [remoteview_audio_source_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu kanału audio, który znajduje się pod przekazanym indeksem.
 - [RE_API](#) int32_t [remoteview_audio_source_name](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **name)
Pobranie opisowa nazwy kanału audio, który znajduje się pod przekazanym indeksem.
 - [RE_API](#) int32_t [remoteview_video_source_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu źródła video, które znajduje się pod przekazanym indeksem.
 - [RE_API](#) int32_t [remoteview_video_source_name](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **name)
Pobranie opisowa nazwy źródła video, które znajduje się pod przekazanym indeksem.
 - [RE_API](#) int32_t [remoteview_audio_level_channel_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu kanału audio, dla którego posiadamy informacje o poziomie dźwięku pod podanym indeksem.
 - [RE_API](#) int32_t [remoteview_audio_levels](#) ([RemoteViewHandle](#) handle, float *levels, uint32_t max_level_count)
Pobranie aktualnych wartości poziomów dźwięku.

1.3.2 Opis szczegółowy

Funkcje te służą do inicjalizacji modułu oraz obiektu RemoteView.

Funkcje te służą również do połączenia do serwera, do kontroli tego połączenia oraz do odczytu danych, które są przez niego udostępnione.

1.3.3 Dokumentacja funkcji

1.3.3.1 [RE_API](#) uint32_t remoteview_api_version ()

Funkcja zwraca aktualną wersję API, używaną przez moduł.

Funkcja pozwala na wczesne wykrycie zmian w API i odmowę działania, jeśli aplikacja obsługuje API w wersji innej niż moduł.

Zwraca:

Aktualna wartość stałej [REMOTEVIEW_API_VERSION](#) (w chwili kompilacji)

Zobacz również:

[REMOTEVIEW_API_VERSION](#)

1.3.3.2 [RE_API](#) int32_t remoteview_initialize (const char * data_dir)

Funkcja wywoływana aby zainicjalizować bibliotekę.

Parametry:

in	data_dir	Pełna ścieżka do katalogu w którym zostaną zapisane dane aplikacji. Do danych aplikacji należą między innymi logi oraz ewentualne zrzuty pamięci. Przekazanie wartości NULL spowoduje wykorzystanie domyślnej ścieżki.
----	----------	--

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - inicjalizacja przebiegła pomyślnie
- [REMOTEVIEW_ERROR_INITIALIZATION_FAILED](#) - inicjalizacja nie powiodła się
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazana ścieżka jest niepoprawna
- [REMOTEVIEW_ERROR_INVALID_LICENSE](#) - brak ważnej licencji na produkt

Spostrzeżenia:

Funkcja musi być wywołana przed wywołaniem dowolnej innej funkcji. Po zawołaniu remoteview_module_shutdown() funkcja musi być wywołana ponownie. Funkcja może być wywołana więcej niż raz. Aby poprawnie zakończyć pracę należy wywołać funkcję remoteview_module_shutdown() dokładnie tyle samo razy.

Zobacz również:

remoteview_module_shutdown

1.3.3.3 [RE_API](#) int32_t remoteview_shutdown ()

Zamknięcie modułu RemoteView.

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - zamknięcie przebiegło pomyślnie
- [REMOTEVIEW_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Spostrzeżenia:

Funkcja musi być wywołana jeden raz dla każdego poprawnego wywołania funkcji remoteview_module_initialize(), Przed wywołaniem funkcji należy zniszczyć wszystkie utworzone obiekty.

Zobacz również:

remoteview_module_initialize

1.3.3.4 [RE_API](#) `int32_t remoteview_create (RemoteViewHandle * handle, RemoteViewConnectedCallback connected_callback, const char * address, uint32_t port)`

Utworzenie nowego obiektu RemoteView, który od razu będzie próbował się łączyć do podanego adresu.

Próba połączenia nie następuje w momencie utworzenia obiektu, ale jest dokonywana w tle. Moment połączenia lub rozłączenia będzie sygnalizowany za pomocą przekazanego callbacka.

Parametry:

out	<i>handle</i>	Wskaźnik, do którego zostanie zapisany utworzony obiekt RemoteView
in	<i>connected_callback</i>	Callback, który zostanie zawołany w momencie zmiany stanu połączenia
in	<i>address</i>	Adres IP komputera, na którym działa serwer udostępniający dane
in	<i>port</i>	Port, pod którym serwer udostępniający dane nasłuchuje połączeń (domyślnie 9666)

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie utworzono obiekt
- [REMOTEVIEW_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany callback ma wartość NULL
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany adres ma wartość NULL
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany adres jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany port jest niepoprawny
- [REMOTEVIEW_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [REMOTEVIEW_ERROR_OPERATION_FAILED](#) - próba otwarcia połączenia nie powiodła się

Spostrzeżenia:

Jeśli funkcja ta zwróci poprawny obiekt, należy go zwolnić za pomocą funkcji [remoteview_destroy](#).

Obiekt od razu będzie próbował nawiązać połączenie po utworzeniu. Przed zawołaniem innych funkcji należy upewnić się, że połączenie jest aktywne. Najlepiej dokonać tego oczekując na połączenie za pomocą funkcji [remoteview_wait_for_connection](#).

Obsługiwane są adresy IPv4 oraz adresy IPv6.

Zobacz również:

[remoteview_destroy](#), [remoteview_wait_for_connection](#)

[RemoteViewConnectedCallback](#)

1.3.3.5 [RE_API](#) int32_t remoteview_destroy ([RemoteViewHandle](#) handle)

Zniszczenie obiektu RemoteView, utworzonego wcześniej za pomocą remoteview_create.

Parametry:

in	handle	Uchwyt do obiektu RemoteView
----	--------	------------------------------

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie zniszczono obiekt
- [REMOTEVIEW_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

Spostrzeżenia:

Przed zawołaniem tej funkcji należy upewnić się, że wszystkie odtwarzacze stworzone za pomocą tego obiektu są zniszczone.

1.3.3.6 [RE_API](#) int32_t remoteview_wait_for_connection ([RemoteViewHandle](#) handle, float seconds)

Oczekiwanie na nawiązanie połączenia z serwerem.

Funkcja oczekuje maksymalnie podaną ilość sekund, informując, czy udało się nawiązać połączenie. Funkcja może zakończyć się przed tym czasem, jeśli pomyślnie nawiązano połączenie.

Parametry:

in	handle	Uchwyt do obiektu RemoteView
in	seconds	Maksymalna ilość sekund oczekiwania na połączenie

Zwraca:

Wartość 1 oznacza, że pomyślnie nawiązano połączenie.

Wartość 0 oznacza, że w podanym czasie nie udało się nawiązać połączenia.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3.3.7 [RE_API](#) int32_t remoteview_is_connected ([RemoteViewHandle](#) handle)

Sprawdzenie, czy połączenie do serwera jest aktywne.

Parametry:

in	handle	Uchwyt do obiektu RemoteView
----	--------	------------------------------

Zwraca:

Wartość 1 oznacza, że połączenie jest aktywne.

Wartość 0 oznacza, że aktualnie brak jest połączenia.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3.3.8 [RE_API int32_t remoteview_num_audio_sources \(RemoteViewHandle handle\)](#)

Pobranie ilości źródeł audio, przekazywanych do nas przez serwer.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
----	---------------	------------------------------

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość źródeł audio udostępnionych przez serwer. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

1.3.3.9 [RE_API int32_t remoteview_num_video_sources \(RemoteViewHandle handle\)](#)

Pobranie ilości źródeł video, przekazywanych do nas przez serwer.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
----	---------------	------------------------------

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość źródeł video udostępnionych przez serwer. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

1.3.3.10 [RE_API int32_t remoteview_num_audio_levels \(RemoteViewHandle handle\)](#)

Pobranie ilości źródeł audio, dla których posiadamy informację o aktualnym poziomie głośności.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
----	---------------	------------------------------

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość źródeł audio, dla których posiadamy informację o ich poziomach głośności. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

[REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

[REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Ilość otrzymywanych informacji o poziomach głośności kanałów nie musi być identyczna z ilością kanałów audio udostępnionych do odtwarzania.

1.3.3.11 [RE_API](#) `int32_t remoteview_audio_source_code` ([RemoteViewHandle](#) *handle*, `uint32_t index`, `const char ** code`)

Pobranie kodu kanału audio, który znajduje się pod przekazanym indeksem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>index</i>	Wewnętrzny indeks kanału audio w obiekcie RemoteView
out	<i>code</i>	Wskaźnik, do którego zostanie zapisany kod kanału audio

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie pobrano kod kanału audio
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału audio jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod kanału ma wartość NULL
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Kod kanału audio jest tożsamy z tym zdefiniowanym w serwerze, jednak jego indeks jest wewnętrznym indeksem używanym przez RemoteView.

Indeksy kanałów audio są numerowane od 0 do wartości zwróconej przez funkcję [remoteview_num_audio_sources\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu kanału audio jest zarządzana przez RemoteView i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [remoteview_destroy](#) lub do otrzymania informacji (callback) o utracie połączenia do serwera.

1.3.3.12 [RE_API](#) `int32_t remoteview_audio_source_name` ([RemoteViewHandle](#) *handle*, `uint32_t index`, `const char ** name`)

Pobranie opisowa nazwy kanału audio, który znajduje się pod przekazanym indeksem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>index</i>	Wewnętrzny indeks kanału audio w obiekcie RemoteView
out	<i>name</i>	Wskaźnik, do którego zostanie zapisana opisowa nazwa kanału audio

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie pobrano nazwę kanału audio
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany indeks kanału audio jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę kanału ma wartość NULL
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Indeksy kanałów audio są numerowane od 0 do wartości zwróconej przez funkcję [remoteview_num_audio_sources\(\)](#) - 1.

Pamięć, w której zapisana jest nazwa kanału audio jest zarządzana przez RemoteView i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [remoteview_destroy](#) lub do otrzymania informacji (callback) o utracie połączenia do serwera.

1.3.3.13 [RE_API](#) `int32_t remoteview_video_source_code (RemoteViewHandle handle, uint32_t index, const char ** code)`

Pobranie kodu źródła video, które znajduje się pod przekazanym indeksem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>index</i>	Wewnętrzny indeks źródła video w obiekcie RemoteView
out	<i>code</i>	Wskaźnik, do którego zostanie zapisany kod źródła video

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie pobrano kod źródła video
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany indeks źródła video jest niepoprawny



- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod źródła video ma wartość NULL
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Kod źródła video jest tożsamy z tym zdefiniowanym w serwerze, jednak jego indeks jest wewnętrznym indeksem używanym przez RemoteView.

Indeksy źródeł video są numerowane od 0 do wartości zwróconej przez funkcję [remoteview_num_video_sources\(\)](#) - 1.

Pamięć, w której zapisana jest kod źródła video jest zarządzana przez RemoteView i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zwołania [remoteview_destroy](#) lub do otrzymania informacji (callback) o utracie połączenia do serwera.

1.3.3.14 [RE_API](#) int32_t remoteview_video_source_name ([RemoteViewHandle](#) handle, uint32_t index, const char ** name)

Pobranie opisowa nazwy źródła video, które znajduje się pod przekazanym indeksem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>index</i>	Wewnętrzny indeks źródła video w obiekcie RemoteView
out	<i>name</i>	Wskaźnik, do którego zostanie zapisana opisowa nazwa źródła video

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie pobrano nazwę źródła video
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany indeks źródła video jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na nazwę źródła video ma wartość NULL
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Indeksy źródeł video są numerowane od 0 do wartości zwróconej przez funkcję [remoteview_num_video_sources\(\)](#) - 1.

Pamięć, w której zapisana jest nazwa źródła video jest zarządzana przez RemoteView i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zwołania [remoteview_destroy](#) lub do otrzymania informacji (callback) o utracie połączenia do serwera.

1.3.3.15 [RE_API](#) `int32_t remoteview_audio_level_channel_code` ([RemoteViewHandle](#) *handle*, `uint32_t index`, `const char ** code`)

Pobranie kodu kanału audio, dla którego posiadamy informacje o poziomie dźwięku pod podanym indeksem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>index</i>	Wewnętrzny indeks poziomu dźwięku kanału audio w obiekcie RemoteView
out	<i>code</i>	Wskaźnik, do którego zostanie zapisany kod kanału audio

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie pobrano kod kanału audio
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany indeks poziomu dźwięku kanału audio jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na kod kanału ma wartość NULL
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Kod kanału audio jest tożsamy z tym zdefiniowanym w serwerze, jednak jego indeks jest wewnętrznym indeksem używanym przez RemoteView.

Indeksy kanałów audio są numerowane od 0 do wartości zwróconej przez funkcję [remoteview_num_audio_levels\(\)](#) - 1.

Pamięć, w której zapisana jest wartość kodu kanału audio jest zarządzana przez RemoteView i nie należy jej zwalniać. Wskaźnik będzie ważny do momentu zawołania [remoteview_destroy](#) lub do otrzymania informacji (callback) o utracie połączenia do serwera.

1.3.3.16 [RE_API](#) `int32_t remoteview_audio_levels` ([RemoteViewHandle](#) *handle*, `float * levels`, `uint32_t max_level_count`)

Pobranie aktualnych wartości poziomów dźwięku.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
out	<i>levels</i>	Tablica, do której zostaną zapisane aktualne poziomy dźwięku dla wszystkich kanałów



in	<code>max_level_count</code>	Ilość elementów tablicy, do której zostaną zapisane poziomy dźwięku; rozmiar ten musi wynosić co najmniej tyle, co wartość zwrócona przez funkcję remoteview_num_audio_levels
----	------------------------------	---

Zwraca:

Jeśli wartość ≥ 0 , to jest to ilość zapisanych wartości poziomów audio. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazana tablica ma wartość NULL
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - przekazany rozmiar tablicy jest zbyt mały
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Poziom sygnałów jest zwracany jako wartość z przedziału [0.0, 1.0].

Poszczególne wartości poziomu sygnału audio zapisywane są w kolejności, w jakiej źródła audio odczytywane są przez funkcję [remoteview_audio_level_channel_code](#).

1.4 Funkcje odtwarzacza

Funkcje te służą zarządzania odtwarzaczami RemoteView.

1.4.1 Funkcje

- [RE_API](#) int32_t [remoteview_create_player](#) ([RemoteViewHandle](#) handle, [RemoteViewPlayerHandle](#) *player, const char *audio_source_code, const char *video_source_code, intptr_t window)
Utworzenie nowego odtwarzacza dla podanych źródeł audio i video.
- [RE_API](#) int32_t [remoteview_destroy_player](#) ([RemoteViewHandle](#) handle, [RemoteViewPlayerHandle](#) player)
Zniszczenie odtwarzacza utworzonego przez funkcję [remoteview_create_player](#).
- [RE_API](#) int32_t [remoteview_player_resize_video](#) ([RemoteViewPlayerHandle](#) handle, uint32_t width, uint32_t height)
Zmiana rozmiaru wyjściowego okna, w którym wyświetlany jest obraz.
- [RE_API](#) int32_t [remoteview_player_clear_video](#) ([RemoteViewPlayerHandle](#) handle)
Wyczyszczenie obrazu w podglądzie video w odtwarzaczu.
- [RE_API](#) int32_t [remoteview_player_refresh_video](#) ([RemoteViewPlayerHandle](#) handle)
Odświeżenie obrazu w podglądzie video w odtwarzaczu.

1.4.2 Opis szczegółowy

Funkcje te służą zarządzania odtwarzaczami RemoteView.

Odtwarzacze utworzone za pomocą tych funkcji wyświetlają obraz oraz odtwarzają dźwięk zgodnie z wybranymi źródłami.



1.4.3 Dokumentacja funkcji

1.4.3.1 [RE_API](#) `int32_t remoteview_create_player (RemoteViewHandle handle, RemoteViewPlayerHandle * player, const char * audio_source_code, const char * video_source_code, intptr_t window)`

Utworzenie nowego odtwarzacza dla podanych źródeł audio i video.

Dla danego źródła audio oraz video można utworzyć tylko jeden odtwarzacz, tzn. nie można używać dwóch odtwarzaczy z tym samym kodem audio lub video.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
out	<i>player</i>	Wskaźnik, do którego zostanie zapisany uchwyt na obiekt odtwarzacza
in	<i>audio_source_code</i>	Kod kanału audio, który będzie odtwarzany przez odtwarzacz (lub NULL, jeśli chcemy odtwarzać tylko obraz)
in	<i>video_source_code</i>	Kod źródła obrazu, które będzie odtwarzany przez odtwarzacz (lub NULL, jeśli chcemy odtwarzać tylko dźwięk)
in	<i>window</i>	Uchwyt do okna, w którym zostanie utworzony odtwarzacz obrazu

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie utworzono odtwarzacz
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - zarówno kod źródeł audio jak i video mają wartość NULL
- [REMOTEVIEW_ERROR_INVALID_PARAMETER](#) - podano kod źródła obrazu ale uchwyt do okna ma wartość 0
- [REMOTEVIEW_ERROR_OUT_OF_MEMORY](#) - brak wolnej pamięci aby wykonać tę operację
- [REMOTEVIEW_ERROR_OPERATION_FAILED](#) - utworzenie odtwarzacza nie powiodło się
- [REMOTEVIEW_ERROR_NOT_CONNECTED](#) - brak połączenia do serwera

Spostrzeżenia:

Po pomyślnym utworzeniu odtwarzacza należy go zniszczyć za pomocą funkcji [remoteview_create_player](#). Zniszczenie odtwarzacza wymaga obiektu RemoteView, dlatego należy to zrobić przed zawołaniem funkcji [remoteview_destroy](#).

Uchwyt do okna jest zależny od platformy, na której uruchomiony jest RemoteView. W przypadku systemu Windows jest to HWND, a w przypadku systemu Linux Window (X11). W przypadku przekazania wartości NULL jako źródła obrazu można podać wartość 0.

Pomyślnie utworzony odtwarzacz będzie automatycznie przerywał i wznowiał odtwarzanie po rozłączeniu/połączeniu do serwera.

Zobacz również:

[remoteview create player](#)

1.4.3.2 [RE_API](#) `int32_t remoteview_destroy_player` ([RemoteViewHandle](#) *handle*,
[RemoteViewPlayerHandle](#) *player*)

Zniszczenie odtwarzacza utworzonego przez funkcję [remoteview create player](#).

Parametry:

in	<i>handle</i>	Uchwyt do obiektu RemoteView
in	<i>player</i>	Uchwyt do obiektu odtwarzacza

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie zniszczono odtwarzacz
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu RemoteView jest niepoprawny
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt do obiektu odtwarzacza jest niepoprawny

1.4.3.3 [RE_API](#) `int32_t remoteview_player_resize_video` ([RemoteViewPlayerHandle](#) *handle*,
`uint32_t width`, `uint32_t height`)

Zmiana rozmiaru wyjściowego okna, w którym wyświetlany jest obraz.

Funkcja ta musi zostać zawołana, jeśli zmienił się rozmiar okna, w którym jest osadzony odtwarzacz.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu odtwarzacza
in	<i>width</i>	Nowa szerokość okna
in	<i>height</i>	Nowa wysokość okna

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie przekazano nowy rozmiar okna
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [REMOTEVIEW_ERROR_OPERATION_FAILED](#) - przekazanie nowego rozmiaru okna nie powiodło się

1.4.3.4 [RE_API](#) `int32_t remoteview_player_clear_video` ([RemoteViewPlayerHandle](#) *handle*)

Wyczyszczenie obrazu w podglądzie video w odtwarzaczu.



Po zawołaniu tej funkcji wyświetlany obraz zostaje wyczyszczony, wyświetlając czarny ekran. Po otrzymaniu nowej klatki obrazu z serwera zostanie ona wyświetlona normalnie.

Parametry:

in	handle	Uchwyt do obiektu odtwarzacza
----	--------	-------------------------------

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie wyczyszczono podgląd
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.4.3.5 [RE_API](#) int32_t remoteview_player_refresh_video ([RemoteViewPlayerHandle](#) handle)

Odświeżenie obrazu w podglądzie video w odtwarzaczu.

Po zawołaniu tej funkcji wyświetlany obraz zostaje odświeżony. Funkcja ma zastosowanie, jeśli wyświetlany obraz musi zostać wyświetlony ponownie (np. po zasłonięciu części okna przez inne okno).

Parametry:

in	handle	Uchwyt do obiektu odtwarzacza
----	--------	-------------------------------

Zwraca:

- [REMOTEVIEW_SUCCESS](#) - pomyślnie odświeżono
- [REMOTEVIEW_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

2 Dokumentacja plików

2.1 Dokumentacja pliku RemoteView.h

Plik zawierający publiczne API dla modułu RemoteView.

```
#include <stdint.h>
```

2.1.1 Definicje

- #define [RE_API](#)
Makro eksportujące funkcje API w bibliotece *.dll/*.so.

2.1.2 Definicje typów

- typedef void * [RemoteViewHandle](#)
Uchwyt do obiektu RemoteView.
- typedef void * [RemoteViewPlayerHandle](#)
Uchwyt do obiektu odtwarzacza, tworzonego przez obiekt RemoteView.
- typedef void(* [RemoteViewConnectedCallback](#)) (uint32_t connected)
Callback, który jest zostanie zawołany w momencie zmiany stanu połączenia z serwerem.

2.1.3 Wyliczenia

- enum [RemoteViewStatus](#) { [REMOTEVIEW_SUCCESS](#) = 0,
[REMOTEVIEW_ERROR_INITIALIZATION_FAILED](#) = -1,



[REMOTEVIEW_ERROR_NOT_INITIALIZED](#) = -2, [REMOTEVIEW_ERROR_INTERNAL](#) = -3,
[REMOTEVIEW_ERROR_OUT_OF_MEMORY](#) = -4,
[REMOTEVIEW_ERROR_INVALID_PARAMETER](#) = -5,
[REMOTEVIEW_ERROR_INVALID_HANDLE](#) = -6, [REMOTEVIEW_ERROR_NOT_CONNECTED](#)
= -7, [REMOTEVIEW_ERROR_OPERATION_FAILED](#) = -8,
[REMOTEVIEW_ERROR_INVALID_INDEX](#) = -9, [REMOTEVIEW_ERROR_INVALID_LICENSE](#) = -
1000 }Enumeracja możliwych wartości zwracanych z funkcji API.

2.1.4 Funkcje

- [RE_API](#) uint32_t [remoteview_api_version](#) ()
Funkcja zwraca aktualną wersję API, używaną przez moduł.
- [RE_API](#) int32_t [remoteview_initialize](#) (const char *data_dir)
Funkcja wywoływana aby zainicjalizować bibliotekę.
- [RE_API](#) int32_t [remoteview_shutdown](#) ()
Zamknięcie modułu RemoteView.
- [RE_API](#) int32_t [remoteview_create](#) ([RemoteViewHandle](#) *handle, [RemoteViewConnectedCallback](#) connected_callback, const char *address, uint32_t port)
Utworzenie nowego obiektu RemoteView, który od razu będzie próbował się łączyć do podanego adresu.
- [RE_API](#) int32_t [remoteview_destroy](#) ([RemoteViewHandle](#) handle)
Zniszczenie obiektu RemoteView, utworzonego wcześniej za pomocą [remoteview_create](#).
- [RE_API](#) int32_t [remoteview_wait_for_connection](#) ([RemoteViewHandle](#) handle, float seconds)
Oczekiwanie na nawiązanie połączenia z serwerem.
- [RE_API](#) int32_t [remoteview_is_connected](#) ([RemoteViewHandle](#) handle)
Sprawdzenie, czy połączenie do serwera jest aktywne.
- [RE_API](#) int32_t [remoteview_num_audio_sources](#) ([RemoteViewHandle](#) handle)
Pobranie ilości źródeł audio, przekazywanych do nas przez serwer.
- [RE_API](#) int32_t [remoteview_num_video_sources](#) ([RemoteViewHandle](#) handle)
Pobranie ilości źródeł video, przekazywanych do nas przez serwer.
- [RE_API](#) int32_t [remoteview_num_audio_levels](#) ([RemoteViewHandle](#) handle)
Pobranie ilości źródeł audio, dla których posiadamy informację o aktualnym poziomie głośności.
- [RE_API](#) int32_t [remoteview_audio_source_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu kanału audio, który znajduje się pod przekazanym indeksem.
- [RE_API](#) int32_t [remoteview_audio_source_name](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **name)
Pobranie opisowej nazwy kanału audio, który znajduje się pod przekazanym indeksem.
- [RE_API](#) int32_t [remoteview_video_source_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu źródła video, które znajduje się pod przekazanym indeksem.



- [RE_API](#) int32_t [remoteview_video_source_name](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **name)
Pobranie opisowa nazwy źródła video, które znajduje się pod przekazanym indeksem.
- [RE_API](#) int32_t [remoteview_audio_level_channel_code](#) ([RemoteViewHandle](#) handle, uint32_t index, const char **code)
Pobranie kodu kanału audio, dla którego posiadamy informacje o poziomie dźwięku pod podanym indeksem.
- [RE_API](#) int32_t [remoteview_audio_levels](#) ([RemoteViewHandle](#) handle, float *levels, uint32_t max_level_count)
Pobranie aktualnych wartości poziomów dźwięku.
- [RE_API](#) int32_t [remoteview_create_player](#) ([RemoteViewHandle](#) handle, [RemoteViewPlayerHandle](#) *player, const char *audio_source_code, const char *video_source_code, intptr_t window)
Utworzenie nowego odtwarzacza dla podanych źródeł audio i video.
- [RE_API](#) int32_t [remoteview_destroy_player](#) ([RemoteViewHandle](#) handle, [RemoteViewPlayerHandle](#) player)
Zniszczenie odtwarzacza utworzonego przez funkcję [remoteview_create_player](#).
- [RE_API](#) int32_t [remoteview_player_resize_video](#) ([RemoteViewPlayerHandle](#) handle, uint32_t width, uint32_t height)
Zmiana rozmiaru wyjściowego okna, w którym wyświetlany jest obraz.
- [RE_API](#) int32_t [remoteview_player_clear_video](#) ([RemoteViewPlayerHandle](#) handle)
Wyczyszczenie obrazu w podglądzie video w odtwarzaczu.
- [RE_API](#) int32_t [remoteview_player_refresh_video](#) ([RemoteViewPlayerHandle](#) handle)
Odświeżenie obrazu w podglądzie video w odtwarzaczu.

2.1.5 Zmienne

- const uint32_t [REMOTEVIEW_API_VERSION](#) = 1
Zmienna określająca wersję API.

2.1.6 Opis szczegółowy

Plik zawierający publiczne API dla modułu RemoteView.

Definicja w pliku [RemoteView.h](#).

2.2 RemoteView.h

```

1 #pragma once
2
3 #include <stdint.h>
4
42 #if defined(_WIN32) || defined(WIN32)
43     #if defined(REMOTEVIEW_BUILD)
44         #define RE_API __declspec(dllexport)
45     #else /* REMOTEVIEW_BUILD */
46         #define RE_API __declspec(dllimport)

```



```
47 #endif /* REMOTEVIEW_BUILD */
48 #elif defined(__linux__)
49 #if defined(REMOTEVIEW_BUILD)
50 #define RE_API __attribute__((visibility("default")))
51 #else /* REMOTEVIEW_BUILD */
52 #define RE_API
53 #endif /* REMOTEVIEW_BUILD */
54 #else
55 #define RE_API
56 #error "Unsupported platform!"
57 #endif
58
59 #if defined(__cplusplus)
60 extern "C" {
61 #endif /* __cplusplus */
62
63
64
65
66
67
68
69
70
71
72
73 const uint32_t REMOTEVIEW\_API\_VERSION = 1;
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103 enum RemoteViewStatus {
104 REMOTEVIEW\_SUCCESS = 0,
105
106 REMOTEVIEW\_ERROR\_INITIALIZATION\_FAILED = -1,
107 REMOTEVIEW\_ERROR\_NOT\_INITIALIZED = -2,
108
109 REMOTEVIEW\_ERROR\_INTERNAL = -3,
110 REMOTEVIEW\_ERROR\_OUT\_OF\_MEMORY = -4,
111 REMOTEVIEW\_ERROR\_INVALID\_PARAMETER = -5,
112 REMOTEVIEW\_ERROR\_INVALID\_HANDLE = -6,
113 REMOTEVIEW\_ERROR\_NOT\_CONNECTED = -7,
114 REMOTEVIEW\_ERROR\_OPERATION\_FAILED = -8,
115 REMOTEVIEW\_ERROR\_INVALID\_INDEX = -9,
116
117 REMOTEVIEW\_ERROR\_INVALID\_LICENSE = -1000
118 };
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143 typedef void* RemoteViewHandle;
144
145
146
147
148 typedef void* RemoteViewPlayerHandle;
149
150
151
152
153
154
155 typedef void (*RemoteViewConnectedCallback)(uint32_t connected);
156
157
158
159
160
161 /* -----
```



```
162 * | Main RemoteView API
163 * -----
164 */
165
176 RE API uint32_t remoteview\_api\_version();
177
197 RE API int32_t remoteview\_initialize(const char* data_dir);
198
211 RE API int32_t remoteview\_shutdown();
212
245 RE API int32_t remoteview\_create(RemoteViewHandle* handle,
RemoteViewConnectedCallback connected_callback, const char* address, uint32_t port);
246
260 RE API int32_t remoteview\_destroy(RemoteViewHandle handle);
261
278 RE API int32_t remoteview\_wait\_for\_connection(RemoteViewHandle handle, float
seconds);
279
291 RE API int32_t remoteview\_is\_connected(RemoteViewHandle handle);
292
304 RE API int32_t remoteview\_num\_audio\_sources(RemoteViewHandle handle);
305
317 RE API int32_t remoteview\_num\_video\_sources(RemoteViewHandle handle);
318
335 RE API int32_t remoteview\_num\_audio\_levels(RemoteViewHandle handle);
336
359 RE API int32_t remoteview\_audio\_source\_code(RemoteViewHandle handle, uint32_t
index, const char** code);
360
381 RE API int32_t remoteview\_audio\_source\_name(RemoteViewHandle handle, uint32_t
index, const char** name);
382
405 RE API int32_t remoteview\_video\_source\_code(RemoteViewHandle handle, uint32_t
index, const char** code);
406
427 RE API int32_t remoteview\_video\_source\_name(RemoteViewHandle handle, uint32_t
index, const char** name);
428
452 RE API int32_t remoteview\_audio\_level\_channel\_code(RemoteViewHandle handle,
uint32_t index, const char** code);
453
```



```
475 RE API int32_t remoteview\_audio\_levels(RemoteViewHandle handle, float* levels,  
uint32_t max_level_count);  
476  
477  
478 /* -----  
479 * | Player API  
480 * -----  
481 */  
482  
516 RE API int32_t remoteview\_create\_player(RemoteViewHandle handle,  
RemoteViewPlayerHandle* player, const char* audio_source_code, const char*  
video_source_code, intptr_t window);  
517  
529 RE API int32_t remoteview\_destroy\_player(RemoteViewHandle handle,  
RemoteViewPlayerHandle player);  
530  
545 RE API int32_t remoteview\_player\_resize\_video(RemoteViewPlayerHandle handle,  
uint32_t width, uint32_t height);  
546  
559 RE API int32_t remoteview\_player\_clear\_video(RemoteViewPlayerHandle handle);  
560  
574 RE API int32_t remoteview\_player\_refresh\_video(RemoteViewPlayerHandle handle);  
575  
576 #if defined(__cplusplus)  
577 } /* extern "C" */  
578 #endif /* __cplusplus */  
579
```

Dokumentacja API dostarczanego przez moduł SpeechSynthesis

1 Dokumentacja grup

1.1 Typy

Typy wykorzystywane przez bibliotekę.

1.1.1 Definicje typów

- typedef struct SpeechSynthesisInternal * [SpeechSynthesisHandle](#)
Uchwyt do obiektu syntezy mowy.

1.1.2 Wyliczenia

- enum [SpeechSynthesisStatus](#) { [SPEECHSYNTHESIS_SUCCESS](#) = 0,
[SPEECHSYNTHESIS_ERROR_INITIALIZATION_FAILED](#) = -1,
[SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED](#) = -2, [SPEECHSYNTHESIS_ERROR_INTERNAL](#)
= -3, [SPEECHSYNTHESIS_ERROR_OUT_OF_MEMORY](#) = -4,
[SPEECHSYNTHESIS_ERROR_INVALID_PARAMETER](#) = -5,
[SPEECHSYNTHESIS_ERROR_FAILED_TO_LOAD_VOICE](#) = -10,
[SPEECHSYNTHESIS_ERROR_INVALID_GONG_FILE](#) = -11,
[SPEECHSYNTHESIS_ERROR_NO_GONG_FILE](#) = -12,
[SPEECHSYNTHESIS_ERROR_AUDIO_DEVICE_FORMAT_NOT_SUPPORTED](#) = -20,
[SPEECHSYNTHESIS_ERROR_NO_VALID_AUDIO_DEVICE](#) = -21,
[SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) = -100,
[SPEECHSYNTHESIS_ERROR_INVALID_LICENSE](#) = -1000 } *Enumeracja możliwych wartości zwracanych z funkcji API.*

1.1.3 Opis szczegółowy

Typy wykorzystywane przez bibliotekę.

1.1.4 Dokumentacja typów wyliczanych

1.1.4.1 enum [SpeechSynthesisStatus](#)

Enumeracja możliwych wartości zwracanych z funkcji API.

Wartości błędów zawsze są mniejsze od 0, dzięki czemu możliwe jest odróżnienie wystąpienia błędu od poprawnej wartości. W opisie każdej z funkcji jest dostępna lista błędów możliwych do wystąpienia.

Wartości wyliczeń

- ***SPEECHSYNTHESIS_SUCCESS*** Funkcja zakończyła się poprawnie.
- ***SPEECHSYNTHESIS_ERROR_INITIALIZATION_FAILED*** Inicjalizacja biblioteki nie powiodła się
- ***SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED*** Próbowano użyć funkcji bez inicjalizacji biblioteki.
- ***SPEECHSYNTHESIS_ERROR_INTERNAL*** Wewnętrzny błąd biblioteki.



- **SPEECHSYNTHESIS_ERROR_OUT_OF_MEMORY** Nie można zaalokować wystarczającej ilości pamięci, aby wykonać operację
- **SPEECHSYNTHESIS_ERROR_INVALID_PARAMETER** Przekazany parametr jest niepoprawny.
- **SPEECHSYNTHESIS_ERROR_FAILED_TO_LOAD_VOICE** Nie udało się załadować pliku z głosem.
- **SPEECHSYNTHESIS_ERROR_INVALID_GONG_FILE** Plik gongu jest niepoprawny.
- **SPEECHSYNTHESIS_ERROR_NO_GONG_FILE** Nie udało się odnaleźć pliku gongu.
- **SPEECHSYNTHESIS_ERROR_AUDIO_DEVICE_FORMAT_NOT_SUPPORTED** Karta dźwiękowa nie obsługuje żadnego z wymaganych formatów dźwięku.
- **SPEECHSYNTHESIS_ERROR_NO_VALID_AUDIO_DEVICE** Nie znaleziono karty dźwiękowej.
- **SPEECHSYNTHESIS_ERROR_INVALID_HANDLE** Przekazany uchwyt do obiektu syntezy mowy jest niepoprawny.
- **SPEECHSYNTHESIS_ERROR_INVALID_LICENSE** Brak ważnej licencji na produkt.
- Aplikacja nie może kontynuować.

Definicja w linii [117](#) pliku [SpeechSynthesis.h](#).

1.2 Funkcje inicjalizacyjne

funkcje, które pozwalają stworzyć oraz zniszczyć obiekt syntezy mowy.

1.2.1 Funkcje

- [SPEECHSYNTHESIS_EXPORT](#) uint32_t [speechsynthesis_api_version](#) ()
Zwrócenie aktualnej wersji API, używanej przez moduł.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_initialize](#) ()
Zainicjalizowanie modułu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_shutdown](#) ()
Zamknięcie modułu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_create](#) ([SpeechSynthesisHandle](#) *handle)
Utworzenie i otwarcie obiektu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_free](#) ([SpeechSynthesisHandle](#) handle)
Zamknięcie i zniszczenie obiektu syntezy mowy.

1.2.2 Opis szczegółowy

Funkcje, które pozwalają stworzyć oraz zniszczyć obiekt syntezy mowy.

1.2.3 Dokumentacja funkcji

1.2.3.1 [SPEECHSYNTHESIS_EXPORT](#) uint32_t [speechsynthesis_api_version](#) ()

Zwrócenie aktualnej wersji API, używanej przez moduł.

Funkcja pozwala na wczesne wykrycie zmian w API i odmowę działania, jeśli aplikacja obsługuje API w wersji innej niż moduł.

Zwraca:

- Aktualna wartość stałej [SPEECHSYNTHESIS_API_VERSION](#) (w chwili kompilacji)

Zobacz również:

[SPEECHSYNTHESIS_API_VERSION](#)

1.2.3.2 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_initialize](#) ()

Zainicjalizowanie modułu syntezy mowy.

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - inicjalizacja przebiegła pomyślnie
- [SPEECHSYNTHESIS_ERROR_INITIALIZATION_FAILED](#) - inicjalizacja nie powiodła się
- [SPEECHSYNTHESIS_ERROR_INVALID_LICENSE](#) - brak ważnej licencji na produkt

Spostrzeżenia:

Funkcja musi być wywołana przed wywołaniem dowolnej innej funkcji. Po zawołaniu [speechsynthesis_shutdown\(\)](#) funkcja musi być wywołana ponownie. Funkcja może być wywołana więcej niż raz. Aby poprawnie zakończyć pracę należy wywołać funkcję [speechsynthesis_shutdown\(\)](#) dokładnie tyle samo razy.

Zobacz również:

[speechsynthesis_shutdown](#)

1.2.3.3 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_shutdown](#) ()

Zamknięcie modułu syntezy mowy.

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - zamknięcie przebiegło pomyślnie
- [SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany

Spostrzeżenia:

Funkcja musi być wywołana jeden raz dla każdego poprawnego wywołania funkcji [speechsynthesis_initialize\(\)](#), Przed wywołaniem funkcji należy zniszczyć wszystkie utworzone obiekty.

Zobacz również:

[speechsynthesis_initialize](#)

1.2.3.4 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_create](#) ([SpeechSynthesisHandle](#) *
handle)

Utworzenie i otwarcie obiektu syntezy mowy.

Parametry:

out	handle	Wskaźnik, do którego zostanie zapisany uchwyt do obiektu syntezy
-----	--------	--

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie utworzono i otwarto obiekt syntezy
- [SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany
- [SPEECHSYNTHESIS_ERROR_INVALID_PARAMETER](#) - przekazany wskaźnik na uchwyt ma wartość NULL
- [SPEECHSYNTHESIS_ERROR_OUT_OF_MEMORY](#) - brak pamięci, aby wykonać tę operację
- [SPEECHSYNTHESIS_ERROR_FAILED_TO_LOAD_VOICE](#) - nie udało się załadować pliku z głosem
- [SPEECHSYNTHESIS_ERROR_NO_GONG_FILE](#) - plik z gongiem nie istnieje
- [SPEECHSYNTHESIS_ERROR_INVALID_GONG_FILE](#) - plik z gongiem jest nieprawidłowy
- [SPEECHSYNTHESIS_ERROR_AUDIO_DEVICE_FORMAT_NOT_SUPPORTED](#) - karta dźwiękowa nie obsługuje żadnego z wymaganych formatów
- [SPEECHSYNTHESIS_ERROR_NO_VALID_AUDIO_DEVICE](#) - nie znaleziono karty dźwiękowej

1.2.3.5 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_free](#) ([SpeechSynthesisHandle handle](#))

Zamknięcie i zniszczenie obiektu syntezy mowy.

Parametry:

in	handle	Uchwyt do obiektu syntezy mowy
----	--------	--------------------------------

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie zamknięto i zniszczono obiekt syntezy
- [SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED](#) - moduł nie został zainicjalizowany
- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3 Funkcje do kontroli procesu odczytu

Funkcje, które pozwalają na odczytanie tekstu oraz na zarządzanie parametrami odczytu.

1.3.1 Funkcje

- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_read_text](#) ([SpeechSynthesisHandle handle](#), const char *text)
Odczyt przekazanego tekstu.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_stop_reading](#) ([SpeechSynthesisHandle handle](#))
Natychmiastowe zakończenie odczytywania przekazanego wcześniej tekstu.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_set_sampling_rate](#) ([SpeechSynthesisHandle handle](#), uint32_t sampling_rate)
Ustawienie częstotliwości próbkowania, z jaką będzie generowany odczytywany tekst.



- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_get_sampling_rate](#)
([SpeechSynthesisHandle](#) handle)
Pobranie aktualnie ustawionej częstotliwości próbkowania, z jaką jest generowany odczytywany tekst.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_set_gong_enabled](#)
([SpeechSynthesisHandle](#) handle, int32_t enabled)
Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytywanym tekstem.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_get_gong_enabled](#)
([SpeechSynthesisHandle](#) handle)
Pobranie informacji, czy odtwarzanie dźwięku gongu przed odczytywanym tekstem jest aktywne.

1.3.2 Opis szczegółowy

Funkcje, które pozwalają na odczytanie tekstu oraz na zarządzanie parametrami odczytu.

1.3.3 Dokumentacja funkcji

- 1.3.3.1 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_read_text](#)
([SpeechSynthesisHandle](#) handle, const char * text)

Odczyt przekazanego tekstu.

Parametry:

in	handle	Uchwyt do obiektu syntezy mowy
in	text	Tekst w formacie UTF-8, który zostanie odczytany przez syntezy mowy

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie zamknięto i zniszczono obiekt syntezy mowy
- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [SPEECHSYNTHESIS_ERROR_OUT_OF_MEMORY](#) - brak pamięci, aby wykonać tę operację

- 1.3.3.2 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_stop_reading](#)
([SpeechSynthesisHandle](#) handle)

Natychmiastowe zakończenie odczytywania przekazanego wcześniej tekstu.

Parametry:

in	handle	Uchwyt do obiektu syntezy mowy
----	--------	--------------------------------

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie zamknięto i zniszczono obiekt syntezy mowy
- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3.3.3 [SPEECHSYNTHESIS_EXPORT](#) `int32_t` `speechsynthesis_set_sampling_rate` ([SpeechSynthesisHandle](#) *handle*, `uint32_t` *sampling_rate*)

Ustawienie częstotliwości próbkowania, z jaką będzie generowany odczytywany tekst.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu syntezy mowy
in	<i>sampling_rate</i>	<p>Częstotliwość próbkowania, z jaką ma być generowany odczytywany tekst. Obsługiwane wartości:</p> <ul style="list-style-type: none"> ▪ 8000 ▪ 16000 ▪ 22050 ▪ 44100 ▪ 48000

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie zamknięto i zniszczono obiekt syntezy mowy
- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny
- [SPEECHSYNTHESIS_ERROR_INVALID_PARAMETER](#) - przekazana częstotliwość jest niepoprawna

1.3.3.4 [SPEECHSYNTHESIS_EXPORT](#) `int32_t` `speechsynthesis_get_sampling_rate` ([SpeechSynthesisHandle](#) *handle*)

Pobranie aktualnie ustawionej częstotliwości próbkowania, z jaką jest generowany odczytywany tekst.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu syntezy mowy
----	---------------	--------------------------------

Zwraca:

Jeśli wartość ≥ 0 , to jest to aktualnie ustawiona częstotliwość próbkowania. Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędów:

[SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3.3.5 [SPEECHSYNTHESIS_EXPORT](#) `int32_t` `speechsynthesis_set_gong_enabled` ([SpeechSynthesisHandle](#) *handle*, `int32_t` *enabled*)

Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytywanym tekstem.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu syntezy mowy
----	---------------	--------------------------------



in	<i>enabled</i>	Wartość określająca czy gong należy włączyć (1), czy też wyłączyć (0)
----	----------------	---

Zwraca:

- [SPEECHSYNTHESIS_SUCCESS](#) - poprawnie zamknięto i zniszczono obiekt syntezatora
- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

1.3.3.6 [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_get_gong_enabled](#)
([SpeechSynthesisHandle](#) handle)

Pobranie informacji, czy odtwarzanie dźwięku gongu przed odczytywanym tekstem jest aktywne.

Parametry:

in	<i>handle</i>	Uchwyt do obiektu syntezatora mowy
----	---------------	------------------------------------

Zwraca:

Wartość 0 oznacza, że dźwięk gongu jest wyłączony.

Wartość 1 oznacza, że dźwięk gongu jest włączony.

Jeśli wartość jest mniejsza od 0, to jest to jeden z następujących kodów błędu:

- [SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) - przekazany uchwyt jest niepoprawny

2 Dokumentacja plików

2.1 Dokumentacja pliku `SpeechSynthesis.h`

Plik zawierający publiczne API dla modułu syntezy mowy.

```
#include <stdint.h>
```

2.1.1 Definicje

- `#define SPEECHSYNTHESIS_EXPORT`
Makro eksportujące funkcje API w bibliotece `*.dll/*.so`.
- `#define SPEECHSYNTHESIS_API_VERSION 1`
Zmienna określająca wersję API.

2.1.2 Definicje typów

- `typedef struct SpeechSynthesisInternal * SpeechSynthesisHandle`
Uchwyt do obiektu syntezatora mowy.

2.1.3 Wyliczenia

- `enum SpeechSynthesisStatus { SPEECHSYNTHESIS_SUCCESS = 0,
SPEECHSYNTHESIS_ERROR_INITIALIZATION_FAILED = -1,
SPEECHSYNTHESIS_ERROR_NOT_INITIALIZED = -2, SPEECHSYNTHESIS_ERROR_INTERNAL
= -3, SPEECHSYNTHESIS_ERROR_OUT_OF_MEMORY = -4,
SPEECHSYNTHESIS_ERROR_INVALID_PARAMETER = -5,`



[SPEECHSYNTHESIS_ERROR_FAILED_TO_LOAD_VOICE](#) = -10,
[SPEECHSYNTHESIS_ERROR_INVALID_GONG_FILE](#) = -11,
[SPEECHSYNTHESIS_ERROR_NO_GONG_FILE](#) = -12,
[SPEECHSYNTHESIS_ERROR_AUDIO_DEVICE_FORMAT_NOT_SUPPORTED](#) = -20,
[SPEECHSYNTHESIS_ERROR_NO_VALID_AUDIO_DEVICE](#) = -21,
[SPEECHSYNTHESIS_ERROR_INVALID_HANDLE](#) = -100,
[SPEECHSYNTHESIS_ERROR_INVALID_LICENSE](#) = -1000 }Enumeracja możliwych wartości
zwracanych z funkcji API.

2.1.4 Funkcje

- [SPEECHSYNTHESIS_EXPORT](#) uint32_t [speechsynthesis_api_version](#) ()
Zwrócenie aktualnej wersji API, używanej przez moduł.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_initialize](#) ()
Zainicjalizowanie modułu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_shutdown](#) ()
Zamknięcie modułu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_create](#) ([SpeechSynthesisHandle](#) *handle)
Utworzenie i otwarcie obiektu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_free](#) ([SpeechSynthesisHandle](#) handle)
Zamknięcie i zniszczenie obiektu syntezy mowy.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_read_text](#) ([SpeechSynthesisHandle](#) handle, const char *text)
Odczyt przekazanego tekstu.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_stop_reading](#) ([SpeechSynthesisHandle](#) handle)
Natychmiastowe zakończenie odczytywania przekazanego wcześniej tekstu.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_set_sampling_rate](#) ([SpeechSynthesisHandle](#) handle, uint32_t sampling_rate)
Ustawienie częstotliwości próbkowania, z jaką będzie generowany odczytywany tekst.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_get_sampling_rate](#) ([SpeechSynthesisHandle](#) handle)
Pobranie aktualnie ustawionej częstotliwości próbkowania, z jaką jest generowany odczytywany tekst.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_set_gong_enabled](#) ([SpeechSynthesisHandle](#) handle, int32_t enabled)
Włączenie lub wyłączenie odtwarzania dźwięku gongu przed odczytywaniem tekstem.
- [SPEECHSYNTHESIS_EXPORT](#) int32_t [speechsynthesis_get_gong_enabled](#) ([SpeechSynthesisHandle](#) handle)
Pobranie informacji, czy odtwarzanie dźwięku gongu przed odczytywaniem tekstem jest aktywne.



2.1.5 Opis szczegółowy

Plik zawierający publiczne API dla modułu syntezy mowy.

Definicja w pliku [SpeechSynthesis.h](#).

2.1.6 Dokumentacja definicji

2.1.6.1 SPEECHSYNTHESIS_API_VERSION 1

Zmienna określająca wersję API.

Wersję tę należy porównać z wartością zwracaną przez funkcję [speechsynthesis_api_version\(\)](#).
Jeśli wartości te różnią się, nie należy używać tego modułu.

Zobacz również:

- [speechsynthesis_api_version](#)

Definicja w linii [61](#) pliku [SpeechSynthesis.h](#).

2.2 SpeechSynthesis.h

```
1 #pragma once
2
3 #include <stdint.h>
4
50 #if defined(_WIN32) || defined(WIN32)
51     #if defined(SPEECHSYNTHESIS_BUILD)
52         #define SPEECHSYNTHESIS_EXPORT __declspec(dllexport)
53     #else /* SPEECHSYNTHESIS_BUILD */
54         #define SPEECHSYNTHESIS_EXPORT __declspec(dllimport)
55     #endif /* SPEECHSYNTHESIS_BUILD */
56 #elif defined(__linux__)
57     #if defined(SPEECHSYNTHESIS_BUILD)
58         #define SPEECHSYNTHESIS_EXPORT __attribute__((visibility("default")))
59     #else /* SPEECHSYNTHESIS_BUILD */
60         #define SPEECHSYNTHESIS_EXPORT
61     #endif /* SPEECHSYNTHESIS_BUILD */
62 #else
63     #define SPEECHSYNTHESIS_EXPORT
64     #error "Unsupported platform!"
65 #endif
66
67 #if defined(__cplusplus)
68 extern "C" {
69 #endif /* defined(__cplusplus) */
70
```




```
61 #define SPEECHSYNTHESIS_API_VERSION 1
62
117 enum SpeechSynthesisStatus {
118     SPEECHSYNTHESIS\_SUCCESS = 0,
119
120     SPEECHSYNTHESIS\_ERROR\_INITIALIZATION\_FAILED = -1,
121     SPEECHSYNTHESIS\_ERROR\_NOT\_INITIALIZED = -2,
122
123     SPEECHSYNTHESIS\_ERROR\_INTERNAL = -3,
124     SPEECHSYNTHESIS\_ERROR\_OUT\_OF\_MEMORY = -4,
125     SPEECHSYNTHESIS\_ERROR\_INVALID\_PARAMETER = -5,
126
127     SPEECHSYNTHESIS\_ERROR\_FAILED\_TO\_LOAD\_VOICE = -10,
128     SPEECHSYNTHESIS\_ERROR\_INVALID\_GONG\_FILE = -11,
129     SPEECHSYNTHESIS\_ERROR\_NO\_GONG\_FILE = -12,
130
131     SPEECHSYNTHESIS\_ERROR\_AUDIO\_DEVICE\_FORMAT\_NOT\_SUPPORTED = -20,
132     SPEECHSYNTHESIS\_ERROR\_NO\_VALID\_AUDIO\_DEVICE = -21,
133
134     SPEECHSYNTHESIS\_ERROR\_INVALID\_HANDLE = -100,
135
136     SPEECHSYNTHESIS\_ERROR\_INVALID\_LICENSE = -1000
137 };
138
142 typedef struct SpeechSynthesisInternal* SpeechSynthesisHandle;
143
158 SPEECHSYNTHESIS\_EXPORT uint32_t speechsynthesis\_api\_version();
159
174 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_initialize();
175
188 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_shutdown();
189
206 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_create(SpeechSynthesisHandle*
handle);
207
218 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_free(SpeechSynthesisHandle handle);
219
231 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_read\_text(SpeechSynthesisHandle
handle, const char* text);
232
242 SPEECHSYNTHESIS\_EXPORT int32_t speechsynthesis\_stop\_reading(SpeechSynthesisHandle
handle);
```




```
243
261 SPEECHSYNTHESIS\_EXPORT int32_t
speechsynthesis\_set\_sampling\_rate(SpeechSynthesisHandle handle, uint32_t sampling_rate);
262
274 SPEECHSYNTHESIS\_EXPORT int32_t
speechsynthesis\_get\_sampling\_rate(SpeechSynthesisHandle handle);
275
286 SPEECHSYNTHESIS\_EXPORT int32_t
speechsynthesis\_set\_gong\_enabled(SpeechSynthesisHandle handle, int32_t enabled);
287
299 SPEECHSYNTHESIS\_EXPORT int32_t
speechsynthesis\_get\_gong\_enabled(SpeechSynthesisHandle handle);
300
301 #if defined(__cplusplus)
302 } /* extern "C" */
303 #endif /* defined(__cplusplus) */
```



Dokumentacja API dostarczanego przez moduł Wideoterminala

1 Dokumentacja grup

1.1 Wartości globalne

Wartości globalne wykorzystywane w module.

1.1.1 Definicje

- `#define DLLEXPORT __declspec(dllexport)`
Makro eksportujące funkcje API w bibliotece dll.

1.1.2 Zmienne

- `const uint32_t VIDEOTERMINAL_API_VERSION = 3`
Zmienna określająca wersję API.

1.1.3 Opis szczegółowy

Wartości globalne wykorzystywane w module.

1.1.4 Dokumentacja zmiennych

1.1.4.1 [VIDEOTERMINAL_API_VERSION](#) = 3

Zmienna określająca wersję API.

Wersję tę należy porównać z wartością zwracaną przez funkcję [vc_getApiVersion\(\)](#). Jeśli wartości te różnią się, nie należy używać tego modułu.

Zobacz również:

[vc_getApiVersion](#)

Definicja w linii [27](#) pliku [conferenceInterface.h](#).

1.2 Typy

Typy wykorzystywane w module.

1.2.1 Struktury danych

- struct [Callbacks](#)
Struktura przechowująca wszystkie wywołania zwrotne od Wideoterminala.
 - struct [Handle](#)
Uchwyt do obiektu Wideokonferencji.

1.2.2 Definicje typów

- typedef uint32_t [bool](#)
Uściślenie typu bool.

1.2.3 Opis szczegółowy

Typy wykorzystywane w module.

1.3 Inicjalizacja

Funkcje służące do inicjalizacji obiektu interfejsu do Wideoterminala.

1.3.1 Wyliczenia

- enum [RegisterStatus](#) { [SUCCESS](#), [TERMINAL_USED](#), [VERSION_MISMATCH](#), [NULL_CALLBACKS](#) } *Enumeracja możliwych efektów rejestracji.*

1.3.2 Funkcje

- [DLLEXPORT uint32_t vc_getApiVersion \(\)](#)
Funkcja zwraca wersję API używaną przez moduł
- [DLLEXPORT Handle * vc_getHandle \(\)](#)
Funkcja służąca do otrzymania uchwytu do obiektu Wideoterminala.
- [DLLEXPORT bool vc_init \(Handle *handle, const char *terminalIP, uint32_t terminalPort\)](#)
Inicjalizowanie interfejsu.
- [DLLEXPORT uint32_t vc_register \(Handle *h, Callbacks *callbacks\)](#)
Funkcja rejestrująca interfejs w Wideoterminalu.

1.3.3 Opis szczegółowy

Funkcje służące do inicjalizacji obiektu interfejsu do Wideoterminala.

Aby zainicjalizować Wideoterminal, należy przejść przez następujące kroki:

- Wywołanie funkcji [vc_getApiVersion\(\)](#) w celu sprawdzenia wersji API
- Wywołanie funkcji [vc_getHandle\(\)](#) w celu otrzymania uchwytu do interfejsu Wideoterminala
- Wywołanie funkcji [vc_init\(\)](#), aby uzyskać połączenie sieciowe z Wideoterminalem
- Wywołanie funkcji [vc_register\(\)](#), aby uwierzytelnić się w Wideoterminalu oraz by uzyskać kanał informacji zwrotnych

Przed wywołaniem pełnej procedury inicjalizacji nie można wykonywać operacji z innych grup (nie dotyczy [vc_cleanup\(\)](#))

1.3.4 Dokumentacja typów wyliczanych

1.3.4.1 enum [RegisterStatus](#)

Enumeracja możliwych efektów rejestracji.

Wartość ta jest zwracana przez funkcję [vc_register\(\)](#)

Wartości wyliczeń

SUCCESS Rejestracja przebiegła pomyślnie

TERMINAL_USED Wideoterminal jest już wykorzystywany

VERSION_MISMATCH Interfejs oraz Wideoterminal posiadają niezgodne numery wersji

NULL_CALLBACKS W strukturze znaleziono nieuzupełnione wywołanie zwrotne

Definicja w linii [215](#) pliku [conferenceInterface.h](#).

1.3.5 Dokumentacja funkcji

1.3.5.1 [DLLEXPORT uint32_t vc_getApiVersion \(\)](#)

Funkcja zwraca wersję API używaną przez moduł

Funkcja pozwala na wczesne wykrycie przez aplikację zmian w API. Jeśli wersja API używana przez aplikację nie jest zgodna z wartością tutaj zwróconą, powinna ona nie wykorzystywać tego modułu

Zwraca:

Wartość stałej [VIDEOTERMINAL_API_VERSION](#)

Zobacz również:

[VIDEOTERMINAL_API_VERSION](#)

1.3.5.2 [DLLEXPORT Handle*](#) `vc_getHandle ()`

Funkcja służąca do otrzymania uchwytu do obiektu Wideoterminala.

Zwraca:

Niezerowy wskaźnik - operacja przebiegła pomyślnie

Wskaźnik zerowy - wystąpił błąd inicjalizacji (inny obiekt typu [Handle](#) już istnieje)

Funkcja pozwala uzyskać obiekt uchwytu, który należy potem przekazywać do każdej funkcji w API. W danym momencie może występować w programie tylko jeden obiekt typu [Handle](#). Otrzymany za pomocą tej funkcji obiekt należy zwolnić przy pomocy funkcji [vc_cleanup\(\)](#)

1.3.5.3 [DLLEXPORT _bool](#) `vc_init (Handle * handle, const char * terminalIP, uint32_t terminalPort)`

Inicjalizowanie interfejsu.

Parametry:

in	<i>handle</i>	prawidłowy uchwyt do obiektu Wideoterminala
in	<i>terminalIP</i>	adres ip Wideoterminala
in	<i>terminalPort</i>	numer portu Wideoterminala

Zwraca:

Funkcja zwraca, czy udało się połączyć z Wideoterminalem

1.3.5.4 [DLLEXPORT uint32_t](#) `vc_register (Handle * h, Callbacks * callbacks)`

Funkcja rejestrująca interfejs w Wideoterminalu.

Parametry:

in	<i>h</i>	prawidłowy uchwyt do obiektu Wideoterminala
in	<i>callbacks</i>	wskaźnik do obiektu wypełnionymi wszystkimi wywołaniami zwrotnymi (Callbacks)

Zwraca:

Wartość zwracana jest enumeratorem [RegisterStatus](#). Możliwe do zwrócenia są wszystkie jego wartości

Aby operacja się powiodła, wszystkie elementy przekazywanej struktury [Callbacks](#) muszą być wypełnione niezerowym wskaźnikiem

1.4 Zakończenie działania

Funkcje służące do zakończenia działania interfejsu.

1.4.1 Definicje typów

- `typedef void(__cdecl * terminalDisconnectedPtr) ()`
Wywołanie otrzymywane w przypadku rozłączenia Wideoterminala.

1.4.2 Funkcje

- `DLLEXPORT void vc_cleanup (Handle *h)`
Zamknięcie interfejsu.

1.4.3 Opis szczegółowy

Funkcje służące do zakończenia działania interfejsu.

Aby zakończyć działanie interfejsu należy wywołać funkcję [vc_cleanup\(\)](#). Dodatkowo, działanie interfejsu należy zakończyć po otrzymaniu wywołania zwrotnego [Callbacks::terminalDisconnected](#).

1.4.4 Dokumentacja definicji typów

1.4.4.1 `typedef void(__cdecl * terminalDisconnectedPtr) ()`

Wywołanie otrzymywane w przypadku rozłączenia Wideoterminala.

Może wystąpić, w momencie gdy Wideoterminal zostanie wyłączony albo utracone zostanie połączenie sieciowe.

Definicja w linii [435](#) pliku [conferenceInterface.h](#).

1.4.5 Dokumentacja funkcji

1.4.5.1 `DLLEXPORT void vc_cleanup (Handle * h)`

Zamknięcie interfejsu.

Parametry:

in	h	poprawny uchwyt do obiektu interfejsu
----	---	---------------------------------------

Funkcję można wywołać w dowolnym momencie inicjalizacji opisanej w [Inicjalizacja](#). Funkcja musi więc dostosować swoje działanie do tego, w którym etapie została wywołana.

1.5 Połączenia

Wykonywanie oraz zarządzanie połączeniami z innymi Wideoterminalami.



1.5.1 Definicje typów

- `typedef void(__cdecl * successPtr) ()`
Informacja o pomyślnym rozpoczęciu połączenia.
- `typedef void(__cdecl * incomingPtr) (const char *ip, bool needsAccept)`
Informacja o połączeniu przychodzącym.
- `typedef void(__cdecl * needsAcceptPtr) ()`
Informacja, że wymagane jest ręczne odebranie połączenia.
- `typedef void(__cdecl * callClearedPtr) (uint8_t reason)`
Informacja, że połączenie zostało zakończone.
- `typedef void(__cdecl * rejectedIncomingConnectionPtr) (const char *address)`
Informacja, że pewne połączenie przychodzące zostało w sposób automatyczny odrzucone.
- `typedef void(__cdecl * onStopReceivingDualStreamPtr) (uint32_t reason)`
Informacja, że Wideoteminal zakończył odbieranie prezentacji.
- `typedef void(__cdecl * onStartReceivingDualStreamPtr) ()`
Informacja, że Wideoteminal zaczął odbierać drugi strumień

1.5.2 Wyliczenia

- `enum CallEndReason { NORMAL, OTHER SIDE ENDED, OTHER SIDE REFUSED, OTHER SIDE UNRESPONSIVE, NO OTHER SIDE, GATEKEEPER ENDED, MEDIA TRANSPORT FAIL, GATEKEEPER NO USER, NO COMMON MEDIA FORMATS, SECURITY DENIED, OTHER }`*Enumeracja ze wszystkimi możliwymi powodami zakończenia połączenia.*
- `enum Protocol { H323, SIP }`*Protokoły sygnalizacyjne.*
- `enum AnswerType { MANUAL, AUTO, REJECT }`*Tryby odbierania połączenia.*
- `enum EndDualStreamReason { eEndedByLocalUser = 0, eEndedByRemoteUser = 1, eEndedByError = 2 }`*Powody zakończenia drugiego strumienia.*
- `enum GatekeeperRegistrationStatus { REGISTERED, SECURITY FAIL, OTHER ERROR, UNREGISTERED }`*Status połączenia z gatekeeperem.*

1.5.3 Funkcje

- `DLLEXPORT bool vc_call (Handle *h, const char *ip, uint32_t protocol)`
Wykonanie połączenia.
- `DLLEXPORT bool vc_acceptConnection (Handle *h, uint32_t isAccept)`
Manualne odebranie połączenia.
- `DLLEXPORT uint8_t vc_getAnswerType (Handle *h)`
Pobranie trybu odbierania połączenia.
- `DLLEXPORT void vc_setAnswerType (Handle *h, uint8_t answerType)`
Ustawienie trybu odbierania połączenia.
- `DLLEXPORT void vc_hangupCall (Handle *h)`
Zakończenie połączenia.
- `DLLEXPORT bool vc_sendDTMF (Handle *h, char sign)`
Wysłanie DTMF.

- [DLLEXPORT](#) [bool](#) [vc_getRemotePartyName](#) ([Handle](#) *h, char *name, uint32_t *size)
Pobranie nazwy strony zdalnej.

1.5.4 Opis szczegółowy

Wykonywanie oraz zarządzanie połączeniami z innymi Wideoterminalami.

1.5.5 Wykonywanie połączeń

Aby wykonać połączenie należy wywołać funkcję [vc_call\(\)](#). Jest to zadanie asynchroniczne, a jego zakończeniem jest otrzymanie wywołania zwrotnego [Callbacks::success](#) w przypadku sukcesu albo [Callbacks::callCleared](#) w przypadku błędu.

1.5.6 Odbieranie połączeń

Działania wykonywane przez interfejs oraz Wideoterminal podczas odbierania połączenia zależą od ustawionego w funkcji [vc_setAnswerType\(\)](#) trybu odbierania połączenia. Tryb automatyczny

Gdy Wideoterminal otrzyma połączenie przychodzące, aplikacja otrzymuje wywołanie zwrotne [Callbacks::incoming](#). Następnie Wideoterminal przechodzi procedurę rozpoczęcia połączenia, która kończy się wywołaniem [Callbacks::success](#). Tryb manualny

Informację o połączeniu przychodzącym aplikacja otrzymuje za pomocą wywołania zwrotnego [Callbacks::incoming](#). Następnie w wywołaniu zwrotnym [Callbacks::needsAccept](#) aplikacja otrzymuje zobowiązanie do zdecydowania czy połączenie zostanie odebrane czy nie. Aplikacja musi to zrobić za pomocą funkcji [vc_acceptConnection\(\)](#). Zakończenie odbierania połączenia zaznaczone jest za pomocą wywołania [Callbacks::success](#). Automatyczne odrzucanie

Informacja o automatycznie odrzuconym połączeniu przychodzącym jest przekazywana do Aplikacji poprzez wywołanie zwrotne [Callbacks::rejectedIncomingConnection](#).

1.5.7 Zakończenie połączenia

Aby przerwać połączenie należy wywołać funkcję [vc_hangupCall\(\)](#). Warto zwrócić uwagę, że funkcji tej można też użyć podczas wykonywania połączenia pomiędzy wywołaniem [vc_call\(\)](#) a otrzymaniem [Callbacks::success](#). O fakcie zakończenia połączenia aplikacja jest informowana poprzez wywołanie [Callbacks::callCleared](#).

1.5.8 Dokumentacja definicji typów

1.5.8.1 typedef void(__cdecl * successPtr) ()

Informacja o pomyślnym rozpoczęciu połączenia.

Otrzymanie tego wywołania zwrotnego oznacza, że połączenie z innym Wideoterminalem zostało rozpoczęte.

Definicja w linii [332](#) pliku [conferenceInterface.h](#).

1.5.8.2 typedef void(__cdecl * incomingPtr) (const char *ip, [bool](#) needsAccept)

Informacja o połączeniu przychodzącym.



Parametry:

<i>ip</i>	adres IP albo alias osoby dzwoniącej
<i>needsAccept</i>	informacja, czy będzie potrzebne ręczne odbieranie połączenia (manualny tryb odbierania połączenia)

Informacja, że rozpoczęto procedurę odbierania połączenia przychodzącego.

Definicja w linii [343](#) pliku [conferenceInterface.h](#).

1.5.8.3 `typedef void(__cdecl * needsAcceptPtr) ()`

Informacja, że wymagane jest ręczne odebranie połączenia.

Po otrzymaniu tego wywołania należy zawołać funkcję [vc_acceptConnection\(\)](#) z decyzją

Definicja w linii [352](#) pliku [conferenceInterface.h](#).

1.5.8.4 `typedef void(__cdecl * callClearedPtr) (uint8_t reason)`

Informacja, że połączenie zostało zakończone.

Parametry:

<i>reason</i>	powód zakończenia połączenia - enumerator CallEndReason
---------------	---

Otrzymanie tego wywołania oznacza, że aplikacja może wykonać albo otrzymać nowe połączenie

Definicja w linii [362](#) pliku [conferenceInterface.h](#).

1.5.8.5 `typedef void(__cdecl * rejectedIncomingConnectionPtr) (const char *address)`

Informacja, że pewne połączenie przychodzące zostało w sposób automatyczny odrzucone.

Parametry:

<i>address</i>	adres IP albo alias terminala dzwoniącego
----------------	---

Definicja w linii [370](#) pliku [conferenceInterface.h](#).

1.5.8.6 `typedef void(__cdecl * onStopReceivingDualStreamPtr) (uint32_t reason)`

Informacja, że Wideoteminal zakończył odbieranie prezentacji.

Parametry:

<i>reason</i>	Enumerator EndDualStreamReason
---------------	--

Definicja w linii [386](#) pliku [conferenceInterface.h](#).

1.5.9 Dokumentacja typów wyliczanych

1.5.9.1 `enum CallEndReason`

Enumeracja ze wszystkimi możliwymi powodami zakończenia połączenia.



Przekazywana do aplikacji przez wywołanie zwrotne [Callbacks::callCleared](#)

Wartości wyliczeń

NORMAL Standardowe zakończenie połączenia przez funkcję [vc_hangupCall\(\)](#)

OTHER_SIDE_ENDED Standardowe zakończenie połączenia przez drugą stronę

OTHER_SIDE_REFUSED Druga strona odrzuciła połączenie

OTHER_SIDE_UNRESPONSIVE Druga strona nie odpowiada. Otrzymywane, gdy druga strona zbyt długo zwleka z odebraniem bądź odrzuceniem połączenia

NO_OTHER_SIDE Brak połączenia z adresem do którego dzwoni aplikacja albo nie jest tam uruchomiony terminal obsługujący wybrany protokół

GATEKEEPER_ENDED Gatekeeper zakończył trwające połączenie

MEDIA_TRANSPORT_FAIL Wystąpił problem z kanałem do transmisji multimedialnych

GATEKEEPER_NO_USER Wybrany wewnątrz `vc_call` użytkownik nie istnieje

NO_COMMON_MEDIA_FORMATS Wideoterminal nie posiada wspólnych formatów mediów z drugim terminalem

SECURITY_DENIED Nie było możliwe wykonanie połączenia z zadany poziomem bezpieczeństwa

OTHER Inne błędy

Definicja w linii [228](#) pliku [conferenceInterface.h](#).

1.5.9.2 enum [Protocol](#)

Protokoły sygnalizacyjne.

Wszystkie obsługiwane przez Wideoterminal protokoły sygnalizacyjne

Wartości wyliczeń

H323 Protokół H323

SIP Protokół SIP

Definicja w linii [249](#) pliku [conferenceInterface.h](#).

1.5.9.3 enum [AnswerType](#)

Tryby odbierania połączenia.

Wartości wyliczeń

MANUAL Odbieranie ręczne

AUTO Odbieranie automatyczne

REJECT Automatyczne odrzucanie połączeń

Definicja w linii [258](#) pliku [conferenceInterface.h](#).

1.5.9.4 enum [EndDualStreamReason](#)

Powody zakończenia drugiego strumienia.

Wartości wyliczeń

eEndedByLocalUser Wideoterminal zakończył odbieranie drugiego strumienia

eEndedByRemoteUser Druga strona zakończyła wysyłanie drugiego strumienia

eEndedByError Wystąpił błąd podczas odbierania drugiego strumienia

Definicja w linii [309](#) pliku [conferenceInterface.h](#).

1.5.9.5 enum [GatekeeperRegistrationStatus](#)

Status połączenia z gatekeeperem.

Wartości wyliczeń

REGISTERED Zarejestrowano w gatekeeperze

SECURITY_FAIL Niepoprawne ustawienie zabezpieczeń (w tym nieprawidłowa para użytkownik - hasło)

OTHER_ERROR Inny błąd połączenia z gatekeeperem

UNREGISTERED Odrejestrowano z gatekeepera

Definicja w linii [319](#) pliku [conferenceInterface.h](#).

1.5.10 Dokumentacja funkcji

1.5.10.1 [DLLEXPORT](#) [bool](#) [vc_call](#) ([Handle](#) * *h*, const char * *ip*, uint32_t *protocol*)

Wykonanie połączenia.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>ip</i>	adres IP albo alias, do którego aplikacja chce zadzwonić
in	<i>protocol</i>	wybrany protokół sygnalizacyjny Protocol

Zwraca:

Funkcja zwraca, czy można było rozpocząć wykonywanie połączenia

W każdym momencie Wideoterminal może posiadać tylko jedno połączenie. Połączenie zaczyna się albo wywołaniem [vc_call\(\)](#) albo przyjęciem wywołania [Callbacks::incoming](#), a kończy przyjęciem wywołania [Callbacks::callCleared](#).

1.5.10.2 [DLLEXPORT](#) [bool](#) [vc_acceptConnection](#) ([Handle](#) * *h*, uint32_t *isAccept*)

Manualne odebranie połączenia.



Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>isAccept</i>	decyzja, czy połączenie należy odebrać, czy je odrzucić

Zwraca:

Funkcja zwraca, czy było jakieś połączenie do akceptacji

1.5.10.3 [DLLEXPORT](#) `uint8_t vc_getAnswerType (Handle * h)`

Pobranie trybu odbierania połączenia.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
----	----------	------------------------------

Zwraca:

Tryb odbierania połączenia

Zobacz również:

[AnswerType](#)

1.5.10.4 [DLLEXPORT](#) `void vc_setAnswerType (Handle * h, uint8_t answerType)`

Ustawienie trybu odbierania połączenia.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>answerType</i>	tryb odbierania połączenia

Zobacz również:

[AnswerType](#)

1.5.10.5 [DLLEXPORT](#) `void vc_hangupCall (Handle * h)`

Zakończenie połączenia.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
----	----------	------------------------------

Funkcja służy do zakończenia trwającego połączenia. Powrót z niej nie jest jednoznaczny z tym, że połączenie się zakończyło, ponieważ zakończenie połączenia jest oznaczone przez odebranie [Callbacks::callCleared](#)

1.5.10.6 [DLLEXPORT](#) `bool vc_sendDTMF (Handle * h, char sign)`

Wysłanie DTMF.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>sign</i>	Wysyłany znak

Zwraca:

Funkcja zwraca, czy udało się wysłać DTMF

W przypadku połączenia za pomocą protokołu H.323, funkcja wysyła pakiet H.245 UserIntpuIndication. W przypadku połączenia za pomocą protokołu SIP, funkcja wysyła DTMF zgodnie z RFC2833, a w przypadku gdy strona zdalna go nie obsługuje, wysyła pakiet INFO o typie application/dtmf

1.5.10.7 [DLLEXPORT](#) [_bool](#) [vc_getRemotePartyName](#) ([Handle](#) * *h*, char * *name*, uint32_t * *size*)

Pobranie nazwy strony zdalnej.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>name</i>	Bufor, do którego zostanie zapisana nazwa
in,out	<i>size</i>	Rozmiar bufora

Zwraca:

Funkcja zwraca, czy udało się pobrać nazwę strony zdalnej

Funkcja zwraca nazwę strony zdalnej. W przypadku, gdy Wideoterminal jest zarejestrowany w gatekeeperze zostanie zwrócone h323_id. W przeciwnym razie zostanie zwrócony adres IP.

1.6 Zarządzanie źródłami

Fukcje służące do zarządzania źródłami obrazu oraz dźwięku.

1.6.1 Struktury danych

- struct [SourceDescriptor](#)

1.6.2 Opis źródła.

Definicje typów:

- typedef void(__cdecl * [sourceEventPtr](#)) (uint32_t code, uint8_t event)
Informacja, że mógł się zmienić stan źródła.
- typedef void(__cdecl * [onClosedSourceMediaStreamPtr](#)) (uint8_t type)
Informacja, że został zamknięty strumień wyjściowy.

1.6.3 Wyliczenia

- enum [StreamType](#) { [AUDIO](#), [VIDEO](#), [PRESENTATION](#) } *Wszystkie rodzaje strumieni wyjściowych.*
- enum [SourceEvent](#) { [ACTIVE](#) = 1, [INACTIVE](#) } *Aktywność źródła.*



- enum [SourceType](#) { [AUDIO_SOURCE](#), [VIDEO_SOURCE](#), [DUALSTREAM_SOURCE](#) } *Rodzaje strumieni.*

1.6.4 Funkcje

- [DLLEXPORT](#) void [vc_getAvailableSources](#) ([Handle](#) *h, uint32_t type, [SourceDescriptor](#) *descriptors, uint8_t *descriptorsSize)
Pobranie listy dostępnych źródeł
- [DLLEXPORT](#) bool [vc_setSource](#) ([Handle](#) *h, uint8_t type, uint32_t code)
Ustawienie źródła dla strumienia.

1.6.5 Opis szczegółowy

Fukcje służące do zarządzania źródłami obrazu oraz dźwięku.

1.6.6 Podstawowe pojęcia

- Strumień wyjściowy - ciąg danych multimedialnych wysyłany do strony zdalnej
- Źródło - generator danych multimedialnych.

1.6.7 Strumienie a źródła

Wideokonferencja posiada trzy typy strumieni wyjściowych - audio, podstawowy wideo oraz prezentację. Każdy ze strumieni posiada przypisane źródło, które jest uaktywniane podczas wideokonferencji, aby generować dane wysyłane do strony zdalnej. Strumienie posiadają listę źródeł, które można dla nich uaktywnić. Istnieje dodatkowe, puste źródło o kodzie 0, które nie generuje danych i powoduje, że strumień danego typu nie zostanie otwarty. Domyślnie przypisanym źródłem dla każdego ze strumieni jest źródło puste.

1.6.8 Aktywność źródeł

Każde ze źródeł może być aktywne albo nie, a także może zmienić ten stan podczas działania programu. Źródła nieaktywne cechują się tym, że nie może być ono przypisane do żadnego strumienia.

1.6.9 Reprezentacja strumieni i źródeł w interfejsie

Strumienie w interfejsie reprezentowane są za pomocą enumeratora [StreamType](#). Reprezentacją źródeł jest struktura [SourceDescriptor](#). Posiada ona kod źródła, który jest wykorzystywany podczas komunikacji z Wideoterminalem, nazwę, oraz stan aktywności. Listę wszystkich źródeł dla danego typu strumienia można uzyskać za pomocą funkcji [vc_getAvailableSources\(\)](#), a przypisanie źródła do strumienia za pomocą [vc_setSource\(\)](#). Aplikacja o zmianie aktywności źródła jest powiadamiana za pomocą wywołania zwrotnego [Callbacks::sourceEvent](#).

1.6.10 Podgląd obrazu

Wideoterminal umożliwia uzyskanie podglądu wysyłanego obrazu za pomocą modułu RemoteView. Jeśli podstawowy strumień wideo posiada przypisane niepuste źródło, wtedy musi udostępniać dla RemoteView źródło o nazwie kodowej preview. Gdy moduł RemoteView je uruchomi, wówczas musi zostać w Wideoterminale uruchomione źródło przypisane do tego strumienia, niezależnie od tego, czy aktualnie trwa konferencja czy nie. Jeśli źródło przypisane do tego strumienia stanie się nieaktywne bądź użytkownik ustawi je na puste, wtedy należy zerwać



połączenie z RemoteView. Wideoterminal musi umożliwiać jednoczesne podłączenie kilku takich modułów.

1.6.11 Dokumentacja definicji typów

1.6.11.1 typedef void(__cdecl * sourceEventPtr) (uint32_t code, uint8_t event)

Informacja, że mógł się zmienić stan źródła.

Parametry:

<i>code</i>	kod źródła, którego dotyczy zdarzenie
<i>event</i>	enumerators SourceEvent

Informacja, że aktywność źródła o pewnym kodzie mogła się zmienić. Dozwolone jest wysłanie tego powiadomienia, gdy stan źródła się nie zmienił.

Definicja w linii [416](#) pliku [conferenceInterface.h](#).

1.6.11.2 typedef void(__cdecl * onClosedSourceMediaStreamPtr) (uint8_t type)

Informacja, że został zamknięty strumień wyjściowy.

Parametry:

<i>type</i>	enumerators SourceType
-------------	--

Informacja o tym, że strumień o pewnym typie został zamknięty. Źródło przypisane do tego strumienia jest zmieniane na puste.

Definicja w linii [427](#) pliku [conferenceInterface.h](#).

1.6.12 Dokumentacja typów wyliczanych

1.6.12.1 enum [StreamType](#)

Wszystkie rodzaje strumieni wyjściowych.

Wartości wyliczeń

AUDIO Strumień audio

VIDEO Strumień wideo

PRESENTATION Strumień prezentacji (tzw. dual-stream)

Definicja w linii [268](#) pliku [conferenceInterface.h](#).

1.6.12.2 enum [SourceEvent](#)

Aktywność źródła.

Wartości wyliczeń

ACTIVE Źródło aktywne

INACTIVE Źródło nieaktywne



Definicja w linii [278](#) pliku [conferenceInterface.h](#).

1.6.12.3 enum [SourceType](#)

Rodzaje strumieni.

Wartości wyliczeń

AUDIO_SOURCE Strumień audio

VIDEO_SOURCE Strumień wideo

DUALSTREAM_SOURCE Strumień prezentacji

Definicja w linii [299](#) pliku [conferenceInterface.h](#).

1.6.13 Dokumentacja funkcji

1.6.13.1 **DLLEXPORT** void `vc_getAvailableSources (Handle * h, uint32_t type, SourceDescriptor * descriptors, uint8_t * descriptorsSize)`

Pobranie listy dostępnych źródeł

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>type</i>	rodzaj strumienia (SourceType)
out	<i>descriptors</i>	otrzymane deskryptory źródeł
in,out	<i>descriptorsSize</i>	na wejściu wielkość tablicy descriptors, na wyjściu rzeczywista ilość deskryptorów

1.6.13.2 **DLLEXPORT** **bool** `vc_setSource (Handle * h, uint8_t type, uint32_t code)`

Ustawienie źródła dla strumienia.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>type</i>	strumień, któremu chcemy ustawić źródło (SourceType)
in	<i>code</i>	kod źródła

Zwraca:

Funkcja zwraca, czy udało się poprawnie ustawić źródło. Podczas rozmowy wideokonferencyjnej oraz podczas wysyłania danych do RemoteView sprawdzane jest także, czy poprawnie udało się je uruchomić.

W przypadku, gdy nie uda się ustawić źródła, pozostawiane jest poprzednie.



1.7 Rozmiar wideokonferencji

Funkcje służące do zmiany maksymalnego rozmiaru obrazu obsługiwane przez Wideoterminal.

1.7.1 Struktury danych

- struct [C_Resolution](#)

1.7.2 Rozdzielczość Funkcje

- [DLLEXPORT](#) void [vc_getAvailableResolutions](#) ([Handle](#) *h, [C_Resolution](#) *resolutions, uint32_t *resolutionsSize)

Funkcja pobierająca możliwe do ustawienia rozdzielczości maksymalne.

- [DLLEXPORT](#) [bool](#) [vc_setResolution](#) ([Handle](#) *h, [C_Resolution](#) resolution)

Funkcja ustawiająca rozdzielczość maksymalną.

1.7.3 Opis szczegółowy

Funkcje służące do zmiany maksymalnego rozmiaru obrazu obsługiwane przez Wideoterminal.

Wideoterminal udostępnia klientowi listę rozdzielczości, które można w nim ustawić jako maksymalne. Nie należy traktować tej listy jako listy wszystkich obsługiwanych przez Wideoterminal rozdzielczości.

W odniesieniu do obrazu wychodzącego, rozdzielczość maksymalna oznacza tylko tyle, że nie będzie wysyłany większy od ustawionego obraz. Jeśli źródło nie jest w stanie wygenerować tak dużego obrazu, będzie wysyłało największy możliwy.

W odniesieniu do obrazu przychodzącego, rozdzielczość maksymalna oznacza, że poprawnie działająca strona zdalna nie będzie przysyłała do tego Wideoterminala większego rozmiaru.

1.7.4 Dokumentacja funkcji

- #### 1.7.4.1 [DLLEXPORT](#) void [vc_getAvailableResolutions](#) ([Handle](#) * h, [C_Resolution](#) * resolutions, uint32_t * resolutionsSize)

Funkcja pobierająca możliwe do ustawienia rozdzielczości maksymalne.

Parametry:

in	<i>h</i>	prawidłowy uchwyt do obiektu Wideoterminala
out	<i>resolutions</i>	tablica możliwych do ustawienia rozdzielczości
in,out	<i>resolutionsSize</i>	na wejściu rozmiar tablicy resolutions, a na wyjściu ilość dostępnych rozdzielczości

- #### 1.7.4.2 [DLLEXPORT](#) [bool](#) [vc_setResolution](#) ([Handle](#) * h, [C_Resolution](#) resolution)

Funkcja ustawiająca rozdzielczość maksymalną.

Parametry:

in	<i>h</i>	prawidłowy uchwyt do obiektu Wideoterminala
----	----------	---



in

resolution

wybrana rozdzielczość maksymalna

Zwraca:

Funkcja zwraca, czy udało się ustawić nową rozdzielczość.

Wprowadzona tutaj rozdzielczość maksymalna powinna występować w tablicy zwracanej przez [vc_getAvailableResolutions\(\)](#).

1.8 Książki adresowe oraz historia połączeń

Funkcje służące do zarządzania książkami adresowymi oraz historią połączeń

1.8.1 Definicje typów

- `typedef void(__cdecl * globalAddressbookRefreshSuccessPtr) ()`
Globalna książka adresowa została z powodzeniem odświeżona.
- `typedef void(__cdecl * globalAddressbookRefreshErrorPtr) ()`
Nie udało się odświeżyć globalnej książki adresowej.

1.8.2 Wyliczenia

- `enum ConnectionType { Received, Made, RejectedByListener, IncomingRejected, All }`
Rodzaje połączeń zapisywane w historii połączeń

1.8.3 Funkcje

- `DLLEXPORT bool vc_globalAddressbookRefresh (Handle *h)`
Odświeżenie globalnej książki adresowej.
- `DLLEXPORT bool vc_dictionarySelectLike (Handle *h, const char *courtType, const char *courtPrefix, uint32_t *responseSize)`
Pobranie danych słownika tłumaczącego numery E164.
- `DLLEXPORT bool vc_globalAddressbookSelectLike (Handle *h, const char *search, const char *number, bool active, uint32_t *responseSize)`
Pobieranie danych z globalnej książki adresowej.
- `DLLEXPORT bool vc_historyDeleteAll (Handle *h)`
Usunięcie wszystkich wpisów z historii połączeń
- `DLLEXPORT bool vc_historyDeleteID (Handle *h, uint32_t ID)`
Usunięcie pojedynczego wpisu z historii połączeń
- `DLLEXPORT bool vc_historySelectMode (Handle *h, char mode, uint32_t *responseSize)`
Pobranie wpisów z historii połączeń
- `DLLEXPORT bool vc_localAddressbookDeleteID (Handle *h, uint32_t ID)`
Usuwanie wpisu z lokalnej książki adresowej.
- `DLLEXPORT bool vc_localAddressbookInsert (Handle *h, const char *organization, const char *name, const char *address)`
Dodawanie wpisu do lokalnej książki adresowej.
- `DLLEXPORT bool vc_localAddressbookSelectLike (Handle *h, const char *organization, const char *name, const char *address, uint32_t *responseSize)`
Pobieranie i filtracja wpisów z lokalnej książki adresowej.

1.8.4 Opis szczegółowy

Funkcje służące do zarządzania książkami adresowymi oraz historią połączeń

1.8.5 Książki adresowe

Wideoterminal udostępnia aplikacji dwie książki adresowe - lokalną oraz globalną. Globalna książka adresowa posiada informacje o wszystkich Wideoterminalach znajdujących się w salach sądowych, w tym o ich obecnej aktywności. Lokalna książka adresowa to domyślnie pusty zbiór innych kontaktów. Globalna książka adresowa

Aby było możliwe pobranie danych z globalnej książki najpierw należy ją odświeżyć. Pierwsze odświeżenie odbywa się podczas wywołania [vc_register\(\)](#), a potem można dodatkowo odświeżać za pomocą funkcji [vc_globalAddressbookRefresh\(\)](#). Podczas odświeżenia Wideoterminal sprawdza dostępność innych Wideoterminali. Zakończeniem odświeżania jest otrzymanie wywołania zwrócenego [Callbacks::globalAddressbookRefreshSuccess](#) albo [Callbacks::globalAddressbookRefreshError](#).

Wideoterminal musi zapewniać obsługę słownika tłumaczącego numer E164 Wideoterminali na ich rzeczywistą lokalizację za pomocą funkcji [vc_dictionarySelectLike\(\)](#). Pobieranie oraz filtrowanie danych musi być zrealizowane w funkcji [vc_globalAddressbookSelectLike\(\)](#)

1.8.5.1 Lokalna książka adresowa

Lokalna książka adresowa to prosty zbiór kontaktów, z których każdy posiada nazwę organizacji, nazwę oraz adres. Pobranie oraz filtrowanie jest udostępniane w [vc_localAddressbookSelectLike\(\)](#), dodawanie danych w [vc_localAddressbookInsert\(\)](#), a usuwanie w [vc_localAddressbookDeleteID\(\)](#)

1.8.6 Historia połączeń

Wideoterminal ma za zadanie zapisywać wszystkie wykonywane oraz odbierane połączenia. Każdy wpis powinien posiadać swój typ ([ConnectionType](#)), czas oraz kontakt zwrotny. Kontaktem zwrotnym, w przypadku gdy Wideoterminal jest podłączony do gatekeepera jest h323_id, a w przeciwnym razie adres IP. Pobranie i filtrowanie kontaktów ma być udostępnione w funkcji [vc_historySelectMode\(\)](#), a usuwanie w [vc_historyDeleteID\(\)](#) oraz [vc_historyDeleteAll\(\)](#)

1.8.7 Format danych

Wszystkie funkcje z tej grupy których zadaniem jest pobranie danych zwracają rozmiar odpowiedzi. Aplikacja wtedy musi pobrać otrzymane dane w funkcji [vc_getData\(\)](#). Format otrzymanych danych zależy od wywołanej funkcji i należy je odpowiednio sparsować. Większość otrzymywanych danych to tekst, który rozpoczyna się jednym bajtem rozmiaru, za którym znajduje się dokładnie tyle bajtów danych o kodowaniu UTF-8.

1.8.8 Dokumentacja typów wyliczanych

1.8.8.1 enum [ConnectionType](#)

Rodzaje połączeń zapisywane w historii połączeń

Wartości wyliczeń

Received Połączenie odebrane

Made Połączenie wykonane z powodzeniem

RejectedByListener Połączenie zostało odrzucone przez stronę zdalną

IncomingRejected Połączenie zostało przez nas odrzucone

All Brak filtrowania

Definicja w linii [287](#) pliku [conferenceInterface.h](#).

1.8.9 Dokumentacja funkcji

1.8.9.1 [DLLEXPORT](#) [_bool](#) [vc_globalAddressbookRefresh](#) ([Handle](#) * *h*)

Odświeżenie globalnej książki adresowej.

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
----	----------	------------------------------

Zwraca:

Funkcja zwraca, czy udało się rozpocząć odświeżanie

Funkcja jedynie rozpoczyna odświeżanie. Aplikacja otrzyma informacje o zakończeniu odświeżania w wywołaniu zwrotnym [Callbacks::globalAddressbookRefreshSuccess](#) albo [Callbacks::globalAddressbookRefreshError](#).

1.8.9.2 [DLLEXPORT](#) [_bool](#) [vc_dictionarySelectLike](#) ([Handle](#) * *h*, const char * *courtType*, const char * *courtPrefix*, uint32_t * *responseSize*)

Pobranie danych słownika tłumaczącego numery E164.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>courtType</i>	Rodzaj sądu. Możliwe wartości to "SA" - sąd apelacyjny, "SO" - sąd okręgowy oraz "SR" - sąd rejonowy
in	<i>courtPrefix</i>	prefiks numeru E164 sądu. Można używać składni instrukcji LIKE w wersji SQLite.
out	<i>responseSize</i>	Rozmiar odpowiedzi. Bufor tego rozmiaru należy podać do funkcji vc_getData() .

Zwraca:

Funkcja zwraca, czy udało się pobrać dane

Prefiks numeru E164 ma następującą składnię:

- Pierwsze dwa znaki to 15

- Następne dwa znaki określają kod sądu apelacyjnego
- Kolejne dwa znaki określają kod sądu okręgowego
- Kolejne dwa znaki określają kod sądu rejonowego

Gdy operacja się powiodła, wtedy pole `responseSize` zawiera rozmiar danych do pobrania. Należy to zrobić w funkcji [vc_getData\(\)](#). Otrzymane dane to tablica danych zawierająca najpierw tekst prefiksu, a następnie tekst nazwy sądu. Tekst to najpierw jeden bajt rozmiaru, a następnie tekst w kodowaniu UTF-8 takiej wielkości. Podczas parsowania dane należy podawać do tablicy tak długo, jak są jeszcze dane do przeczytania.

1.8.9.3 [DLLEXPORT](#) [_bool](#) [vc_globalAddressbookSelectLike](#) ([Handle](#) * *h*, [const char](#) * *search*, [const char](#) * *number*, [_bool](#) *active*, [uint32_t](#) * *responseSize*)

Pobieranie danych z globalnej książki adresowej.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>search</i>	Tekst wyszukiwania (używa składni instrukcji LIKE z SQLite)
in	<i>number</i>	Prefiks numeru E164 (używa składni instrukcji LIKE z SQLite)
in	<i>active</i>	Prawda, jeśli aplikacja chce otrzymać tylko aktywne Wideoterminale
out	<i>responseSize</i>	Rozmiar odpowiedzi

Zwraca:

Funkcja zwraca, czy udało się pobrać dane

Wypełnianie pola `search` powoduje, że wyniki są zawężane do tych, w których miasto, nazwa, numer E164 lub numer pokoju zawiera wyszukiwany tekst. Jeśli to pole zawiera kilka słów oddzielonych spacjami, warunek ten musi być spełniony przez każde ze słów. Pole `number` pozwala na dodatkowe filtrowanie za pomocą prefiksów numerów E164 opisanych w opisie [vc_dictionarySelectLike\(\)](#).

Gdy operacja się powiodła, wtedy pole `responseSize` zawiera rozmiar danych do pobrania. Należy to zrobić w funkcji [vc_getData\(\)](#). Odpowiedź to tablica rekordów, które składają się kolejno z:

- tekstu z numerem E164
- tekstu z pełną nazwą sądu
- tekstu z lokacją (każdy sąd może mieć kilka oddziałów)
- tekstu z numerem pokoju
- tekstu z numerem Wideoterminala



- tekstu z typem Terminala ("recourt" albo "dodatkowy")
- tekstu z rodzajem sądu ("SA" - sąd apelacyjny, "SO" - sąd okręgowy, "SR" - sąd rejonowy)
- tekstu z nazwą miejscowości
- wartością typu bool z informacją, czy Wideoterminal jest aktywny

Tekst to bajt rozmiaru, a następnie dane kodowane w UTF-8. Wartość bool to bajt, którego niezerowa wartość oznacza prawdę, a zero fałsz.

Dane należy odczytywać dopóki w buforze są jeszcze jakieś dane do przeczytania.

1.8.9.4 [DLLEXPORT](#) [_bool](#) [vc_historyDeleteAll](#) ([Handle](#) * *h*)

Usunięcie wszystkich wpisów z historii połączeń

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
----	----------	------------------------------

Zwraca:

Funkcja zwraca, czy udało się usunąć dane.

1.8.9.5 [DLLEXPORT](#) [_bool](#) [vc_historyDeleteID](#) ([Handle](#) * *h*, [uint32_t](#) *ID*)

Usunięcie pojedynczego wpisu z historii połączeń

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>ID</i>	Id obiektu do usunięcia

Zwraca:

Funkcja zwraca, czy udało się usunąć obiekt.

1.8.9.6 [DLLEXPORT](#) [_bool](#) [vc_historySelectMode](#) ([Handle](#) * *h*, [char](#) *mode*, [uint32_t](#) * *responseSize*)

Pobranie wpisów z historii połączeń

Parametry:

in	<i>h</i>	uchwyt do obiektu interfejsu
in	<i>mode</i>	Enumerator ConnectionType
out	<i>responseSize</i>	Rozmiar odpowiedzi

Zwraca:

Funkcja zwraca, czy udało się pobrać dane

Funkcja służy do pobierania wpisów z historii połączeń. Gdy operacja się powiodła, wtedy pole `responseSize` zawiera rozmiar danych do pobrania. Należy to zrobić w funkcji [vc_getData\(\)](#). Dane to tablica rekordów, z których każdy zawiera:

- id elementu (do wykorzystania w funkcji [vc_historyDeleteID\(\)](#)) - 4 bajty
- tekst zawierający nazwę strony zdalnej z lokalnej książki adresowej
- tekst zawierający kontakt zwrotny strony zdalnej
- tekst zawierający czas rozpoczęcia wideokonferencji
- [ConnectionType](#) - 1 bajt

Dane należy odczytywać dopóki w buforze znajdują się jeszcze jakieś dane. Tekst to bajt rozmiaru, a następnie dane kodowane w UTF-8.

1.8.9.7 [DLLEXPORT _bool vc_localAddressbookDeleteID \(Handle * h, uint32_t ID\)](#)

Usuwanie wpisu z lokalnej książki adresowej.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>ID</i>	Id elementu do usunięcia

Zwraca:

Funkcja zwraca, czy udało się usunąć obiekt

Id elementu można otrzymać za pomocą funkcji [vc_localAddressbookSelectLike\(\)](#).

1.8.9.8 [DLLEXPORT _bool vc_localAddressbookInsert \(Handle * h, const char * organization, const char * name, const char * address\)](#)

Dodawanie wpisu do lokalnej książki adresowej.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>organization</i>	Nazwa organizacji
in	<i>name</i>	Nazwa
in	<i>address</i>	Adres IP albo h323_id strony zdalnej

Zwraca:

Funkcja zwraca, czy udało się dodać obiekt

Funkcja służy, do dodawania wpisów do lokalnej książki adresowej. Wideoterminal ma za zadanie sprawdzać, czy adresy oraz pary organizacja - nazwa są unikalne.



1.8.9.9 [DLLEXPORT](#) [_bool](#) [vc_localAddressbookSelectLike](#) ([Handle](#) * [h](#), [const char](#) * [organization](#), [const char](#) * [name](#), [const char](#) * [address](#), [uint32_t](#) * [responseSize](#))

Pobieranie i filtracja wpisów z lokalnej książki adresowej.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
in	<i>organization</i>	Filtr nazwy organizacji. Wykorzystuje składnię instrukcji LIKE z SQLite
in	<i>name</i>	Filtr nazwy terminala. Wykorzystuje składnię instrukcji LIKE z SQLite
in	<i>address</i>	Filtr adresu. Wykorzystuje składnię instrukcji LIKE z SQLite
out	<i>responseSize</i>	Rozmiar odpowiedzi

Zwraca:

Funkcja zwraca, czy udało się pobrać dane

Funkcja służy do pobierania oraz filtracji wpisów z lokalnej książki adresowej.

Gdy operacja się powiodła, wtedy pole `responseSize` zawiera rozmiar danych do pobrania. Należy to zrobić w funkcji [vc_getData\(\)](#). Dane to tablica rekordów, z których każdy zawiera:

- Id elementu - do późniejszego wykorzystania w [vc_localAddressbookDeleteID\(\)](#) - 4 bajty
- Tekst z nazwą organizacji
- Tekst z nazwą terminala
- Tekst z adresem

Dane należy odczytywać dopóki w buforze znajdują się jeszcze jakieś dane. Tekst to bajt rozmiaru, a następnie dane kodowane w UTF-8.

1.9 Status pracy Wideoterminala

Funkcje służące do pobierania statusu pracy Wideoterminala.

1.9.1 Funkcje

- [DLLEXPORT StatusHandle](#) * [vc_getStatusHandle](#) ([const char](#) *[ip](#), [uint16_t](#) [port](#))
Pobranie uchwytu do obiektu statusu Wideoterminala.
- [DLLEXPORT _bool](#) [vc_getTerminalStatus](#) ([StatusHandle](#) *[h](#))
Pobranie statusu pracy Wideoterminala.
- [DLLEXPORT _bool](#) [vc_isGatekeeperConnected](#) ([StatusHandle](#) *[h](#))
Sprawdzenie, czy Wideoterminal jest połączony do gatekeepera.

- [DLLEXPORT](#) [bool](#) [vc_getTerminalName](#) ([StatusHandle](#) *h, char *buffer, uint32_t &bufferSize)

Pobranie nazwy Wideoterminala w infrastrukturze wideokonferencyjnej.

- [DLLEXPORT](#) [void](#) [vc_destroyStatusHandle](#) ([StatusHandle](#) *h)

Usunięcie obiektu statusu Wideokonferencji.

1.9.2 Opis szczegółowy

Funkcje służące do pobierania statusu pracy Wideoterminala.

Wideoterminal udostępnia informacje o stanie działania niektórych komponentów. Możliwa do pobrania jest informacja o:

- Możliwości podłączenia się do Wideoterminala przez ReCourta ([vc_getTerminalStatus\(\)](#))
- Statusie połączenia z gatekeeperem ([vc_isGatekeeperConnected\(\)](#))
- Ustawionej w infrastrukturze wideokonferencyjnej nazwie Wideoterminala ([vc_getTerminalName\(\)](#))

Dane te są dostępne za pomocą interfejsu niezależnego z pozostałą częścią interfejsu. Aby uzyskać te informacje należy skorzystać z obiektu [StatusHandle](#)

1.9.3 Dokumentacja funkcji

1.9.3.1 [DLLEXPORT](#) [StatusHandle](#)* [vc_getStatusHandle](#) (const char * ip, uint16_t port)

Pobranie uchwytu do obiektu statusu Wideoterminala.

Parametry:

in	<i>ip</i>	Adres IP Wideoterminala
in	<i>port</i>	Numer portu serwera statusu Wideoterminala

Zwraca:

Funkcja zwraca status do obiektu Wideoterminala lub NULL w przypadku niepowodzenia

Funkcja umożliwia pobieranie informacji o statusie pracy Wideotermianla w przyszłości za pomocą pozostałych funkcji API statusu. Zwrócony w tej funkcji obiekt należy zwolnić za pomocą funkcji [vc_destroyStatusHandle\(\)](#)

1.9.3.2 [DLLEXPORT](#) [bool](#) [vc_getTerminalStatus](#) ([StatusHandle](#) * h)

Pobranie statusu pracy Wideoterminala.

Parametry:

in	<i>h</i>	Uchwyt do obiektu statusu Wideokonferencji
----	----------	--

Zwraca:

Funkcja zwraca, czy ReCourt jest w stanie się połączyć z Wideoterminalem



1.9.3.3 [DLLEXPORT](#) [_bool](#) [vc_isGatekeeperConnected](#) ([StatusHandle](#) * *h*)

Sprawdzenie, czy Wideoterminal jest połączony do gatekeepera.

Parametry:

in	<i>h</i>	Uchwyt do obiektu statusu Wideokonferencji
----	----------	--

Zwraca:

Funkcja zwraca, czy Wideoterminal jest połączony do gatekeepera

Za pomocą tej funkcji można otrzymać informację, czy Wideoterminal będzie działał z wykorzystaniem infrastruktury wideokonferencyjnej

1.9.3.4 [DLLEXPORT](#) [_bool](#) [vc_getTerminalName](#) ([StatusHandle](#) * *h*, [char](#) * *buffer*, [uint32_t](#) & *bufferSize*)

Pobranie nazwy Wideoterminala w infrastrukturze wideokonferencyjnej.

Parametry:

in	<i>h</i>	Uchwyt do obiektu statusu Wideokonferencji
in	<i>buffer</i>	Bufor do którego zostanie zapisana nazwa Wideoterminala
in,out	<i>bufferSize</i>	Jako parametr wejściowy oznacza przekazany rozmiar bufora, a jako parametr wyjściowy oznacza faktyczny rozmiar nazwy (w szczególności wartość ta może być większa niż rozmiar bufora)

Zwraca:

Funkcja zwraca, czy udało się pobrać nazwę

Za pomocą tej funkcji można sprawdzić nazwę, za pomocą której Wideoterminal przedstawia się w infrastrukturze wideokonferencyjnej

1.9.3.5 [DLLEXPORT](#) [void](#) [vc_destroyStatusHandle](#) ([StatusHandle](#) * *h*)

Usunięcie obiektu statusu Wideokonferencji.

Parametry:

in	<i>h</i>	Uchwyt do obiektu statusu Wideokonferencji
----	----------	--

Funkcja zwalnia wszystkie zasoby wykorzystywane przez przekazany obiekt statusu Wideoterminala

1.10 Inne

1.10.1 Definicje typów

- typedef void(__cdecl * [gatekeeperRegistrationStatusPtr](#)) (uint32_t status)



Informacja o zmianie statusu połączenia z gatekeeperem.

1.10.2 Funkcje

- [DLLEXPORT](#) const char * [vc_getCameraIP](#) ([Handle](#) *h)

Pobranie adresu IP kamery.

1.10.3 Opis szczegółowy

1.10.4 Dokumentacja definicji typów

1.10.4.1 typedef void(__cdecl * gatekeeperRegistrationStatusPtr) (uint32_t status)

Informacja o zmianie statusu połączenia z gatekeeperem.

Parametry:

<i>status</i>	Enumerator GatekeeperRegistrationStatus
---------------	---

Definicja w linii [378](#) pliku [conferenceInterface.h](#).

1.10.5 Dokumentacja funkcji

1.10.5.1 [DLLEXPORT](#) const char* [vc_getCameraIP](#) ([Handle](#) * h)

Pobranie adresu IP kamery.

Parametry:

in	<i>h</i>	Uchwyt do obiektu interfejsu
----	----------	------------------------------

Zwraca:

Funkcja zwraca adres IP kamery zdalnej

2 Dokumentacja struktur danych

2.1 Dokumentacja struktury C_Resolution

Rozdzielczość

```
#include <conferenceInterface.h>
```

2.1.1 Pola danych

- uint32_t [width](#)
- uint32_t [height](#)

2.1.2 Opis szczegółowy

Rozdzielczość

Definicja w linii [477](#) pliku [conferenceInterface.h](#).

2.1.3 Dokumentacja pól

2.1.3.1 uint32_t C_Resolution::width

Szerokość

Definicja w linii [478](#) pliku [conferenceInterface.h](#).



2.1.3.2 uint32_t C_Resolution::height

Wysokość

Definicja w linii [479](#) pliku [conferenceInterface.h](#).

2.1.3.3 Dokumentacja dla tej struktury została wygenerowana z pliku:

- [conferenceInterface.h](#)

2.2 Dokumentacja struktury Callbacks

Struktura przechowująca wszystkie wywołania zwrotne od Wideoterminala.

```
#include <conferenceInterface.h>
```

2.2.1 Pola danych

- [successPtr success](#)
- [incomingPtr incoming](#)
- [needsAcceptPtr needsAccept](#)
- [callClearedPtr callCleared](#)
- [rejectedIncomingConnectionPtr rejectedIncomingConnection](#)
- [gatekeeperRegistrationStatusPtr gatekeeperRegistrationStatus](#)
- [onStopReceivingDualStreamPtr onStopReceivingDualStream](#)
- [onStartReceivingDualStreamPtr onStartReceivingDualStream](#)
- [sourceEventPtr sourceEvent](#)
- [onClosedSourceMediaStreamPtr onClosedSourceMediaStream](#)
- [globalAddressbookRefreshSuccessPtr globalAddressbookRefreshSuccess](#)
- [globalAddressbookRefreshErrorPtr globalAddressbookRefreshError](#)
- [terminalDisconnectedPtr terminalDisconnected](#)

2.2.2 Opis szczegółowy

Struktura przechowująca wszystkie wywołania zwrotne od Wideoterminala.

Musi być utworzona przez aplikację i przekazana do interfejsu w funkcji [vc_register\(\)](#). Żadne z jej pól nie może być wskaźnikiem zerowym.

Definicja w linii [444](#) pliku [conferenceInterface.h](#).

2.2.3 Dokumentacja pól

2.2.3.1 [successPtr](#) Callbacks::success

Połączenie zostało rozpoczęte

Zobacz również:

[successPtr](#)

Definicja w linii [445](#) pliku [conferenceInterface.h](#).



2.2.3.2 [incomingPtr](#) Callbacks::incoming

Rozpoczęto odbieranie połączenia

Zobacz również:

[incomingPtr](#)

Definicja w linii [446](#) pliku [conferenceInterface.h](#).

2.2.3.3 [needsAcceptPtr](#) Callbacks::needsAccept

Aktualne połączenie przychodzące wymaga ręcznego odebrania

Zobacz również:

[needsAcceptPtr](#)

Definicja w linii [447](#) pliku [conferenceInterface.h](#).

2.2.3.4 [callClearedPtr](#) Callbacks::callCleared

Połączenie zostało zakończone

Zobacz również:

[callClearedPtr](#)

Definicja w linii [448](#) pliku [conferenceInterface.h](#).

2.2.3.5 [rejectedIncomingConnectionPtr](#) Callbacks::rejectedIncomingConnection

Połączenie zostało odrzucone w sposób automatyczny

Zobacz również:

[rejectedIncomingConnectionPtr](#)

Definicja w linii [449](#) pliku [conferenceInterface.h](#).

2.2.3.6 [gatekeeperRegistrationStatusPtr](#) Callbacks::gatekeeperRegistrationStatus

Zmienił się stan połączenia z gatekeeperem

Zobacz również:

[gatekeeperRegistrationStatusPtr](#)

Definicja w linii [450](#) pliku [conferenceInterface.h](#).

2.2.3.7 [onStopReceivingDualStreamPtr](#) Callbacks::onStopReceivingDualStream

Zakończono odbieranie drugiego strumienia

Zobacz również:

[onStopReceivingDualStreamPtr](#)

Definicja w linii [451](#) pliku [conferenceInterface.h](#).



2.2.3.8 [onStartReceivingDualStreamPtr](#) Callbacks::onStartReceivingDualStream

Rozpoczęto odbieranie drugiego strumienia

Zobacz również:

[onStartReceivingDualStreamPtr](#)

Definicja w linii [452](#) pliku [conferenceInterface.h](#).

2.2.3.9 [sourceEventPtr](#) Callbacks::sourceEvent

Zmiana stanu aktywności źródła.

Zobacz również:

[sourceEventPtr](#)

Definicja w linii [453](#) pliku [conferenceInterface.h](#).

2.2.3.10 [onClosedSourceMediaStreamPtr](#) Callbacks::onClosedSourceMediaStream

Zamknięto strumień wyjściowy.

Zobacz również:

[onClosedSourceMediaStreamPtr](#)

Definicja w linii [454](#) pliku [conferenceInterface.h](#).

2.2.3.11 [globalAddressbookRefreshSuccessPtr](#) Callbacks::globalAddressbookRefreshSuccess

Odświeżono globalną książkę adresową.

Zobacz również:

[globalAddressbookRefreshSuccessPtr](#)

Definicja w linii [455](#) pliku [conferenceInterface.h](#).

2.2.3.12 [globalAddressbookRefreshErrorPtr](#) Callbacks::globalAddressbookRefreshError

Nie udało się odświeżyć globalnej książki adresowej.

Zobacz również:

[globalAddressbookRefreshErrorPtr](#)

Definicja w linii [456](#) pliku [conferenceInterface.h](#).

2.2.3.13 [terminalDisconnectedPtr](#) Callbacks::terminalDisconnected

Utracono połączenie z Wideoterminalem

Zobacz również:

[terminalDisconnectedPtr](#)

Definicja w linii [457](#) pliku [conferenceInterface.h](#).



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



2.2.3.14 Dokumentacja dla tej struktury została wygenerowana z pliku:

- [conferenceInterface.h](#)



2.3 Dokumentacja struktury Handle

Uchwyt do obiektu Wideokonferencji.

```
#include <conferenceInterface.h>
```

2.3.1 Opis szczegółowy

Uchwyt do obiektu Wideokonferencji.

W każdym momencie działania programu może występować co najwyżej jeden obiekt tego typu. Można go otrzymać w funkcji [vc_getHandle\(\)](#) oraz należy go zwolnić w funkcji [vc_cleanup\(\)](#)

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [conferenceInterface.h](#)

2.4 Dokumentacja struktury SourceDescriptor

Opis źródła.

```
#include <conferenceInterface.h>
```

2.4.1 Pola danych

- char [name](#) [32]
- uint32_t [code](#)
- [bool](#) [active](#)

2.4.2 Opis szczegółowy

Opis źródła.

Definicja w linii [466](#) pliku [conferenceInterface.h](#).

2.4.3 Dokumentacja pól

2.4.3.1 char SourceDescriptor::name[32]

Nazwa czytelna dla użytkownika.

Definicja w linii [467](#) pliku [conferenceInterface.h](#).

2.4.3.2 uint32_t SourceDescriptor::code

Kod źródła. Jest on globalnie unikalny.

Definicja w linii [468](#) pliku [conferenceInterface.h](#).

2.4.3.3 [bool](#) SourceDescriptor::active

Informacja, czy źródło jest aktualnie aktywne.

Definicja w linii [469](#) pliku [conferenceInterface.h](#).

2.4.3.4 Dokumentacja dla tej struktury została wygenerowana z pliku:

- [conferenceInterface.h](#)



2.5 Dokumentacja struktury StatusHandle

Uchwyt do obiektu statusu Wideokonferencji.

```
#include <conferenceInterface.h>
```

2.5.1 Opis szczegółowy

Uchwyt do obiektu statusu Wideokonferencji.

Obiekt służący do pobierania statusu działania Wideoterminala. Tworzony za pomocą [vc_getStatusHandle\(\)](#) oraz zwalniany za pomocą [vc_destroyStatusHandle\(\)](#) Można tworzyć wiele obiektów tego typu i mogą działać niezależnie od struktury [Handle](#)

Definicja w linii [56](#) pliku [conferenceInterface.h](#).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [conferenceInterface.h](#)

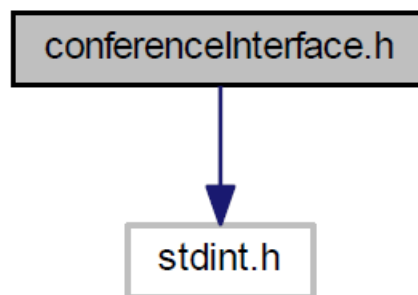
3 Dokumentacja plików

3.1 Dokumentacja pliku conferenceInterface.h

Plik zawierający publiczne API dostępu do terminala wideokonferencyjnego.

```
#include <stdint.h>
```

Wykres zależności załączania dla conferenceInterface.h:



3.1.1 Struktury danych

- struct [Callbacks](#)
- *Struktura przechowująca wszystkie wywołania zwrotne od Wideoterminala.* struct [SourceDescriptor](#)
- *Opis źródła.* struct [C_Resolution](#)



3.1.2 Rozdzielczość Definicje

- #define [DLLEXPORT](#) __declspec(dllexport)

Makro eksportujące funkcje API w bibliotece dll.

3.1.3 Definicje typów

- typedef uint32_t [bool](#)

Uściślenie typu bool.

- typedef void(__cdecl * [successPtr](#)) ()

Informacja o pomyślnym rozpoczęciu połączenia.

- typedef void(__cdecl * [incomingPtr](#)) (const char *ip, [bool](#) needsAccept)

Informacja o połączeniu przychodzącym.

- typedef void(__cdecl * [needsAcceptPtr](#)) ()

Informacja, że wymagane jest ręczne odebranie połączenia.

- typedef void(__cdecl * [callClearedPtr](#)) (uint8_t reason)

Informacja, że połączenie zostało zakończone.

- typedef void(__cdecl * [rejectedIncomingConnectionPtr](#)) (const char *address)

Informacja, że pewne połączenie przychodzące zostało w sposób automatyczny odrzucone.

- typedef void(__cdecl * [gatekeeperRegistrationStatusPtr](#)) (uint32_t status)

Informacja o zmianie statusu połączenia z gatekeeperem.

- typedef void(__cdecl * [onStopReceivingDualStreamPtr](#)) (uint32_t reason)

Informacja, że Wideoteminal zakończył odbieranie prezentacji.

- typedef void(__cdecl * [onStartReceivingDualStreamPtr](#)) ()

Informacja, że Wideoteminal zaczął odbierać drugi strumień

- typedef void(__cdecl * [globalAddressbookRefreshSuccessPtr](#)) ()

Globalna książka adresowa została z powodzeniem odświeżona.

- typedef void(__cdecl * [globalAddressbookRefreshErrorPtr](#)) ()

Nie udało się odświeżyć globalnej książki adresowej.

- typedef void(__cdecl * [sourceEventPtr](#)) (uint32_t code, uint8_t event)

Informacja, że mógł się zmienić stan źródła.

- typedef void(__cdecl * [onClosedSourceMediaStreamPtr](#)) (uint8_t type)

Informacja, że został zamknięty strumień wyjściowy.

- typedef void(__cdecl * [terminalDisconnectedPtr](#)) ()

Wywołanie otrzymane w przypadku rozłączenia Wideoterminala.

3.1.4 Wyliczenia

- enum [RegisterStatus](#) { [SUCCESS](#), [TERMINAL USED](#), [VERSION MISMATCH](#), [NULL CALLBACKS](#) }Enumeracja możliwych efektów rejestracji.

- enum [CallEndReason](#) { [NORMAL](#), [OTHER SIDE ENDED](#), [OTHER SIDE REFUSED](#), [OTHER SIDE UNRESPONSIVE](#), [NO OTHER SIDE](#), [GATEKEEPER ENDED](#), [MEDIA TRANSPORT FAIL](#), [GATEKEEPER NO USER](#), [NO COMMON MEDIA FORMATS](#), [SECURITY DENIED](#), [OTHER](#) }Enumeracja ze wszystkimi możliwymi powodami zakończenia połączenia.



- enum [Protocol](#) { [H323](#), [SIP](#) }Protokoły sygnalizacyjne.
- enum [AnswerType](#) { [MANUAL](#), [AUTO](#), [REJECT](#) }Tryby odbierania połączenia.
- enum [StreamType](#) { [AUDIO](#), [VIDEO](#), [PRESENTATION](#) }Wszystkie rodzaje strumieni wyjściowych.
- enum [SourceEvent](#) { [ACTIVE](#) = 1, [INACTIVE](#) }Aktywność źródła.
- enum [ConnectionType](#) { [Received](#), [Made](#), [RejectedByListener](#), [IncomingRejected](#), [All](#) }Rodzaje połączeń zapisywane w historii połączeń
- enum [SourceType](#) { [AUDIO_SOURCE](#), [VIDEO_SOURCE](#), [DUALSTREAM_SOURCE](#) }Rodzaje strumieni.
- enum [EndDualStreamReason](#) { [eEndedByLocalUser](#) = 0, [eEndedByRemoteUser](#) = 1, [eEndedByError](#) = 2 }Powody zakończenia drugiego strumienia.
- enum [GatekeeperRegistrationStatus](#) { [REGISTERED](#), [SECURITY_FAIL](#), [OTHER_ERROR](#), [UNREGISTERED](#) }Status połączenia z gatekeeperem.

3.1.5 Funkcje

- [DLLEXPORT](#) [uint32_t](#) [vc_getApiVersion](#) ()
Funkcja zwraca wersję API używaną przez moduł
- [DLLEXPORT](#) [Handle](#) * [vc_getHandle](#) ()
Funkcja służąca do otrzymania uchwytu do obiektu Wideoterminala.
- [DLLEXPORT](#) void [vc_cleanup](#) ([Handle](#) *h)
Zamknięcie interfejsu.
- [DLLEXPORT](#) [bool](#) [vc_init](#) ([Handle](#) *handle, const char *terminalIP, [uint32_t](#) terminalPort)
Inicjalizowanie interfejsu.
- [DLLEXPORT](#) void [vc_getAvailableResolutions](#) ([Handle](#) *h, [C_Resolution](#) *resolutions, [uint32_t](#) *resolutionsSize)
Funkcja pobierające możliwe do ustawienia rozdzielczości maksymalne.
- [DLLEXPORT](#) [bool](#) [vc_setResolution](#) ([Handle](#) *h, [C_Resolution](#) resolution)
Funkcja ustawiająca rozdzielczość maksymalną.
- [DLLEXPORT](#) [uint32_t](#) [vc_register](#) ([Handle](#) *h, [Callbacks](#) *callbacks)
Funkcja rejestrująca interfejs w Wideoterminalu.
- [DLLEXPORT](#) [bool](#) [vc_call](#) ([Handle](#) *h, const char *ip, [uint32_t](#) protocol)
Wykonanie połączenia.
- [DLLEXPORT](#) [bool](#) [vc_acceptConnection](#) ([Handle](#) *h, [uint32_t](#) isAccept)
Manualne odebranie połączenia.
- [DLLEXPORT](#) [uint8_t](#) [vc_getAnswerType](#) ([Handle](#) *h)
Pobranie trybu odbierania połączenia.
- [DLLEXPORT](#) void [vc_setAnswerType](#) ([Handle](#) *h, [uint8_t](#) answerType)
Ustawienie trybu odbierania połączenia.
- [DLLEXPORT](#) void [vc_hangupCall](#) ([Handle](#) *h)
Zakończenie połączenia.
- [DLLEXPORT](#) [bool](#) [vc_sendDTMF](#) ([Handle](#) *h, char sign)



Wysłanie DTMF.

- [DLLEXPORT](#) const char * [vc_getCameraIP](#) ([Handle](#) *h)
Pobranie adresu IP kamery.
- [DLLEXPORT](#) void [vc_getAvailableSources](#) ([Handle](#) *h, uint32_t type, [SourceDescriptor](#) *descriptors, uint8_t *descriptorsSize)
Pobranie listy dostępnych źródeł
- [DLLEXPORT](#) [bool](#) [vc_setSource](#) ([Handle](#) *h, uint8_t type, uint32_t code)
Ustawienie źródła dla strumienia.
- [DLLEXPORT](#) [bool](#) [vc_getRemotePartyName](#) ([Handle](#) *h, char *name, uint32_t *size)
Pobranie nazwy strony zdalnej.
- [DLLEXPORT](#) void [vc_getData](#) ([Handle](#) *h, void *data)
Pobieranie danych otrzymanych w wyniku ostatniego zapytania do bazy.
- [DLLEXPORT](#) [bool](#) [vc_globalAddressbookRefresh](#) ([Handle](#) *h)
Odświeżenie globalnej książki adresowej.
- [DLLEXPORT](#) [bool](#) [vc_dictionarySelectLike](#) ([Handle](#) *h, const char *courtType, const char *courtPrefix, uint32_t *responseSize)
Pobranie danych słownika tłumaczącego numery E164.
- [DLLEXPORT](#) [bool](#) [vc_globalAddressbookSelectLike](#) ([Handle](#) *h, const char *search, const char *number, [bool](#) active, uint32_t *responseSize)
Pobieranie danych z globalnej książki adresowej.
- [DLLEXPORT](#) [bool](#) [vc_historyDeleteAll](#) ([Handle](#) *h)
Usunięcie wszystkich wpisów z historii połączeń
- [DLLEXPORT](#) [bool](#) [vc_historyDeleteID](#) ([Handle](#) *h, uint32_t ID)
Usunięcie pojedynczego wpisu z historii połączeń
- [DLLEXPORT](#) [bool](#) [vc_historySelectMode](#) ([Handle](#) *h, char mode, uint32_t *responseSize)
Pobranie wpisów z historii połączeń
- [DLLEXPORT](#) [bool](#) [vc_localAddressbookDeleteID](#) ([Handle](#) *h, uint32_t ID)
Usuwanie wpisu z lokalnej książki adresowej.
- [DLLEXPORT](#) [bool](#) [vc_localAddressbookInsert](#) ([Handle](#) *h, const char *organization, const char *name, const char *address)
Dodawanie wpisu do lokalnej książki adresowej.
- [DLLEXPORT](#) [bool](#) [vc_localAddressbookSelectLike](#) ([Handle](#) *h, const char *organization, const char *name, const char *address, uint32_t *responseSize)
Pobieranie i filtracja wpisów z lokalnej książki adresowej.
- [DLLEXPORT](#) [StatusHandle](#) * [vc_getStatusHandle](#) (const char *ip, uint16_t port)
Pobranie uchwytu do obiektu statusu Wideoterminala.
- [DLLEXPORT](#) [bool](#) [vc_getTerminalStatus](#) ([StatusHandle](#) *h)
Pobranie statusu pracy Wideoterminala.
- [DLLEXPORT](#) [bool](#) [vc_isGatekeeperConnected](#) ([StatusHandle](#) *h)
Sprawdzenie, czy Wideoterminal jest połączony do gatekeepera.



- [DLLEXPORT](#) [bool](#) [vc_getTerminalName](#) ([StatusHandle](#) *h, char *buffer, uint32_t &bufferSize)

Pobranie nazwy Wideoterminala w infrastrukturze wideokonferencyjnej.

- [DLLEXPORT](#) [void](#) [vc_destroyStatusHandle](#) ([StatusHandle](#) *h)

Usunięcie obiektu statusu Wideokonferencji.

3.1.6 Zmienne

- `const uint32_t VIDEOTERMINAL_API_VERSION = 3`

Zmienna określająca wersję API.

3.1.7 Opis szczegółowy

Plik zawierający publiczne API dostępu do termiala wideokonferencyjnego.

Definicja w pliku [conferenceInterface.h](#).

3.2 conferenceInterface.h

```
1 #pragma once
2 #include <stdint.h>
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 const uint32_t VIDEOTERMINAL\_API\_VERSION = 3;
28
29
30
31
32
33
34 #define DLLEXPORT __declspec(dllexport)
35
36
37
38
39
40
41
42
43
44
45 typedef uint32_t bool;
46
47
48
49
50
51
52
53
54
55
56 struct Handle;
57
58
59
60
61
62
63
64
65
66
67 struct StatusHandle;
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125 enum RegisterStatus {
126     SUCCESS,
127     TERMINAL\_USED,
128     VERSION\_MISMATCH,
129     NULL\_CALLBACKS
130 };
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228 enum CallEndReason {
229     NORMAL,
230     OTHER\_SIDE\_ENDED,
231     OTHER\_SIDE\_REFUSED,
232     OTHER\_SIDE\_UNRESPONSIVE,
233     NO\_OTHER\_SIDE,
234     GATEKEEPER\_ENDED,
235     MEDIA\_TRANSPORT\_FAIL,
236     GATEKEEPER\_NO\_USER,
237     NO\_COMMON\_MEDIA\_FORMATS,
238     SECURITY\_DENIED,
239     OTHER
240 };
241
242
243
244
245
246
247
248
249 enum Protocol {
250     H323,
251     SIP
252 };
253
254
255
256
257
258 enum AnswerType {
259     MANUAL,
260     AUTO,
261     REJECT
262 };
263
264
265
266
267
268 enum StreamType {
269     AUDIO,
```



```
270     VIDEO,  
271     PRESENTATION  
272 };  
273  
274 enum SourceEvent {  
275     ACTIVE = 1,  
276     INACTIVE  
277 };  
278  
279 enum ConnectionType {  
280     Received,  
281     Made,  
282     RejectedByListener,  
283     IncomingRejected,  
284     All  
285 };  
286  
287 enum SourceType {  
288     AUDIO_SOURCE,  
289     VIDEO_SOURCE,  
290     DUALSTREAM_SOURCE  
291 };  
292  
293 enum EndDualStreamReason {  
294     eEndedByLocalUser = 0,  
295     eEndedByRemoteUser = 1,  
296     eEndedByError = 2  
297 };  
298  
299 enum GatekeeperRegistrationStatus {  
300     REGISTERED,  
301     SECURITY_FAIL,  
302     OTHER_ERROR,  
303     UNREGISTERED  
304 };  
305  
306 typedef void( cdecl *successPtr) ();  
307  
308 typedef void(__cdecl *incomingPtr)(const char* ip, _bool needsAccept);  
309  
310 typedef void( cdecl *needsAcceptPtr) ();  
311  
312 typedef void( cdecl *callClearedPtr)(uint8_t reason);  
313  
314 typedef void(__cdecl *rejectedIncomingConnectionPtr)(const char* address);  
315  
316 typedef void( cdecl *gatekeeperRegistrationStatusPtr)(uint32_t status);  
317  
318 typedef void(__cdecl *onStopReceivingDualStreamPtr)(uint32_t reason);  
319  
320 typedef void(__cdecl *onStartReceivingDualStreamPtr) ();  
321  
322 typedef void(__cdecl *globalAddressbookRefreshSuccessPtr) ();  
323  
324 typedef void(__cdecl *globalAddressbookRefreshErrorPtr) ();  
325  
326 typedef void(__cdecl *sourceEventPtr)(uint32_t code, uint8_t event);  
327  
328 typedef void(__cdecl *onClosedSourceMediaStreamPtr)(uint8_t type);  
329  
330 typedef void(__cdecl *terminalDisconnectedPtr) ();  
331  
332 struct Callbacks {  
333     successPtr success;  
334     incomingPtr incoming;  
335     needsAcceptPtr needsAccept;  
336     callClearedPtr callCleared;  
337     rejectedIncomingConnectionPtr rejectedIncomingConnection;  
338     gatekeeperRegistrationStatusPtr gatekeeperRegistrationStatus;  
339 }
```



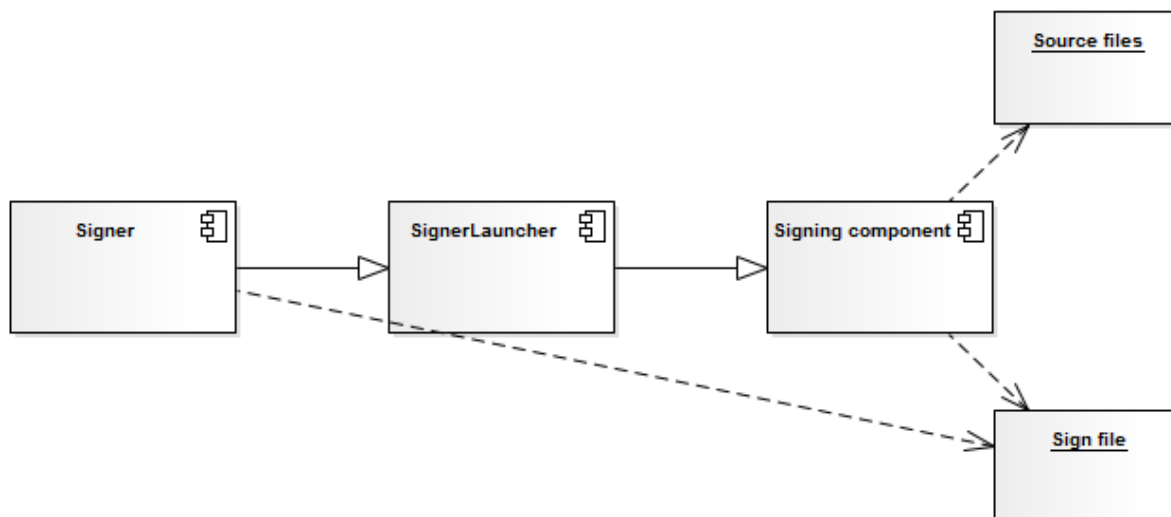
```
451 onStopReceivingDualStreamPtr onStopReceivingDualStream;  
452 onStartReceivingDualStreamPtr onStartReceivingDualStream;  
453 sourceEventPtr sourceEvent;  
454 onClosedSourceMediaStreamPtr onClosedSourceMediaStream;  
455 globalAddressbookRefreshSuccessPtr globalAddressbookRefreshSuccess;  
456 globalAddressbookRefreshErrorPtr globalAddressbookRefreshError;  
457 terminalDisconnectedPtr terminalDisconnected;  
458 };  
459  
460 #pragma pack(4)  
461  
462 struct SourceDescriptor {  
463     char name[32];  
464     uint32_t code;  
465     bool active;  
466 };  
467  
468 struct C\_Resolution {  
469     uint32_t width;  
470     uint32_t height;  
471 };  
472  
473 #pragma pack()  
474  
475 extern "C" { //wszystkie funkcje w tym bloku sa wywoływane w konwencji cdecl  
476  
477 DLLEXPORT uint32_t vc\_getApiVersion();  
478  
479 DLLEXPORT Handle *vc\_getHandle();  
480  
481 DLLEXPORT void vc\_cleanup(Handle* h);  
482  
483 DLLEXPORT bool vc\_init(Handle *handle, const char* terminalIP, uint32_t terminalPort);  
484  
485 DLLEXPORT void vc\_getAvailableResolutions(Handle *h, C\_Resolution *resolutions, uint32_t  
486 *resolutionsSize);  
487  
488 DLLEXPORT bool vc\_setResolution(Handle *h, C\_Resolution resolution);  
489  
490 DLLEXPORT uint32_t vc\_register(Handle *h, Callbacks *callbacks);  
491  
492 DLLEXPORT bool vc\_call(Handle* h, const char* ip, uint32_t protocol);  
493  
494 DLLEXPORT bool vc\_acceptConnection(Handle* h, uint32_t isAccept);  
495  
496 DLLEXPORT uint8_t vc\_getAnswerType(Handle *h);  
497  
498 DLLEXPORT void vc\_setAnswerType(Handle *h, uint8_t answerType);  
499  
500 DLLEXPORT void vc\_hangupCall(Handle* h);  
501  
502 DLLEXPORT bool vc\_sendDTMF(Handle* h, char sign);  
503  
504 DLLEXPORT const char *vc\_getCameraIP(Handle *h);  
505  
506 DLLEXPORT void vc\_getAvailableSources(Handle *h, uint32_t type, SourceDescriptor  
507 *descriptors, uint8_t *descriptorsSize);  
508  
509 DLLEXPORT bool vc\_setSource(Handle *h, uint8_t type, uint32_t code);  
510  
511 DLLEXPORT bool vc\_getRemotePartyName(Handle *h, char *name, uint32_t *size);  
512  
513 //database interface  
514  
515 DLLEXPORT void vc\_getData(Handle* h, void * data);  
516  
517 DLLEXPORT bool vc\_globalAddressbookRefresh(Handle* h);  
518  
519
```



```
728 DLLEXPORT _bool vc_dictionarySelectLike(Handle* h, const char* courtType, const char*
courtPrefix, uint32_t* responseSize);
729
763 DLLEXPORT _bool vc_globalAddressbookSelectLike(Handle* h, const char* search, const char*
number, _bool active, uint32_t* responseSize);
764
772 DLLEXPORT _bool vc_historyDeleteAll(Handle* h);
773
782 DLLEXPORT _bool vc_historyDeleteID(Handle* h, uint32_t ID);
783
808 DLLEXPORT _bool vc_historySelectMode(Handle* h, char mode, uint32_t* responseSize);
809
820 DLLEXPORT _bool vc_localAddressbookDeleteID(Handle* h, uint32_t ID);
821
834 DLLEXPORT _bool vc_localAddressbookInsert(Handle* h, const char* organization, const char*
name, const char* address);
835
859 DLLEXPORT _bool vc_localAddressbookSelectLike(Handle* h, const char* organization, const
char* name, const char* address, uint32_t* responseSize);
860
873 DLLEXPORT StatusHandle *vc_getStatusHandle(const char *ip, uint16_t port);
874
882 DLLEXPORT _bool vc_getTerminalStatus(StatusHandle *h);
883
893 DLLEXPORT _bool vc_isGatekeeperConnected(StatusHandle *h);
894
906 DLLEXPORT _bool vc_getTerminalName(StatusHandle *h, char *buffer, uint32_t &bufferSize);
907
916 DLLEXPORT void vc_destroyStatusHandle(StatusHandle *h);
917
918 }
919
920 #undef DLLEXPORT
```

III. Interfejs komponentu podpisu

Podpis jako zewnętrzny komponent lub aplikacja nie ma ścisłego połączenia z modułami ReCourt. Do współpracy wykorzystuje się oprogramowanie „mostka”, pośredniczące pomiędzy modułami ReCourt, a samym komponentem/aplikacją podpisu elektronicznego.



Rysunek 1 Interfejs komponentu podpisu

W celu integracji nowego komponentu podpisu należy przygotować aplikację mostka (SignerLauncher.exe) zgodnie z poniższymi wytycznymi.

Wymagania dotyczące wykonania podpisu elektronicznego:

- podpisanie pliku udostępnionego lokalnie na dysku
- podpisanie pliku udostępnionego protokołem http (<http://domena/> lub <http://adresIP/>)
- przekazanie do komponentu listy (i ścieżek) plików do podpisania
- utworzenie pliku podpisu z rozszerzeniem .XAdES bez znakowania czasem i udostępnienie go we wskazanej lokalizacji na dysku (pliki wyjściowe wg nazw przekazanych plików)

Komponent mostka podpisu jest aplikacją konsolową przyjmującą parametry wg określonego schematu, (które są zdefiniowane jako jeden wspólny dla wszystkich komponentów). Aplikacja ta, musi mieć nazwę SignerLauncher.exe. Umieszczona jest w odpowiednim podkatalogu (nazwa bez spacji, tylko litery i cyfry z podstawowego alfabetu) w podkatalogu „signer” w głównym katalogu aplikacji modułu ReCourt.

SignerLauncher powinien obsługiwać następujące parametry:

- f – nazwy źródłowe rozdzielone przecinkiem
- d – katalog wyjściowy – miejsce utworzenia plików podpisu

3.1 Przykład zdefiniowanego parametru dla komponentu:

np: -f:%In1,%In2 -d:%Out

gdzie mogą wystąpić



%In1 – to przekazana pełna ścieżka z nazwą plik ogg (znaki rozdzielające foldery to „/”)

%In2 – to przekazana pełna ścieżka z nazwą pliku oga (znaki rozdzielające foldery to „/”)

%Out – to wskazana ścieżka wyjściowa, gdzie będą umieszczone pliki *.XAdES (z końcowym znakiem „/”)

%in1 – to przekazana tylko nazwa pliku ogg

%in2 – to przekazana tylko nazwa pliku oga

%out – to wskazana ścieżka wyjściowa, gdzie będą umieszczone pliki *.XAdES (bez końcowego znaku „/”)

Po zakończeniu działania zwraca kod ExitCode, wg:

=8 – podpis wykonany prawidłowo

<>8 – wystąpił błąd wykonania podpisu

3.2 Przykładowe wywołanie:

```
SignerLauncher.exe -f:c:/dir/fileA.ogg,c:/dir/fileB.oga -d:c:/dir/
```

Wyjście:

```
C:/dir/fileA.ogg.XAdES
```

```
C:/dir/fileB.oga.XAdES
```

ExitCode = 8

Lub

```
SignerLauncher.exe -f:http://127.0.0.1/fileA.ogg,http://127.0.0.1/fileB.oga -d:c:/dir/
```

Wyjście:

```
C:/dir/fileA.ogg.XAdES
```

```
C:/dir/fileB.oga.XAdES
```

ExitCode = 8



IV. Interfejsy RCS dla systemów i programów zewnętrznych

- 1 Interfejsy zewnętrzne RCS
- 2 Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base

2.1 Interfejsy

	Interfejs	Opis
→	ICourtRepositoryCommunicationService	Interfejs odpowiada za zarządzanie salami sądowymi. Wchodzi w niego skład usuwanie, ustawianie, pobieranie sal, zasobów oraz wyróżników.

3 Interfejs ICourtRepositoryCommunicationService

Interfejs odpowiada za zarządzanie salami sądowymi. Wchodzi w niego skład usuwanie, ustawianie, pobieranie sal, zasobów oraz wyróżników.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
 Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
 ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)













3.1 Składnia

C#










```
public interface ICourtRepositoryCommunicationService
```

ICourtRepositoryCommunicationService prezentuje składowe.

3.2 Metody

	Nazwa	Opis
	AddDiscriminant	Dodaje nowy wyróżnik.
	AddStorage	Dodaje zasób.
	CheckTrialsExistsForDiscriminant	Sprawdza czy istnieją jakieś sprawy dla podanego wyróżnika wydziału.
	GetCourtDepartments	Zwraca listę wydziałów.
	GetCourtRoomNames	Zwraca listę nazw sal sądowych.
	GetCourtRoomsVO	Zwraca listę sali sądowych.
	GetCourts	Zwraca listę sądów.
	GetDepartments	Zwraca listę wydziałów dla danych kodów wyróżników
	GetDiscriminantByCode	Zwraca wyróżnik określony kodem.
	GetDiscriminantById	Pobiera wyróżnik określony przez identyfikator.
	GetDiscriminants	Zwraca listę wyróżników.
	GetDiscriminantsVO	Zwraca listę wyróżników.



 GetDiscriminantsVOPagger	Zwraca stronicowaną listę wyróżników.
 GetStorages	Zwraca listę zasobów.
 GetStoragesByType	Zwraca zasoby określonego typu.
 GetStoragesPagger	Zwraca stronicowaną listę zasobów.
 RemoveCourtRoom	Usuwa określoną salę sądową.
 RemoveDiscriminant	Usuwa wyróżnik określony podanym identyfikatorem.
 RemoveStorage	Usuwa określony identyfikatorem zasób.
 SetCourtRoomsVO	Ustawia listę sali sądowych.
 UpdateDiscriminant	Aktualizuje wyróżnik.
 UpdateStorage	Aktualizuje zasób.

3.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

4 Metody






ICourtRepositoryCommunicationService.ICourtRepositoryCommunicationService

[ICourtRepositoryCommunicationService](#) prezentuje składowe.

4.1 Metody

Nazwa	Opis
AddDiscriminant	Dodaje nowy wyróżnik.
AddStorage	Dodaje zasób.
CheckTrialsExistsForDiscriminant	Sprawdza czy istnieją jakieś sprawy dla podanego wyróżnika wydziału.
GetCourtDepartments	Zwraca listę wydziałów.
GetCourtRoomNames	Zwraca listę nazw sal sądowych.
GetCourtRoomsVO	Zwraca listę sali sądowych.
GetCourts	Zwraca listę sądów.
GetDepartments	Zwraca listę wydziałów dla danych kodów wyróżników
GetDiscriminantByCode	Zwraca wyróżnik określony kodem.
GetDiscriminantById	Pobiera wyróżnik określony przez identyfikator.
GetDiscriminants	Zwraca listę wyróżników.
GetDiscriminantsVO	Zwraca listę wyróżników.
GetDiscriminantsVOPagger	Zwraca stronicowaną listę wyróżników.
GetStorages	Zwraca listę zasobów.
GetStoragesByType	Zwraca zasoby określonego typu.
GetStoragesPagger	Zwraca stronicowaną listę zasobów.
RemoveCourtRoom	Usuwa określoną salę sądową.



 RemoveDiscriminant	Usuwa wyróżnik określony podanym identyfikatorem.
 RemoveStorage	Usuwa określony identyfikatorem zasób.
 SetCourtRoomsVO	Ustawia listę sali sądowych.
 UpdateDiscriminant	Aktualizuje wyróżnik.
 UpdateStorage	Aktualizuje zasób.

4.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

5 Metoda ICourtRepositoryCommunicationService.AddDiscriminant

Dodaje nowy wyróżnik.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

5.1 Składnia

```
C#  
  
int AddDiscriminant (  
    ReCourtDiscriminant discriminant  
)
```

5.1.1.1 Parametry

discriminant

Typ: **ReCourtDiscriminant**

Wyróżnik

5.1.1.2 Zwraca

Typ:

Int32

Identyfikator nowego wyróżnika.

5.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



6 Metoda ICourtRepositoryCommunicationService.AddStorage

Dodaje zasób.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

6.1 Składnia

C#	
<pre>int ReCourtStorage)</pre>	<pre>AddStorage(storage</pre>

6.1.1.1 Parametry

storage

Typ: **ReCourtStorage**

Zasób

6.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanego zasobu.

6.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

7 Metoda

ICourtRepositoryCommunicationService.CheckTrialsExistsForDiscriminant

Sprawdza czy istnieją jakieś sprawy dla podanego wyróżnika wydziału.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

7.1 Składnia

```
C#  
  
bool          CheckTrialsExistsForDiscriminant(  
    int        discriminantId  
)
```

7.1.1.1 Parametry

discriminantId

Typ: System.Int32

Id wyróżnika

7.1.1.2 Zwraca

Typ:

Boolean

True jeśli sprawy istnieją, w przeciwnym razie false.

7.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



8 Metoda ICourtRepositoryCommunicationService.GetCourtDepartments

Zwraca listę wydziałów.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

8.1 Składnia

C#

```
List<ReCourtCourtDepartment> GetCourtDepartments ()
```

8.1.1.1 Zwraca

Typ:

List(ReCourtCourtDepartment)

Lista sali sądowych.

8.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



9 Metoda ICourtRepositoryCommunicationService.GetCourtRoomNames

Zwraca listę nazw sal sądowych.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

9.1 Składnia

C#

```
List<string> GetCourtRoomNames ()
```

9.1.1.1 Zwraca

Typ:

List(String)

Lista nazw sal sądowych.

9.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



10 Metoda ICourtRepositoryCommunicationService.GetCourtRoomsVO

Zwraca listę sali sądowych.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

10.1 Składnia

C#

```
List<ReCourtCourtRoomVO> GetCourtRoomsVO ()
```

10.1.1.1 Zwraca

Typ: **List(ReCourtCourtRoomVO)**
Lista sali sądowych.

10.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



11 Metoda ICourtRepositoryCommunicationService.GetCourts

Zwraca listę sądów.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

11.1 Składnia

C#

```
List<ReCourtCourt> GetCourts ()
```

11.1.1.1 Zwraca

Typ:
Lista sali sądowych.

List(ReCourtCourt)

11.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



12 Metoda ICourtRepositoryCommunicationService.GetDepartments

Zwraca listę wydziałów dla danych kodów wyróżników

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

12.1 Składnia

C#	
<pre>List<ReCourtCourtDepartment> IEnumerable<string>)</pre>	<pre>GetDepartments (<i>discriminantCodes</i>)</pre>

12.1.1.1 Parametry

discriminantCodes

Typ: System.Collections.Generic.IEnumerable(**String**)

Kolekcja z kodami wyróżników

12.1.1.2 Zwraca

Typ:

List(ReCourtCourtDepartment)

Lista wydziałów

12.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



13 Metoda ICourtRepositoryCommunicationService.GetDiscriminantByCode

Zwraca wyróżnik określony kodem.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

13.1 Składnia

C#	
ReCourtDiscriminant string)	GetDiscriminantByCode (<i>code</i>

13.1.1.1 Parametry

code

Typ: System.String

Kod

13.1.1.2 Zwraca

Typ:

ReCourtDiscriminant

Obiekt wyróżnika.

13.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

14 Metoda ICourtRepositoryCommunicationService.GetDiscriminantById

Pobiera wyróżnik określony przez identyfikator.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

14.1 Składnia

C#	
ReCourtDiscriminant int)	GetDiscriminantById(id

14.1.1.1 Parametry

id

Typ: System.Int32

Identyfikator wyróżnika.

14.1.1.2 Zwraca

Typ:

ReCourtDiscriminant

Wyróżnik określony przez dany identyfikator

14.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



15 Metoda ICourtRepositoryCommunicationService.GetDiscriminants

Zwraca listę wyróżników.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

15.1 Składnia

C#

```
List<ReCourtDiscriminant> GetDiscriminants ()
```

15.1.1.1 Zwraca

Typ:
Lista wyróżników.

List(ReCourtDiscriminant)

15.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

16 Metoda ICourtRepositoryCommunicationService.GetDiscriminantsVO

Zwraca listę wyróżników.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

16.1 Składnia

C#

```
List<ReCourtDiscriminantVO> GetDiscriminantsVO()
```

16.1.1.1 Zwraca

Typ:

List(ReCourtDiscriminantVO)

Lista wyróżników.

16.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



17 Metoda ICourtRepositoryCommunicationService.GetDiscriminantsVOPagger

Zwraca stronicowaną listę wyróżników.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
 Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
 ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

17.1 Składnia

C#	
PagedList<ReCourtDiscriminantVO> int int SortCriteria)	GetDiscriminantsVOPagger (<i>pageNum</i> , <i>recordsPerPage</i> , <i>sortCriteria</i>)

17.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

17.1.1.2 Zwraca

Typ:

PagedList(ReCourtDiscriminantVO)

Stronicowana lista użytkowników.

17.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



18 Metoda ICourtRepositoryCommunicationService.GetStorages

Zwraca listę zasobów.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

18.1 Składnia

C#

```
List<ReCourtStorage> GetStorages ()
```

18.1.1.1 Zwraca

Typ:

List(ReCourtStorage)

Lista zasobów.

18.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

19 Metoda `ICourtRepositoryCommunicationService.GetStoragesByType`

Zwraca zasoby określonego typu.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: `ReCourt.Services.Communication.CourtRepository.Base` (w `ReCourt.Services.Communication.CourtRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

19.1 Składnia

C#	
<pre>List<ReCourtStorage> string)</pre>	<pre>GetStoragesByType (storageType</pre>

19.1.1.1 Parametry

storageType

Typ: `System.String`

Typ

19.1.1.2 Zwraca

Typ:

List(ReCourtStorage)

Lista zasobów.

19.2 Zobacz również

[Interfejs `ICourtRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.CourtRepository.Base`](#)

20 Metoda ICourtRepositoryCommunicationService.GetStoragesPager

Zwraca stronicowaną listę zasobów.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

20.1 Składnia

C#	
<pre>PagedList<ReCourtStorage> int int SortCriteria)</pre>	<pre>GetStoragesPager (pageNum, recordsPerPage, sortCriteria</pre>

20.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

20.1.1.2 Zwraca

Typ:

Lista stronicowanych zasobów.

PagedList(ReCourtStorage)

20.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

21 Metoda ICourtRepositoryCommunicationService.RemoveCourtRoom

Usuwa określoną salę sądową.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

21.1 Składnia

```
C#  
  
void RemoveCourtRoom(  
    string courtRoom  
)
```

21.1.1.1 Parametry

courtRoom

Typ: System.String

21.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



22 Metoda ICourtRepositoryCommunicationService.RemoveDiscriminant

Usuwa wyróżnik określony podanym identyfikatorem.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

22.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveDiscriminant(discriminantId</pre>

22.1.1.1 Parametry

discriminantId

Typ: System.Int32

Id wyróżnika

22.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

22.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

23 Metoda ICourtRepositoryCommunicationService.RemoveStorage

Usuwa określony identyfikatorem zasób.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

23.1 Składnia

```
C#  
  
bool RemoveStorage(  
    int storageId  
)
```

23.1.1.1 Parametry

storageId

Typ: System.Int32

Id zasobu

23.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

23.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

24 Metoda ICourtRepositoryCommunicationService.SetCourtRoomsVO

Ustawia listę sali sądowych.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

24.1 Składnia

```
C#  
  
void SetCourtRoomsVO (rooms  
    List<ReCourtCourtRoomVO>  
)
```

24.1.1.1 Parametry

rooms

Typ: System.Collections.Generic.List(**ReCourtCourtRoomVO**)

Sale

24.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

25 Metoda ICourtRepositoryCommunicationService.UpdateDiscriminant

Aktualizuje wyróżnik.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

25.1 Składnia

C#	
<pre>bool ReCourtDiscriminant)</pre>	<pre>UpdateDiscriminant(discriminant</pre>

25.1.1.1 Parametry

discriminant

Typ: **ReCourtDiscriminant**

Wyróżnik

25.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

25.2 Zobacz również

[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)

26 Metoda ICourtRepositoryCommunicationService.UpdateStorage

Aktualizuje zasób.

Przestrzeń nazw: [ReCourt.Services.Communication.CourtRepository.Base](#)
Moduł: ReCourt.Services.Communication.CourtRepository.Base (w
ReCourt.Services.Communication.CourtRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

26.1 Składnia

C#	
<pre>bool ReCourtStorage)</pre>	<pre>UpdateStorage(storage</pre>

26.1.1.1 Parametry

storage

Typ: **ReCourtStorage**

Zasób

26.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

26.2 Zobacz również


[Interfejs ICourtRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.CourtRepository.Base](#)



27 Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base

27.1 Interfejsy

	Interfejs	Opis
	DictionaryRepositoryCommunicationService	Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelników akt.

28 Interfejs IDictionaryRepositoryCommunicationService

Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelnia akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
 Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
 ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)










28.1 Składnia

C#








```
public interface IDictionaryRepositoryCommunicationService
```

IDictionaryRepositoryCommunicationService prezentuje składowe.


















28.2 Metody

	Nazwa	Opis
	AddAnnotation	Dodaje określoną adnotację.
	AddBenchRole	Dodaje rolę ław sędziowskich. Element słownika
	AddCaseType	Dodaje typ sprawy.
	AddComplexAnnotation	Dodaje nową adnotację złożoną.
	AddDocumentTemplate	Dodaje szablon dokumentu.
	AddParticipantFunction	Dodaje funkcję uczestnika postępowania.
	AddReadingRoom	Dodaje czytelnię akt.
	GetAnnotations	Zwraca listę adnotacji.
	GetAnnotationsPager	Zwraca stronicowaną listę adnotacji.
	GetBenchRoles	Zwraca listę ról ław sędziowskich.
	GetBenchRolesPager	Zwraca stronicowaną listę ról ław sędziowskich.



 GetCaseTypes	Zwraca typy spraw.
 GetCaseTypesPager	Zwraca stronicowaną listę typów spraw.
 GetComplexAnnotations	Pobiera listę adnotacji złożonych.
 GetComplexAnnotationsPager	Pobiera stronicowaną listę adnotacji złożonych.
 GetComplexAnnotationUsedShortcuts	Pobiera listę skrótów klawiszowych przypisanych do adnotacji.
 GetDocumentTemplate	Zwraca szablon określonego dokumentu.
 GetDocumentTemplates	Zwraca listę szablonów dokumentów.
 GetDocumentTemplatesVO	Zwraca listę szablonów dokumentów.
 GetFreeDays	Zwraca listę dni wolnych.
 GetMachineInstanceForMainDiscriminant	Zwraca instancję dla pozycji słownika
 GetMachineInstances	Pobiera listę instancji rcs
 GetParticipantFunctions	Zwraca listę funkcji uczestnika postępowania;
 GetParticipantFunctionsPager	Zwraca stronicowaną listę funkcji uczestnika postępowania.
 GetReadingRoom	Zwraca określoną identyfikatorem czytelnię akt.
 GetReadingRoomNames	Zwraca nazwy czytelnii akt.
 GetReadingRooms	Zwraca listę czytelnii akt.
 GetReadingRoomsPager	Zwraca stronicowaną listę czytelnii akt.
 GetReadingRoomsVO	Zwraca listę czytelnii akt.
 IndexDocumentTemplates	Indeksuje szablony dokumentów.



 IsMainDiscriminantCorrect	Zwraca informację o głównym wyróżniku
 RemoveAnnotation	Usuwa adnotację.
 RemoveBenchRole	Usuwa określoną rolę ław sędziowskich.
 RemoveCaseType	Usuwa określony typ spraw.
 RemoveComplexAnnotation	Usuwa adnotację złożoną.
 RemoveDocumentTemplate	Usuwa szablon dokumentu.
 RemoveParticipantFunction	Usuwa funkcję uczestnika postępowania.
 RemoveReadingRoom	Usuwa czytelnię akt.
 SetFreeDays	Ustawia listę dni wolnych.
 UpdateAnnotation	Aktualizuje adnotację.
 UpdateBenchRole	Aktualizuje rolę ław sędziowskich.
 UpdateCaseType	Aktualizuje typ spraw.
 UpdateComplexAnnotation	Aktualizuje adnotację złożoną.
 UpdateDocumentTemplate	Aktualizuje szablon dokumentu.
 UpdateMachineInstances	Aktualizuje dane instancji
 UpdateParticipantFunction	Aktualizuje funkcję uczestnika postępowania.
 UpdateReadingRoom	Aktualizuje czytelnię akt.

28.3 Zobacz również
















[Przeźren nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

29 Metody


IDictionaryRepositoryCommunicationService.IDictionaryRepositoryCommunicationService

[IDictionaryRepositoryCommunicationService](#) prezentuje składowe.













29.1 Metody

Nazwa	Opis
 AddAnnotation	Dodaje określoną adnotację.
 AddBenchRole	Dodaje rolę ław sędziowskich. Element słownika
 AddCaseType	Dodaje typ sprawy.
 AddComplexAnnotation	Dodaje nową adnotację złożoną.
 AddDocumentTemplate	Dodaje szablon dokumentu.
 AddParticipantFunction	Dodaje funkcję uczestnika postępowania.
 AddReadingRoom	Dodaje czytelnię akt.
 GetAnnotations	Zwraca listę adnotacji.
 GetAnnotationsPager	Zwraca stronicowaną listę adnotacji.
 GetBenchRoles	Zwraca listę ról ław sędziowskich.
 GetBenchRolesPager	Zwraca stronicowaną listę ról ław sędziowskich.
 GetCaseTypes	Zwraca typy spraw.
 GetCaseTypesPager	Zwraca stronicowaną listę typów spraw.
 GetComplexAnnotations	Pobiera listę adnotacji złożonych.
 GetComplexAnnotationsPager	Pobiera stronicowaną listę adnotacji złożonych.



 GetComplexAnnotationUsedShortcuts	Pobiera listę skrótów klawiszowych przypisanych do adnotacji.
 GetDocumentTemplate	Zwraca szablon określonego dokumentu.
 GetDocumentTemplates	Zwraca listę szablonów dokumentów.
 GetDocumentTemplatesVO	Zwraca listę szablonów dokumentów.
 GetFreeDays	Zwraca listę dni wolnych.
 GetMachineInstanceForMainDiscriminant	Zwraca instancję dla pozycji słownika
 GetMachineInstances	Pobiera listę instancji rcs
 GetParticipantFunctions	Zwraca listę funkcji uczestnika postępowania;
 GetParticipantFunctionsPager	Zwraca stronicowaną listę funkcji uczestnika postępowania.
 GetReadingRoom	Zwraca określoną identyfikatorem czytelnię akt.
 GetReadingRoomNames	Zwraca nazwy czytelni akt.
 GetReadingRooms	Zwraca listę czytelni akt.
 GetReadingRoomsPager	Zwraca stronicowaną listę czytelni akt.
 GetReadingRoomsVO	Zwraca listę czytelni akt.
 IndexDocumentTemplates	Indeksuje szablony dokumentów.
 IsMainDiscriminantCorrect	Zwraca informację o głównym wyróżnika
 RemoveAnnotation	Usuwa adnotację.
 RemoveBenchRole	Usuwa określoną rolę ław sędziowskich.
 RemoveCaseType	Usuwa określony typ spraw.
 RemoveComplexAnnotation	Usuwa adnotację złożoną.



 RemoveDocumentTemplate	Usuwa szablon dokumentu.
 RemoveParticipantFunction	Usuwa funkcję uczestnika postępowania.
 RemoveReadingRoom	Usuwa czytelnię akt.
 SetFreeDays	Ustawia listę dni wolnych.
 UpdateAnnotation	Aktualizuje adnotację.
 UpdateBenchRole	Aktualizuje rolę ław sędziowskich.
 UpdateCaseType	Aktualizuje typ spraw.
 UpdateComplexAnnotation	Aktualizuje adnotację złożoną.
 UpdateDocumentTemplate	Aktualizuje szablon dokumentu.
 UpdateMachineInstances	Aktualizuje dane instancji
 UpdateParticipantFunction	Aktualizuje funkcję uczestnika postępowania.
 UpdateReadingRoom	Aktualizuje czytelnię akt.

29.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

30 Metoda IDictionaryRepositoryCommunicationService.AddAnnotation

Dodaje określoną adnotację.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

30.1 Składnia

```
C#  
  
int AddAnnotation(  
    ReCourtDictionaryItem item,  
    ReCourtUser user  
)
```

30.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

user

Typ: **ReCourtUser**

Użytkownik

30.1.1.2 Zwraca

Typ:

Identyfikator dodanej adnotacji.

Int32

30.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

31 Metoda IDictionaryRepositoryCommunicationService.AddBenchRole

Dodaje rolę ław sędziowskich. Element słownika

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

31.1 Składnia

C#	
<pre>int ReCourtDictionaryItem)</pre>	<pre>AddBenchRole (item</pre>

31.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

31.1.1.2 Zwraca

Typ:

Identyfikator nowej roli ław sędziowskich.

Int32

31.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

32 Metoda IDictionaryRepositoryCommunicationService.AddCaseType

Dodaje typ sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

32.1 Składnia

```
C#  
  
int AddCaseType (  
    ReCourtDictionaryItem item  
)
```

32.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

32.1.1.2 Zwraca

Typ:

Int32

Identyfikator typu sprawy.

32.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



33 Metoda IDictionaryRepositoryCommunicationService.AddComplexAnnotation

Dodaje nową adnotację złożoną.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

33.1 Składnia

C#	
<pre>int ReCourtDictionaryItem)</pre>	<pre>AddComplexAnnotation(item</pre>

33.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

33.1.1.2 Zwraca

Typ:

Identyfikator nowej adnotacji.

Int32

33.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

34 Metoda IDictionaryRepositoryCommunicationService.AddDocumentTemplate

Dodaje szablon dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

34.1 Składnia

C#	
ReCourtDocumentTemplate	AddDocumentTemplate (
string	<i>templateName</i> ,
string	<i>templateContent</i>
))

34.1.1.1 Parametry

templateName

Typ: System.String

Nazwa szablonu

templateContent

Typ: System.String

Treść

34.1.1.2 Zwraca

Typ:

ReCourtDocumentTemplate

Obiekt szablonu dokumentu.

34.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



35 Metoda IDictionaryRepositoryCommunicationService.AddParticipantFunction

Dodaje funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)

Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w

ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

35.1 Składnia

```
C#  
  
int AddParticipantFunction(  
    ReCourtDictionaryItem item  
)
```

35.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

35.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanej funkcji.

35.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

36 Metoda IDictionaryRepositoryCommunicationService.AddReadingRoom

Dodaje czytelnię akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

36.1 Składnia

C#	
<pre>int ReCourtReadingRoom)</pre>	<pre>AddReadingRoom(room</pre>

36.1.1.1 Parametry

room

Typ: **ReCourtReadingRoom**

Sala

36.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanej czytelni akt.

36.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

37 Metoda IDictionaryRepositoryCommunicationService.GetAnnotations

Zwraca listę adnotacji.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

37.1 Składnia

C#	
List<ReCourtDictionaryItem> ReCourtUser)	GetAnnotations (user)

37.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

37.1.1.2 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista adnotacji.

37.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

38 Metoda IDictionaryRepositoryCommunicationService.GetAnnotationsPager

Zwraca stronicowaną listę adnotacji.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

38.1 Składnia

```
C#  
  
PagedList<ReCourtDictionaryItem>          GetAnnotationsPager (  
    ReCourtUser                             reCourtUser,  
    int                                       pageNum,  
    int                                       recordsPerPage,  
    SortCriteria                             sortCriteria  
)
```

38.1.1.1 Parametry

reCourtUser

Typ: **ReCourtUser**

Bieżący użytkownik.

pageNum

Typ: System.Int32

Numer strony.

recordsPerPage

Typ: System.Int32

Liczba wierszy danych na stronę.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania danych.

38.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Stronicowana lista adnotacji.

38.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



39 Metoda IDictionaryRepositoryCommunicationService.GetBenchRoles

Zwraca listę ról ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

39.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetBenchRoles()
```

39.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista roli ław sędziowskich.

39.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

40 Metoda IDictionaryRepositoryCommunicationService.GetBenchRolesPager

Zwraca stronicowaną listę ról ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

40.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetBenchRolesPager (pageNum, recordsPerPage, sortCriteria</pre>

40.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

40.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Lista stronicowanych ról ław sędziowskich.

40.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

41 Metoda IDictionaryRepositoryCommunicationService.GetCaseTypes

Zwraca typy spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

41.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetCaseTypes ()
```

41.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista typów spraw.

41.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

42 Metoda IDictionaryRepositoryCommunicationService.GetCaseTypesPager

Zwraca stronicowaną listę typów spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

42.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetCaseTypesPager (pageNum, recordsPerPage, sortCriteria</pre>

42.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

42.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Lista stronicowanych typów spraw.

42.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



43 Metoda IDictionaryRepositoryCommunicationService.GetComplexAnnotations

Pobiera listę adnotacji złożonych.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

43.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetComplexAnnotations()
```

43.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista adnotacji złożonych.

43.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

44 Metoda

IDictionaryRepositoryCommunicationService.GetComplexAnnotationsPager

Pobiera stronicowaną listę adnotacji złożonych.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

44.1 Składnia

```
C#  
  
PagedList<ReCourtDictionaryItem> GetComplexAnnotationsPager (  
    int                pageNum,  
    int                recordsPerPage,  
    SortCriteria       sortCriteria  
)
```

44.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

44.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Stronicowana lista adnotacji złożonych.

44.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

45 Metoda

IDictionaryRepositoryCommunicationService.GetComplexAnnotationUsedShortcuts

Pobiera listę skrótów klawiszowych przypisanych do adnotacji.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

45.1 Składnia

C#

```
List<string> GetComplexAnnotationUsedShortcuts ()
```

45.1.1.1 Zwraca

Typ:

List(String)

Lista skrótów klawiszowych.

45.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

46 Metoda IDictionaryRepositoryCommunicationService.GetDocumentTemplate

Zwraca szablon określonego dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

46.1 Składnia

```
C#  
  
ReCourtDocumentTemplate          GetDocumentTemplate (  
    int                             id  
)
```

46.1.1.1 Parametry

id

Typ: System.Int32

Id szablonu

46.1.1.2 Zwraca

Typ:

ReCourtDocumentTemplate

Obiekt szablonu dokumentu.

46.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



47 Metoda IDictionaryRepositoryCommunicationService.GetDocumentTemplates

Zwraca listę szablonów dokumentów.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

47.1 Składnia

C#

```
List<ReCourtDocumentTemplate> GetDocumentTemplates ()
```

47.1.1.1 Zwraca

Typ:

List(ReCourtDocumentTemplate)

Lista szablonów dokumentów.

47.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

48 Metoda

IDictionaryRepositoryCommunicationService.GetDocumentTemplatesVO

Zwraca listę szablonów dokumentów.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

48.1 Składnia

C#

```
List<ReCourtDocumentTemplateVO> GetDocumentTemplatesVO ()
```

48.1.1.1 Zwraca

Typ: **List(ReCourtDocumentTemplateVO)**
Lista szablonów dokumentów.

48.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



49 Metoda IDictionaryRepositoryCommunicationService.GetFreeDays

Zwraca listę dni wolnych.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

49.1 Składnia

C#

```
List<DateTime> GetFreeDays ()
```

49.1.1.1 Zwraca

Typ:
Lista dni wolnych.

List(DateTime)

49.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

50 Metoda

IDictionaryRepositoryCommunicationService.GetMachineInstanceForMainDiscriminant

Zwraca instancję dla pozycji słownika

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

50.1 Składnia

```
C#  
  
List<ReCourtMachineInstance>  
GetMachineInstanceForMainDiscriminant (  
    int codId  
)
```

50.1.1.1 Parametry

codId

Typ: System.Int32

Id wydziału

50.1.1.2 Zwraca

Typ: **List(ReCourtMachineInstance)**

50.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



51 Metoda IDictionaryRepositoryCommunicationService.GetMachineInstances

Pobiera listę instancji rcs

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

51.1 Składnia

C#

```
List<ReCourtMachineInstance> GetMachineInstances ()
```

51.1.1.1 Zwraca

Typ: **List(ReCourtMachineInstance)**
Lista instancji

51.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



52 Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctions

Zwraca listę funkcji uczestnika postępowania;

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

52.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetParticipantFunctions()
```

52.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista funkcji uczestnika postępowania.

52.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

53 Metoda

IDictionaryRepositoryCommunicationService.GetParticipantFunctionsPager

Zwraca stronicowaną listę funkcji uczestnika postępowania.

Przeźreń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

53.1 Składnia

```
C#  
  
PagedList<ReCourtDictionaryItem> GetParticipantFunctionsPager (  
    int                pageNum,  
    int                recordsPerPage,  
    SortCriteria       sortCriteria  
)
```

53.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

53.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Lista stronicowanych funkcji uczestnika postępowania.

53.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przeźreń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

54 Metoda IDictionaryRepositoryCommunicationService.GetReadingRoom

Zwraca określoną identyfikatorem czytelnię akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

54.1 Składnia

C#	
ReCourtReadingRoom	GetReadingRoom (
int	id
))

54.1.1.1 Parametry

id

Typ: System.Int32

Id sali

54.1.1.2 Zwraca

Typ:

ReCourtReadingRoom

Obiekt czytelni akt.

54.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



55 Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomNames

Zwraca nazwy czytelni akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

55.1 Składnia

C#

```
List<string> GetReadingRoomNames ()
```

55.1.1.1 Zwraca

Typ:

List(String)

Lista nazw czytelni akt.

55.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



56 Metoda IDictionaryRepositoryCommunicationService.GetReadingRooms

Zwraca listę czytelni akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

56.1 Składnia

C#

```
List<ReCourtReadingRoom> GetReadingRooms ()
```

56.1.1.1 Zwraca

Typ:

List(ReCourtReadingRoom)

Lista czytelni akt.

56.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

57 Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomsPager

Zwraca stronicowaną listę czytelników akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

57.1 Składnia

C#	
<pre>PagedList<ReCourtReadingRoom> int int SortCriteria)</pre>	<pre>GetReadingRoomsPager (pageNum, recordsPerPage, sortCriteria</pre>

57.1.1.1 Parametry

pageNum

Typ: System.Int32

Numer strony danych do pobrania.

recordsPerPage

Typ: System.Int32

Liczba wierszy danych na stronę.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania danych.

57.1.1.2 Zwraca

Typ:

Stronicowana lista czytelników akt.

PagedList(ReCourtReadingRoom)

57.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



58 Metoda IDictionaryRepositoryCommunicationService.GetReadingRoomsVO

Zwraca listę czytelni akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

58.1 Składnia

C#

```
List<ReCourtReadingRoomVO> GetReadingRoomsVO ()
```

58.1.1.1 Zwraca

Typ:

List(ReCourtReadingRoomVO)

Lista czytelni akt.

58.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

59 Metoda

IDictionaryRepositoryCommunicationService.IndexDocumentTemplates

Indeksuje szablony dokumentów.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

59.1 Składnia

```
C#  
  
string                IndexDocumentTemplates (  
    ReCourtUser        user,  
    string              fileName,  
    string              content,  
    string              type  
)
```

59.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

fileName

Typ: System.String

Nazwa pliku

content

Typ: System.String

Treść

type

Typ: System.String

Typ

59.1.1.2 Zwraca

Typ:

Wynik aktualizacji.

String

59.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

60 Metoda

IDictionaryRepositoryCommunicationService.IsMainDiscriminantCorrect

Zwraca informację o głównym wyróżniku

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

60.1 Składnia

```
C#  
  
int IsMainDiscriminantCorrect(  
    ReCourtDiscriminant discriminant  
)
```

60.1.1.1 Parametry

discriminant

Typ: **ReCourtDiscriminant**

Wyróżnik

60.1.1.2 Zwraca

Typ:

Int32

Czy główny

60.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

61 Metoda IDictionaryRepositoryCommunicationService.RemoveAnnotation

Usuwa adnotację.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

61.1 Składnia

```
C#  
  
bool RemoveAnnotation(  
    int itemId  
)
```

61.1.1.1 Parametry

itemId

Typ: System.Int32

Id elementu słownika

61.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

61.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

62 Metoda IDictionaryRepositoryCommunicationService.RemoveBenchRole

Usuwa określoną rolę ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

62.1 Składnia

```
C#  
  
bool RemoveBenchRole (  
    int itemId  
)
```

62.1.1.1 Parametry

itemId

Typ: System.Int32

Id elementu słownika

62.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

62.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

63 Metoda IDictionaryRepositoryCommunicationService.RemoveCaseType

Usuwa określony typ spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

63.1 Składnia

```
C#  
  
bool RemoveCaseType (  
    int itemId  
)
```

63.1.1.1 Parametry

itemId

Typ: System.Int32

Element słownika Id

63.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

63.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



64 Metoda

IDictionaryRepositoryCommunicationService.RemoveComplexAnnotation

Usuwa adnotację złożoną.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

64.1 Składnia

```
C#  
  
bool RemoveComplexAnnotation(  
    int itemId  
)
```

64.1.1.1 Parametry

itemId

Typ: System.Int32

Id elementu słownika

64.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

64.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

65 Metoda

IDictionaryRepositoryCommunicationService.RemoveDocumentTemplate

Usuwa szablon dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

65.1 Składnia

```
C#  
  
bool RemoveDocumentTemplate (  
    int id  
)
```

65.1.1.1 Parametry

id

Typ: System.Int32

Id szablonu

65.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

65.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

66 Metoda

IDictionaryRepositoryCommunicationService.RemoveParticipantFunction

Usuwa funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

66.1 Składnia

```
C#  
  
bool RemoveParticipantFunction(  
    int itemId  
)
```

66.1.1.1 Parametry

itemId

Typ: System.Int32

Id elementu słownika

66.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

66.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

67 Metoda IDictionaryRepositoryCommunicationService.RemoveReadingRoom

Usuwa czytelnię akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)

Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w

ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

67.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveReadingRoom(roomId</pre>

67.1.1.1 Parametry

roomId

Typ: System.Int32

Id sali

67.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

67.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

68 Metoda IDictionaryRepositoryCommunicationService.SetFreeDays

Ustawia listę dni wolnych.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

68.1 Składnia

```
C#  
  
bool SetFreeDays (  
    List<DateTime> days  
)
```

68.1.1.1 Parametry

days

Typ: System.Collections.Generic.List(**DateTime**)

Dni wolne

68.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

68.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

69 Metoda IDictionaryRepositoryCommunicationService.UpdateAnnotation

Aktualizuje adnotację.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

69.1 Składnia

C#	
<pre>bool ReCourtDictionaryItem)</pre>	<pre>UpdateAnnotation(item</pre>

69.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

69.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

69.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

70 Metoda IDictionaryRepositoryCommunicationService.UpdateBenchRole

Aktualizuje rolę ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

70.1 Składnia

C#	
<pre>bool ReCourtDictionaryItem)</pre>	<pre>UpdateBenchRole(item</pre>

70.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

70.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

70.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

71 Metoda IDictionaryRepositoryCommunicationService.UpdateCaseType

Aktualizuje typ spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

71.1 Składnia

C#	
<pre>bool ReCourtDictionaryItem)</pre>	<pre>UpdateCaseType (item</pre>

71.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

71.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

71.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

72 Metoda

IDictionaryRepositoryCommunicationService.UpdateComplexAnnotation

Aktualizuje adnotację złożoną.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

72.1 Składnia

```
C#  
  
bool UpdateComplexAnnotation(  
    ReCourtDictionaryItem item  
)
```

72.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

72.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

72.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

73 Metoda

IDictionaryRepositoryCommunicationService.UpdateDocumentTemplate

Aktualizuje szablon dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

73.1 Składnia

C#	
ReCourtDocumentTemplate	UpdateDocumentTemplate (
int	id,
string	templateContent
)	

73.1.1.1 Parametry

id

Typ: System.Int32

Id szablonu

templateContent

Typ: System.String

Treść

73.1.1.2 Zwraca

Typ:

ReCourtDocumentTemplate

Obiekt szablonu dokumentu.

73.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

74 Metoda

IDictionaryRepositoryCommunicationService.UpdateMachineInstances

Aktualizuje dane instancji

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

74.1 Składnia

```
C#  
  
void UpdateMachineInstances (  
    List<ReCourtMachineInstance> instances  
)
```

74.1.1.1 Parametry

instances

Typ: System.Collections.Generic.List(**ReCourtMachineInstance**)

Dane instancji

74.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

75 Metoda

IDictionaryRepositoryCommunicationService.UpdateParticipantFunction

Aktualizuje funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

75.1 Składnia

```
C#  
  
bool UpdateParticipantFunction(  
    ReCourtDictionaryItem item  
)
```

75.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Element słownika

75.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

75.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)

76 Metoda IDictionaryRepositoryCommunicationService.UpdateReadingRoom

Aktualizuje czytelnię akt.

Przestrzeń nazw: [ReCourt.Services.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Communication.DictionaryRepository.Base (w
ReCourt.Services.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

76.1 Składnia

C#	
<pre>bool ReCourtReadingRoom)</pre>	<pre>UpdateReadingRoom(room</pre>

76.1.1.1 Parametry

room

Typ: **ReCourtReadingRoom**

Sala

76.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

76.2 Zobacz również


[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.DictionaryRepository.Base](#)



77 Przestrzeń nazw ReCourt.Services.Communication.Documents.Base

77.1 Interfejsy

	Interfejs	Opis
	IDocumentsCommunicationService	Interfejs odpowiada za pobieranie dokumentów (np. powiązanych z konkretną czytelnią akt bądź użytkownikiem).

78 Interfejs IDocumentsCommunicationService

Interfejs odpowiada za pobieranie dokumentów (np. powiązanych z konkretną czytelnią akt bądź użytkownikiem).

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
 Moduł: ReCourt.Services.Communication.Documents.Base (w
 ReCourt.Services.Communication.Documents.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

78.1 Składnia

C#

```
public interface IDocumentsCommunicationService
```

IDocumentsCommunicationService prezentuje składowe.

78.2 Metody

	Nazwa	Opis
	AreNewDocumentAvailable	Sprawdza, czy istnieją nowe dokumenty dla podanego użytkownika.
	GetDocument	Zwraca dokument.
	GetDocumentForReadingRoom	Zwraca dokument o zadanych parametrach.
	GetUserDocuments	Zwraca dokumenty użytkownika.
	GetUserDocumentsVO	Zwraca dokumenty użytkownika VO.

78.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)

79 Metody

IDocumentsCommunicationService.IDocumentsCommunicationService

[IDocumentsCommunicationService](#) prezentuje składowe.

79.1 Metody

	Nazwa	Opis
	AreNewDocumentAvailable	Sprawdza, czy istnieją nowe dokumenty dla podanego użytkownika.
	GetDocument	Zwraca dokument.
	GetDocumentForReadingRoom	Zwraca dokument o zadanych parametrach.
	GetUserDocuments	Zwraca dokumenty użytkownika.
	GetUserDocumentsVO	Zwraca dokumenty użytkownika VO.

79.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)



80 Metoda IDocumentsCommunicationService.AreNewDocumentAvailable

Sprawdza, czy istnieją nowe dokumenty dla podanego użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
Moduł: ReCourt.Services.Communication.Documents.Base (w
ReCourt.Services.Communication.Documents.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

80.1 Składnia

```
C#  
  
bool AreNewDocumentAvailable(  
    int userId,  
    DateTime lastChecked  
)
```

80.1.1.1 Parametry

userId

Typ: System.Int32

Identyfikator użytkownika.

lastChecked

Typ: System.DateTime

Data ostatniego sprawdzania.

80.1.1.2 Zwraca

Typ:

Boolean

Informację, czy istnieją nowe dokumenty.

80.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)



81 Metoda IDocumentsCommunicationService.GetDocument

Zwraca dokument.

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
Moduł: ReCourt.Services.Communication.Documents.Base (w
ReCourt.Services.Communication.Documents.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

81.1 Składnia

C#	
Document long)	GetDocument (<i>id</i>

81.1.1.1 Parametry

id

Typ: System.Int64

Id dokumentu

81.1.1.2 Zwraca

Typ:

Document

Pobrany dokument.

81.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)



82 Metoda IDocumentsCommunicationService.GetDocumentForReadingRoom

Zwraca dokument o zadanych parametrach.

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
Moduł: ReCourt.Services.Communication.Documents.Base (w
ReCourt.Services.Communication.Documents.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

82.1 Składnia

```
C#  
  
Document GetDocumentForReadingRoom (  
    int roomId,  
    string token  
)
```

82.1.1.1 Parametry

roomId

Typ: System.Int32

Id czytelnia

token

Typ: System.String

Token

82.1.1.2 Zwraca

Typ:

Pobrano dokument.

Document

82.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)

83 Metoda IDocumentsCommunicationService.GetUserDocuments

Zwraca dokumenty użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
 Moduł: ReCourt.Services.Communication.Documents.Base (w
 ReCourt.Services.Communication.Documents.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

83.1 Składnia

```

C#
PagedList<Document> GetUserDocuments (
    DocumentFilter     filter,
    int                 userId,
    int                 pageNum,
    int                 recordsPerPage,
    SortCriteria        sortCriteria
)
    
```

83.1.1.1 Parametry

filter

Typ: **DocumentFilter**

Obiekt filtru.

userId

Typ: System.Int32

Identyfikator użytkownika, którego zadania zostaną pobrane.

pageNum

Typ: System.Int32

Numer strony.

recordsPerPage

Typ: System.Int32

Ilość rekordów na stronie.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania.

83.1.1.2 Zwraca

Typ:

PagedList(Document)

Lista zawierająca dokumenty użytkownika.

83.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)

84 Metoda `IDocumentsCommunicationService.GetUserDocumentsVO`

Zwraca dokumenty użytkownika VO.

Przestrzeń nazw: [ReCourt.Services.Communication.Documents.Base](#)
Moduł: `ReCourt.Services.Communication.Documents.Base` (w
`ReCourt.Services.Communication.Documents.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

84.1 Składnia

```
C#
PagedList<DocumentWithStateVO> GetUserDocumentsVO (
    DocumentFilter filter,
    int userId,
    int pageNum,
    int recordsPerPage,
    SortCriteria sortCriteria
)
```

84.1.1.1 Parametry

filter

Typ: **DocumentFilter**

Obiekt filtru.

userId

Typ: `System.Int32`

Identyfikator użytkownika, którego zadania zostaną pobrane.

pageNum

Typ: `System.Int32`

Numer strony.

recordsPerPage

Typ: `System.Int32`

Ilość rekordów na stronie.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania.

84.1.1.2 Zwraca

Typ:

PagedList(DocumentWithStateVO)

Lista zawierająca dokumenty użytkownika VO.

84.2 Zobacz również

[Interfejs IDocumentsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Documents.Base](#)



85 Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base

85.1 Klasy

	Klasa	Opis
	FileTransferInfo	Dane pliku
	SendFileRequestMessage	Dane pliku

85.2 Interfejsy

	Interfejs	Opis
	IFileTransferCommunicationService	Interfejs odpowiada za zarządzanie plikami (choćby pobieranie, usuwanie, wysyłanie).

85.3 Wyliczenia

	Wyliczenie	Opis
	FileType	Typ pliku

86 Klasa FileTransferInfo

Dane pliku

86.1 Hierarchia

System.Object

ReCourt.Services.Communication.FileTransfer.Base.FileTransferInfo

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)

Moduł: ReCourt.Services.Communication.FileTransfer.Base (w

ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

86.2 Składnia

C#






```
public class FileTransferInfo
```

FileTransferInfo prezentuje składowe.

86.3 Konstruktory

	Nazwa	Opis
	FileTransferInfo	Konstruktor

86.4 Właściwości

	Nazwa	Opis
	Checksum	Suma kontrolna
	FileTransferDestination	Docelowy katalog
	Length	Wielkość
	Name	Nazwa
	SubDirName	Podkatalog

86.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



87 Konstruktor FileTransferInfo

Konstruktor

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

87.1 Składnia

C#

```
public FileTransferInfo()
```

87.2 Zobacz również






[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

88 Właściwości FileTransferInfo.FileTransferInfo

[FileTransferInfo](#) prezentuje składowe.

88.1 Właściwości

	Nazwa	Opis
	Checksum	Suma kontrolna
	FileTransferDestination	Docelowy katalog
	Length	Wielkość
	Name	Nazwa
	SubDirName	Podkatalog

88.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przeźreń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

89 Właściwość FileTransferInfo.Checksum

Suma kontrolna

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

89.1 Składnia

C#

```
public byte[] Checksum { get; set; }
```

89.1.1.1 Zwraca

Typ: **Byte[]**

89.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

90 Właściwość FileTransferInfo.FileTransferDestination

Docelowy katalog

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

90.1 Składnia

C#

```
public FileType FileTransferDestination { get; set; }
```

90.1.1.1 Zwraca

Typ: [FileType](#)

90.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

91 Właściwość FileTransferInfo.Length

Wielkość

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

91.1 Składnia

C#

```
public long Length { get; set; }
```

91.1.1.1 Zwraca

Typ: **Int64**

91.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



92 Właściwość FileTransferInfo.Name

Nazwa

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

92.1 Składnia

C#

```
public string Name { get; set; }
```

92.1.1.1 Zwraca

Typ: **String**

92.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

93 Właściwość FileTransferInfo.SubDirName

Podkatalog

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

93.1 Składnia

C#

```
public string SubDirName { get; set; }
```

93.1.1.1 Zwraca

Typ: **String**

93.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

94 Wyliczenie FileType

Typ pliku

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

94.1 Składnia

C#

```
public enum FileType
```

94.2 Składowe

Nazwa	Wartość	Opis
Report	0	Raport
Temp	1	Tymczasowy

94.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

95 Interfejs IFileTransferCommunicationService

Interfejs odpowiada za zarządzanie plikami (choćby pobieranie, usuwanie, wysyłanie).

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
 Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
 ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)




95.1 Składnia

C#

```
public interface IFileTransferCommunicationService
```

IFileTransferCommunicationService prezentuje składowe.

95.2 Metody

	Nazwa	Opis
	CalculateFileSize	Zwraca informację o rozmiarze plików
	DropDirectory	Porzuca (usuwa) dany katalog.
	DropFiles	Porzuca (usuwa) dane pliki.
	GetFile	Zwraca plik.
	GetTextFileContent	Zwraca treść pliku tekstowego.
	SendFile	Wysyła plik.

95.3 Zobacz również







[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

96 Metody

IFileTransferCommunicationService.IFileTransferCommunicationService

[IFileTransferCommunicationService](#) prezentuje składowe.

96.1 Metody

	Nazwa	Opis
	CalculateFileSize	Zwraca informację o rozmiarze plików
	DropDirectory	Porzuca (usuwa) dany katalog.
	DropFiles	Porzuca (usuwa) dane pliki.
	GetFile	Zwraca plik.
	GetTextFileContent	Zwraca treść pliku tekstowego.
	SendFile	Wysyła plik.

96.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



97 Metoda IFileTransferCommunicationService.CalculateFileSize

Zwraca informację o rozmiarze plików

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

97.1 Składnia

C#	
<pre>long List<string>)</pre>	<pre>CalculateFileSize(files</pre>

97.1.1.1 Parametry

files

Typ: System.Collections.Generic.List(**String**)

Lista plików do pobrania

97.1.1.2 Zwraca

Typ:

Int64

Rozmiar plików

97.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

98 Metoda IFileTransferCommunicationService.DropDirectory

Porzuca (usuwa) dany katalog.

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

98.1 Składnia

C#	
<pre>bool string FileType)</pre>	<pre>DropDirectory(path, type</pre>

98.1.1.1 Parametry

path

Typ: System.String

ścieżka

type

Typ: [ReCourt.Services.Communication.FileTransfer.Base.FileType](#)

Typ

98.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

98.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

99 Metoda IFileTransferCommunicationService.DropFiles

Porzuca (usuwa) dane pliki.

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

99.1 Składnia

C#	
<pre>bool List<string>)</pre>	<pre>DropFiles (files</pre>

99.1.1.1 Parametry

files

Typ: System.Collections.Generic.List(**String**)

Nazwy plików

99.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

99.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



100Metoda IFileTransferCommunicationService.GetFiles

Zwraca plik.

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

100.1 Składnia

```
C#  
  
SendFileRequestMessage GetFile(  
    SendFileRequestMessage fileToGet  
)
```

100.1.1.1 Parametry

fileToGet

Typ: [ReCourt.Services.Communication.FileTransfer.Base.SendFileRequestMessage](#)

Dane pliku

100.1.1.2 Zwraca

Typ: [SendFileRequestMessage](#)

Pobrano plik.

100.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



101 Metoda `IFileTransferCommunicationService.GetTextFileContent`

Zwraca treść pliku tekstowego.

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

101.1 Składnia

```
C#  
  
string GetTextFileContent (  
    string fileName  
)
```

101.1.1.1 Parametry

fileName

Typ: System.String

Nazwa pliku

101.1.1.2 Zwraca

Typ:

String

Treść pliku.

101.2 Zobacz również

[Interfejs `IFileTransferCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.FileTransfer.Base`](#)



102 Metoda IFileTransferCommunicationService.SendFile

Wysyła plik.

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

102.1 Składnia

```
C#  
  
void SendFile(  
    SendFileRequestMessage request  
)
```

102.1.1.1 Parametry

request

Typ: [ReCourt.Services.Communication.FileTransfer.Base.SendFileRequestMessage](#)

102.2 Zobacz również

[Interfejs IFileTransferCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



103 Klasa `SendFileRequestMessage`

Dane pliku

103.1 Hierarchia

System.Object

ReCourt.Services.Communication.FileTransfer.Base.SendFileRequestMessage

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)

Moduł: ReCourt.Services.Communication.FileTransfer.Base (w

ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

103.2 Składnia

```
C#
public class SendFileRequestMessage
```

SendFileRequestMessage prezentuje składowe.

103.3 Konstruktory

	Nazwa	Opis
	SendFileRequestMessage	Inicjalizuje nową instancję klasy SendFileRequestMessage

103.4 Pola

	Nazwa	Opis
	FileData	Zawartość
	FileInfo	Dane pliku

103.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)

104 Konstruktor `SendFileRequestMessage`

Inicjalizuje nową instancję klasy [SendFileRequestMessage](#)

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

104.1 Składnia

C#

```
public SendFileRequestMessage ()
```

104.2 Zobacz również

[Klasa `SendFileRequestMessage`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.FileTransfer.Base`](#)



105 Pola `SendMessage.SendMessage`

[SendMessage](#) prezentuje składowe.

105.1 Pola

	Nazwa	Opis
	FileData	Zawartość
	FileInfo	Dane pliku

105.2 Zobacz również

[Klasa SendMessage](#)

[Przestrzeń nazw ReCourt.Services.Communication.FileTransfer.Base](#)



106 Pole `SendMessage.FileData`

Zawartość

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

106.1 Składnia

C#

```
public Stream FileData
```

106.1.1.1 Zwraca

Typ: **Stream**

106.2 Zobacz również

[Klasa `SendMessage`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.FileTransfer.Base`](#)

107 Pole `SendMessage.FileInfo`

Dane pliku

Przestrzeń nazw: [ReCourt.Services.Communication.FileTransfer.Base](#)
Moduł: ReCourt.Services.Communication.FileTransfer.Base (w
ReCourt.Services.Communication.FileTransfer.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

107.1 Składnia

C#

```
public FileTransferInfo FileInfo
```

107.1.1.1 Zwraca

Typ: [FileTransferInfo](#)

107.2 Zobacz również


[Klasa `SendMessage`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.FileTransfer.Base`](#)



108Przestrzeń nazw ReCourt.Services.Communication.Logging.Base

108.1 Interfejsy

	Interfejs	Opis
	LoggingCommunicationService	Interfejs odpowiada za logowanie błędów i komunikatów.



109 Interfejs ILoggingCommunicationService

Interfejs odpowiada za logowanie błędów i komunikatów.

Przestrzeń nazw: [ReCourt.Services.Communication.Logging.Base](#)
Moduł: ReCourt.Services.Communication.Logging.Base (w
ReCourt.Services.Communication.Logging.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

109.1 Składnia

C#

```
public interface ILoggingCommunicationService
```

ILoggingCommunicationService prezentuje składowe.

109.2 Metody

	Nazwa	Opis
	LogAction	Logowanie informacji
	LogError	Logowanie informacji



109.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Logging.Base](#)

110 Metody `ILoggingCommunicationService.ILoggingCommunicationService`

[ILoggingCommunicationService](#) prezentuje składowe.

110.1 Metody

	Nazwa	Opis
	LogAction	Logowanie informacji
	LogError	Logowanie informacji

110.2 Zobacz również

[Interfejs `ILoggingCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Logging.Base`](#)

111 Metoda `ILoggingCommunicationService.LogAction`

Logowanie informacji

Przestrzeń nazw: [ReCourt.Services.Communication.Logging.Base](#)
Moduł: ReCourt.Services.Communication.Logging.Base (w
ReCourt.Services.Communication.Logging.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

111.1 Składnia

```
C#  
  
void LogAction(  
    int cjrId,  
    string room,  
    string message  
)
```

111.1.1.1 Parametry

cjrId

Typ: System.Int32

Id CJR

room

Typ: System.String

Sala

message

Typ: System.String

Wiadomość

111.2 Zobacz również

[Interfejs `ILoggingCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Logging.Base`](#)

112 Metoda `ILoggingCommunicationService.LogError`

Logowanie informacji

Przestrzeń nazw: [ReCourt.Services.Communication.Logging.Base](#)
Moduł: ReCourt.Services.Communication.Logging.Base (w
ReCourt.Services.Communication.Logging.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

112.1 Składnia

```
C#  
  
void LogError(  
    int cjrId,  
    string room,  
    string message,  
    string stack  
)
```

112.1.1.1 Parametry

cjrId

Typ: System.Int32

Id CJR

room

Typ: System.String

Sala

message

Typ: System.String

Wiadomość

stack

Typ: System.String

Stos

112.2 Zobacz również

[Interfejs `ILoggingCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Logging.Base`](#)



113Przestrzeń

nazw

ReCourt.Services.Communication.MediaServerConfiguration.Base

113.1 Interfejsy

	Interfejs	Opis
	IMediaServerConfiguration	Zarządzanie konfiguracją

114 Interfejs IMediaServerConfiguration

Zarządzanie konfiguracją

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
 Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
 ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
 (1.0.0.0)

114.1 Składnia

C#

```
public interface IMediaServerConfiguration
```

IMediaServerConfiguration prezentuje składowe.

114.2 Metody

	Nazwa	Opis
	CheckConfigs	Sprawdzenie konfiguracji
	GetAllMachines	Pobranie listy maszyn
	GetConfig	Pobierz konfigurację
	GetMachines	Pobranie listy maszyn
	MarkConfigAsInvalid	Oznaczenie konfiguracji do aktualizacji
	MarkConfigAsLoaded	Oznaczenie konfiguracji
	SaveConfig	Zapisanie konfiguracji
	SetMachineRoom	Ustawienie sali

114.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



115 Metody IMediaServerConfiguration.IMediaServerConfiguration

[IMediaServerConfiguration](#) prezentuje składowe.

115.1 Metody

	Nazwa	Opis
	CheckConfigs	Sprawdzenie konfiguracji
	GetAllMachines	Pobranie listy maszyn
	GetConfig	Pobierz konfigurację
	GetMachines	Pobranie listy maszyn
	MarkConfigAsInvalid	Oznaczenie konfiguracji do aktualizacji
	MarkConfigAsLoaded	Oznaczenie konfiguracji
	SaveConfig	Zapisanie konfiguracji
	SetMachineRoom	Ustawienie sali

115.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)

116 Metoda IMediaServerConfiguration.CheckConfigs

Sprawdzenie konfiguracji

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

116.1 Składnia

```
C#  
  
Dictionary<ConfigType,           byte[]>           CheckConfigs(  
    Guid                          machine,  
    Dictionary<ConfigType,       byte[]>           fileHashes  
)
```

116.1.1.1 Parametry

machine

Typ: System.Guid

Maszyna

fileHashes

Typ: System.Collections.Generic.Dictionary(**ConfigType**, **Byte[]**)

Hashe

116.1.1.2 Zwraca

Typ: **Dictionary(ConfigType, Byte[])**

Informacja o poprawności

116.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



117 Metoda IMediaServerConfiguration.GetAllMachines

Pobranie listy maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

117.1 Składnia

C#

```
List<MediaServerMachine> GetAllMachines()
```

117.1.1.1 Zwraca

Typ:

List(MediaServerMachine)

Lista maszyn

117.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



118Metoda IMediaServerConfiguration.GetConfig

Pobierz konfigurację

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
 Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
 ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
 (1.0.0.0)

118.1 Składnia

C#	
ConfigFile	GetConfig(<i>machine</i> , <i>configType</i>)

118.1.1.1 Parametry

machine

Typ: System.Guid

Maszyna

configType

Typ: **ConfigType**

Typ

118.1.1.2 Zwraca

Typ:

Dane konfiguracji

ConfigFile

118.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)

119 Metoda IMediaServerConfiguration.GetMachines

Pobranie listy maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

119.1 Składnia

C#	
<pre>PagedList<MediaServerMachine> int int SortCriteria)</pre>	<pre>GetMachines (pageNum, recordsPerPage, sortCriteria</pre>

119.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

119.1.1.2 Zwraca

Typ: **PagedList(MediaServerMachine)**

119.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)

120Metoda IMediaServerConfiguration.MarkConfigAsInvalid

Oznaczenie konfiguracji do aktualizacji

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

120.1 Składnia

```
C#  
  
void MarkConfigAsInvalid(  
    Guid machine,  
    ConfigType configType  
)
```

120.1.1.1 Parametry

machine

Typ: System.Guid

Maszyna

configType

Typ: **ConfigType**

Typ

120.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



121Metoda IMediaServerConfiguration.MarkConfigAsLoaded

Oznaczenie konfiguracji

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

121.1 Składnia

```
C#  
  
void MarkConfigAsLoaded(  
    Guid machine,  
    ConfigType configType  
)
```

121.1.1.1 Parametry

machine

Typ: System.Guid

Maszyna

configType

Typ: **ConfigType**

Typ

121.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)

122 Metoda IMediaServerConfiguration.SaveConfig

Zapisanie konfiguracji

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

122.1 Składnia

```
C#  
  
void SaveConfig(  
    Guid machine,  
    ConfigType configType,  
    byte[] contents  
)
```

122.1.1.1 Parametry

machine

Typ: System.Guid

Maszyna

configType

Typ: **ConfigType**

Typ

contents

Typ: System.Byte[]

Zawartość

122.2 Zobacz również

[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



123 Metoda IMediaServerConfiguration.SetMachineRoom

Ustawienie sali

Przestrzeń nazw: [ReCourt.Services.Communication.MediaServerConfiguration.Base](#)
Moduł: ReCourt.Services.Communication.MediaServerConfiguration.Base (w
ReCourt.Services.Communication.MediaServerConfiguration.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

123.1 Składnia

```
C#  
  
void SetMachineRoom(  
    Guid identifier,  
    string room  
)
```

123.1.1.1 Parametry

identifier

Typ: System.Guid

Maszyna

room

Typ: System.String

Sala

123.2 Zobacz również


[Interfejs IMediaServerConfiguration](#)

[Przestrzeń nazw ReCourt.Services.Communication.MediaServerConfiguration.Base](#)



124Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base

124.1 Interfejsy

	Interfejs	Opis
	IRcsSettingsCommunicationService	Interfejs odpowiada za obsługę zadań pobrania wydziałów sądów, sądów oraz komunikację.



125 Interfejs IRcsSettingsCommunicationService

Interfejs odpowiada za obsługę zadań pobrania wydziałów sądów, sądów oraz komunikację.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

125.1 Składnia
















```
C#  
  
public interface IRcsSettingsCommunicationService
```

IRcsSettingsCommunicationService prezentuje składowe.

125.2 Metody

	Nazwa	Opis
	AddInformation	Dodaje komunikat wyświetlany użytkownikowi.
	AddLink	Dodaje odnośnik wyświetlany użytkownikowi.
	AddServerInstance	Dodaje nową instancję serwera.
	AddTrialListView	Dodaje rozprawę (widok listy).
	GetActiveCourtDepartmentForDiscriminant	Zwraca listę aktualnych ReCourtCourtDepartment dla listy wyróżników
	GetConfiguredCourtDepartments	Zwraca skonfigurowane wydziały sądu.
	GetCourtAddress	Adres IP.
	GetCurrentTimeFromServer	Pobiera aktualny czas serwera.
	GetDashBoardData	Pobiera zbiór danych dla panelu użytkownika.
	GetInformations	Zwraca listę komunikatów dla użytkownika.



 GetLinks	Zwraca listę odnośników.
 GetServerInstance	Zwraca instancję serwera odpowiadającym identyfikatorowi sądu.
 GetServerInstanceForDepartment	Zwraca instancję serwera odpowiedniego wydziału.
 GetServerInstanceVos	Zwraca instancje serwera.
 GetServerInstanceVosPagger	Zwraca stronicowane instancje serwera.
 GetSignerComponents	Zwraca komponenty sygnatariuszy.
 GetTrialListView	Zwraca listę rozpraw (widok listy).
 RemoveInformation	Usuwa komunikat.
 RemoveLink	Usuwa odnośnik.
 RemoveServerInstance	Usuwa instancję serwera.
 RemoveTrialListView	Usuwa rozprawę.
 UpdateInformation	Aktualizuje komunikat.
 UpdateLink	Aktualizuje odnośnik.
 UpdateServerInstance	Aktualizuje instancję serwera.
 UpdateTrialListView	Aktualizuje rozprawę.

125.3 Zobacz również

[Przeźrenie nazw ReCourt.Services.Communication.RcsSettings.Base](#)

126 Metody












IRcsSettingsCommunicationService.IRcsSettingsCommunicationService

[IRcsSettingsCommunicationService](#) prezentuje składowe.

126.1 Metody

	Nazwa	Opis
⇒	AddInformation	Dodaje komunikat wyświetlany użytkownikowi.
⇒	AddLink	Dodaje odnośnik wyświetlany użytkownikowi.
⇒	AddServerInstance	Dodaje nową instancję serwera.
⇒	AddTrialListView	Dodaje rozprawę (widok listy).
⇒	GetActiveCourtDepartmentForDiscriminant	Zwraca listę aktualnych ReCourtCourtDepartment dla listy wyróżników
⇒	GetConfiguredCourtDepartments	Zwraca skonfigurowane wydziały sądu.
⇒	GetCourtAddress	Adres IP.
⇒	GetCurrentTimeFromServer	Pobiera aktualny czas serwera.
⇒	GetDashBoardData	Pobiera zbiór danych dla panelu użytkownika.
⇒	GetInformations	Zwraca listę komunikatów dla użytkownika.
⇒	GetLinks	Zwraca listę odnośników.
⇒	GetServerInstance	Zwraca instancję serwera odpowiadającym identyfikatorowi sądu.
⇒	GetServerInstanceForDepartment	Zwraca instancję serwera odpowiedniego wydziału.
⇒	GetServerInstanceVos	Zwraca instancje serwera.



 GetServerInstanceVosPagger	Zwraca stronicowane instancje serwera.
 GetSignerComponents	Zwraca komponenty sygnatariuszy.
 GetTrialListView	Zwraca listę rozpraw (widok listy).
 RemoveInformation	Usuwa komunikat.
 RemoveLink	Usuwa odnośnik.
 RemoveServerInstance	Usuwa instancję serwera.
 RemoveTrialListView	Usuwa rozprawę.
 UpdateInformation	Aktualizuje komunikat.
 UpdateLink	Aktualizuje odnośnik.
 UpdateServerInstance	Aktualizuje instancję serwera.
 UpdateTrialListView	Aktualizuje rozprawę.

126.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



127Metoda IRcsSettingsCommunicationService.AddInformation

Dodaje komunikat wyświetlany użytkownikowi.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

127.1 Składnia

C#	
<pre>int ReCourtInformation)</pre>	<pre>AddInformation(information</pre>

127.1.1.1 Parametry

information

Typ: **ReCourtInformation**

Informacja

127.1.1.2 Zwraca

Typ:

Int32

Identyfikator komunikatu.

127.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

128Metoda IRcsSettingsCommunicationService.AddLink

Dodaje odnośnik wyświetlany użytkownikowi.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

128.1 Składnia

C#	
<pre>int ReCourtLink</pre>	<pre>AddLink(link</pre>
<pre>)</pre>	

128.1.1.1 Parametry

link

Typ: **ReCourtLink**

Link

128.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanego odnośnika.

128.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



129Metoda IRcsSettingsCommunicationService.AddServerInstance

Dodaje nową instancję serwera.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

129.1 Składnia

```
C#  
  
int AddServerInstance(  
    ReCourtServerInstance instance  
)
```

129.1.1.1 Parametry

instance

Typ: **ReCourtServerInstance**

Instancja

129.1.1.2 Zwraca

Typ:

Int32

Identyfikator nowej instancji.

129.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

130Metoda IRcsSettingsCommunicationService.AddTrialListView

Dodaje rozprawę (widok listy).

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

130.1 Składnia

```
C#  
  
int AddTrialListView(  
    ReCourtTrialListView trialListView,  
    int userId,  
    bool asGlobal  
)
```

130.1.1.1 Parametry

trialListView

Typ: **ReCourtTrialListView**

Widok

userId

Typ: System.Int32

Id użytkownika

asGlobal

Typ: System.Boolean

Czy globalne

130.1.1.2 Zwraca

Typ:

Identyfikator dodanej rozprawy.

Int32

130.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

131 Metoda

`IRcsSettingsCommunicationService.GetActiveCourtDepartmentForDiscriminant`

Zwraca listę aktualnych `ReCourtCourtDepartment` dla listy wyróżników

Przeźren nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: `ReCourt.Services.Communication.RcsSettings.Base` (w
`ReCourt.Services.Communication.RcsSettings.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

131.1 Składnia

```
C#  
  
List<ReCourtCourtDepartment>  
GetActiveCourtDepartmentForDiscriminant (  
    List<string> discriminantsList  
)
```

131.1.1.1 Parametry

discriminantsList

Typ: `System.Collections.Generic.List<String>`

131.1.1.2 Zwraca

Typ: `List<ReCourtCourtDepartment>`
Lista wydziałów

131.2 Zobacz również

[Interfejs `IRcsSettingsCommunicationService`](#)

[Przeźren nazw `ReCourt.Services.Communication.RcsSettings.Base`](#)

132 Metoda IRcsSettingsCommunicationService.GetConfiguredCourtDepartments

Zwraca skonfigurowane wydziały sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

132.1 Składnia

```
C#  
  
List<ReCourtCourtDepartment> GetConfiguredCourtDepartments (  
    int excludeInsId  
)
```

132.1.1.1 Parametry

excludeInsId

Typ: System.Int32

Bez instancji o id

132.1.1.2 Zwraca

Typ: **List(ReCourtCourtDepartment)**

Lista obiektów wydziałów sądu.

132.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



133 Metoda IRcsSettingsCommunicationService.GetCourtAddress

Adres IP.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

133.1 Składnia

```
C#  
  
string GetCourtAddress (  
    int icourtId,  
    int departmentId  
)
```

133.1.1.1 Parametry

icourtId

Typ: System.Int32

Id sądu

departmentId

Typ: System.Int32

Id wydziału

133.1.1.2 Zwraca

Typ:

String

Adres IP

133.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



135 Metoda IRcsSettingsCommunicationService.GetDashboardData

Pobiera zbiór danych dla panelu użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

135.1 Składnia

C#	
ReCourtDashboardData Nullable<int>)	GetDashboardData (<i>userId</i>)

135.1.1.1 Parametry

userId

Typ: System.Nullable(**Int32**)

Identyfikator użytkownika.

135.1.1.2 Zwraca

Typ:

ReCourtDashboardData

Dane do panelu użytkownika.

135.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



136Metoda IRcsSettingsCommunicationService.GetInformations

Zwraca listę komunikatów dla użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

136.1 Składnia

```
C#  
  
List<ReCourtInformation> GetInformations (  
    bool onlyActive  
)
```

136.1.1.1 Parametry

onlyActive

Typ: System.Boolean

Czy tylko aktywne

136.1.1.2 Zwraca

Typ:

Lista komunikatów.

List(ReCourtInformation)

136.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



137Metoda IRcsSettingsCommunicationService.GetLinks

Zwraca listę odnośników.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

137.1 Składnia

C#

```
List<ReCourtLink> GetLinks ()
```

137.1.1.1 Zwraca

Typ:

List(ReCourtLink)

Lista odnośników.

137.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



138Metoda IRcsSettingsCommunicationService.GetServerInstance

Zwraca instancję serwera odpowiadającym identyfikatorowi sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

138.1 Składnia

C#	
ReCourtServerInstance	GetServerInstance (
int	<i>courtId</i>
))

138.1.1.1 Parametry

courtId

Typ: System.Int32

Id sądu

138.1.1.2 Zwraca

Typ:

ReCourtServerInstance

Obiekt instancji serwera.

138.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

139 Metoda IRcsSettingsCommunicationService.GetServerInstanceForDepartment

Zwraca instancję serwera odpowiedniego wydziału.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

139.1 Składnia

C#	
<pre>ReCourtServerInstance int)</pre>	<pre>GetServerInstanceForDepartment(departmentId</pre>

139.1.1.1 Parametry

departmentId

Typ: System.Int32

Id wydziału

139.1.1.2 Zwraca

Typ:

ReCourtServerInstance

Obiekt instancji serwera.

139.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

141 Metoda IRcsSettingsCommunicationService.GetServerInstanceVosPagger

Zwraca stronicowane instancje serwera.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

141.1 Składnia

```
C#  
  
PagedList<ReCourtServerInstanceVO> GetServerInstanceVosPagger (  
    ReCourtCourtFilter filter,  
    int pageNum,  
    int recordsPerPage,  
    SortCriteria sortCriteria  
)
```

141.1.1.1 Parametry

filter

Typ: **ReCourtCourtFilter**

Filtr

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

141.1.1.2 Zwraca

Typ: **PagedList(ReCourtServerInstanceVO)**

Stronicowana lista instancji.

141.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

142 Metoda IRcsSettingsCommunicationService.GetSignerComponents

Zwraca komponenty sygnatariuszy.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

142.1 Składnia

C#

```
List<ReCourtSigner> GetSignerComponents ()
```

142.1.1.1 Zwraca

Typ:

List(ReCourtSigner)

Lista obiektów sygnatariuszy.

142.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

143 Metoda IRcsSettingsCommunicationService.GetTrialListViews

Zwraca listę rozpraw (widok listy).

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

143.1 Składnia

C#	
List<ReCourtTrialListView>	GetTrialListViews (
int	<i>userId</i> ,
bool	<i>globalToo</i>
))

143.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

globalToo

Typ: System.Boolean

Czy globalne

143.1.1.2 Zwraca

Typ:

List(ReCourtTrialListView)

Lista rozpraw (widok listy).

143.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

144 Metoda IRcsSettingsCommunicationService.RemoveInformation

Usuwa komunikat.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

144.1 Składnia

```
C#  
  
bool RemoveInformation(  
    int informationId  
)
```

144.1.1.1 Parametry

informationId

Typ: System.Int32

Id informacji

144.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

144.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



145 Metoda IRcsSettingsCommunicationService.RemoveLink

Usuwa odnośnik.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

145.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveLink(linkId</pre>

145.1.1.1 Parametry

linkId

Typ: System.Int32

Id linku

145.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

145.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

146 Metoda IRcsSettingsCommunicationService.RemoveServerInstance

Usuwa instancję serwera.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

146.1 Składnia

```
C#  
  
bool RemoveServerInstance(  
    int instanceId  
)
```

146.1.1.1 Parametry

instanceId

Typ: System.Int32

Id instancji

146.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

146.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

147Metoda IRcsSettingsCommunicationService.RemoveTrialListView

Usuwa rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

147.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveTrialListView(trialListViewId</pre>

147.1.1.1 Parametry

trialListViewId

Typ: System.Int32

Id widoku

147.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

147.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

148Metoda IRcsSettingsCommunicationService.UpdateInformation

Aktualizuje komunikat.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

148.1 Składnia

C#	
<pre>bool ReCourtInformation)</pre>	<pre>UpdateInformation(information</pre>

148.1.1.1 Parametry

information

Typ: **ReCourtInformation**

Informacja

148.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

148.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

149Metoda IRcsSettingsCommunicationService.UpdateLink

Aktualizuje odnośnik.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

149.1 Składnia

C#	
<pre>bool ReCourtLink)</pre>	<pre>UpdateLink(link</pre>

149.1.1.1 Parametry

link

Typ: **ReCourtLink**

Link

149.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

149.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



150Metoda IRcsSettingsCommunicationService.UpdateServerInstance

Aktualizuje instancję serwera.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

150.1 Składnia

```
C#  
  
bool UpdateServerInstance (  
    ReCourtServerInstance instance  
)
```

150.1.1.1 Parametry

instance

Typ: **ReCourtServerInstance**

Instancja

150.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

150.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)



151Metoda IRcsSettingsCommunicationService.UpdateTrialListView

Aktualizuje rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.RcsSettings.Base](#)
Moduł: ReCourt.Services.Communication.RcsSettings.Base (w
ReCourt.Services.Communication.RcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

151.1 Składnia

C#	
<pre>bool ReCourtTrialListView)</pre>	<pre>UpdateTrialListView(trialListView</pre>

151.1.1.1 Parametry

trialListView

Typ: **ReCourtTrialListView**

Widok

151.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.


151.2 Zobacz również

[Interfejs IRcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.RcsSettings.Base](#)

152Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base

152.1 Interfejsy

	Interfejs	Opis
	IReadingRoom	Interfejs odpowiada za sprawdzanie symboli.



153 Interfejs IReadingRoom

Interfejs odpowiada za sprawdzanie symboli.

Przestrzeń nazw: [ReCourt.Services.Communication.ReadingRoom.Base](#)
Moduł: ReCourt.Services.Communication.ReadingRoom.Base (w
ReCourt.Services.Communication.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)



153.1 Składnia

C#

```
public interface IReadingRoom
```

IReadingRoom prezentuje składowe.

153.2 Metody

	Nazwa	Opis
	GetToken	Pobiera symbol.
	IsTokenValid	Sprawdza czy symbol jest poprawny.

153.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base](#)



154 Metody IReadingRoom.IReadingRoom

[IReadingRoom](#) prezentuje składowe.

154.1 Metody

	Nazwa	Opis
	GetToken	Pobiera symbol.
	IsTokenValid	Sprawdza czy symbol jest poprawny.

154.2 Zobacz również

[Interfejs IReadingRoom](#)

[Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base](#)



155 Metoda IReadingRoom.GetToken

Pobiera symbol.

Przestrzeń nazw: [ReCourt.Services.Communication.ReadingRoom.Base](#)
Moduł: ReCourt.Services.Communication.ReadingRoom.Base (w
ReCourt.Services.Communication.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

155.1 Składnia

C#	
Token	GetToken (
string	<i>token</i> ,
string	<i>documentNumber</i>
)	

155.1.1.1 Parametry

token

Typ: System.String

Token

documentNumber

Typ: System.String

Numer dokumentu

155.1.1.2 Zwraca

Typ:

Obiekt symbolu.

Token

155.2 Zobacz również

[Interfejs IReadingRoom](#)

[Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base](#)

156 Metoda IReadingRoom.IsTokenValid

Sprawdza czy symbol jest poprawny.

Przestrzeń nazw: [ReCourt.Services.Communication.ReadingRoom.Base](#)
Moduł: ReCourt.Services.Communication.ReadingRoom.Base (w
ReCourt.Services.Communication.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

156.1 Składnia

C#	
<pre>bool string out)</pre>	<pre>IsTokenValid(token, documentType</pre>

156.1.1.1 Parametry

token

Typ: System.String

Token

documentType

Typ: System.String

Typ dokumentu

156.1.1.2 Zwraca

Typ:

Boolean

True jeśli symbol poprawny, w przeciwnym razie false.


156.2 Zobacz również

[Interfejs IReadingRoom](#)

[Przestrzeń nazw ReCourt.Services.Communication.ReadingRoom.Base](#)

157Przestrzeń nazw ReCourt.Services.Communication.Reports.Base

157.1 Interfejsy

	Interfejs	Opis
	IReportsCommunicationService	Interfejs odpowiada za zarządzanie raportami oraz obsługę uprawnień oraz ról.



158 Interfejs IReportsCommunicationService

Interfejs odpowiada za zarządzanie raportami oraz obsługę uprawnień oraz ról.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
 Moduł: ReCourt.Services.Communication.Reports.Base (w
 ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

158.1 Składnia




















```
C#
public interface IReportsCommunicationService
```




IReportsCommunicationService prezentuje składowe.

158.2 Metody

	Nazwa	Opis
	AddCategory	Dodaje kategoryę.
	AddReportRightToCategory	Dodaje przywilej raportu do danej kategorii.
	AddReportRightToReport	Dodaje przywilej raportu do konkretnego raportu.
	AddReportSchedule	Dodaje harmonogram raportu.
	AreNewReportsAvailable	Zwraca informację czy dostępne są nowe raporty.
	DeleteCategory	Usuwa kategorię o zadany identyfikatorze.
	DeleteReport	Usuwa raport o podany identyfikatorze.
	DeleteReportRight	Usuwa przywilej powiązany z raportem.
	ExportReports	Exportuje raporty o określonych identyfikatorach.
	GetAllVersionForReport	Zwraca wszystkie wersje raportu o danym kodzie.
	GetReport	Zwraca raport.
	GetReportByCode	Zwraca raport o danym kodzie.
	GetReportCategories	Zwraca kategorie raportów.



 GetReportHistoryByUser	Zwraca historię raportów powiązanych z użytkownikiem.
 GetReportHistoryByUserPager	Zwraca stronicowaną historię raportów powiązanych z użytkownikiem.
 GetReports	Zwraca listę z raportami.
 GetReportSchedule	Zwraca harmonogram raportu.
 GetReportSchedulerAdminPager	Zwraca stronicowany harmonogram raportów ułożony względem administratora.
 GetReportSchedulerByUserPager	Zwraca stronicowany harmonogram raportów ułożony względem użytkowników.
 GetReportsPager	Zwraca stronicowaną listę raportów.
 GetReportWithRights	Zwraca listę raportów z przywilejami.
 GetReportWithRightsPager	Zwraca stronicowaną listę raportów z przywilejami.
 GetRolesRights	Zwraca listę roli z przywilejami.
 GetUserRights	Zwraca listę przywilejów użytkowników.
 ImportReports	Importuje podane raporty.
 PerformQuietReportOnServer	Tworzy cicho raport na serwerze.
 PerformReport	Tworzy raport o określonych parametrach.
 PerformReportOnServer	Tworzy raport .repx dla wskazanego raportu.
 RemoveAllReportHistory	Usuwa całą historię raportów dla danego użytkownika.
 RemoveReportHistory	Usuwa historię powiązaną z konkretnym raportem.
 RemoveReportSchedule	Usuwa harmonogram raportu.
 SwitchReportRight	Zmienia przywileje powiązane z raportem.

 UpdateCategory	Aktualizuje daną kategorię.
 UpdateReport	Aktualizuje konkretny raport.
 UpdateReportSchedule	Aktualizuje harmonogram raportu.

158.3 Zobacz również

[Przeźren nazw ReCourt.Services.Communication.Reports.Base](#)

159 Metody IReportsCommunicationService.IReportsCommunicationService

[IReportsCommunicationService](#) prezentuje składowe.

159.1 Metody

Nazwa	Opis
AddCategory	Dodaje kategorie.
AddReportRightToCategory	Dodaje przywilej raportu do danej kategorii.
AddReportRightToReport	Dodaje przywilej raportu do konkretnego raportu.
AddReportSchedule	Dodaje harmonogram raportu.
AreNewReportsAvailable	Zwraca informację czy dostępne są nowe raporty.
DeleteCategory	Usuwa kategorię o zadanym identyfikatorze.
DeleteReport	Usuwa raport o podanym identyfikatorze.
DeleteReportRight	Usuwa przywilej powiązany z raportem.
ExportReports	Exportuje raporty o określonych identyfikatorach.
GetAllVersionForReport	Zwraca wszystkie wersje raportu o danym kodzie.
GetReport	Zwraca raport.
GetReportByCode	Zwraca raport o danym kodzie.
GetReportCategories	Zwraca kategorie raportów.
GetReportHistoryByUser	Zwraca historię raportów powiązanych z użytkownikiem.
GetReportHistoryByUserPager	Zwraca stronicowaną historię raportów powiązanych z użytkownikiem.
GetReports	Zwraca listę z raportami.
GetReportSchedule	Zwraca harmonogram raportu.



GetReportSchedulerAdminPager	Zwraca stronicowany harmonogram raportów ułożony względem administratora.
GetReportSchedulerByUserPager	Zwraca stronicowany harmonogram raportów ułożony względem użytkowników.
GetReportsPager	Zwraca stronicowaną listę raportów.
GetReportWithRights	Zwraca listę raportów z przywilejami.
GetReportWithRightsPager	Zwraca stronicowaną listę raportów z przywilejami.
GetRolesRights	Zwraca listę roli z przywilejami.
GetUserRights	Zwraca listę przywilejów użytkowników.
ImportReports	Importuje podane raporty.
PerformQuietReportOnServer	Tworzy cicho raport na serwerze.
PerformReport	Tworzy raport o określonych parametrach.
PerformReportOnServer	Tworzy raport .repx dla wskazanego raportu.
RemoveAllReportHistory	Usuwa całą historię raportów dla danego użytkownika.
RemoveReportHistory	Usuwa historię powiązaną z konkretnym raportem.
RemoveReportSchedule	Usuwa harmonogram raportu.
SwitchReportRight	Zmienia przywileje powiązane z raportem.
UpdateCategory	Aktualizuje daną kategorię.
UpdateReport	Aktualizuje konkretny raport.
UpdateReportSchedule	Aktualizuje harmonogram raportu.

159.2 Zobacz również

[Interfejs IReportsCommunicationService](#)



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

160Metoda IReportsCommunicationService.AddCategory

Dodaje kategorie.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

160.1 Składnia

C#	
<pre>bool ReCourtReportCategoryVO)</pre>	<pre>AddCategory(categoryVO</pre>

160.1.1.1 Parametry

categoryVO

Typ: **ReCourtReportCategoryVO**

Kategoria

160.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

160.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

161Metoda IReportsCommunicationService.AddReportRightToCategory

Dodaje przywilej raportu do danej kategorii.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

161.1 Składnia

```
C#  
  
bool AddReportRightToCategory(  
    ReCourtReportRight reportRight  
)
```

161.1.1.1 Parametry

reportRight

Typ: **ReCourtReportRight**

Uprawnienia

161.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

161.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

162 Metoda IReportsCommunicationService.AddReportRightToReport

Dodaje przywilej raportu do konkretnego raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

162.1 Składnia

```
C#  
  
bool AddReportRightToReport (  
    ReCourtReportRight reportRight  
)
```

162.1.1.1 Parametry

reportRight

Typ: **ReCourtReportRight**

Uprawnienia

162.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

162.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

163 Metoda `IReportsCommunicationService.AddReportSchedule`

Dodaje harmonogram raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)

Moduł: `ReCourt.Services.Communication.Reports.Base` (w

`ReCourt.Services.Communication.Reports.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

163.1 Składnia

```
C#  
  
int AddReportSchedule (  
    ReCourtReportSchedule schedule  
)
```

163.1.1.1 Parametry

schedule

Typ: **ReCourtReportSchedule**

Harmonogram

163.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanego harmonogramu raportu.

163.2 Zobacz również

[Interfejs `IReportsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Reports.Base`](#)



164 Metoda IReportsCommunicationService.AreNewReportsAvailable

Zwraca informację czy dostępne są nowe raporty.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

164.1 Składnia

```
C#  
  
bool AreNewReportsAvailable(  
    int userId,  
    DateTime lastChecked  
)
```

164.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

lastChecked

Typ: System.DateTime

Data ostatniego sprawdzenia

164.1.1.2 Zwraca

Typ:

Boolean

True jeśli są dostępne, w przeciwnym razie false.

164.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

165 Metoda IReportsCommunicationService.DeleteCategory

Usuwa kategorię o zadanim identyfikatorze.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

165.1 Składnia

```
C#  
  
bool DeleteCategory(  
    int categoryId  
)
```

165.1.1.1 Parametry

categoryId

Typ: System.Int32

Id kategorii

165.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

165.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

166Metoda IReportsCommunicationService.DeleteReport

Usuwa raport o podanym identyfikatorze.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

166.1 Składnia

```
C#  
  
bool DeleteReport(  
    int reportId  
)
```

166.1.1.1 Parametry

reportId

Typ: System.Int32

Id raportu

166.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

166.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

167Metoda IReportsCommunicationService.DeleteReportRight

Usuwa przywilej powiązany z raportem.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

167.1 Składnia

```
C#  
  
bool DeleteReportRight (  
    int rightId  
)
```

167.1.1.1 Parametry

rightId

Typ: System.Int32

Id uprawnienia

167.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

167.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

168Metoda IReportsCommunicationService.ExportReports

Exportuje raporty o określonych identyfikatorach.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

168.1 Składnia

C#	
<pre>ICollection<ReCourtReport> ICollection<int>)</pre>	<pre>ExportReports (reportIds</pre>

168.1.1.1 Parametry

reportIds

Typ: System.Collections.Generic.ICollection(Int32)

Identyfikatory raportów

168.1.1.2 Zwraca

Typ: **ICollection(ReCourtReport)**
Kolekcja wyeksportowanych raportów.

168.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

169Metoda IReportsCommunicationService.GetAllVersionForReport

Zwraca wszystkie wersje raportu o danym kodzie.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

169.1 Składnia

```
C#  
  
List<ReCourtReport> GetAllVersionForReport (  
    string reportCode  
)
```

169.1.1.1 Parametry

reportCode

Typ: System.String

Kod raportu

169.1.1.2 Zwraca

Typ:

Lista raportów.

List(ReCourtReport)

169.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)



170Metoda IReportsCommunicationService.GetReport

Zwraca raport.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

170.1 Składnia

C#	
ReCourtReport int)	GetReport (<i>reportId</i>

170.1.1.1 Parametry

reportId

Typ: System.Int32

Id raportu

170.1.1.2 Zwraca

Typ:

ReCourtReport

Obiekt raportu.

170.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

171Metoda IReportsCommunicationService.GetReportByCode

Zwraca raport o danym kodzie.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

171.1 Składnia

C#	
<pre>ReCourtReport string)</pre>	<pre>GetReportByCode (reportCode</pre>

171.1.1.1 Parametry

reportCode

Typ: System.String

Kod raportu

171.1.1.2 Zwraca

Typ:

ReCourtReport

Obiekt raportu.

171.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

172Metoda IReportsCommunicationService.GetReportCategories

Zwraca kategorie raportów.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

172.1 Składnia

C#

```
List<ReCourtReportCategoryVO> GetReportCategories()
```

172.1.1.1 Zwraca

Typ: **List(ReCourtReportCategoryVO)**
Lista obiektów kategorii raportów.

172.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

173 Metoda `IReportsCommunicationService.GetReportHistoryByUser`

Zwraca historię raportów powiązanych z użytkownikiem.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: `ReCourt.Services.Communication.Reports.Base` (w `ReCourt.Services.Communication.Reports.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

173.1 Składnia

```
C#  
  
List<ReportHistoryVO> GetReportHistoryByUser (  
    int                userId,  
    Nullable<int>     lastRecordsCount  
)
```

173.1.1.1 Parametry

userId

Typ: `System.Int32`

Id użytkownika

lastRecordsCount

Typ: `System.Nullable(Int32)`

Ostatnia ilość rekordów

173.1.1.2 Zwraca

Typ:

List(ReportHistoryVO)

Lista raportów powiązanych z konkretnym użytkownikiem.

173.2 Zobacz również

[Interfejs `IReportsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Reports.Base`](#)

174 Metoda `IReportsCommunicationService.GetReportHistoryByUserPagger`

Zwraca stronicowaną historię raportów powiązanych z użytkownikiem.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: `ReCourt.Services.Communication.Reports.Base` (w
`ReCourt.Services.Communication.Reports.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

174.1 Składnia

C#	
<pre>PagedList<ReportHistoryVO> int int int SortCriteria)</pre>	<pre>GetReportHistoryByUserPagger (<i>userId</i>, <i>pageNum</i>, <i>recordsPerPage</i>, <i>sortCriteria</i>)</pre>

174.1.1.1 Parametry

userId

Typ: `System.Int32`

Id użytkownika

pageNum

Typ: `System.Int32`

Strona

recordsPerPage

Typ: `System.Int32`

Ilość na stronie

sortCriteria

Typ: `SortCriteria`

Sortowanie

174.1.1.2 Zwraca

Typ: `PagedList<ReportHistoryVO>`

Stronicowana historia raportów powiązanych z użytkownikiem.

174.2 Zobacz również

[Interfejs `IReportsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Reports.Base`](#)

175 Metoda IReportsCommunicationService.GetReports

Zwraca listę z raportami.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

175.1 Składnia

```
C#  
  
List<ReCourtReportVO> GetReports (  
    ReCourtReportFilter filter  
)
```

175.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

175.1.1.2 Zwraca

Typ:

List(ReCourtReportVO)

Lista z raportami.

175.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

176Metoda IReportsCommunicationService.GetReportSchedule

Zwraca harmonogram raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

176.1 Składnia

C#	
ReCourtReportSchedule int)	GetReportSchedule(<i>scheduleId</i>)

176.1.1.1 Parametry

scheduleId

Typ: System.Int32

Id harmonogramu

176.1.1.2 Zwraca

Typ:

ReCourtReportSchedule

Obiekt harmonogramu raportu.

176.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

179Metoda IReportsCommunicationService.GetReportsPager

Zwraca stronicowaną listę raportów.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

179.1 Składnia

C#	
PagedList<ReCourtReportVO> ReCourtReportFilter int int SortCriteria)	GetReportsPager (<i>filter</i> , <i>pageNum</i> , <i>recordsPerPage</i> , <i>sortCriteria</i>)

179.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

179.1.1.2 Zwraca

Typ:

PagedList(ReCourtReportVO)

Stronicowana lista z raportami.

179.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

180Metoda IReportsCommunicationService.GetReportWithRights

Zwraca listę raportów z przywilejami.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

180.1 Składnia

```
C#  
  
List<ReCourtReportVO> GetReportWithRights (  
    ReCourtReportFilter filter  
)
```

180.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

180.1.1.2 Zwraca

Typ:

List(ReCourtReportVO)

Lista raportów z przywilejami.

180.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

181Metoda IReportsCommunicationService.GetReportWithRightsPager

Zwraca stronicowaną listę raportów z przywilejami.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

181.1 Składnia

C#	
<pre>PagedList<ReCourtReportVO> ReCourtReportFilter int int SortCriteria)</pre>	<pre>GetReportWithRightsPager (filter, pageNum, recordsPerPage, sortCriteria</pre>

181.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

181.1.1.2 Zwraca

Typ:

PagedList(ReCourtReportVO)

Stronicowana lista raportów z przywilejami.

181.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

182 Metoda IReportsCommunicationService.GetRolesRights

Zwraca listę ról z przywilejami.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

182.1 Składnia

```
C#  
  
List<ReCourtReportRoleRight> GetRolesRights (  
    ReCourtReportFilter filter  
)
```

182.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

182.1.1.2 Zwraca

Typ: **List(ReCourtReportRoleRight)**

Lista ról z przywilejami.

182.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

183 Metoda IReportsCommunicationService.GetUserRights

Zwraca listę przywilejów użytkowników.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

183.1 Składnia

```
C#  
  
List<ReCourtReportUserRight> GetUserRights (  
    ReCourtReportFilter filter  
)
```

183.1.1.1 Parametry

filter

Typ: **ReCourtReportFilter**

Filtr

183.1.1.2 Zwraca

Typ:

List(ReCourtReportUserRight)

Lista przywilejów użytkowników.

183.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)



184Metoda IReportsCommunicationService.ImportReports

Importuje podane raporty.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

184.1 Składnia

C#	
<pre>bool ICollection<ReCourtReport>)</pre>	<pre>ImportReports(reportDefinitionList</pre>

184.1.1.1 Parametry

reportDefinitionList

Typ: System.Collections.Generic.ICollection(**ReCourtReport**)

Lista definicji

184.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

184.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)



185 Metoda IReportsCommunicationService.PerformQuietReportOnServer

Tworzy cicho raport na serwerze.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

185.1 Składnia

```
C#  
  
StreamReport                      PerformQuietReportOnServer (  
    StreamReport                      report  
)
```

185.1.1.1 Parametry

report

Typ: **StreamReport**

Raport

185.1.1.2 Zwraca

Typ:

StreamReport

Stworzony raport.

185.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

186 Metoda IReportsCommunicationService.PerformReport

Tworzy raport o określonych parametrach.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

186.1 Składnia

C#	
<pre>DataTable int ICollection<ReCourtReportParameterVO>)</pre>	<pre>PerformReport (reportId, parameters</pre>

186.1.1.1 Parametry

reportId

Typ: System.Int32

Id raportu

parameters

Typ: System.Collections.Generic.ICollection(**ReCourtReportParameterVO**)

Parametry

186.1.1.2 Zwraca

Typ:

Tabela z danymi.

DataTable

186.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

187Metoda IReportsCommunicationService.PerformReportOnServer

Tworzy raport .repx dla wskazanego raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
 Moduł: ReCourt.Services.Communication.Reports.Base (w
 ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

187.1 Składnia

```
C#
string PerformReportOnServer (
    ReCourtUser user,
    string reportCode,
    string templateFile,
    ICollection<ReCourtReportParameterVO> parameters,
    List<IQueryParser> customQueryParsers
)
```

187.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

reportCode

Typ: System.String

Kod raportu

templateFile

Typ: System.String

Szablon

parameters

Typ: System.Collections.Generic.ICollection(**ReCourtReportParameterVO**)

Parametry

customQueryParsers

Typ: System.Collections.Generic.List(**IQueryParser**)

Parsery parametrów

187.1.1.2 Zwraca

Typ:

String

Nazwa pliku .repx na serwerze.



187.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

188Metoda IReportsCommunicationService.RemoveAllReportHistory

Usuwa całą historię raportów dla danego użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

188.1 Składnia

```
C#  
  
bool RemoveAllReportHistory(  
    int userId  
)
```

188.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

188.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

188.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)



189Metoda IReportsCommunicationService.RemoveReportHistory

Usuwa historię powiązaną z konkretnym raportem.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

189.1 Składnia

```
C#  
  
bool RemoveReportHistory(  
    int userId,  
    int reportHistoryId  
)
```

189.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

reportHistoryId

Typ: System.Int32

Id historii

189.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

189.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

190Metoda IReportsCommunicationService.RemoveReportSchedule

Usuwa harmonogram raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

190.1 Składnia

```
C#  
  
bool RemoveReportSchedule(  
    int scheduleId  
)
```

190.1.1.1 Parametry

scheduleId

Typ: System.Int32

Id harmonogramu

190.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

190.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)



191Metoda IReportsCommunicationService.SwitchReportRight

Zmienia przywileje powiązane z raportem.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

191.1 Składnia

```
C#  
  
bool SwitchReportRight (  
    int rightId  
)
```

191.1.1.1 Parametry

rightId

Typ: System.Int32

Id uprawnienia

191.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

191.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

192Metoda IReportsCommunicationService.UpdateCategory

Aktualizuje daną kategorię.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

192.1 Składnia

C#	
<pre>bool ReCourtReportCategoryVO)</pre>	<pre>UpdateCategory(categoryVO</pre>

192.1.1.1 Parametry

categoryVO

Typ: **ReCourtReportCategoryVO**

Kategoria

192.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

192.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

193Metoda IReportsCommunicationService.UpdateReport

Aktualizuje konkretny raport.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: ReCourt.Services.Communication.Reports.Base (w
ReCourt.Services.Communication.Reports.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

193.1 Składnia

C#	
<pre>bool ReCourtReport)</pre>	<pre>UpdateReport (report</pre>

193.1.1.1 Parametry

report

Typ: **ReCourtReport**

Raport

193.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

193.2 Zobacz również

[Interfejs IReportsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Reports.Base](#)

194 Metoda `IReportsCommunicationService.UpdateReportSchedule`

Aktualizuje harmonogram raportu.

Przestrzeń nazw: [ReCourt.Services.Communication.Reports.Base](#)
Moduł: `ReCourt.Services.Communication.Reports.Base` (w
`ReCourt.Services.Communication.Reports.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

194.1 Składnia

```
C#  
  
bool UpdateReportSchedule (  
    ReCourtReportSchedule schedule  
)
```

194.1.1.1 Parametry

schedule

Typ: **ReCourtReportSchedule**

Harmonogram

194.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.


194.2 Zobacz również

[Interfejs `IReportsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Reports.Base`](#)

195 Przestrzeń nazw ReCourt.Services.Communication.Settings.Base

195.1 Interfejsy

	Interfejs	Opis
	ISettingsCommunicationService	Interfejs odpowiada za obsługę żądań pobrania wydziałów sądów, sądów oraz komunikację.



196 Interfejs ISettingsCommunicationService

Interfejs odpowiada za obsługę zadań pobrania wydziałów sądów, sądów oraz komunikację.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
 Moduł: ReCourt.Services.Communication.Settings.Base (w
 ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

196.1 Składnia

```
C#
public interface ISettingsCommunicationService
```

ISettingsCommunicationService prezentuje składowe.

196.2 Metody

	Nazwa	Opis
	GetCurrentServerName	Pobiera aktualną nazwę instancji.
	GetCurrentTimeFromServer	Data serwera.
	GetGlobalSettings	Zwraca globalne ustawienia.
	GetRefCodes	Zwraca kody odniesienia.
	GetUserSettings	Zwraca wszystkie ustawienia danego użytkownika.
	IsInstanceAuthorized	Czy instancja RCS jest autoryzowana
	SetGlobalSetting	Ustawienia konkretne ustawienie.
	SetUserSetting	Ustawia ustawienie.

196.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Settings.Base](#)

197 Metody `ISettingsCommunicationService`.`ISettingsCommunicationService`

[ISettingsCommunicationService](#) prezentuje składowe.

197.1 Metody

	Nazwa	Opis
	GetCurrentServerName	Pobiera aktualną nazwę instancji.
	GetCurrentTimeFromServer	Data serwera.
	GetGlobalSettings	Zwraca globalne ustawienia.
	GetRefCodes	Zwraca kody odniesienia.
	GetUserSettings	Zwraca wszystkie ustawienia danego użytkownika.
	IsInstanceAuthorized	Czy instancja RCS jest autoryzowana
	SetGlobalSetting	Ustawienia konkretne ustawienie.
	SetUserSetting	Ustawia ustawienie.

197.2 Zobacz również

[Interfejs `ISettingsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Settings.Base`](#)

198Metoda ISettingsCommunicationService.GetCurrentServerName

Pobiera aktualną nazwę instancji.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: ReCourt.Services.Communication.Settings.Base (w
ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

198.1 Składnia

C#

```
ReCourtDataInstance GetCurrentServerName ()
```

198.1.1.1 Zwraca

Typ:
Obiekt instancji sądu.

ReCourtDataInstance

198.2 Zobacz również

[Interfejs ISettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Settings.Base](#)

199Metoda ISettingsCommunicationService.GetCurrentTimeFromServer

Data serwera.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: ReCourt.Services.Communication.Settings.Base (w
ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

199.1 Składnia

C#

```
DateTime GetCurrentTimeFromServer ()
```

199.1.1.1 Zwraca

Typ: **DateTime**
Data.

199.2 Zobacz również

[Interfejs ISettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Settings.Base](#)

200Metoda ISettingsCommunicationService.GetGlobalSettings

Zwraca globalne ustawienia.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: ReCourt.Services.Communication.Settings.Base (w
ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

200.1 Składnia

```
C#  
  
Dictionary<string, ReCourtSystemVariableSetting>  
GetGlobalSettings ()
```

200.1.1.1 Zwraca

Typ: **Dictionary(String, ReCourtSystemVariableSetting)**
Obiekt ustawień.

200.2 Zobacz również

[Interfejs ISettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Settings.Base](#)



201Metoda ISettingsCommunicationService.GetRefCodes

Zwraca kody odniesienia.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: ReCourt.Services.Communication.Settings.Base (w
ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

201.1 Składnia

C#

```
List<ReCourtRefCode> GetRefCodes ()
```

201.1.1.1 Zwraca

Typ:
Lista kodów.

List(ReCourtRefCode)

201.2 Zobacz również

[Interfejs ISettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Settings.Base](#)



202 Metoda `ISettingsCommunicationService.GetUserSettings`

Zwraca wszystkie ustawienia danego użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: ReCourt.Services.Communication.Settings.Base (w
ReCourt.Services.Communication.Settings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

202.1 Składnia

```
C#  
  
List<ReCourtSetting> GetUserSettings (  
    int userId  
)
```

202.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

202.1.1.2 Zwraca

Typ:

List(ReCourtSetting)

Lista obiektów ustawień.

202.2 Zobacz również

[Interfejs `ISettingsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Settings.Base`](#)



203 Metoda `ISettingsCommunicationService.IsInstanceAuthorized`

Czy instancja RCS jest autoryzowana

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: `ReCourt.Services.Communication.Settings.Base` (w
`ReCourt.Services.Communication.Settings.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

203.1 Składnia

C#

```
bool IsInstanceAuthorized()
```

203.1.1.1 Zwraca

Typ: **Boolean**
Informacja o autoryzacji

203.2 Zobacz również

[Interfejs `ISettingsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Settings.Base`](#)



204 Metoda `ISettingsCommunicationService.SetGlobalSetting`

Ustawienia konkretne ustawienie.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: `ReCourt.Services.Communication.Settings.Base` (w
`ReCourt.Services.Communication.Settings.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

204.1 Składnia

```
C#  
  
bool SetGlobalSetting(  
    string code,  
    ReCourtSystemVariableSetting value  
)
```

204.1.1.1 Parametry

code

Typ: `System.String`

Kod

value

Typ: `ReCourtSystemVariableSetting`

Wartość

204.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

204.2 Zobacz również

[Interfejs `ISettingsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Settings.Base`](#)

205 Metoda `ISettingsCommunicationService.SetUserSetting`

Ustawia ustawienie.

Przestrzeń nazw: [ReCourt.Services.Communication.Settings.Base](#)
Moduł: `ReCourt.Services.Communication.Settings.Base` (w
`ReCourt.Services.Communication.Settings.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

205.1 Składnia

```
C#  
  
bool SetUserSetting(  
    int     userId,  
    string  code,  
    string  value  
)
```

205.1.1.1 Parametry

userId

Typ: `System.Int32`

Id użytkownika

code

Typ: `System.String`

Kod

value

Typ: `System.String`

Wartość

205.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

205.2 Zobacz również


[Interfejs `ISettingsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Settings.Base`](#)



206Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base

206.1 Interfejsy

	Interfejs	Opis
	ITaskListCommunicationService	Interfejs odpowiada za obsługę stanów zadań (np. sprawdzenie ich dostępności).

207 Interfejs ITaskListCommunicationService

Interfejs odpowiada za obsługę stanów zadań (np. sprawdzenie ich dostępności).

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
 Moduł: ReCourt.Services.Communication.TaskList.Base (w
 ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)


207.1 Składnia

C#

```
public interface ITaskListCommunicationService
```

ITaskListCommunicationService prezentuje składowe.

207.2 Metody

	Nazwa	Opis
	AreNewTasksAvailable	Sprawdza, czy od ostatniego pobrania listy zadań użytkownik dodał nowe zadania do tasklisty.
	GetTask	Pobiera zadanie z tasklisty o wybranym id.
	GetTaskParameter	Pobiera zadanie z tasklisty o wybranym id.
	GetTaskProgress	Pobiera wartość postępu zadania.
	GetUserTasks	Pobiera zadania tasklisty dla użytkownika.
	RemoveAllTasks	Usuwa wszystkie zadania użytkownika.
	RemoveTask	Usuwa zadanie z tasklisty użytkownika.
	UpdateTaskParams	Aktualizacja listy parametrów.

207.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base](#)

208 Metody `ITaskListCommunicationService.ITaskListCommunicationService`

[ITaskListCommunicationService](#) prezentuje składowe.

208.1 Metody

	Nazwa	Opis
⇒	AreNewTasksAvailable	Sprawdza, czy od ostatniego pobrania listy zadań użytkownik dodał nowe zadania do tasklisty.
⇒	GetTask	Pobiera zadanie z tasklisty o wybranym id.
⇒	GetTaskParameter	Pobiera zadanie z tasklisty o wybranym id.
⇒	GetTaskProgress	Pobiera wartość postępu zadania.
⇒	GetUserTasks	Pobiera zadania tasklisty dla użytkownika.
⇒	RemoveAllTasks	Usuwa wszystkie zadania użytkownika.
⇒	RemoveTask	Usuwa zadanie z tasklisty użytkownika.
⇒	UpdateTaskParams	Aktualizacja listy parametrów.

208.2 Zobacz również

[Interfejs `ITaskListCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskList.Base`](#)

209 Metoda `ITaskListCommunicationService.AreNewTasksAvailable`

Sprawdza, czy od ostatniego pobrania listy zadań użytkownik dodał nowe zadania do tasklisty.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)

Moduł: `ReCourt.Services.Communication.TaskList.Base` (w

`ReCourt.Services.Communication.TaskList.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

209.1 Składnia

C#	
<pre>bool int DateTime)</pre>	<pre>AreNewTasksAvailable(userId, lastChecked</pre>

209.1.1.1 Parametry

userId

Typ: `System.Int32`

Identyfikator użytkownika.

lastChecked

Typ: `System.DateTime`

Data oraz czas ostatniego pobrania listy.

209.1.1.2 Zwraca

Typ:

Boolean

True jeśli nowe zadania zostały dodane, false w przeciwnym wypadku.

209.2 Zobacz również

[Interfejs `ITaskListCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskList.Base`](#)

210Metoda ITaskListCommunicationService.GetTask

Pobiera zadanie z tasklisty o wybranym id.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: ReCourt.Services.Communication.TaskList.Base (w
ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

210.1 Składnia

C#	
ReCourtTask long)	GetTask (<i>id</i>

210.1.1.1 Parametry

id

Typ: System.Int64

Identyfikator zadania.

210.1.1.2 Zwraca

Typ:

ReCourtTask

Zadanie tasklisty użytkownika.

210.2 Zobacz również

[Interfejs ITaskListCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base](#)

211 Metoda `ITaskListCommunicationService.GetTaskParameter`

Pobiera zadanie z tasklisty o wybranym id.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: ReCourt.Services.Communication.TaskList.Base (w
ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

211.1 Składnia

C#	
<code>TaskListParameterBase</code>	<code>GetTaskParameter (</code>
<code> long</code>	<code> <i>id</i></code>
<code>)</code>	

211.1.1.1 Parametry

id

Typ: `System.Int64`

Identyfikator zadania.

211.1.1.2 Zwraca

Typ:

TaskListParameterBase

Zadanie tasklisty użytkownika.

211.2 Zobacz również

[Interfejs `ITaskListCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskList.Base`](#)



212Metoda ITaskListCommunicationService.GetTaskProgress

Pobiera wartość postępu zadania.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: ReCourt.Services.Communication.TaskList.Base (w
ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

212.1 Składnia

C#	
<pre>float TaskId)</pre>	<pre>GetTaskProgress (id</pre>

212.1.1.1 Parametry

id

Typ: **TaskId**

Identyfikator zadania.

212.1.1.2 Zwraca

Typ:

Single

Wartość postępu z zakresu 0.0 - 1.0.

212.2 Zobacz również

[Interfejs ITaskListCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base](#)

213 Metoda `ITaskListCommunicationService.GetUserTasks`

Pobiera zadania tasklisty dla użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: ReCourt.Services.Communication.TaskList.Base (w
ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

213.1 Składnia

```
C#
PagedList<ReCourtTask> GetUserTasks (
    TaskFilter           filter,
    int                  userId,
    int                  pageNum,
    int                  recordsPerPage,
    SortCriteria         sortCriteria
)
```

213.1.1.1 Parametry

filter

Typ: **TaskFilter**

Obiekt filtru.

userId

Typ: System.Int32

Identyfikator użytkownika, którego zadania zostaną pobrane.

pageNum

Typ: System.Int32

Numer strony.

recordsPerPage

Typ: System.Int32

Ilość rekordów na stronie.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania.

213.1.1.2 Zwraca

Typ:

PagedList(ReCourtTask)

Lista zawierająca zadania tasklisty użytkownika.



213.2 Zobacz również

[Interfejs ITaskListCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base](#)

214 Metoda `ITaskListCommunicationService.RemoveAllTasks`

Usuwa wszystkie zadania użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)

Moduł: `ReCourt.Services.Communication.TaskList.Base` (w

`ReCourt.Services.Communication.TaskList.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

214.1 Składnia

```
C#  
  
void RemoveAllTasks (  
    int userId  
)
```

214.1.1.1 Parametry

userId

Typ: `System.Int32`

Identyfikator użytkownika.

214.2 Zobacz również

[Interfejs `ITaskListCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskList.Base`](#)

215 Metoda `ITaskListCommunicationService.RemoveTask`

Usuwa zadanie z tasklisty użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: `ReCourt.Services.Communication.TaskList.Base` (w `ReCourt.Services.Communication.TaskList.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

215.1 Składnia

```
C#  
  
bool RemoveTask (  
    int     userId,  
    long    task  
)
```

215.1.1.1 Parametry

userId

Typ: `System.Int32`

Identyfikator użytkownika.

task

Typ: `System.Int64`

Identyfikator taska do usunięcia.

215.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

215.2 Zobacz również

[Interfejs `ITaskListCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskList.Base`](#)

216Metoda ITaskListCommunicationService.UpdateTaskParams

Aktualizacja listy parametrów.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskList.Base](#)
Moduł: ReCourt.Services.Communication.TaskList.Base (w
ReCourt.Services.Communication.TaskList.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

216.1 Składnia

C#	
<pre>bool ReCourtTask TaskListParameterBase)</pre>	<pre>UpdateTaskParams (task, param</pre>

216.1.1.1 Parametry

task

Typ: **ReCourtTask**

Zadanie z aktualnymi parametrami.

param

Typ: **TaskListParameterBase**

Parametr

216.1.1.2 Zwraca

Typ:

Boolean

True, jeśli operacja się powiedzie. W przeciwny razie false.


216.2 Zobacz również

[Interfejs ITaskListCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskList.Base](#)

217Przestrzeń nazw ReCourt.Services.Communication.TaskManager.Base

217.1 Interfejsy

Interfejs	Opis
 ITaskManagerCommunicationService	Interfejs odpowiada za zarządzanie zadaniami.



218 Interfejs `ITaskManagerCommunicationService`

Interfejs odpowiada za zarządzanie zadaniami.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskManager.Base](#)
Moduł: `ReCourt.Services.Communication.TaskManager.Base` (w
`ReCourt.Services.Communication.TaskManager.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

218.1 Składnia

C#

```
public interface ITaskManagerCommunicationService
```

`ITaskManagerCommunicationService` prezentuje składowe.

218.2 Metody

	Nazwa	Opis
	GetPendingTaskList	Pobiera listę zadań oczekujących.
	TryToCancelTask	Próbuje anulować zadanie.

218.3 Zobacz również



[Przestrzeń nazw `ReCourt.Services.Communication.TaskManager.Base`](#)

219 Metody

ITaskManagerCommunicationService.ITaskManagerCommunicationService

[ITaskManagerCommunicationService](#) prezentuje składowe.

219.1 Metody

	Nazwa	Opis
	GetPendingTaskList	Pobiera listę zadań oczekujących.
	TryToCancelTask	Próbuje anulować zadanie.

219.2 Zobacz również

[Interfejs ITaskManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskManager.Base](#)



220Metoda `ITaskManagerCommunicationService.GetPendingTaskList`

Pobiera listę zadań oczekujących.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskManager.Base](#)
Moduł: `ReCourt.Services.Communication.TaskManager.Base` (w
`ReCourt.Services.Communication.TaskManager.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

220.1 Składnia

C#

```
IList<TaskVO> GetPendingTaskList ()
```

220.1.1.1 Zwraca

Typ:

`IList(TaskVO)`

Lista zadań.

220.2 Zobacz również

[Interfejs `ITaskManagerCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TaskManager.Base`](#)

221Metoda ITaskManagerCommunicationService.TryToCancelTask

Próbuje anulować zadanie.

Przestrzeń nazw: [ReCourt.Services.Communication.TaskManager.Base](#)
Moduł: ReCourt.Services.Communication.TaskManager.Base (w
ReCourt.Services.Communication.TaskManager.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

221.1 Składnia

C#	
<pre>bool TaskId)</pre>	<pre>TryToCancelTask(id</pre>

221.1.1.1 Parametry

id

Typ: **TaskId**

Identyfikator anulowanego zadania.

221.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.


221.2 Zobacz również

[Interfejs ITaskManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TaskManager.Base](#)

222Przestrzeń nazw ReCourt.Services.Communication.TransferManager

222.1 Interfejsy

	Interfejs	Opis
	ITransferManagerCommunicationService	Interfejs odpowiada za zarządzanie przesyłem danych.

223 Interfejs ITransferManagerCommunicationService

Interfejs odpowiada za zarządzanie przesyłem danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
 Moduł: ReCourt.Services.Communication.TransferManager (w
 ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)










223.1 Składnia

C#

```
public interface ITransferManagerCommunicationService
```

ITransferManagerCommunicationService prezentuje składowe.

223.2 Metody

	Nazwa	Opis
	CheckIsTrialExists	Sprawdza czy sprawa o podanym trialId istnieje.
	FinalizeTransfer	Kończy przesyłanie danych.
	GetCaseFileHashes	Sprawdza czy sprawa o podanym trialId istnieje.
	IsAlive	Sprawdza czy sprawa o podanym trialId istnieje.
	RegisterTransfer	Rejestruje przesyłanie danych.
	SendContent	Wysyła zawartość.
	SynchronizationLockTrial	
	UpdateSynchronizeDestination	
	UpdateTranscriptionFiles	

223.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)

224 Metody

`ITransferManagerCommunicationService.ITransferManagerCommunicationService`

[ITransferManagerCommunicationService](#) prezentuje składowe.

224.1 Metody

	Nazwa	Opis
⇒	CheckIsTrialExists	Sprawdza czy sprawa o podanym trialId istnieje.
⇒	FinalizeTransfer	Kończy przesyłanie danych.
⇒	GetCaseFileHashes	Sprawdza czy sprawa o podanym trialId istnieje.
⇒	IsAlive	Sprawdza czy sprawa o podanym trialId istnieje.
⇒	RegisterTransfer	Rejestruje przesyłanie danych.
⇒	SendContent	Wysyła zawartość.
⇒	SynchronizationLockTrial	
⇒	UpdateSynchronizeDestination	
⇒	UpdateTranscriptionFiles	

224.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)



225 Metoda `ITransferManagerCommunicationService.CheckIsTrialExists`

Sprawdza czy sprawa o podanym `trialId` istnieje.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
Moduł: `ReCourt.Services.Communication.TransferManager` (w
`ReCourt.Services.Communication.TransferManager.dll`) Wersja: 1.0.0.0 (1.0.0.0)

225.1 Składnia

C#	
<pre>bool string)</pre>	<pre>CheckIsTrialExists(trialId</pre>

225.1.1.1 Parametry

trialId

Typ: `System.String`

Identyfikator sprawy

225.1.1.2 Zwraca

Typ:

Boolean

True jeśli istnieje, w przeciwnym razie false.

225.2 Zobacz również

[Interfejs `ITransferManagerCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferManager`](#)

226Metoda ITransferManagerCommunicationService.FinalizeTransfer

Kończy przesłanie danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
Moduł: ReCourt.Services.Communication.TransferManager (w
ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

226.1 Składnia

```
C#  
  
string FinalizeTransfer(  
    string guid,  
    long totalFileSize,  
    TrialInfo trialInfo  
)
```

226.1.1.1 Parametry

guid

Typ: System.String

Id transferu

totalFileSize

Typ: System.Int64

Rozmiar plików

trialInfo

Typ: **TrialInfo**

Dane sprawy

226.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

String

226.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)



227Metoda ITransferManagerCommunicationService.GetCaseFileHashes

Sprawdza czy sprawa o podanym trialId istnieje.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
Moduł: ReCourt.Services.Communication.TransferManager (w
ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

227.1 Składnia

C#		
Dictionary<string, string >	string>	GetCaseFileHashes (<i>trialId</i>)

227.1.1.1 Parametry

trialId

Typ: System.String

Identyfikator sprawy

227.1.1.2 Zwraca

Typ: Dictionary(String, String)

True jeśli istnieje, w przeciwnym razie false.

227.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)



228Metoda ITransferManagerCommunicationService.IsAlive

Sprawdza czy sprawa o podanym trialId istnieje.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
Moduł: ReCourt.Services.Communication.TransferManager (w
ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

228.1 Składnia

C#

```
bool IsAlive()
```

228.1.1.1 Zwraca

Typ:

Boolean

True jeśli istnieje, w przeciwnym razie false.

228.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)

229Metoda ITransferManagerCommunicationService.RegisterTransfer

Rejestruje przesłanie danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
 Moduł: ReCourt.Services.Communication.TransferManager (w
 ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

229.1 Składnia

```

C#
string RegisterTransfer (
    Guid source,
    string sourceName,
    string destinationDepartmentCode,
    string sourceCourtName,
    string trialId,
    string caseSignature,
    string discriminantCode,
    string courtName,
    string caseDescription,
    bool admin,
    string mode,
    string synchronizationType,
    long totalSize,
    Guid transferCode
)
  
```

229.1.1.1 Parametry

source

Typ: System.Guid

Guid rcs źródłowy

sourceName

Typ: System.String

Nazwa źródłowa rcs

destinationDepartmentCode

Typ: System.String

Docelowy wydział

sourceCourtName

Typ: System.String

Nazwa sądu źródłowego



trialId

Typ: System.String

Identyfikator sprawy

caseSignature

Typ: System.String

Sygnatura

discriminantCode

Typ: System.String

Wydział

courtName

Typ: System.String

Nazwa sądu

caseDescription

Typ: System.String

Opis sprawy

admin

Typ: System.Boolean

Czy transfera idministracyjny

mode

Typ: System.String

Tryb przesyłania

synchronizationType

Typ: System.String

Tryb synchronizacji

totalSize

Typ: System.Int64

Rozmiar sprawy

transferCode

Typ: System.Guid

[229.1.1.2](#) [Zwraca](#)

Typ:

Identyfikator transferu.

String

229.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)



231Metoda ITransferManagerCommunicationService.SynchronizationLockTrial

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)

Moduł: ReCourt.Services.Communication.TransferManager (w

ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

231.1 Składnia

```
C#
bool SynchronizationLockTrial(
    string trialId,
    bool isLocked
)
```

231.1.1.1 Parametry

trialId

Typ: System.String

isLocked

Typ: System.Boolean

231.1.1.2 Zwraca

Typ: **Boolean**

231.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)

232Metoda

ITransferManagerCommunicationService.UpdateSynchronizeDestination

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)
Moduł: ReCourt.Services.Communication.TransferManager (w
ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

232.1 Składnia

C#	
ReCourtTransfer	UpdateSynchronizeDestination (
ReCourtTransfer	<i>transfer</i>
))

232.1.1.1 Parametry

transfer

Typ: **ReCourtTransfer**

232.1.1.2 Zwraca

Typ: **ReCourtTransfer**

232.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)

233Metoda ITransferManagerCommunicationService.UpdateTranscriptionFiles

Przestrzeń nazw: [ReCourt.Services.Communication.TransferManager](#)

Moduł: ReCourt.Services.Communication.TransferManager (w

ReCourt.Services.Communication.TransferManager.dll) Wersja: 1.0.0.0 (1.0.0.0)

233.1 Składnia

C#	
<pre>bool string Dictionary<string, string)</pre>	<pre>UpdateTranscriptionFiles(discriminantCode, localTranscriptionFiles, trialId</pre>

233.1.1.1 Parametry

discriminantCode

Typ: System.String

localTranscriptionFiles

Typ: System.Collections.Generic.Dictionary(**String**, **String**)

trialId

Typ: System.String

233.1.1.2 Zwraca

Typ: **Boolean**


233.2 Zobacz również

[Interfejs ITransferManagerCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferManager](#)

234Przestrzeń nazw ReCourt.Services.Communication.TransferRepository.Base

234.1 Interfejsy

	Interfejs	Opis
	ITransferRepositoryCommunicationService	Interfejs odpowiada za obsługę przesyłu danych.

235 Interfejs ITransferRepositoryCommunicationService

Interfejs odpowiada za obsługę przesyłu danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)

Moduł: ReCourt.Services.Communication.TransferRepository.Base (w

ReCourt.Services.Communication.TransferRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)





235.1 Składnia

C#

```
public interface ITransferRepositoryCommunicationService
```

ITransferRepositoryCommunicationService prezentuje składowe.

235.2 Metody

	Nazwa	Opis
	CancelTransfer	Anuluje transfer.
	DeleteByType	Usuwa przez typ.
	GetFinishedTransfers	Pobiera listę skończonych transferów.
	GetFinishedTransfersPager	Pobiera stronicowaną listę skończonych transferów.
	GetTransfers	Pobiera listę transferów.
	GetTransfersPager	Pobiera stronicowaną listę transferów.
	IsAlive	Sprawdza połączenie.
	TransferCase	Wykonuje transfer sprawy.

235.3 Zobacz również



[Przestrzeń nazw ReCourt.Services.Communication.TransferRepository.Base](#)

236 Metody

`ITransferRepositoryCommunicationService`.`ITransferRepositoryCommunicationService`

[ITransferRepositoryCommunicationService](#) prezentuje składowe.

236.1 Metody

	Nazwa	Opis
	CancelTransfer	Anuluje transfer.
	DeleteByType	Usuwa przez typ.
	GetFinishedTransfers	Pobiera listę skończonych transferów.
	GetFinishedTransfersPager	Pobiera stronicowaną listę skończonych transferów.
	GetTransfers	Pobiera listę transferów.
	GetTransfersPager	Pobiera stronicowaną listę transferów.
	IsAlive	Sprawdza połączenie.
	TransferCase	Wykonuje transfer sprawy.

236.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)

237Metoda ITransferRepositoryCommunicationService.CancelTransfer

Anuluje transfer.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)
Moduł: ReCourt.Services.Communication.TransferRepository.Base (w
ReCourt.Services.Communication.TransferRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

237.1 Składnia

```
C#  
  
void CancelTransfer(  
    int transferId  
)
```

237.1.1.1 Parametry

transferId

Typ: System.Int32

Id transferu

237.2 Zobacz również

[Interfejs ITransferRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferRepository.Base](#)

238 Metoda `ITransferRepositoryCommunicationService.DeleteByType`

Usuwa przez typ.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)
Moduł: ReCourt.Services.Communication.TransferRepository.Base (w
ReCourt.Services.Communication.TransferRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

238.1 Składnia

```
C#  
  
void DeleteByType (  
    string type  
)
```

238.1.1.1 Parametry

type

Typ: System.String

Typ

238.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)



239 Metoda `ITransferRepositoryCommunicationService.GetFinishedTransfers`

Pobiera listę skończonych transferów.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)

Moduł: `ReCourt.Services.Communication.TransferRepository.Base` (w

`ReCourt.Services.Communication.TransferRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

239.1 Składnia

C#

```
List<ReCourtTransfer> GetFinishedTransfers ()
```

239.1.1.1 Zwraca

Typ:

List(ReCourtTransfer)

Lista transferów.

239.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)



241 Metoda `ITransferRepositoryCommunicationService.GetTransfers`

Pobiera listę transferów.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)

Moduł: `ReCourt.Services.Communication.TransferRepository.Base` (w

`ReCourt.Services.Communication.TransferRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

241.1 Składnia

C#

```
List<ReCourtTransfer> GetTransfers ()
```

241.1.1.1 Zwraca

Typ:

List(ReCourtTransfer)

Lista transferów.

241.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)

242 Metoda `ITransferRepositoryCommunicationService.GetTransfersPager`

Pobiera stronicowaną listę transferów.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TransferRepository.Base` (w
`ReCourt.Services.Communication.TransferRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

242.1 Składnia

C#	
<pre>PagedList<ReCourtTransfer> ReCourtTransferCriteria int int SortCriteria)</pre>	<pre>GetTransfersPager(transferCriteria, pageNum, recordsPerPage, sortCriteria</pre>

242.1.1.1 Parametry

transferCriteria

Typ: **ReCourtTransferCriteria**

Filtr

pageNum

Typ: `System.Int32`

Numer strony.

recordsPerPage

Typ: `System.Int32`

Ilość rekordów na stronie.

sortCriteria

Typ: **SortCriteria**

Kryteria sortowania.

242.1.1.2 Zwraca

Typ: **PagedList(ReCourtTransfer)**
Lista transferów.

242.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)

243 Metoda `ITransferRepositoryCommunicationService.IsAlive`

Sprawdza połączenie.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)
Moduł: ReCourt.Services.Communication.TransferRepository.Base (w
ReCourt.Services.Communication.TransferRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

243.1 Składnia

```
C#  
  
bool                                     IsAlive(  
    string                               addressRoot,  
    string                               binding  
)
```

243.1.1.1 Parametry

addressRoot

Typ: System.String

binding

Typ: System.String

243.1.1.2 Zwraca

Typ: **Boolean**

243.2 Zobacz również

[Interfejs `ITransferRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TransferRepository.Base`](#)

244 Metoda ITransferRepositoryCommunicationService.TransferCase

Wykonuje transfer sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TransferRepository.Base](#)
 Moduł: ReCourt.Services.Communication.TransferRepository.Base (w
 ReCourt.Services.Communication.TransferRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

244.1 Składnia

```

C#
bool TransferCase (
    ReCourtUser user,
    int caseId,
    Guid destination,
    string destinationName,
    string destinationDepartmentCode,
    string destinationCourtName,
    Nullable<DateTime> startDate,
    string trialId,
    string caseSignature,
    string discriminantCode,
    string courtName,
    string caseDescription,
    bool admin,
    string mode,
    string synchronizationType,
    Guid transerCase
)
    
```

244.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: System.Int32

Id sprawy

destination

Typ: System.Guid

Docelowy rcs

destinationName

Typ: System.String

Nazwa docelowego rcs

destinationDepartmentCode

Typ: System.String

Docelowy wydział

destinationCourtName

Typ: System.String

Docelowy sąd

startDate

Typ: System.Nullable(**DateTime**)

Data

trialId

Typ: System.String

Identyfikator sprawy

caseSignature

Typ: System.String

Sygnatura

discriminantCode

Typ: System.String

Wydział

courtName

Typ: System.String

Sąd

caseDescription

Typ: System.String

Opis

admin

Typ: System.Boolean

Czy tryb admina

mode

Typ: System.String

Tryb przesyłania

synchronizationType

Typ: System.String

Tryb synchronizacji

transerCase

Typ: System.Guid

244.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

244.2 Zobacz również


[Interfejs ITransferRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TransferRepository.Base](#)



245 Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base

245.1 Interfejsy

	Interfejs	Opis
	ITrialAuthoringCommunicationService	Interfejs odpowiada za obsługę rozpraw (tworzenie, zmiana pokoju bądź wyróżników).



246 Interfejs ITrialAuthoringCommunicationService

Interfejs odpowiada za obsługę rozpraw (tworzenie, zmiana pokoju bądź wyróżników).

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)

Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w

ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

246.1 Składnia

```
C#
public interface ITrialAuthoringCommunicationService
```

ITrialAuthoringCommunicationService prezentuje składowe.

246.2 Metody

	Nazwa	Opis
	ChangeDiscriminant	Zmienia wyróżnik sprawy.
	ChangeRoom	Zmienia pokój.
	ChangeSignature	Zmienia sygnaturę sprawy.
	CopyAudioAfterSignature	Kopiowanie plików po podpisie
	EraseFragment	Wymazuje fragment nagrania.
	ExtractAudio	Wyodrębnia dane audio.
	ExtractFragment	Wyodrębnia fragment nagrania.
	ExtractSeparatedAudio	Rozdziela wyodrębnione dane audio.
	ImportAudioWithExportMixed	Pobiera dane audio.
	OpenTrial	Otwiera sprawę.

246.3 Zobacz również











[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

247 Metody

ITrialAuthoringCommunicationService.ITrialAuthoringCommunicationService

[ITrialAuthoringCommunicationService](#) prezentuje składowe.

247.1 Metody

	Nazwa	Opis
	ChangeDiscriminant	Zmienia wyróżnik sprawy.
	ChangeRoom	Zmienia pokój.
	ChangeSignature	Zmienia sygnaturę sprawy.
	CopyAudioAfterSignature	Kopiowanie plików po podpisie
	EraseFragment	Wymazuje fragment nagrania.
	ExtractAudio	Wyodrębnia dane audio.
	ExtractFragment	Wyodrębnia fragment nagrania.
	ExtractSeparatedAudio	Rozdziela wyodrębnione dane audio.
	ImportAudioWithExportMixed	Pobiera dane audio.
	OpenTrial	Otwiera sprawę.

247.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

248Metoda ITrialAuthoringCommunicationService.ChangeDiscriminant

Zmienia wyróżnik sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

248.1 Składnia

C#	
<pre>bool ReCourtUser TrialInfo ReCourtDiscriminantVO string)</pre>	<pre>ChangeDiscriminant(user, trialInfo, newDiscriminant, taskListActionType</pre>

248.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

newDiscriminant

Typ: **ReCourtDiscriminantVO**

Nowy wydział

taskListActionType

Typ: System.String

Kontroler akcji

248.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

248.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

249Metoda ITrialAuthoringCommunicationService.ChangeRoom

Zmienia pokój.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
 Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
 ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

249.1 Składnia

C#	
bool	ChangeRoom (
ReCourtUser	<i>user</i> ,
TrialInfo	<i>trialInfo</i> ,
string	<i>newRoom</i> ,
string	<i>taskListActionType</i>
))

249.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

newRoom

Typ: System.String

Nowa sala

taskListActionType

Typ: System.String

Kontroler akcji

249.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

249.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

250Metoda ITrialAuthoringCommunicationService.ChangeSignature

Zmienia sygnaturę sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

250.1 Składnia

C#	
<pre>bool ReCourtUser TrialInfo string string)</pre>	<pre>ChangeSignature(user, trialInfo, newSignature, taskListActionType</pre>

250.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

newSignature

Typ: System.String

Nowa sygnatura

taskListActionType

Typ: System.String

Kontroler akcji

250.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

250.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

251Metoda ITrialAuthoringCommunicationService.CopyAudioAfterSignature

Kopiowanie plików po podpisie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
 Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
 ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

251.1 Składnia

C#	
bool	CopyAudioAfterSignature (
ReCourtUser	<i>user</i> ,
TrialInfo	<i>trialInfo</i> ,
string	<i>subDirName</i> ,
string	<i>outputFileName</i> ,
string	<i>fileExtensionSuffix</i> ,
string	<i>trialStatisticComment</i> ,
string	<i>taskListActionType</i> ,
string	<i>actionCode</i> ,
long	<i>relatedTaskId</i>
)	

251.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

subDirName

Typ: System.String

Nazwa katalogu

outputFileName

Typ: System.String

Fragment nazwy pliku

fileExtensionSuffix

Typ: System.String

Przedrostek nazwy

trialStatisticComment



Typ: System.String

taskListActionType

Typ: System.String

Kontroler akcji

actionCode

Typ: System.String

Kod akcji

relatedTaskId

Typ: System.Int64

Id zadania zależnego

[251.1.1.2](#) [Zwraca](#)

Typ:

Informacja o powodzeniu operacji.

Boolean

[251.2](#) [Zobacz również](#)

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

252Metoda ITrialAuthoringCommunicationService.EraseFragment

Wymazuje fragment nagrania.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
 Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
 ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

252.1 Składnia

```

C#
bool EraseFragment (
    ReCourtUser user,
    TrialInfo trialInfo,
    string audioCodec,
    float audioQuality,
    string videoCodec,
    float videoQuality,
    long eraseStartMillisecond,
    long eraseEndMillisecond,
    string taskListActionType
)
    
```

252.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

audioCodec

Typ: System.String

Kodek

audioQuality

Typ: System.Single

Jakość

videoCodec

Typ: System.String

Kodek

videoQuality



Typ: System.Single

Jakość

eraseStartMillisecond

Typ: System.Int64

Zamazanie od

eraseEndMillisecond

Typ: System.Int64

Zamazanie do

taskListActionType

Typ: System.String

Kontroler akcji

[252.1.1.2](#) [Zwraca](#)

Typ:

Informacja o powodzeniu operacji.

Boolean

[252.2](#) Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

253Metoda ITrialAuthoringCommunicationService.ExtractAudio

Wyodrębnia dane audio.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

253.1 Składnia

```
C#
bool ExtractAudio(
    ReCourtUser user,
    TrialInfo trialInfo,
    string codec,
    float quality,
    int bitrate,
    bool vbr,
    bool[] channels,
    string taskListActionType
)
```

253.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

codec

Typ: System.String

Kodek

quality

Typ: System.Single

Jakość

bitrate

Typ: System.Int32

Bitrate

vbr

Typ: System.Boolean

Zmienny bitrate

channels

Typ: System.Boolean[]

Kanały

taskListActionType

Typ: System.String

Kontroler akcji

[253.1.1.2](#) [Zwraca](#)

Typ:

Informacja o powodzeniu operacji.

Boolean

[253.2](#) Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

254Metoda ITrialAuthoringCommunicationService.ExtractFragment

Wyodrębnia fragment nagrania.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

254.1 Składnia

```
C#
bool ExtractFragment (
    ReCourtUser user,
    TrialInfo trialInfo,
    TimeSpan start,
    TimeSpan end,
    bool printLabel,
    TrialExportMode mode,
    string taskListActionType
)
```

254.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

start

Typ: System.TimeSpan

Od

end

Typ: System.TimeSpan

Do

printLabel

Typ: System.Boolean

Czy drukować opis

mode

Typ: **TrialExportMode**

Tryb

taskListActionType

Typ: System.String

Kontroler akcji

[254.1.1.2](#) [Zwraca](#)

Typ:

Informacja o powodzeniu operacji.

Boolean

[254.2](#) Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

255Metoda ITrialAuthoringCommunicationService.ExtractSeparatedAudio

Rozdziela wydrebnione dane audio.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

255.1 Składnia

C#	
<pre>bool ReCourtUser TrialInfo string)</pre>	<pre>ExtractSeparatedAudio(user, trialInfo, taskListActionType</pre>

255.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

taskListActionType

Typ: System.String

Kontroler akcji

255.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

255.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

256Metoda ITrialAuthoringCommunicationService.ImportAudioWithExportMixed

Pobiera dane audio.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
 Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
 ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

256.1 Składnia

```

C#
bool ImportAudioWithExportMixed(
    ReCourtUser user,
    TrialInfo trialInfo,
    string codec,
    float quality,
    string tempSubDirName,
    string taskListActionType
)
  
```

256.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

codec

Typ: System.String

Kodek

quality

Typ: System.Single

Jakość

tempSubDirName

Typ: System.String

Nazwa katalogu z plikami

taskListActionType

Typ: System.String

Kontroler akcji

256.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

256.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

257Metoda ITrialAuthoringCommunicationService.OpenTrial

Otwiera sprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialAuthoring.Base](#)
 Moduł: ReCourt.Services.Communication.TrialAuthoring.Base (w
 ReCourt.Services.Communication.TrialAuthoring.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

257.1 Składnia

```

C#
bool OpenTrial(
    ReCourtUser user,
    TrialInfo trialInfo,
    int userIdTo,
    DateTime dateTo,
    int annotationRegionStart,
    int annotationRegionEnd
)
  
```

257.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

userIdTo

Typ: System.Int32

Dla użytkownika

dateTo

Typ: System.DateTime

Do

annotationRegionStart

Typ: System.Int32

Miejsce adnotacji

annotationRegionEnd

Typ: System.Int32

Miejsce adnotacji

257.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean


257.2 Zobacz również

[Interfejs ITrialAuthoringCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialAuthoring.Base](#)

258Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base

258.1 Interfejsy

	Interfejs	Opis
	ITrialRepositoryCommunicationService	Interfejs odpowiada za zarządzanie danymi powiązаныmi z rozprawą.

259 Interfejs ITrialRepositoryCommunicationService

Interfejs odpowiada za zarządzanie danymi powiązаныmi z rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)

Moduł: ReCourt.Services.Communication.TrialRepository.Base (w

ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)













259.1 Składnia

C#

```
public interface ITrialRepositoryCommunicationService
```

ITrialRepositoryCommunicationService prezentuje składowe.
















259.2 Metody

	Nazwa	Opis
	ChangeAuthorizedDepartmentCase	Tworzy zadanie
	ChangeCapeDiscriminantRcsCase	Tworzy zadanie
	ChangeCapeSignatureCase	Tworzy zadanie
	ChangeCapeSupplyRcsCase	Tworzy zadanie
	ChangeCreatorRcsCase	Tworzy zadanie
	CheckDataIntegration	Sprawdza integrację danych.
	CloseTrial	Zamyka rozprawy o podanych identyfikatorach.
	CloseTrialByData	Zamyka podaną rozprawę.
	CreateChangeAuthorizedDepartmentGlobalTask	Tworzy zadanie
	CreateChangeCapeDiscriminantRcsGlobalTask	Tworzy zadanie
	CreateChangeCapeSupplyRcsGlobalTask	Tworzy zadanie
	CreateChangeCreatorRcsGlobalTask	Tworzy zadanie



 CreateDataIntegrationCheckTask	Tworzy zadanie sprawdzenia integracji danych.
 DeleteTestTrial	Usuwa sprawę testową z listy.
 EvenData	Wyrównuje dane sprawy w kierunku określonym parametrem.
 EvenDataForDiscriminantChange	Wyrównuje dane sprawy dla wyróżnika
 GetCaseIndexingResults	Zwraca informację o wynikach indeksowania wskazanych spraw.
 GetDataIntegrationResults	Zwraca informację o wynikach sprawdzania integralności wskazanych spraw.
 GetDiscriminantsByCode	Zwraca wyróżniki odpowiadające kodom.
 GetRecordingDataDescription	Usuwa pliki sprawy.
 GetTrial	Zwraca rozprawę.
 GetTrialAnnotations	Zwraca adnotacje powiązane z rozprawą.
 GetTrialAnnotationsHistory	Zwraca historię adnotacji powiązanych z zadaną rozprawą.
 GetTrialCapeStateHistory	Zwraca listę statusów przetwarzania sprawy w CAPE
 GetTrialDocument	Zwraca dokument powiązany z rozprawą.
 GetTrialDocumentsVO	Zwraca dokumenty powiązane z rozprawą.
 GetTrialFiles	Zwraca pliki powiązane z rozprawą.
 GetTrialProtocol	Zwraca protokół rozprawy.



 GetTrialProtocolHistory	Zwraca historię protokołu powiązanego z zadaną rozprawą.
 GetTrials	Zwraca stronicowane rozprawy.
 GetTrialsForRecording	Zwraca rozprawy do nagrania.
 GetTrialTimeline	Zwraca oś czasu rozprawy.
 IndexTrials	Indeksuje rozprawy.
 MarkTrialAsMovedToCKIS	Oznacza sprawę jako przeniesioną do CKIS.
 ReCloseTrial	Ponownie zamyka rozprawę.
 ReIndexTrial	Reindeksuje rozprawę.
 RemoveTrialFiles	Usuwa pliki sprawy.
 SetSetCaseCapeStatus	Ustawia status cape
 StoreBurnResult	Zapisuje wynik nagrania spraw.
 StoreIndexingResult	Zapisuje wynik indeksowania sprawy.
 UpdateDataIntegrationResult	Aktualizuje wynik sprawdzania integralności sprawy.
 UpdateTrialAnnotations	Aktualizuje listę adnotacji rozprawy.
 UpdateTrialData	Aktualizuje dane sprawy.

259.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

260Metody

















ITrialRepositoryCommunicationService.ITrialRepositoryCommunicationService

[ITrialRepositoryCommunicationService](#) prezentuje składowe.












260.1 Metody

	Nazwa	Opis
⇒	ChangeAuthorizedDepartmentCase	Tworzy zadanie
⇒	ChangeCapeDiscriminantRcsCase	Tworzy zadanie
⇒	ChangeCapeSignatureCase	Tworzy zadanie
⇒	ChangeCapeSupplyRcsCase	Tworzy zadanie
⇒	ChangeCreatorRcsCase	Tworzy zadanie
⇒	CheckDataIntegration	Sprawdza integrację danych.
⇒	CloseTrial	Zamyka rozprawy o podanych identyfikatorach.
⇒	CloseTrialByData	Zamyka podaną rozprawę.
⇒	CreateChangeAuthorizedDepartmentGlobalTask	Tworzy zadanie
⇒	CreateChangeCapeDiscriminantRcsGlobalTask	Tworzy zadanie
⇒	CreateChangeCapeSupplyRcsGlobalTask	Tworzy zadanie
⇒	CreateChangeCreatorRcsGlobalTask	Tworzy zadanie
⇒	CreateDataIntegrationCheckTask	Tworzy zadanie sprawdzenia integracji danych.
⇒	DeleteTestTrial	Usuwa sprawę testową z listy.
⇒	EvenData	Wyrównuje dane sprawy w kierunku określonym parametrem.
⇒	EvenDataForDiscriminantChange	Wyrównuje dane sprawy dla wyróżnika



 GetCaseIndexingResults	Zwraca informację o wynikach indeksowania wskazanych spraw.
 GetDataIntegrationResults	Zwraca informację o wynikach sprawdzania integralności wskazanych spraw.
 GetDiscriminantsByCode	Zwraca wyróżniki odpowiadające kodom.
 GetRecordingDataDescription	Usuwa pliki sprawy.
 GetTrial	Zwraca rozprawę.
 GetTrialAnnotations	Zwraca adnotacje powiązane z rozprawą.
 GetTrialAnnotationsHistory	Zwraca historię adnotacji powiązanych z zadaną rozprawą.
 GetTrialCapeStateHistory	Zwraca listę statusów przetwarzania sprawy w CAPE
 GetTrialDocument	Zwraca dokument powiązany z rozprawą.
 GetTrialDocumentsVO	Zwraca dokumenty powiązane z rozprawą.
 GetTrialFiles	Zwraca pliki powiązane z rozprawą.
 GetTrialProtocol	Zwraca protokół rozprawy.
 GetTrialProtocolHistory	Zwraca historię protokołu powiązanego z zadaną rozprawą.
 GetTrials	Zwraca stronicowane rozprawy.
 GetTrialsForRecording	Zwraca rozprawy do nagrania.
 GetTrialTimeline	Zwraca oś czasu rozprawy.



 IndexTrials	Indeksuje rozprawę.
 MarkTrialAsMovedToCKIS	Oznacza sprawę jako przeniesioną do CKIS.
 ReCloseTrial	Ponownie zamyka rozprawę.
 ReIndexTrial	Reindeksuje rozprawę.
 RemoveTrialFiles	Usuwa pliki sprawy.
 SetSetCaseCapeStatus	Ustawia status cape
 StoreBurnResult	Zapisuje wynik nagania spraw.
 StoreIndexingResult	Zapisuje wynik indeksowania sprawy.
 UpdateDataIntegrationResult	Aktualizuje wynik sprawdzania integralności sprawy.
 UpdateTrialAnnotations	Aktualizuje listę adnotacji rozpraw.
 UpdateTrialData	Aktualizuje dane sprawy.

260.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

261Metoda

ITrialRepositoryCommunicationService.ChangeAuthorizedDepartmentCase

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

261.1 Składnia

```
C#  
  
bool ChangeAuthorizedDepartmentCase (  
    ReCourtUser user,  
    int caseId,  
    List<string> authorizedDepartments  
)
```

261.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: System.Int32

Id sprawy

authorizedDepartments

Typ: System.Collections.Generic.List(**String**)

Nowa wartość

261.1.1.2 Zwraca

Typ:

Boolean

Informacja o poprawności

261.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

262Metoda

ITrialRepositoryCommunicationService.ChangeCapeDiscriminantRcsCase

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

262.1 Składnia

```
C#  
  
bool ChangeCapeDiscriminantRcsCase (  
    ReCourtUser user,  
    int caseId,  
    string newValue  
)
```

262.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: System.Int32

Id sprawy

newValue

Typ: System.String

Nowa wartość

262.1.1.2 Zwraca

Typ:

Boolean

Informacja o poprawności

262.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

263 Metoda `ITrialRepositoryCommunicationService.ChangeCapeSignatureCase`

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w
`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

263.1 Składnia

```
C#  
  
bool ChangeCapeSignatureCase (  
    ReCourtUser user,  
    int caseId,  
    string newValue  
)
```

263.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: `System.Int32`

Id sprawy

newValue

Typ: `System.String`

Nowa wartość

263.1.1.2 Zwraca

Typ:

Informacja o poprawności

Boolean

263.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

264 Metoda `ITrialRepositoryCommunicationService.ChangeCapeSupplyRcsCase`

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w
`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

264.1 Składnia

```
C#  
  
bool ChangeCapeSupplyRcsCase (  
    ReCourtUser user,  
    int caseId,  
    string newValue  
)
```

264.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: `System.Int32`

Id sprawy

newValue

Typ: `System.String`

Nowa wartość

264.1.1.2 Zwraca

Typ:

Informacja o poprawności

Boolean

264.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

265 Metoda `ITrialRepositoryCommunicationService.ChangeCreatorRcsCase`

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w
`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

265.1 Składnia

```
C#  
  
bool ChangeCreatorRcsCase (  
    ReCourtUser user,  
    int caseId,  
    string newValue  
)
```

265.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: System.Int32

Id sprawy

newValue

Typ: System.String

Nowa wartość

265.1.1.2 Zwraca

Typ:

Informacja o poprawności

Boolean

265.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

266Metoda ITrialRepositoryCommunicationService.CheckDataIntegration

Sprawdza integrację danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

266.1 Składnia

```
C#  
  
DataIntegration CheckDataIntegration (  
    ReCourtUser user,  
    int id  
)
```

266.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

id

Typ: System.Int32

Id integracji

266.1.1.2 Zwraca

Typ:

Wynik sprawdzenia.

DataIntegration

266.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



267Metoda ITrialRepositoryCommunicationService.CloseTrial

Zamyka rozprawy o podanych identyfikatorach.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

267.1 Składnia

```
C#  
  
bool CloseTrial(  
    ReCourtUser user,  
    int ids  
)
```

267.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

ids

Typ: System.Int32

Id sprawy

267.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

267.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

268Metoda ITrialRepositoryCommunicationService.CloseTrialByData

Zamyka podaną rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

268.1 Składnia

```
C#  
  
bool CloseTrialByData (  
    ReCourtUser user,  
    TrialInfo ids  
)
```

268.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

ids

Typ: **TrialInfo**

Sprawa

268.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

268.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

269 Metoda

ITrialRepositoryCommunicationService.CreateChangeAuthorizedDepartmentGlobalTask

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

269.1 Składnia

```
C#  
  
void CreateChangeAuthorizedDepartmentGlobalTask (  
    ReCourtUser user,  
    List<string> discriminantCodes,  
    bool onlyEmpty,  
    List<string> authorizedDepartments  
)
```

269.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

onlyEmpty

Typ: System.Boolean

Czy puste

authorizedDepartments

Typ: System.Collections.Generic.List(**String**)

Nowa wartość

269.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

270Metoda

ITrialRepositoryCommunicationService.CreateChangeCapeDiscriminantRcsGlobalTask

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

270.1 Składnia

```
C#  
  
void CreateChangeCapeDiscriminantRcsGlobalTask(  
    ReCourtUser user,  
    List<string> discriminantCodes,  
    bool onlyEmpty,  
    string newValue  
)
```

270.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

onlyEmpty

Typ: System.Boolean

Czy puste

newValue

Typ: System.String

Nowa wartość

270.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

271Metoda

`ITrialRepositoryCommunicationService.CreateChangeCapeSupplyRcsGlobalTask`

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w `ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

271.1 Składnia

```
C#  
  
void CreateChangeCapeSupplyRcsGlobalTask (  
    ReCourtUser user,  
    List<string> discriminantCodes,  
    bool onlyEmpty,  
    string newValue  
)
```

271.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminantCodes

Typ: `System.Collections.Generic.List<String>`

Wydziały

onlyEmpty

Typ: `System.Boolean`

Czy puste

newValue

Typ: `System.String`

Nowa wartość

271.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

272Metoda

ITrialRepositoryCommunicationService.CreateChangeCreatorRcsGlobalTask

Tworzy zadanie

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

272.1 Składnia

```
C#  
  
void CreateChangeCreatorRcsGlobalTask(  
    ReCourtUser user,  
    List<string> discriminantCodes,  
    bool onlyEmpty,  
    string newValue  
)
```

272.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

onlyEmpty

Typ: System.Boolean

Czy puste

newValue

Typ: System.String

Nowa wartość

272.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

273 Metoda

ITrialRepositoryCommunicationService.CreateDataIntegrationCheckTask

Tworzy zadanie sprawdzenia integracji danych.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

273.1 Składnia

```
C#  
  
void CreateDataIntegrationCheckTask(  
    ReCourtUser user,  
    List<string> discriminantCodes  
)
```

273.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

273.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



274Metoda ITrialRepositoryCommunicationService.DeleteTestTrial

Usuwa sprawę testową z listy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

274.1 Składnia

C#	
<pre>bool ReCourtUser int)</pre>	<pre>DeleteTestTrial(currentUser, casId</pre>

274.1.1.1 Parametry

currentUser

Typ: **ReCourtUser**

Użytkownik

casId

Typ: System.Int32

Id sprawy

274.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

274.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

275 Metoda `ITrialRepositoryCommunicationService.EventData`

Wyrównuje dane sprawy w kierunku określonym parametrem.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

275.1 Składnia

```
C#
bool EvenData (
    ReCourtUser user,
    DataIntegration dataIntegration,
    UpdateDirection direction
)
```

275.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

dataIntegration

Typ: **DataIntegration**

Wynik integracji

direction

Typ: **UpdateDirection**

Kierunek

275.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

275.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

276Metoda

ITrialRepositoryCommunicationService.EventDataForDiscriminantChange

Wyrównuje dane sprawy dla wyróżnika

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

276.1 Składnia

```
C#
bool EvenDataForDiscriminantChange (
    ReCourtUser user,
    ReCourtDiscriminant discriminant,
    bool isIndividualDiscriminant = false
)
```

276.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

discriminant

Typ: **ReCourtDiscriminant**

Wydział

isIndividualDiscriminant (Optional)

Typ: System.Boolean

276.1.1.2 Zwraca

Typ:

Informacja o powodzeniu

Boolean

276.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

277Metoda ITrialRepositoryCommunicationService.GetCaseIndexingResults

Zwraca informację o wynikach indeksowania wskazanych spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

277.1 Składnia

C#	
<pre>PagedList<CaseIndexingResult> List<string> int int)</pre>	<pre>GetCaseIndexingResults (<i>discriminantCodes</i>, <i>pageNum</i>, <i>recordsPerPage</i>)</pre>

277.1.1.1 Parametry

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

277.1.1.2 Zwraca

Typ: **PagedList(CaseIndexingResult)**

Lista wyników indeksowania.

277.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

278Metoda ITrialRepositoryCommunicationService.GetDataIntegrationResults

Zwraca informację o wynikach sprawdzania integralności wskazanych spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

278.1 Składnia

C#	
PagedList<DataIntegration> List<string> int int SortCriteria)	GetDataIntegrationResults (<i>discriminantCodes</i> , <i>pageNum</i> , <i>recordsPerPage</i> , <i>sortCriteria</i>

278.1.1.1 Parametry

discriminantCodes

Typ: System.Collections.Generic.List(**String**)

Wydziały

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

278.1.1.2 Zwraca

Typ:

Lista wyników sprawdzania integralności.

PagedList(DataIntegration)

278.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

279 Metoda `ITrialRepositoryCommunicationService.GetDiscriminantsByCode`

Zwraca wyróżniki odpowiadające kodom.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

279.1 Składnia

C#	
<pre>List<ReCourtDiscriminantVO> string[])</pre>	<pre>GetDiscriminantsByCode (<i>discriminantCodes</i>)</pre>

279.1.1.1 Parametry

discriminantCodes

Typ: System.String[]

Wydziały

279.1.1.2 Zwraca

Typ: **List(ReCourtDiscriminantVO)**
Lista obiektów wyróżników.

279.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

280Metoda ITrialRepositoryCommunicationService.GetRecordingDataDescription

Usuwa pliki sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

280.1 Składnia

C#	
DataDescription ReCourtUser int)	GetRecordingDataDescription (<i>user</i> , <i>casId</i>)

280.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

casId

Typ: System.Int32

Id sprawy

280.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

DataDescription

280.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



281Metoda ITrialRepositoryCommunicationService.GetTrial

Zwraca rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

281.1 Składnia

C#	
<pre>TrialInfo int)</pre>	<pre>GetTrial(id</pre>

281.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

281.1.1.2 Zwraca

Typ:

TrialInfo

Obiekt informacji o rozprawie.

281.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

282Metoda ITrialRepositoryCommunicationService.GetTrialAnnotations

Zwraca adnotacje powiązane z rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

282.1 Składnia

C#	
<pre>List<Annotation> int string)</pre>	<pre>GetTrialAnnotations(caseId, type</pre>

282.1.1.1 Parametry

caseId

Typ: System.Int32

Id sprawy

type

Typ: System.String

Rodzaj adnotacji

282.1.1.2 Zwraca

Typ:

Lista adnotacji.

List(Annotation)

282.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

283 Metoda `ITrialRepositoryCommunicationService.GetTrialAnnotationsHistory`

Zwraca historię adnotacji powiązanych zadaną rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w
`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

283.1 Składnia

```
C#  
  
List<AnnotationHistory> GetTrialAnnotationsHistory(  
    int trialId,  
    string type  
)
```

283.1.1.1 Parametry

trialId

Typ: `System.Int32`

Id sprawy

type

Typ: `System.String`

Rodzaj adnotacji

283.1.1.2 Zwraca

Typ:

Lista obiektów historii adnotacji.

List(AnnotationHistory)

283.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

284 Metoda `ITrialRepositoryCommunicationService.GetTrialCapeStateHistory`

Zwraca listę statusów przetwarzania sprawy w CAPE

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)

Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w

`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

284.1 Składnia

C#	
<pre>List<TrialCapeState> int)</pre>	<pre>GetTrialCapeStateHistory(trialId</pre>

284.1.1.1 Parametry

trialId

Typ: `System.Int32`

Id sprawy

284.1.1.2 Zwraca

Typ:

lista statusów przetwarzania sprawy w CAPE

List(TrialCapeState)

284.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)

285 Metoda `ITrialRepositoryCommunicationService.GetTrialDocument`

Zwraca dokument powiązany z rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

285.1 Składnia

C#	
<code>TrialDocument</code>	<code>GetTrialDocument (</code>
<code>int</code>	<code>docId</code>
<code>)</code>	

285.1.1.1 Parametry

docId

Typ: System.Int32

Id dokumentu

285.1.1.2 Zwraca

Typ:

TrialDocument

Obiekt dokumentu.

285.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)



286 Metoda ITrialRepositoryCommunicationService.GetTrialDocumentsVO

Zwraca dokumenty powiązane z rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)

Moduł: ReCourt.Services.Communication.TrialRepository.Base (w

ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

286.1 Składnia

C#	
List<TrialDocumentVO> int)	GetTrialDocumentsVO (<i>id</i>

286.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

286.1.1.2 Zwraca

Typ:

List(TrialDocumentVO)

Lista dokumentów.

286.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



287Metoda ITrialRepositoryCommunicationService.GetTrialFiles

Zwraca pliki powiązane z rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

287.1 Składnia

```
C#  
  
List<TrialFile> GetTrialFiles (  
    int id  
)
```

287.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

287.1.1.2 Zwraca

Typ:

List(TrialFile)

Lista plików rozprawy.

287.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

288Metoda ITrialRepositoryCommunicationService.GetTrialProtocol

Zwraca protokół rozprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

288.1 Składnia

C#	
<pre>TrialDocument int Nullable<int>)</pre>	<pre>GetTrialProtocol(id, version = null</pre>

288.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

version (Optional)

Typ: System.Nullable(**Int32**)

Nr wersji

288.1.1.2 Zwraca

Typ:

Obiekt protokołu rozprawy.

TrialDocument

288.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

289 Metoda ITrialRepositoryCommunicationService.GetTrialProtocolHistory

Zwraca historię protokołu powiązanego z zadaną rozprawą.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

289.1 Składnia

```
C#  
  
List<TrialDocument> GetTrialProtocolHistory(  
    int id  
)
```

289.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

289.1.1.2 Zwraca

Typ:

List(TrialDocument)

Lista protokołów.

289.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



290Metoda ITrialRepositoryCommunicationService.GetTrials

Zwraca stronicowane rozprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

290.1 Składnia

C#	
<pre>PagedList<TrialInfoVO> TrialCriteria ReCourtUser int int)</pre>	<pre>GetTrials(criteria, user, pageNum, recordsPerPage</pre>

290.1.1.1 Parametry

criteria

Typ: **TrialCriteria**

Filtr

user

Typ: **ReCourtUser**

Użytkownik

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

290.1.1.2 Zwraca

Typ:

Stronicowana lista rozpraw.

PagedList(TrialInfoVO)

290.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

291Metoda ITrialRepositoryCommunicationService.GetTrialsForRecording

Zwraca rozprawy do nagrania.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

291.1 Składnia

C#	
<pre>PagedList<TrialInfoVO> TrialCriteria)</pre>	<pre>GetTrialsForRecording(<i>trialCriteria</i>)</pre>

291.1.1.1 Parametry

trialCriteria

Typ: **TrialCriteria**

Filtr

291.1.1.2 Zwraca

Typ:

PagedList(TrialInfoVO)

Stronicowana lista rozpraw.

291.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

292Metoda ITrialRepositoryCommunicationService.GetTrialTimeline

Zwraca oś czasu rozprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

292.1 Składnia

C#	
<pre>ICollection<byte>[] int)</pre>	<pre>GetTrialTimeline(id</pre>

292.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

292.1.1.2 Zwraca

Typ:

ICollection<Byte>[]

Dane osi czasu.

292.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

293 Metoda `ITrialRepositoryCommunicationService.IndexTrials`

Indeksuje rozprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: `ReCourt.Services.Communication.TrialRepository.Base` (w
`ReCourt.Services.Communication.TrialRepository.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

293.1 Składnia

```
C#  
  
bool IndexTrials(  
    ReCourtUser user,  
    List<ReCourtDiscriminantVO> source  
)
```

293.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

source

Typ: `System.Collections.Generic.List<ReCourtDiscriminantVO>`

Wydziały

293.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

293.2 Zobacz również

[Interfejs `ITrialRepositoryCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.TrialRepository.Base`](#)



294Metoda ITrialRepositoryCommunicationService.MarkTrialAsMovedToCKIS

Oznacza sprawę jako przeniesioną do CKIS.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
 Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
 ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

294.1 Składnia

C#	
bool	MarkTrialAsMovedToCKIS (
ReCourtUser	<i>user</i> ,
int	<i>trialId</i> ,
string	<i>location</i>
))

294.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialId

Typ: System.Int32

Id sprawy

location

Typ: System.String

Opis

294.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

294.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



295 Metoda ITrialRepositoryCommunicationService.ReCloseTrial

Ponownie zamyka rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

295.1 Składnia

C#	
bool	ReCloseTrial(
ReCourtUser	<i>user</i> ,
int	<i>casId</i>
))

295.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

casId

Typ: System.Int32

Id sprawy

295.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

295.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

296Metoda ITrialRepositoryCommunicationService.ReIndexTrial

Reindeksuje rozprawę.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

296.1 Składnia

C#	
<pre>bool ReCourtUser TrialInfo)</pre>	<pre>ReIndexTrial(user, trial</pre>

296.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trial

Typ: **TrialInfo**

Dane sprawy

296.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

296.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

297Metoda ITrialRepositoryCommunicationService.RemoveTrialFiles

Usuwa pliki sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

297.1 Składnia

```
C#  
  
bool RemoveTrialFiles (  
    ReCourtUser user,  
    int casId  
)
```

297.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

casId

Typ: System.Int32

Id sprawy

297.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

297.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

298Metoda ITrialRepositoryCommunicationService.SetSetCaseCapeStatus

Ustawia status cape

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

298.1 Składnia

```
C#  
  
bool SetSetCaseCapeStatus (  
    ReCourtUser reCourtUser,  
    int trialId,  
    string state  
)
```

298.1.1.1 Parametry

reCourtUser

Typ: **ReCourtUser**

Użytkownik

trialId

Typ: System.Int32

Id sprawy

state

Typ: System.String

Status

298.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

298.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

299Metoda ITrialRepositoryCommunicationService.StoreBurnResult

Zapisuje wynik nagania spraw.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
 Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
 ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

299.1 Składnia

C#	
bool	StoreBurnResult (
ReCourtUser	<i>user</i> ,
List<Tuple<int, TimeSpan>>	<i>trialRecordings</i> ,
bool	<i>burnResult</i> ,
string	<i>type</i> ,
string	<i>exportMode</i>
)	

299.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

trialRecordings

Typ: System.Collections.Generic.List(**Tuple(Int32, TimeSpan)**)

Fragmenty

burnResult

Typ: System.Boolean

Wynik

type

Typ: System.String

ISO lub DVD.

exportMode

Typ: System.String

Tryb

299.1.1.2 Zwraca

Typ: **Boolean**
 Informacja o powodzeniu operacji.

299.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



300Metoda ITrialRepositoryCommunicationService.StoreIndexingResult

Zapisuje wynik indeksowania sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

300.1 Składnia

C#	
<pre>bool CaseIndexingResult)</pre>	<pre>StoreIndexingResult(caseIndexing</pre>

300.1.1.1 Parametry

caseIndexing

Typ: **CaseIndexingResult**

Wynik indeksowania

300.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

300.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)



301Metoda ITrialRepositoryCommunicationService.UpdateDataIntegrationResult

Aktualizuje wynik sprawdzania integralności sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)

Moduł: ReCourt.Services.Communication.TrialRepository.Base (w

ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

301.1 Składnia

C#	
<pre>bool DataIntegration)</pre>	<pre>UpdateDataIntegrationResult(dataIntegration</pre>

301.1.1.1 Parametry

dataIntegration

Typ: **DataIntegration**

Wynik sprawdzania integralności

301.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

301.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

302 Metoda ITrialRepositoryCommunicationService.UpdateTrialAnnotations

Aktualizuje listę adnotacji rozprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
 Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
 ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

302.1 Składnia

```
C#
bool UpdateTrialAnnotations (
    ReCourtUser user,
    int caseId,
    string annotationsType,
    List<Annotation> annotations,
    string protocolDocumentContent
)
```

302.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

caseId

Typ: System.Int32

Id sprawy

annotationsType

Typ: System.String

Rodzaj adnotacji

annotations

Typ: System.Collections.Generic.List(**Annotation**)

Adnotacje

protocolDocumentContent

Typ: System.String

Treść protokołu

302.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

302.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

303 Metoda ITrialRepositoryCommunicationService.UpdateTrialData

Aktualizuje dane sprawy.

Przestrzeń nazw: [ReCourt.Services.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Communication.TrialRepository.Base (w
ReCourt.Services.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

303.1 Składnia

```
C#  
  
bool UpdateTrialData (  
    ReCourtUser currentUser,  
    TrialInfo trialInfo  
)
```

303.1.1.1 Parametry

currentUser

Typ: **ReCourtUser**

Użytkownik

trialInfo

Typ: **TrialInfo**

Dane sprawy

303.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.




303.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)





[Przestrzeń nazw ReCourt.Services.Communication.TrialRepository.Base](#)

304Przestrzeń nazw ReCourt.Services.Communication.Update.Base

304.1 Klasy

	Klasa	Opis
	FileTransferInfo	
	MachineHash	
	SendFileRequestMessage	

304.2 Interfejsy

	Interfejs	Opis
	IMachineCatalogueService	Rejestracja maszyn
	IReCourtUpdateCommunicationService	Interfejs odpowiada za zarządzanie żądaniami.
	IUpdateCommunicationService	Interfejs odpowiada za zarządzanie ustawieniami aplikacji.
	IUpdateService	



305 Klasa FileTransferInfo

305.1 Hierarchia

System.Object

ReCourt.Services.Communication.Update.Base.FileTransferInfo

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

305.2 Składnia

C#

```
public class FileTransferInfo
```

FileTransferInfo prezentuje składowe.

305.3 Konstruktory

	Nazwa	Opis
	FileTransferInfo	Inicjalizuje nową instancję klasy FileTransferInfo

305.4 Właściwości

	Nazwa	Opis
	Checksum	
	ChunkSize	
	CreateFile	
	FileFinished	
	Name	
	RunParam	

305.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



306 Konstruktor FileTransferInfo

Inicjalizuje nową instancję klasy [FileTransferInfo](#)

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

306.1 Składnia

C#

```
public FileTransferInfo ()
```

306.2 Zobacz również

[Klasa FileTransferInfo](#)


[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



307 Właściwości FileTransferInfo.FileTransferInfo

[FileTransferInfo](#) prezentuje składowe.

307.1 Właściwości

	Nazwa	Opis
	Checksum	
	ChunkSize	
	CreateFile	
	FileFinished	
	Name	
	RunParam	

307.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



308 Właściwość FileTransferInfo.Checksum

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

308.1 Składnia

C#

```
public byte[] Checksum { get; set; }
```

308.1.1.1 Zwraca

Typ: **Byte[]**

308.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



309 Właściwość FileTransferInfo.ChunkSize

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

309.1 Składnia

C#

```
public int ChunkSize { get; set; }
```

309.1.1.1 Zwraca

Typ: **Int32**

309.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



310 Właściwość FileTransferInfo.CreateFile

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

310.1 Składnia

C#

```
public bool CreateFile { get; set; }
```

310.1.1.1 Zwraca

Typ: **Boolean**

310.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

311 Właściwość FileTransferInfo.FileFinished

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

311.1 Składnia

C#

```
public bool FileFinished { get; set; }
```

311.1.1.1 Zwraca

Typ: **Boolean**

311.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



312 Właściwość FileTransferInfo.Name

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

312.1 Składnia

C#

```
public string Name { get; set; }
```

312.1.1.1 Zwraca

Typ: **String**

312.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



313 Właściwość FileTransferInfo.RunParam

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

313.1 Składnia

C#

```
public string RunParam { get; set; }
```

313.1.1.1 Zwraca

Typ: **String**

313.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

314 Interfejs IMachineCatalogueService

Rejestracja maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

314.1 Składnia

```
C#  
  
public interface IMachineCatalogueService
```

IMachineCatalogueService prezentuje składowe.

314.2 Metody

Nazwa	Opis
CheckMachineRegistration	Weryfikacja rejestracji
ForceUpdateMachine	Wymuszenie aktualizacji
GetAllMachines	Zwraca listę maszyn
GetAllMachinesForType	Zwraca listę maszyn
GetMachine	Pobiera dane maszyny
GetMachines(Int32, Int32, SortCriteria, ICollection(String))	Zwraca listę maszyn
GetMachines(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn
GetMachinesWithTypePrefix(String)	Zwraca listę maszyn
GetMachinesWithTypePrefix(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn
RegisterMachine	Rejestracja maszyny

314.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



315 Metody IMachineCatalogueService.IMachineCatalogueService

[IMachineCatalogueService](#) prezentuje składowe.

315.1 Metody

Nazwa	Opis
CheckMachineRegistration	Weryfikacja rejestracji
ForceUpdateMachine	Wymuszenie aktualizacji
GetAllMachines	Zwraca listę maszyn
GetAllMachinesForType	Zwraca listę maszyn
GetMachine	Pobiera dane maszyny
GetMachines(Int32, Int32, SortCriteria, ICollection(String))	Zwraca listę maszyn
GetMachines(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn
GetMachinesWithTypePrefix(String)	Zwraca listę maszyn
GetMachinesWithTypePrefix(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn
RegisterMachine	Rejestracja maszyny

315.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

316Metoda IMachineCatalogueService.CheckMachineRegistration

Weryfikacja rejestracji

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

316.1 Składnia

C#	
ResponseStatus	CheckMachineRegistration (
Guid	<i>identifier</i> ,
byte[]	<i>machineHash</i> ,
string	<i>ip</i> ,
string	<i>executableVersion</i> ,
string	<i>systemName</i>
)	

316.1.1.1 Parametry

identifier

Typ: System.Guid

Guid

machineHash

Typ: System.Byte[]

Hash

ip

Typ: System.String

IP

executableVersion

Typ: System.String

Wersja exe

systemName

Typ: System.String

System

316.1.1.2 Zwraca

Typ: **ResponseStatus**

316.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

317Metoda IMachineCatalogueService.ForceUpdateMachine

Wymuszenie aktualizacji

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

317.1 Składnia

```
C#  
  
void ForceUpdateMachine (  
    Machine machine,  
    byte[] machineHash  
)
```

317.1.1.1 Parametry

machine

Typ: **Machine**

Maszyna

machineHash

Typ: System.Byte[]

Hash

317.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

318Metoda IMachineCatalogueService.GetAllMachines

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

318.1 Składnia

C#	
<pre>PagedList<Machine> int int SortCriteria)</pre>	<pre>GetAllMachines (pageNum, recordsPerPage, sortCriteria</pre>

318.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

318.1.1.2 Zwraca

Typ:

Lista maszyn

PagedList(Machine)

318.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



319Metoda IMachineCatalogueService.GetAllMachinesForType

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

319.1 Składnia

```
C#  
  
List<Machine> GetAllMachinesForType (  
    string typeName,  
    OperatingSystem os  
)
```

319.1.1.1 Parametry

typeName

Typ: System.String

Typ

os

Typ: **OperatingSystem**

System

319.1.1.2 Zwraca

Typ:

Lista maszyn

List(Machine)

319.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



320Metoda IMachineCatalogueService.GetMachine

Pobiera dane maszyny

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

320.1 Składnia

C#	
Machine Guid)	GetMachine(<i>identifier</i>

320.1.1.1 Parametry

identifier

Typ: System.Guid

Guid

320.1.1.2 Zwraca

Typ:

Machine

Dane maszyny

320.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

321Metoda IMachineCatalogueService.GetMachines

321.1 Przeciżenia

	Nazwa	Opis
	GetMachines(Int32, Int32, SortCriteria, ICollection(String))	Zwraca listę maszyn
	GetMachines(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn

321.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

322Metoda IMachineCatalogueService.GetMachines (Int32, Int32, SortCriteria, ICollection(String))

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

322.1 Składnia

```
C#  
  
PagedList<Machine> GetMachines (  
    int                pageNum,  
    int                recordsPerPage,  
    SortCriteria       sortCriteria,  
    ICollection<string> types  
)
```

322.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

types

Typ: System.Collections.Generic.ICollection(**String**)

Wg typów

322.1.1.2 Zwraca

Typ:

Lista maszyn

PagedList(Machine)

322.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przeciążenia GetMachines](#)



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

323Metoda IMachineCatalogueService.GetMachines (Int32, Int32, SortCriteria, String)

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

323.1 Składnia

C#	
PagedList<Machine>	GetMachines (
int	pageNum,
int	recordsPerPage,
SortCriteria	sortCriteria,
string	type
)	

323.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

type

Typ: System.String

Wg typu

323.1.1.2 Zwraca

Typ:

Lista maszyn

PagedList(Machine)

323.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przeciążenia GetMachines](#)



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**


UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

324 Metoda IMachineCatalogueService.GetMachinesWithTypePrefix

324.1 Przeciżenia

	Nazwa	Opis
	GetMachinesWithTypePrefix(String)	Zwraca listę maszyn
	GetMachinesWithTypePrefix(Int32, Int32, SortCriteria, String)	Zwraca listę maszyn

324.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



325 Metoda `IMachineCatalogueService.GetMachinesWithTypePrefix` (String)

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: `ReCourt.Services.Communication.Update.Base` (w
`ReCourt.Services.Communication.Update.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

325.1 Składnia

```
C#  
  
List<Machine> GetMachinesWithTypePrefix(  
    string typePrefix  
)
```

325.1.1.1 Parametry

typePrefix

Typ: `System.String`

typ

325.1.1.2 Zwraca

Typ: **List(Machine)**
Lista maszyn

325.2 Zobacz również

[Interfejs `IMachineCatalogueService`](#)

[Przeciążenia `GetMachinesWithTypePrefix`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

326 Metoda `IMachineCatalogueService.GetMachinesWithTypePrefix (Int32, Int32, SortCriteria, String)`

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

326.1 Składnia

```
C#  
  
PagedList<Machine> GetMachinesWithTypePrefix (  
    int                pageNum,  
    int                recordsPerPage,  
    SortCriteria       sortCriteria,  
    string             typePrefix  
)
```

326.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

typePrefix

Typ: System.String

Wg typu

326.1.1.2 Zwraca

Typ:

Lista maszyn

PagedList(Machine)

326.2 Zobacz również

[Interfejs `IMachineCatalogueService`](#)

[Przeciążenia `GetMachinesWithTypePrefix`](#)



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

327Metoda IMachineCatalogueService.RegisterMachine

Rejestracja maszyny

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

327.1 Składnia

```
C#
RegisterMachineResponse RegisterMachine (
    Machine machine,
    byte[] machineHash,
    bool hasExtendedParameters = false
)
```

327.1.1.1 Parametry

machine

Typ: **Machine**

Maszyna

machineHash

Typ: System.Byte[]

Hash

hasExtendedParameters (Optional)

Typ: System.Boolean

Informacja o parametrach

327.1.1.2 Zwraca

Typ: **RegisterMachineResponse**

327.2 Zobacz również

[Interfejs IMachineCatalogueService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



328 Interfejs IReCourtUpdateCommunicationService

Interfejs odpowiada za zarządzanie żadaniami.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

328.1 Składnia

C#

```
public interface IReCourtUpdateCommunicationService
```

IReCourtUpdateCommunicationService prezentuje składowe.

328.2 Metody

	Nazwa	Opis
	StatusRequest	Pobiera status żądania.
	UpdateRequest	Aktualizuje żądanie.

328.3 Zobacz również



[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

329 Metody

`IReCourtUpdateCommunicationService.IReCourtUpdateCommunicationService`

[IReCourtUpdateCommunicationService](#) prezentuje składowe.

329.1 Metody

	Nazwa	Opis
	StatusRequest	Pobiera status żądania.
	UpdateRequest	Aktualizuje żądanie.

329.2 Zobacz również

[Interfejs IReCourtUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

332 Interfejs IUpdateCommunicationService

Interfejs odpowiada za zarządzanie ustawieniami aplikacji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)













332.1 Składnia

C#





















```
public interface IUpdateCommunicationService
```

IUpdateCommunicationService prezentuje składowe.

332.2 Metody

	Nazwa	Opis
	AddConfigGroup	Dodaje konfiguracje grupy.
	AddUpdateRequest	Dodaje żądanie aktualizacji.
	AddUpdateRequestForAll	Dodaje żądanie aktualizacji dla wszystkich.
	AddUpdateRequestForGroup	Dodaje żądanie aktualizacji dla grupy.
	CancelUpdateReCourtRequest	Anuluje żądanie aktualizacji.
	CheckUpdateReCourtRequest	Sprawdza czy są dostępne aktualizacje.
	GetAgentConfigGroup	Zwraca agenta konfiguracji grup.
	GetAgentInstance	Zwraca agenta instancji.
	GetAgentInstanceCurrentModules	Zwraca listę obecnych modułów instancji agenta.
	GetAgentInstanceGroupList	Zwraca listę agentów instancji grup.
	GetAudioDevicesConfig	Pobiera konfigurację urządzeń audio.
	GetCodecsConfig	Pobiera konfiguracje kodeków.



 GetCommandMappingsConfig	Pobiera polecenia konfiguracyjne.
 GetConfigGroupList	Zwraca listę agentów konfiguracji grup.
 GetConfigGroupListPager	Zwraca stronicowaną listę agentów konfiguracji grup.
 GetConfigListForGroup	Pobiera konfigurację listy dla grup.
 GetCourtApplicationConfig	Pobiera konfiguracje aplikacji sądu.
 GetCourtDataConfig	Pobiera konfigurację danych sądu.
 GetCourtSettingsConfig	Pobiera konfigurację ustawień sądu.
 GetFontConfig	Pobiera konfigurację czcionki.
 GetMediaServerConfig	Pobiera konfiguracje serwera mediów.
 GetModuleItemWithData	Pobiera moduł z danymi.
 GetModulesForGroup	Zwraca moduły dla konkretnej grupy.
 GetReCourtInstanceList	Pobiera listę instancji.
 GetReCourtInstanceListPager	Pobiera stronicowaną listę instancji.
 GetReCourtServicesInstanceList	Pobiera listę instancji serwisów.
 GetReCourtServicesInstanceListByGroupId	Pobiera listę serwisów aplikacji.
 GetReCourtServicesInstanceListPager	Pobiera stronicowaną listę serwisów aplikacji.
 GetReCourtVersionList	Zwraca wersje programu ReCourt.
 GetTvConfig	Pobiera konfigurację telewizyjną.
 GetUpdateHistory	Zwraca listę obiektów z wersjami instancji agenta.
 GetVideoDevicesConfig	Pobiera konfigurację urządzeń wideo.

 GetVideoLayoutsConfig	Pobiera konfigurację obrazu wideo.
 LogOnReCourtInstance	Logowanie do systemu.
 LogOnReCourtServicesInstance	Logowanie do serwisu systemu.
 RemoveConfigGroup	Usuwa konfigurację grupy.
 SaveAudioDevicesConfig	Zapisuje konfiguracje urządzeń audio.
 SaveCodecsConfig	Zapisuje konfiguracje kodeków.
 SaveCommandMappingsConfig	Zapisuje polecenia z konfiguracji.
 SaveCourtApplicationConfig	Zapisuje konfigurację aplikacji sądu.
 SaveCourtDataConfig	Zapisuje dane konfiguracyjne sądu.
 SaveCourtSettingsConfig	Zapisuje ustawienia sądu.
 SaveFontConfig	Zapisuje konfigurację czcionki.
 SaveMediaServerConfig	Zapisuje konfiguracje serwera mediów.
 SaveTvConfig	Zapisuje ustawienia telewizyjne.
 SaveVideoDevicesConfig	Zapisuje ustawienia urządzenia wideo.
 SaveVideoLayoutsConfig	Zapisuje konfigurację układu obrazu.
 UpdateReCourtRequest	Aktualizuje aplikacje.

332.3 Zobacz również


















[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)























333 Metody IUpdateCommunicationService.IUpdateCommunicationService

[IUpdateCommunicationService](#) prezentuje składowe.

333.1 Metody

Nazwa	Opis
 AddConfigGroup	Dodaje konfiguracje grupy.
 AddUpdateRequest	Dodaje żądanie aktualizacji.
 AddUpdateRequestForAll	Dodaje żądanie aktualizacji dla wszystkich.
 AddUpdateRequestForGroup	Dodaje żądanie aktualizacji dla grupy.
 CancelUpdateReCourtRequest	Anuluje żądanie aktualizacji.
 CheckUpdateReCourtRequest	Sprawdza czy są dostępne aktualizacje.
 GetAgentConfigGroup	Zwraca agenta konfiguracji grup.
 GetAgentInstance	Zwraca agenta instancji.
 GetAgentInstanceCurrentModules	Zwraca listę obecnych modułów instancji agenta.
 GetAgentInstanceGroupList	Zwraca listę agentów instancji grup.
 GetAudioDevicesConfig	Pobiera konfigurację urządzeń audio.
 GetCodecsConfig	Pobiera konfiguracje kodeków.
 GetCommandMappingsConfig	Pobiera polecenia konfiguracyjne.
 GetConfigGroupList	Zwraca listę agentów konfiguracji grup.
 GetConfigGroupListPager	Zwraca stronicowaną listę agentów konfiguracji grup.
 GetConfigListForGroup	Pobiera konfigurację listy dla grup.
 GetCourtApplicationConfig	Pobiera konfiguracje aplikacji sądu.

 GetCourtDataConfig	Pobiera konfigurację danych sądu.
 GetCourtSettingsConfig	Pobiera konfigurację ustawień sądu.
 GetFontConfig	Pobiera konfigurację czcionki.
 GetMediaServerConfig	Pobiera konfiguracje serwera mediów.
 GetModuleItemWithData	Pobiera moduł z danymi.
 GetModulesForGroup	Zwraca moduły dla konkretnej grupy.
 GetReCourtInstanceList	Pobiera listę instancji.
 GetReCourtInstanceListPager	Pobiera stronicowaną listę instancji.
 GetReCourtServicesInstanceList	Pobiera listę instancji serwisów.
 GetReCourtServicesInstanceListByGroupId	Pobiera listę serwisów aplikacji.
 GetReCourtServicesInstanceListPager	Pobiera stronicowaną listę serwisów aplikacji.
 GetReCourtVersionList	Zwraca wersje programu ReCourt.
 GetTvConfig	Pobiera konfigurację telewizyjną.
 GetUpdateHistory	Zwraca listę obiektów z wersjami instancji agenta.
 GetVideoDevicesConfig	Pobiera konfigurację urządzeń wideo.
 GetVideoLayoutsConfig	Pobiera konfigurację obrazu wideo.
 LogOnReCourtInstance	Logowanie do systemu.
 LogOnReCourtServicesInstance	Logowanie do serwisu systemu.
 RemoveConfigGroup	Usuwa konfigurację grupy.
 SaveAudioDevicesConfig	Zapisuje konfiguracje urządzeń audio.



 SaveCodecsConfig	Zapisuje konfiguracje kodeków.
 SaveCommandMappingsConfig	Zapisuje polecenia z konfiguracji.
 SaveCourtApplicationConfig	Zapisuje konfigurację aplikacji sądu.
 SaveCourtDataConfig	Zapisuje dane konfiguracyjne sądu.
 SaveCourtSettingsConfig	Zapisuje ustawienia sądu.
 SaveFontConfig	Zapisuje konfigurację czcionki.
 SaveMediaServerConfig	Zapisuje konfiguracje serwera mediów.
 SaveTvConfig	Zapisuje ustawienia telewizyjne.
 SaveVideoDevicesConfig	Zapisuje ustawienia urządzenia wideo.
 SaveVideoLayoutsConfig	Zapisuje konfigurację układu obrazu.
 UpdateReCourtRequest	Aktualizuje aplikacje.

333.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

334Metoda IUpdateCommunicationService.AddConfigGroup

Dodaje konfiguracje grupy.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

334.1 Składnia

C#	
<pre>bool ReCourtAgentConfigGroup Nullable<int> Nullable<int>)</pre>	<pre>AddConfigGroup(newGroup, copyFromGroupId, copyFromInstanceId</pre>

334.1.1.1 Parametry

newGroup

Typ: **ReCourtAgentConfigGroup**

Dane grupy

copyFromGroupId

Typ: System.Nullable(**Int32**)

Kopiuje z

copyFromInstanceId

Typ: System.Nullable(**Int32**)

Kopiuje z

334.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

334.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

335 Metoda IUpdateCommunicationService.AddUpdateRequest

Dodaje żądanie aktualizacji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

335.1 Składnia

C#	
<pre>bool int int int)</pre>	<pre>AddUpdateRequest(instanceId, moduleId, userId</pre>

335.1.1.1 Parametry

instanceId

Typ: System.Int32

Id instancji

moduleId

Typ: System.Int32

Id grupy

userId

Typ: System.Int32

Id użytkownika

335.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

335.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

336 Metoda `IUpdateCommunicationService.AddUpdateRequestForAll`

Dodaje żądanie aktualizacji dla wszystkich.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: `ReCourt.Services.Communication.Update.Base` (w
`ReCourt.Services.Communication.Update.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

336.1 Składnia

```
C#  
  
bool AddUpdateRequestForAll (  
    int moduleGroupId,  
    int userId  
)
```

336.1.1.1 Parametry

moduleGroupId

Typ: `System.Int32`

Id grupy

userId

Typ: `System.Int32`

Id użytkownika

336.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

336.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

337Metoda IUpdateCommunicationService.AddUpdateRequestForGroup

Dodaje żądanie aktualizacji dla grupy.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

337.1 Składnia

C#	
<pre>bool int int int)</pre>	<pre>AddUpdateRequestForGroup(groupId, moduleGroupId, userId</pre>

337.1.1.1 Parametry

groupId

Typ: System.Int32

Id grupy

moduleGroupId

Typ: System.Int32

Id grupy

userId

Typ: System.Int32

Id użytkownika

337.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

337.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

338Metoda IUpdateCommunicationService.CancelUpdateReCourtRequest

Anuluje żądanie aktualizacji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

338.1 Składnia

C#	
<pre>bool TaskId)</pre>	<pre>CancelUpdateReCourtRequest (taskId</pre>

338.1.1.1 Parametry

taskId

Typ: **TaskId**

Id zadania

338.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

338.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

339Metoda IUpdateCommunicationService.CheckUpdateReCourtRequest

Sprawdza czy są dostępne aktualizacje.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

339.1 Składnia

C#	
ReCourtUpdateResponse	CheckUpdateReCourtRequest (
int	instanceId,
int	userId
)	

339.1.1.1 Parametry

instanceId

Typ: System.Int32

Id instancji

userId

Typ: System.Int32

Id użytkownika

339.1.1.2 Zwraca

Typ:

ReCourtUpdateResponse

Wynik sprawdzenia dostępnych aktualizacji.

339.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

340Metoda IUpdateCommunicationService.GetAgentConfigGroup

Zwraca agenta konfiguracji grup.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

340.1 Składnia

C#	
ReCourtAgentConfigGroup	GetAgentConfigGroup (
int	<i>groupId</i>
))

340.1.1.1 Parametry

groupId

Typ: System.Int32

Id grupy

340.1.1.2 Zwraca

Typ:

Agent konfiguracji grup.

ReCourtAgentConfigGroup

340.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

341Metoda IUpdateCommunicationService.GetAgentInstance

Zwraca agenta instancji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

341.1 Składnia

C#	
ReCourtAgentInstance int)	GetAgentInstance (instanceId

341.1.1.1 Parametry

instanceId

Typ: System.Int32

341.1.1.2 Zwraca

Typ:

Obiekt agenta instancji.

ReCourtAgentInstance

341.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

342 Metoda `IUpdateCommunicationService.GetAgentInstanceCurrentModules`

Zwraca listę obecnych modułów instancji agenta.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: `ReCourt.Services.Communication.Update.Base` (w
`ReCourt.Services.Communication.Update.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

342.1 Składnia

```
C#  
  
List<ReCourtAgentInstanceModuleVO>  
GetAgentInstanceCurrentModules (  
    int instanceId  
)
```

342.1.1.1 Parametry

instanceId

Typ: `System.Int32`

Id instancji

342.1.1.2 Zwraca

Typ: `List(ReCourtAgentInstanceModuleVO)`
Lista modułów instancji agenta.

342.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

343 Metoda `IUpdateCommunicationService.GetAgentInstanceGroupList`

Zwraca listę agentów instancji grup.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

343.1 Składnia

C#

```
List<ReCourtAgentInstanceGroup> GetAgentInstanceGroupList ()
```

343.1.1.1 Zwraca

Typ: **List(ReCourtAgentInstanceGroup)**
Lista agentów instancji grup.

343.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

344 Metoda `IUpdateCommunicationService.GetAudioDevicesConfig`

Pobiera konfigurację urządzeń audio.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

344.1 Składnia

```
C#  
  
ReCourtXmlAudioDevices          GetAudioDevicesConfig(  
    ReCourtAgentConfigSelector    selector  
)
```

344.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

344.1.1.2 Zwraca

Typ: **ReCourtXmlAudioDevices**
Element konfiguracji.

344.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

345 Metoda IUpdateCommunicationService.GetCodecsConfig

Pobiera konfiguracje kodeków.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

345.1 Składnia

```
C#  
  
ReCourtXmlCodecs                GetCodecsConfig(  
    ReCourtAgentConfigSelector    selector  
)
```

345.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

345.1.1.2 Zwraca

Typ:

ReCourtXmlCodecs

Element konfiguracji.

345.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

346 Metoda `IUpdateCommunicationService.GetCommandMappingsConfig`

Pobiera polecenia konfiguracyjne.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

346.1 Składnia

```
C#  
  
ReCourtXmlCommandMappings          GetCommandMappingsConfig (  
    ReCourtAgentConfigSelector      selector  
)
```

346.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

346.1.1.2 Zwraca

Typ: **ReCourtXmlCommandMappings**

Polecenia konfiguracyjne.

346.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)



348Metoda IUpdateCommunicationService.GetConfigGroupListPager

Zwraca stronicowaną listę agentów konfiguracji grup.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

348.1 Składnia

C#	
<pre>PagedList<ReCourtAgentConfigGroup> int int SortCriteria)</pre>	<pre>GetConfigGroupListPager (pageNum, recordsPerPage, sortCriteria</pre>

348.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

348.1.1.2 Zwraca

Typ: **PagedList(ReCourtAgentConfigGroup)**

Stronicowana lista agentów konfiguracji grup.

348.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



349 Metoda `IUpdateCommunicationService.GetConfigListGroup`

Pobiera konfigurację listy dla grup.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

349.1 Składnia

C#	
<pre>List<ReCourtAgentConfigGroupItem> int)</pre>	<pre>GetConfigListGroup(groupId</pre>

349.1.1.1 Parametry

groupId

Typ: System.Int32

Id grupy

349.1.1.2 Zwraca

Typ: **List(ReCourtAgentConfigGroupItem)**
Element konfiguracji.

349.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)



351Metoda IUpdateCommunicationService.GetCourtDataConfig

Pobiera konfigurację danych sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

351.1 Składnia

```
C#  
  
ReCourtXmlCourtData                      GetCourtDataConfig(  
    ReCourtAgentConfigSelector                      selector  
)
```

351.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

351.1.1.2 Zwraca

Typ:

ReCourtXmlCourtData

Element konfiguracji.

351.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

352 Metoda IUpdateCommunicationService.GetCourtSettingsConfig

Pobiera konfigurację ustawień sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

352.1 Składnia

```
C#  
  
ReCourtXmlCourtSettings          GetCourtSettingsConfig(  
    ReCourtAgentConfigSelector    selector  
)
```

352.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

352.1.1.2 Zwraca

Typ: **ReCourtXmlCourtSettings**
Element konfiguracji.

352.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

353 Metoda IUpdateCommunicationService.GetFontConfig

Pobiera konfigurację czcionki.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

353.1 Składnia

```
C#  
  
ReCourtXmlFontConfig                      GetFontConfig(  
    ReCourtAgentConfigSelector                      selector  
)
```

353.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

353.1.1.2 Zwraca

Typ:

ReCourtXmlFontConfig

Element konfiguracji.

353.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



354 Metoda `IUpdateCommunicationService.GetMediaServerConfig`

Pobiera konfigurację serwera mediów.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

354.1 Składnia

```
C#  
  
ReCourtXmlMediaServerConfig GetMediaServerConfig(  
    ReCourtAgentConfigSelector selector  
)
```

354.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

354.1.1.2 Zwraca

Typ: **ReCourtXmlMediaServerConfig**
Element konfiguracji.

354.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

355 Metoda IUpdateCommunicationService.GetModuleItemWithData

Pobiera moduł z danymi.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

355.1 Składnia

C#	
ReCourtAgentModuleItem int)	GetModuleItemWithData (<i>moduleItemId</i>)

355.1.1.1 Parametry

moduleItemId

Typ: System.Int32

Id modułu

355.1.1.2 Zwraca

Typ:

ReCourtAgentModuleItem

Moduł z danymi.

355.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



356 Metoda IUpdateCommunicationService.GetModulesForGroup

Zwraca moduły dla konkretnej grupy.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

356.1 Składnia

C#	
<pre>List<ReCourtAgentModuleItem> int)</pre>	<pre>GetModulesForGroup (groupId</pre>

356.1.1.1 Parametry

groupId

Typ: System.Int32

Id grupy

356.1.1.2 Zwraca

Typ: **List(ReCourtAgentModuleItem)**
Lista agentów.

356.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

358Metoda IUpdateCommunicationService.GetReCourtInstanceListPager

Pobiera stronicowaną listę instancji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

358.1 Składnia

C#	
<pre>PagedList<ReCourtAgentInstance> int int SortCriteria)</pre>	<pre>GetReCourtInstanceListPager (pageNum, recordsPerPage, sortCriteria)</pre>

358.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

358.1.1.2 Zwraca

Typ:

PagedList(ReCourtAgentInstance)

Stronicowana lista instancji.

358.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



359 Metoda IUpdateCommunicationService.GetReCourtServicesInstanceList

Pobiera listę instancji serwisów.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

359.1 Składnia

C#

```
List<ReCourtAgentInstance> GetReCourtServicesInstanceList ()
```

359.1.1.1 Zwraca

Typ:

List(ReCourtAgentInstance)

Lista instancji serwisów.

359.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

360Metoda

IUpdateCommunicationService.GetReCourtServicesInstanceListByGroupId

Pobiera listę serwisów aplikacji.

Przeźren nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

360.1 Składnia

```
C#  
  
List<ReCourtAgentInstance>  
GetReCourtServicesInstanceListByGroupId(  
    int groupId  
)
```

360.1.1.1 Parametry

groupId

Typ: System.Int32

Id grupy

360.1.1.2 Zwraca

Typ: **List(ReCourtAgentInstance)**
Lista serwisów aplikacji.

360.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przeźren nazw ReCourt.Services.Communication.Update.Base](#)



363 Metoda IUpdateCommunicationService.GetTvConfig

Pobiera konfigurację telewizyjną.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

363.1 Składnia

```
C#  
  
ReCourtXmlTvConfig           GetTvConfig(  
    ReCourtAgentConfigSelector selector  
)
```

363.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

363.1.1.2 Zwraca

Typ:

ReCourtXmlTvConfig

Element konfiguracji.

363.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

364 Metoda IUpdateCommunicationService.GetUpdateHistory

Zwraca listę obiektów z wersjami instancji agenta.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

364.1 Składnia

C#	
<pre>List<ReCourtAgentInstanceVersion> int)</pre>	<pre>GetUpdateHistory(instanceId</pre>

364.1.1.1 Parametry

instanceId

Typ: System.Int32

Id instancji

364.1.1.2 Zwraca

Typ: **List(ReCourtAgentInstanceVersion)**

Lista obiektów z wersjami instancji agenta.

364.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



365 Metoda IUpdateCommunicationService.GetVideoDevicesConfig

Pobiera konfigurację urządzeń wideo.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

365.1 Składnia

```
C#  
  
ReCourtXmlVideoDevices                      GetVideoDevicesConfig(  
    ReCourtAgentConfigSelector                      selector  
)
```

365.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

365.1.1.2 Zwraca

Typ: **ReCourtXmlVideoDevices**
Element konfiguracji.

365.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

366 Metoda `IUpdateCommunicationService.GetVideoLayoutsConfig`

Pobiera konfigurację obrazu wideo.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

366.1 Składnia

```
C#  
  
ReCourtXmlVideoLayouts          GetVideoLayoutsConfig(  
    ReCourtAgentConfigSelector    selector  
)
```

366.1.1.1 Parametry

selector

Typ: **ReCourtAgentConfigSelector**

Filtr

366.1.1.2 Zwraca

Typ: **ReCourtXmlVideoLayouts**
Element konfiguracji.

366.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

367Metoda IUpdateCommunicationService.LogOnReCourtInstance

Logowanie do systemu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

367.1 Składnia

C#	
<pre>ReCourtAgentInstance ReCourtSystemIdentity)</pre>	<pre>LogOnReCourtInstance(systemIdentity</pre>

367.1.1.1 Parametry

systemIdentity

Typ: **ReCourtSystemIdentity**

Dane instancji

367.1.1.2 Zwraca

Typ:

ReCourtAgentInstance

Instancja agenta.

367.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

368 Metoda `IUpdateCommunicationService.LogOnReCourtServicesInstance`

Logowanie do serwisu systemu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

368.1 Składnia

```
C#  
  
ReCourtAgentInstance                      LogOnReCourtServicesInstance (  
    ReCourtSystemIdentity                      systemIdentity  
)
```

368.1.1.1 Parametry

systemIdentity

Typ: **ReCourtSystemIdentity**

Dane instancji

368.1.1.2 Zwraca

Typ: **ReCourtAgentInstance**
Instancja agenta.

368.2 Zobacz również

[Interfejs `IUpdateCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)

369Metoda IUpdateCommunicationService.RemoveConfigGroup

Usuwa konfigurację grupy.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

369.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveConfigGroup(configGroupId</pre>

369.1.1.1 Parametry

configGroupId

Typ: System.Int32

Id grupy

369.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

369.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



370Metoda IUpdateCommunicationService.SaveAudioDevicesConfig

Zapisuje konfiguracje urządzeń audio.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

370.1 Składnia

```
C#  
  
bool                                      SaveAudioDevicesConfig(  
    ReCourtXmlAudioDevices                      config,  
    int                                              userId  
)
```

370.1.1.1 Parametry

config

Typ: **ReCourtXmlAudioDevices**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

370.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

370.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

372 Metoda IUpdateCommunicationService.SaveCommandMappingsConfig

Zapisuje polecenia z konfiguracji.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

372.1 Składnia

```
C#  
  
bool SaveCommandMappingsConfig(  
    ReCourtXmlCommandMappings config,  
    int userId  
)
```

372.1.1.1 Parametry

config

Typ: **ReCourtXmlCommandMappings**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

372.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

372.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



373 Metoda IUpdateCommunicationService.SaveCourtApplicationConfig

Zapisuje konfigurację aplikacji sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

373.1 Składnia

```
C#  
  
bool SaveCourtApplicationConfig(  
    ReCourtXmlCourtApplication config,  
    int userId  
)
```

373.1.1.1 Parametry

config

Typ: **ReCourtXmlCourtApplication**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

373.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

373.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



374Metoda IUpdateCommunicationService.SaveCourtDataConfig

Zapisuje dane konfiguracyjne sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

374.1 Składnia

```
C#  
  
bool                                              SaveCourtDataConfig(  
    ReCourtXmlCourtData                      config,  
    int                                              userId  
)
```

374.1.1.1 Parametry

config

Typ: **ReCourtXmlCourtData**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

374.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

374.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

375 Metoda IUpdateCommunicationService.SaveCourtSettingsConfig

Zapisuje ustawienia sądu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

375.1 Składnia

```
C#  
  
bool SaveCourtSettingsConfig(  
    ReCourtXmlCourtSettings config,  
    int userId  
)
```

375.1.1.1 Parametry

config

Typ: **ReCourtXmlCourtSettings**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

375.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

375.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



377Metoda IUpdateCommunicationService.SaveMediaServerConfig

Zapisuje konfiguracje serwera mediów.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

377.1 Składnia

```
C#  
  
bool                                              SaveMediaServerConfig(  
    ReCourtXmlMediaServerConfig                      config,  
    int                                                      userId  
)
```

377.1.1.1 Parametry

config

Typ: **ReCourtXmlMediaServerConfig**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

377.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

377.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

379Metoda IUpdateCommunicationService.SaveVideoDevicesConfig

Zapisuje ustawienia urządzenia wideo.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

379.1 Składnia

```
C#  
  
bool                                      SaveVideoDevicesConfig(  
    ReCourtXmlVideoDevices                      config,  
    int                                              userId  
)
```

379.1.1.1 Parametry

config

Typ: **ReCourtXmlVideoDevices**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

379.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

379.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



380Metoda IUpdateCommunicationService.SaveVideoLayoutsConfig

Zapisuje konfigurację układu obrazu.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

380.1 Składnia

```
C#
bool                                      SaveVideoLayoutsConfig(
    ReCourtXmlVideoLayouts                      config,
    int                                              userId
)
```

380.1.1.1 Parametry

config

Typ: **ReCourtXmlVideoLayouts**

Konfiguracja

userId

Typ: System.Int32

Id użytkownika

380.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

380.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

381Metoda IUpdateCommunicationService.UpdateReCourtRequest

Aktualizuje aplikacje.

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

381.1 Składnia

```
C#  
  
int UpdateReCourtRequest (  
    int instanceId,  
    int userId  
)
```

381.1.1.1 Parametry

instanceId

Typ: System.Int32

Id instancji

userId

Typ: System.Int32

Id użytkownika

381.1.1.2 Zwraca

Typ: **Int32**
Wynik aktualizacji. 0 - OK. 1 - Nie znaleziono instancji. 2 - Brak aplikacji aktualizującej. 3 - Brak połączenia z CJR. 9 - Błąd aktualizacji.

381.2 Zobacz również

[Interfejs IUpdateCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

382 Interfejs IUpdateService

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)











382.1 Składnia

C#





```
public interface IUpdateService
```

IUpdateService prezentuje składowe.

382.2 Metody

	Nazwa	Opis
	AddConfigType	
	ConfigTypes	
	GetConfigType	
	GetManualUpdatePackagesForMachine	
	GetUpdatesHistory	
	HasAnyUpdatesForMachine	
	ProductsForConfigType	
	ProductsForOperatingSystem	
	RegisterDummyPackage	
	RemoveConfigGroup	
	ReportInstallationStatus	
	RescanDropDirectry	
	ScheduleAllPackagesInstallOnMachine	



 SchedulePackageInstallationForConfigType(String, OperatingSystem, List(Int32))	
 SchedulePackageInstallationForConfigType(String, OperatingSystem, Int32)	
 SchedulePackageInstallationOnMachine(Guid, List(Int32))	
 SchedulePackageInstallationOnMachine(Guid, Int32)	
 SetConfigTypeAutoUpdate	
 SetConfigTypeProducts	
 UpdatesForMachine	

382.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

383 Metody IUpdateService.IUpdateService

[IUpdateService](#) prezentuje składowe.

383.1 Metody

Nazwa	Opis
 AddConfigType	
 ConfigTypes	
 GetConfigType	
 GetManualUpdatePackagesForMachine	
 GetUpdatesHistory	
 HasAnyUpdatesForMachine	
 ProductsForConfigType	
 ProductsForOperatingSystem	
 RegisterDummyPackage	
 RemoveConfigGroup	
 ReportInstallationStatus	
 RescanDropDirectry	
 ScheduleAllPackagesInstallOnMachine	
 SchedulePackageInstallationForConfigType(String, OperatingSystem, List(Int32))	
 SchedulePackageInstallationForConfigType(String, OperatingSystem, Int32)	
 SchedulePackageInstallationOnMachine(Guid, List(Int32))	
 SchedulePackageInstallationOnMachine(Guid, Int32)	
 SetConfigTypeAutoUpdate	



 SetConfigTypeProducts	
 UpdatesForMachine	

383.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

384 Metoda IUpdateService.AddConfigType

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

384.1 Składnia

```
C#  
  
void AddConfigType (  
    string name,  
    OperatingSystem os,  
    bool autoUpdate,  
    List<string> products  
)
```

384.1.1.1 Parametry

name

Typ: System.String

os

Typ: **OperatingSystem**

autoUpdate

Typ: System.Boolean

products

Typ: System.Collections.Generic.List(**String**)

384.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

385 Metoda IUpdateService.ConfigTypes

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

385.1 Składnia

C#	
<pre>PagedList<MachineConfigType> int int SortCriteria)</pre>	<pre>ConfigTypes (pageNum, recordsPerPage, sortCriteria</pre>

385.1.1.1 Parametry

pageNum

Typ: System.Int32

recordsPerPage

Typ: System.Int32

sortCriteria

Typ: **SortCriteria**

385.1.1.2 Zwraca

Typ: **PagedList(MachineConfigType)**

385.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

386 Metoda IUpdateService.GetConfigType

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

386.1 Składnia

```
C#  
  
MachineConfigType          GetConfigType (  
    string                   configType,  
    OperatingSystem          os  
)
```

386.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

386.1.1.2 Zwraca

Typ: **MachineConfigType**

386.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



387Metoda IUpdateService.GetManualUpdatePackagesForMachine

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

387.1 Składnia

C#

```
List<Package> GetManualUpdatePackagesForMachine (  
    Guid machineIdentifier  
)
```

387.1.1.1 Parametry

machineIdentifier

Typ: System.Guid

387.1.1.2 Zwraca

Typ: **List(Package)**

387.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



388Metoda IUpdateService.GetUpdatesHistory

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

388.1 Składnia

```
C#  
  
List<ReCourtUpdateHistory> GetUpdatesHistory(  
    Guid machineId  
)
```

388.1.1.1 Parametry

machineId

Typ: System.Guid

388.1.1.2 Zwraca

Typ: **List(ReCourtUpdateHistory)**

388.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



389Metoda IUpdateService.HasAnyUpdatesForMachine

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

389.1 Składnia

C#	
<pre>bool HasAnyUpdatesForMachine (Guid identifier)</pre>	

389.1.1.1 Parametry

identifier

Typ: System.Guid

389.1.1.2 Zwraca

Typ: **Boolean**

389.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



390Metoda IUpdateService.ProductsForConfigType

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

390.1 Składnia

```
C#
List<string> ProductsForConfigType (
    string configType,
    OperatingSystem os
)
```

390.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

390.1.1.2 Zwraca

Typ: **List(String)**

390.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



391Metoda IUpdateService.ProductsForOperatingSystem

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

391.1 Składnia

```
C#  
  
List<string>                ProductsForOperatingSystem (  
    OperatingSystem                os  
)
```

391.1.1.1 Parametry

os

Typ: **OperatingSystem**

391.1.1.2 Zwraca

Typ: **List(String)**

391.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



392 Metoda IUpdateService.RegisterDummyPackage

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

392.1 Składnia

```
C#  
  
void RegisterDummyPackage (  
    string name,  
    string os,  
    Version version  
)
```

392.1.1.1 Parametry

name

Typ: System.String

os

Typ: System.String

version

Typ: System.Version

392.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

393 Metoda IUpdateService.RemoveConfigGroup

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

393.1 Składnia

```
C#  
  
bool RemoveConfigGroup(  
    MachineConfigType configType  
)
```

393.1.1.1 Parametry

configType

Typ: **MachineConfigType**

393.1.1.2 Zwraca

Typ: **Boolean**

393.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



394 Metoda IUpdateService.ReportInstallationStatus

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

394.1 Składnia

```
C#  
  
void ReportInstallationStatus(  
    Guid identifier,  
    long updateId,  
    UpdateStatus status  
)
```

394.1.1.1 Parametry

identifier

Typ: System.Guid

updateId

Typ: System.Int64

status

Typ: **UpdateStatus**

394.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



395 Metoda IUpdateService.RescanDropDirectry

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

395.1 Składnia

C#

```
void RescanDropDirectry ()
```

395.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



396 Metoda IUpdateService.ScheduleAllPackagesInstallOnMachine

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.Update.Base (w

ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

396.1 Składnia

C#

```
void ScheduleAllPackagesInstallOnMachine (
    Guid identifier
)
```

396.1.1.1 Parametry

identifier

Typ: System.Guid

396.2 Zobacz również



[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



397 Metoda IUpdateService.SchedulePackageInstallationForConfigType

397.1 Przeciżenia

Nazwa	Opis
 SchedulePackageInstallationForConfigType(String, OperatingSystem, List(Int32))	
 SchedulePackageInstallationForConfigType(String, OperatingSystem, Int32)	

397.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

398Metoda IUpdateService.SchedulePackageInstallationForConfigType (String, OperatingSystem, List(Int32))

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

398.1 Składnia

C#

```
void SchedulePackageInstallationForConfigType (  
    string configType,  
    OperatingSystem os,  
    List<int> packageIds  
)
```

398.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

packageIds

Typ: System.Collections.Generic.List(**Int32**)

398.2 Zobacz również

[Interfejs IUpdateService](#)

[Przeciążenia SchedulePackageInstallationForConfigType](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

399Metoda IUpdateService.SchedulePackageInstallationForConfigType (String, OperatingSystem, Int32)

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

399.1 Składnia

C#

```
void SchedulePackageInstallationForConfigType (
    string configType,
    OperatingSystem os,
    int packageId
)
```

399.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

packageId

Typ: System.Int32

399.2 Zobacz również

[Interfejs IUpdateService](#)

[Przeciążenia SchedulePackageInstallationForConfigType](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



400Metoda IUpdateService.SchedulePackageInstallationOnMachine

400.1 Przeciżenia

	Nazwa	Opis
	SchedulePackageInstallationOnMachine(Guid, List<Int32>)	
	SchedulePackageInstallationOnMachine(Guid, Int32)	

400.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



401 Metoda `IUpdateService.SchedulePackageInstallationOnMachine` (Guid, List(Int32))

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

401.1 Składnia

C#

```
void SchedulePackageInstallationOnMachine(  
    Guid identifier,  
    List<int> packageIds  
)
```

401.1.1.1 Parametry

identifier

Typ: System.Guid

packageIds

Typ: System.Collections.Generic.List(Int32)

401.2 Zobacz również

[Interfejs IUpdateService](#)

[Przebieżenia SchedulePackageInstallationOnMachine](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



402 Metoda IUpdateService.SchedulePackageInstallationOnMachine (Guid, Int32)

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

402.1 Składnia

```
C#  
  
void SchedulePackageInstallationOnMachine(  
    Guid identifier,  
    int packageId  
)
```

402.1.1.1 Parametry

identifier

Typ: System.Guid

packageId

Typ: System.Int32

402.2 Zobacz również

[Interfejs IUpdateService](#)

[Przeciążenia SchedulePackageInstallationOnMachine](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



403 Metoda IUpdateService.SetConfigTypeAutoUpdate

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

403.1 Składnia

C#

```
void SetConfigTypeAutoUpdate (  
    string configType,  
    OperatingSystem os,  
    bool autoUpdate  
)
```

403.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

autoUpdate

Typ: System.Boolean

403.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

404 Metoda IUpdateService.SetConfigTypeProducts

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

404.1 Składnia

```
C#  
  
void SetConfigTypeProducts (  
    string configType,  
    OperatingSystem os,  
    List<string> products  
)
```

404.1.1.1 Parametry

configType

Typ: System.String

os

Typ: **OperatingSystem**

products

Typ: System.Collections.Generic.List(**String**)

404.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



405 Metoda IUpdateService.UpdatesForMachine

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

405.1 Składnia

C#	
List<UpdateRequest> Guid)	UpdatesForMachine (<i>identifier</i>

405.1.1.1 Parametry

identifier

Typ: System.Guid

405.1.1.2 Zwraca

Typ: **List(UpdateRequest)**

405.2 Zobacz również

[Interfejs IUpdateService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

406 Klasa MachineHash

406.1 Hierarchia

System.Object

ReCourt.Services.Communication.Update.Base.MachineHash

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
 Moduł: ReCourt.Services.Communication.Update.Base (w
 ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

406.2 Składnia

C#

```
public static class MachineHash
```

MachineHash prezentuje składowe.

406.3 Konstruktory

	Nazwa	Opis
	MachineHash	

406.4 Metody

	Nazwa	Opis
	Calculate	
	HashString	

406.5 Pola

	Nazwa	Opis
	Empty	

406.6 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

407Konstruktor MachineHash

Inicjalizuje statyczne pole klasy [MachineHash](#)

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

407.1 Składnia

C#

```
static MachineHash()
```

407.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



408 Metody MachineHash.MachineHash

[MachineHash](#) prezentuje składowe.

408.1 Metody

	Nazwa	Opis
	Calculate	
	HashString	

408.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

409 Metoda MachineHash.Calculate

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

409.1 Składnia

```
C#  
  
public static byte[] Calculate(  
    string machineType,  
    string location = null  
)
```

409.1.1.1 Parametry

machineType

Typ: System.String

location (Optional)

Typ: System.String

409.1.1.2 Zwraca

Typ: **Byte[]**

409.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



410 Metoda MachineHash.HashString

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

410.1 Składnia

```
C#  
  
private          static          void          HashString(  
    MD5  
    string  
)  
    md5,  
    s
```

410.1.1.1 Parametry

md5

Typ: System.Security.Cryptography.MD5

s

Typ: System.String

410.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

411Pola MachineHash.MachineHash

411.1 Pola

	Nazwa	Opis
	Empty	

411.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



412 Pole MachineHash.Empty

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.Update.Base (w
ReCourt.Services.Communication.Update.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

412.1 Składnia

C#

```
private static readonly byte[] Empty
```

412.1.1.1 Zwraca

Typ: **Byte[]**

412.2 Zobacz również

[Klasa MachineHash](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

413 Klasa `SendFileRequestMessage`

413.1 Hierarchia

System.Object

ReCourt.Services.Communication.Update.Base.SendFileRequestMessage

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)

Moduł: ReCourt.Services.Communication.ReCourtUpdate (w

ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)


413.2 Składnia

C#

```
public class SendFileRequestMessage
```

SendFileRequestMessage prezentuje składowe.

413.3 Konstruktory

	Nazwa	Opis
	SendFileRequestMessage	Inicjalizuje nową instancję klasy SendFileRequestMessage

413.4 Pola

	Nazwa	Opis
	FileData	
	FileInfo	

413.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

414 Konstruktor `SendFileRequestMessage`

Inicjalizuje nową instancję klasy [SendFileRequestMessage](#)

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

414.1 Składnia

C#

```
public SendFileRequestMessage ()
```

414.2 Zobacz również

[Klasa `SendFileRequestMessage`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)



415 Pola `SendMessage.SendMessage`

[SendMessage](#) prezentuje składowe.

415.1 Pola

	Nazwa	Opis
	FileData	
	FileInfo	

415.2 Zobacz również

[Klasa SendMessage](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)



416 Pole `SendMessageRequestMessage.FileData`

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

416.1 Składnia

C#

```
public byte[] FileData
```

416.1.1.1 Zwraca

Typ: **Byte[]**

416.2 Zobacz również

[Klasa `SendMessageRequestMessage`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Update.Base`](#)



417 Pole `SendMessage.FileInfo`

Przestrzeń nazw: [ReCourt.Services.Communication.Update.Base](#)
Moduł: ReCourt.Services.Communication.ReCourtUpdate (w
ReCourt.Services.Communication.ReCourtUpdate.dll) Wersja: 1.0.0.0 (1.0.0.0)

417.1 Składnia

C#

```
public FileTransferInfo FileInfo
```

417.1.1.1 Zwraca

Typ: [FileTransferInfo](#)


417.2 Zobacz również

[Klasa SendMessage](#)

[Przestrzeń nazw ReCourt.Services.Communication.Update.Base](#)

418Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base

418.1 Interfejsy

	Interfejs	Opis
	IUserRepositoryCommunicationService	Interfejs odpowiada za zarządzanie użytkownikami. Umożliwia zmianę uprawnień oraz ról.

419 Interfejs IUserRepositoryCommunicationService

Interfejs odpowiada za zarządzanie użytkownikami. Umożliwia zmianę uprawnień oraz ról.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
 Moduł: ReCourt.Services.Communication.UserRepository.Base (w
 ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)













419.1 Składnia

















C#

```
public interface IUserRepositoryCommunicationService
```

IUserRepositoryCommunicationService prezentuje składowe.

419.2 Metody

	Nazwa	Opis
	AddRole	Dodaje nową rolę.
	AddUser	Dodanie użytkownika.
	AuthorizeUser	Uwierzytelnianie użytkownika.
	EnableUser	Włączanie użytkownika lub wyłączenie.
	GetAction	Pobiera akcję na podstawie kodu.
	GetActions	Pobiera listę akcji.
	GetActiveDirectoryUserNames	Pobiera listę użytkowników w domenie.
	GetLastVisitedStatistics	Pobiera informację o ilości nowych zadań, spraw i dokumentów od ostatniej wizyty użytkownika.
	GetReportRights	Pobiera listę raportów.
	GetRole	Pobiera rolę.
	GetRoles	Pobiera role uwzględniając filtr.
	GetUserACL	Pobiera ACL użytkownika.

 GetUserById	Pobiera użytkownika o podanym identyfikatorze.
 GetUserByName	Pobiera użytkownika o podanej nazwie.
 GetUserNames	Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.
 GetUserPrivilegesVO	Pobiera uprawnienia użytkownika.
 GetUserRoles	Pobiera listę ról użytkownika.
 GetUsers	Pobiera użytkowników umożliwiając filtrację listy wynikowej.
 GetUsersForAction	Pobiera listę użytkowników uprawnionych do akcji.
 IsAuthorizedUser	Sprawdza uwierzytelnienie użytkownika.
 IsUserInDomain	Sprawdzenie czy użytkownik jest w domenie.
 RequestNewDiscriminant	Zgłasza żądanie nadania uprawnień do wyróżnika.
 RequestNewRole	Zgłasza żądanie nadania roli.
 RespondToDiscriminantRequest	Odpowiada na żądanie nadania uprawnień do wyróżnika.
 RespondToRoleRequest	Odpowiada na żądanie nadania roli.
 UpdateRole	Aktualizuje dane roli.
 UpdateUser	Aktualizuje dane użytkownika.
 UpdateUserLastVisited	Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

419.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)













420Metody

IUserRepositoryCommunicationService.IUserRepositoryCommunicationService

[IUserRepositoryCommunicationService](#) prezentuje składowe.

420.1 Metody

Nazwa	Opis
AddRole	Dodaje nową rolę.
AddUser	Dodanie użytkownika.
AuthorizeUser	Uwierzytelnianie użytkownika.
EnableUser	Włączanie użytkownika lub wyłączenie.
GetAction	Pobiera akcję na podstawie kodu.
GetActions	Pobiera listę akcji.
GetActiveDirectoryUserNames	Pobiera listę użytkowników w domenie.
GetLastVisitedStatistics	Pobiera informację o ilości nowych zadań, spraw i dokumentów od ostatniej wizyty użytkownika.
GetReportRights	Pobiera listę raportów.
GetRole	Pobiera rolę.
GetRoles	Pobiera role uwzględniając filtr.
GetUserACL	Pobiera ACL użytkownika.
GetUserById	Pobiera użytkownika o podanym identyfikatorze.
GetUserByName	Pobiera użytkownika o podanej nazwie.
GetUserNames	Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.
GetUserPrivilegesVO	Pobiera uprawnienia użytkownika.

 GetUserRoles	Pobiera listę ról użytkownika.
 GetUsers	Pobiera użytkowników umożliwiając filtrację listy wynikowej.
 GetUsersForAction	Pobiera listę użytkowników uprawnionych do akcji.
 IsAuthorizedUser	Sprawdza uwierzytelnienie użytkownika.
 IsUserInDomain	Sprawdzenie czy użytkownik jest w domenie.
 RequestNewDiscriminant	Zgłasza żądanie nadania uprawnień do wyróżnika.
 RequestNewRole	Zgłasza żądanie nadania roli.
 RespondToDiscriminantRequest	Odpowiada na żądanie nadania uprawnień do wyróżnika.
 RespondToRoleRequest	Odpowiada na żądanie nadania roli.
 UpdateRole	Aktualizuje dane roli.
 UpdateUser	Aktualizuje dane użytkownika.
 UpdateUserLastVisited	Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

420.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



421 Metoda IUserRepositoryCommunicationService.AddRole

Dodaje nową rolę.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

421.1 Składnia

C#	
<pre>int ReCourtRole)</pre>	<pre>AddRole(role</pre>

421.1.1.1 Parametry

role

Typ: **ReCourtRole**

Rola

421.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanej roli.

421.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

422 Metoda IUserRepositoryCommunicationService.AddUser

Dodanie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

422.1 Składnia

```
C#  
  
int AddUser (  
    ReCourtUser user,  
    List<ReCourtRole> roles,  
    List<ReCourtUserACL> userAcl  
)
```

422.1.1.1 Parametry

user

Typ: **ReCourtUser**

Dane użytkownika

roles

Typ: System.Collections.Generic.List(**ReCourtRole**)

Role

userAcl

Typ: System.Collections.Generic.List(**ReCourtUserACL**)

Uprawnienia

422.1.1.2 Zwraca

Typ:

Identyfikator dodanego użytkownika.

Int32

422.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

423 Metoda IUserRepositoryCommunicationService.AuthorizeUser

Uwierzytelnianie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

423.1 Składnia

C#	
<pre>bool string string)</pre>	<pre>AuthorizeUser(username, userPassword</pre>

423.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

userPassword

Typ: System.String

Hasło

423.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

423.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

424Metoda IUserRepositoryCommunicationService.EnableUser

Włączanie użytkownika lub wyłączenie.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

424.1 Składnia

```
C#  
  
bool EnableUser (  
    int     userId,  
    bool    enable  
)
```

424.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

enable

Typ: System.Boolean

Czy włączony

424.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

424.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



425 Metoda IUserRepositoryCommunicationService.GetAction

Pobiera akcję na podstawie kodu.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

425.1 Składnia

C#	
<pre>ReCourtAction string)</pre>	<pre>GetAction(code</pre>

425.1.1.1 Parametry

code

Typ: System.String

Kod akcji

425.1.1.2 Zwraca

Typ:

ReCourtAction

Pobrana akcja.

425.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



426 Metoda IUserRepositoryCommunicationService.GetActions

Pobiera listę akcji.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

426.1 Składnia

C#

```
List<ReCourtActionVO> GetActions()
```

426.1.1.1 Zwraca

Typ:

List(ReCourtActionVO)

Lista akcji.

426.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

427Metoda

IUserRepositoryCommunicationService.GetActiveDirectoryUserNames

Pobiera listę użytkowników w domenie.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

427.1 Składnia

```
C#  
  
PagedList<ReCourtDomainUser> GetActiveDirectoryUserNames (  
    string filter,  
    int pageNumber  
)
```

427.1.1.1 Parametry

filter

Typ: System.String

Filtr

pageNumber

Typ: System.Int32

Strona

427.1.1.2 Zwraca

Typ: **PagedList(ReCourtDomainUser)**
Stronicowana lista użytkowników w domenie.

427.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

428 Metoda IUserRepositoryCommunicationService.GetLastVisitedStatistics

Pobiera informację o ilości nowych zadań, spraw i dokumentów od ostatniej wizyty użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

428.1 Składnia

C#		
Dictionary<DataTypeEnum,	int>	GetLastVisitedStatistics(int userId)

428.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

428.1.1.2 Zwraca

Typ: **Dictionary(DataTypeEnum, Int32)**
Ilości nowych elementów od ostatniej wizyty.

428.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

429 Metoda IUserRepositoryCommunicationService.GetReportRights

Pobiera listę raportów.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

429.1 Składnia

```
C#  
  
List<ReCourtReportRightVO> GetReportRights (  
    int userId  
)
```

429.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

429.1.1.2 Zwraca

Typ:

List(ReCourtReportRightVO)

Lista raportów.

429.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



430Metoda IUserRepositoryCommunicationService.GetRole

Pobiera rolę.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Communication.UserRepository.Base (w

ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

430.1 Składnia

C#	
ReCourtRole	GetRole (
int	roleId
))

430.1.1.1 Parametry

roleId

Typ: System.Int32

Id roli

430.1.1.2 Zwraca

Typ:

ReCourtRole

Znaleziona rola.

430.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

431 Metoda IUserRepositoryCommunicationService.GetRoles

Pobiera role uwzględniając filtr.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

431.1 Składnia

```
C#  
  
List<ReCourtRoleVO> GetRoles (  
    ReCourtRoleCriteria filterCriteria  
)
```

431.1.1.1 Parametry

filterCriteria

Typ: **ReCourtRoleCriteria**

Filtr

431.1.1.2 Zwraca

Typ:

List(ReCourtRoleVO)

Lista ról.

431.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

432 Metoda IUserRepositoryCommunicationService.GetUserACL

Pobiera ACL użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

432.1 Składnia

```
C#  
  
List<ReCourtUserACL> GetUserACL(  
    int userId  
)
```

432.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

432.1.1.2 Zwraca

Typ:

List(ReCourtUserACL)

Lista ACL użytkownika.

432.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



433 Metoda IUserRepositoryCommunicationService.GetUserById

Pobiera użytkownika o podanym identyfikatorze.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

433.1 Składnia

```
C#  
  
ReCourtUser          GetUserById(  
    int                userId  
)
```

433.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

433.1.1.2 Zwraca

Typ:

ReCourtUser

Obiekt użytkownika.

433.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



434 Metoda IUserRepositoryCommunicationService.GetUserByName

Pobiera użytkownika o podanej nazwie.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

434.1 Składnia

C#	
ReCourtUser	GetUserByName (
string	<i>userName</i>
))

434.1.1.1 Parametry

userName

Typ: System.String

Nazwa użytkownika

434.1.1.2 Zwraca

Typ:

ReCourtUser

Obiekt użytkownika.

434.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



435 Metoda IUserRepositoryCommunicationService.GetUserNames

Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

435.1 Składnia

C#

```
Dictionary<int, string> GetUserNames()
```

435.1.1.1 Zwraca

Typ: **Dictionary(Int32, String)**
Słownik nazw użytkowników wraz z ich identyfikatorami.

435.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

436 Metoda IUserRepositoryCommunicationService.GetUserPrivilegesVO

Pobiera uprawnienia użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

436.1 Składnia

```
C#  
  
List<ReCourtUserACLVO>          GetUserPrivilegesVO (  
    int                          userId  
)
```

436.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

436.1.1.2 Zwraca

Typ: **List(ReCourtUserACLVO)**

Lista uprawnień użytkownika.

436.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

437Metoda IUserRepositoryCommunicationService.GetUserRoles

Pobiera listę ról użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Communication.UserRepository.Base (w

ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

437.1 Składnia

```
C#  
  
List<ReCourtRole>          GetUserRoles (  
    int                    userId  
)
```

437.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

437.1.1.2 Zwraca

Typ:

List(ReCourtRole)

Lista ról.

437.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

438 Metoda IUserRepositoryCommunicationService.GetUsers

Pobiera użytkowników umożliwiając filtrację listy wynikowej.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

438.1 Składnia

```
C#  
  
PagedList<ReCourtUserVO> GetUsers (  
    ReCourtUserCriteria    filterCriteria,  
    int                    page,  
    int                    recordsPerPage,  
    SortCriteria           sortCriteria  
)
```

438.1.1.1 Parametry

filterCriteria

Typ: **ReCourtUserCriteria**

Filtr

page

Typ: System.Int32

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

438.1.1.2 Zwraca

Typ: **PagedList(ReCourtUserVO)**

Stronicowana lista użytkowników.

438.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

439 Metoda IUserRepositoryCommunicationService.GetUsersForAction

Pobiera listę użytkowników uprawnionych do akcji.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

439.1 Składnia

```
C#  
  
List<ReCourtUser> GetUsersForAction (  
    string          actionName,  
    string          discriminantCode  
)
```

439.1.1.1 Parametry

actionName

Typ: System.String

Kod akcji

discriminantCode

Typ: System.String

Wyróżnik

439.1.1.2 Zwraca

Typ:

Lista użytkowników.

List(ReCourtUser)

439.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

440Metoda IUserRepositoryCommunicationService.IsAuthorizedUser

Sprawdza uwierzytelnienie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

440.1 Składnia

```
C#  
  
bool IsAuthorizedUser(  
    string username  
)
```

440.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

440.1.1.2 Zwraca

Typ:

Boolean

True jeśli użytkownik jest uwierzytelniony, w przeciwnym razie false.

440.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

441 Metoda IUserRepositoryCommunicationService.IsUserInDomain

Sprawdzenie czy użytkownik jest w domenie.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

441.1 Składnia

```
C#  
  
int IsUserInDomain(  
    string username  
)
```

441.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

441.1.1.2 Zwraca

Typ:

Int32

True jeśli użytkownik w domenie, w przeciwnym razie false.

441.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

442 Metoda IUserRepositoryCommunicationService.RequestNewDiscriminant

Zgłasza żądanie nadania uprawnień do wyróżnika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

442.1 Składnia

```
C#  
  
bool RequestNewDiscriminant (  
    ReCourtUser requestingUser,  
    string discriminantCode,  
    string reason  
)
```

442.1.1.1 Parametry

requestingUser

Typ: **ReCourtUser**

Dane użytkownika

discriminantCode

Typ: System.String

Opis

reason

Typ: System.String

Powód

442.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

442.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

443 Metoda IUserRepositoryCommunicationService.RequestNewRole

Zgłasza żądanie nadania roli.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

443.1 Składnia

C#	
<pre>bool ReCourtUser string string)</pre>	<pre>RequestNewRole(requestingUser, roleDescription, reason</pre>

443.1.1.1 Parametry

requestingUser

Typ: **ReCourtUser**

Dane użytkownika

roleDescription

Typ: System.String

Opis

reason

Typ: System.String

Powód

443.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

443.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

444 Metoda

IUserRepositoryCommunicationService.RespondToDiscriminantRequest

Odpowiada na żądanie nadania uprawnień do wyróżnika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

444.1 Składnia

```
C#
bool RespondToDiscriminantRequest (
    string response,
    string taskGuid,
    ReCourtUser respondingUser
)
```

444.1.1.1 Parametry

response

Typ: System.String

Odpowiedź

taskGuid

Typ: System.String

Id zadania

respondingUser

Typ: **ReCourtUser**

Użytkownik

444.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

444.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

445 Metoda IUserRepositoryCommunicationService.RespondToRoleRequest

Odpowiada na żądanie nadania roli.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

445.1 Składnia

C#	
<pre>bool string string ReCourtUser)</pre>	<pre>RespondToRoleRequest (response, taskGuid, respondingUser</pre>

445.1.1.1 Parametry

response

Typ: System.String

Odpowiedź

taskGuid

Typ: System.String

Id zadania

respondingUser

Typ: **ReCourtUser**

Użytkownik

445.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

445.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



446 Metoda IUserRepositoryCommunicationService.UpdateRole

Aktualizuje dane roli.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

446.1 Składnia

C#	
<pre>bool ReCourtRole)</pre>	<pre>UpdateRole(role</pre>

446.1.1.1 Parametry

role

Typ: **ReCourtRole**

Rola

446.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

446.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

447 Metoda IUserRepositoryCommunicationService.UpdateUser

Aktualizuje dane użytkownika.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Communication.UserRepository.Base (w
ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

447.1 Składnia

C#	
bool	UpdateUser (
ReCourtUser	<i>user</i> ,
List<ReCourtRole>	<i>roles</i> ,
List<ReCourtUserACL>	<i>userAcl</i>
))

447.1.1.1 Parametry

user

Typ: **ReCourtUser**

Dane użytkownika

roles

Typ: System.Collections.Generic.List(**ReCourtRole**)

Role

userAcl

Typ: System.Collections.Generic.List(**ReCourtUserACL**)

Uprawnienia

447.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

447.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)

448 Metoda IUserRepositoryCommunicationService.UpdateUserLastVisited

Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

Przestrzeń nazw: [ReCourt.Services.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Communication.UserRepository.Base (w

ReCourt.Services.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

448.1 Składnia

```
C#
bool UpdateUserLastVisited(
    int userId,
    DataTypeEnum dataType
)
```

448.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

dataType

Typ: **DataTypeEnum**

Typ sprawdzenia

448.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

448.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.UserRepository.Base](#)



449Przestrzeń

nazw

ReCourt.Services.Communication.Workflow.ReadingRoom.Base

449.1 Interfejsy

	Interfejs	Opis
→	IDocumentsReadingRoomCommunicationService	Interfejs odpowiada za zarządzanie obiegami dokumentów. Wchodzi w niego skład tworzenie, akceptowania oraz odrzucanie danych obiegów.



450 Interfejs IDocumentsReadingRoomCommunicationService

Interfejs odpowiada za zarządzanie obiegami dokumentów. Wchodzi w niego skład tworzenie, akceptowania oraz odrzucanie danych obiegów.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w
ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

450.1 Składnia

C#

```
public interface IDocumentsReadingRoomCommunicationService
```

IDocumentsReadingRoomCommunicationService prezentuje składowe.

450.2 Metody

	Nazwa	Opis
	AcceptWorkflow	Akceptuje obieg dokumentu.
	AcceptWorkflowISO	Akceptuje obieg dokumentu ISO.
	CloseWorkflow	Zamyka obieg dokumentu.
	CloseWorkflowISO	Zamyka obieg dokumentu ISO.
	CreateWorkflow	Tworzy obieg dokumentu.
	CreateWorkflowISO	Tworzy obieg dokumentu ISO.
	RejectWorkflow	Odrzuca obieg dokumentu z określonego powodu.
	RejectWorkflowISO	Odrzuca obieg dokumentu ISO z określonego powodu.

450.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

451 Metody

IDocumentsReadingRoomCommunicationService.IDocumentsReadingRoomCommunicationService

[IDocumentsReadingRoomCommunicationService](#) prezentuje składowe.

451.1 Metody

	Nazwa	Opis
	AcceptWorkflow	Akceptuje obieg dokumentu.
	AcceptWorkflowISO	Akceptuje obieg dokumentu ISO.
	CloseWorkflow	Zamyka obieg dokumentu.
	CloseWorkflowISO	Zamyka obieg dokumentu ISO.
	CreateWorkflow	Tworzy obieg dokumentu.
	CreateWorkflowISO	Tworzy obieg dokumentu ISO.
	RejectWorkflow	Odrzuca obieg dokumentu z określonego powodu.
	RejectWorkflowISO	Odrzuca obieg dokumentu ISO z określonego powodu.

451.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

452Metoda IDocumentsReadingRoomCommunicationService.AcceptWorkflow

Akceptuje obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w
ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

452.1 Składnia

C#	
<pre>bool Document ReCourtUser)</pre>	<pre>AcceptWorkflow(document, user</pre>

452.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

452.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

452.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

453Metoda IDocumentsReadingRoomCommunicationService.AcceptWorkflowISO

Akceptuje obieg dokumentu ISO.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w
ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

453.1 Składnia

C#	
bool	AcceptWorkflowISO (
Document	document,
ReCourtUser	user
)	

453.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

453.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

453.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

454Metoda IDocumentsReadingRoomCommunicationService.CloseWorkflow

Zamyka obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w
ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

454.1 Składnia

C#	
bool	CloseWorkflow(
Document	<i>document,</i>
ReCourtUser	<i>user</i>
))

454.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

454.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

454.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

455 Metoda IDocumentsReadingRoomCommunicationService.CloseWorkflowISO

Zamyka obieg dokumentu ISO.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w
ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

455.1 Składnia

C#	
<pre>bool Document ReCourtUser)</pre>	<pre>CloseWorkflowISO(document, user</pre>

455.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

455.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

455.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

456Metoda IDocumentsReadingRoomCommunicationService.CreateWorkflow

Tworzy obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

456.1 Składnia

```
C#
bool CreateWorkflow(
    Document document,
    string previewTranscriptionDocumentTaskListItemController,
    string manageTranscriptionDocumentTaskListItemController
)
```

456.1.1.1 Parametry

document

Typ: **Document**

Dokument

previewTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler podglądu

manageTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler zarządzania

456.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

456.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

457Metoda IDocumentsReadingRoomCommunicationService.CreateWorkflowISO

Tworzy obieg dokumentu ISO.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

457.1 Składnia

```
C#
bool CreateWorkflowISO (
    Document document,
    string previewTranscriptionDocumentTaskListItemController,
    string manageTranscriptionDocumentTaskListItemController
)
```

457.1.1.1 Parametry

document

Typ: **Document**

Dokument

previewTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler podglądu

manageTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler zarządzania

457.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

457.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

458Metoda IDocumentsReadingRoomCommunicationService.RejectWorkflow

Odrzuca obieg dokumentu z określonego powodu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

458.1 Składnia

C#	
<pre>bool Document string ReCourtUser)</pre>	<pre>RejectWorkflow(document, reason, user</pre>

458.1.1.1 Parametry

document

Typ: **Document**

Dokument

reason

Typ: System.String

Powód

user

Typ: **ReCourtUser**

Użytkownik

458.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

458.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

459Metoda IDocumentsReadingRoomCommunicationService.RejectWorkflowISO

Odrzuca obieg dokumentu ISO z określonego powodu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.ReadingRoom.Base (w ReCourt.Services.Communication.Workflow.ReadingRoom.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

459.1 Składnia

C#	
bool	RejectWorkflowISO (
Document	<i>document</i> ,
string	<i>reason</i> ,
ReCourtUser	<i>user</i>
))

459.1.1.1 Parametry

document

Typ: **Document**

Dokument

reason

Typ: System.String

Powód

user

Typ: **ReCourtUser**

Użytkownik

459.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

459.2 Zobacz również

[Interfejs IDocumentsReadingRoomCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.ReadingRoom.Base](#)

460Przestrzeń


nazw

ReCourt.Services.Communication.Workflow.Transcription.Base

460.1 Klasy

	Klasa	Opis
	FileTransferInfo	Dane pliku
	UploadFileRequest	Dane pliku

460.2 Interfejsy

	Interfejs	Opis
	IDocumentsTranscriptionCommunicationService	Interfejs odpowiada za zarządzanie transkrypcjami. Umożliwia akceptowania zadania importu, odrzucenie go, wgranie pliku z transkrypcją oraz zarządzanie obiegiem konkretnego dokumentu.

461 Klasa FileTransferInfo

Dane pliku

461.1 Hierarchia

System.Object

ReCourt.Services.Communication.Workflow.Transcription.Base.FileTransferInfo

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)


461.2 Składnia

C#

```
public class FileTransferInfo
```

FileTransferInfo prezentuje składowe.

461.3 Konstruktory

	Nazwa	Opis
	FileTransferInfo	Inicjalizuje nową instancję klasy FileTransferInfo

461.4 Pola

	Nazwa	Opis
	Document	Dla dokumentu
	FileName	Nazwa pliku
	User	Użytkownik

461.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)



462 Konstruktor FileTransferInfo

Inicjalizuje nową instancję klasy [FileTransferInfo](#)

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w

ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

462.1 Składnia

C#

```
public FileTransferInfo ()
```

462.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

463 Pola FileTransferInfo.FileTransferInfo

[FileTransferInfo](#) prezentuje składowe.

463.1 Pola

	Nazwa	Opis
	Document	Dla dokumentu
	FileName	Nazwa pliku
	User	Użytkownik

463.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)



464 Pole FileTransferInfo.Document

Dla dokumentu

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

464.1 Składnia

C#

```
public Document Document
```

464.1.1.1 Zwraca

Typ: **Document**

464.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

465 Pole FileTransferInfo.FileName

Nazwa pliku

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w

ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

465.1 Składnia

C#

```
public string FileName
```

465.1.1.1 Zwraca

Typ: **String**

465.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

466 Pole FileTransferInfo.User

Użytkownik

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w

ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

466.1 Składnia

C#

```
public ReCourtUser User
```

466.1.1.1 Zwraca

Typ: **ReCourtUser**

466.2 Zobacz również

[Klasa FileTransferInfo](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

467 Interfejs IDocumentsTranscriptionCommunicationService

Interfejs odpowiada za zarządzanie transkrypcjami. Umożliwia akceptowania żądania importu, odrzucenie go, wgranie pliku z transkrypcją oraz zarządzanie obiegiem konkretnego dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
 Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
 ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

467.1 Składnia

C#

```
public interface IDocumentsTranscriptionCommunicationService
```

IDocumentsTranscriptionCommunicationService prezentuje składowe.

467.2 Metody

	Nazwa	Opis
	AcceptImportRequest	Akceptuje żądanie importu.
	AcceptWorkflow	Akceptuje obieg dokumentu.
	CreateWorkflow	Tworzy obieg dokumentu.
	RejectImportRequest	Odrzuca żądanie importu.
	RejectWorkflow	Odrzuca obieg dokumentu.
	UploadTranscriptionFile	Wgrywa plik z transkrypcją.

467.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

468 Metody

IDocumentsTranscriptionCommunicationService.IDocumentsTranscriptionCommunicationService

[IDocumentsTranscriptionCommunicationService](#) prezentuje składowe.

468.1 Metody

	Nazwa	Opis
	AcceptImportRequest	Akceptuje żądanie importu.
	AcceptWorkflow	Akceptuje obieg dokumentu.
	CreateWorkflow	Tworzy obieg dokumentu.
	RejectImportRequest	Odrzuca żądanie importu.
	RejectWorkflow	Odrzuca obieg dokumentu.
	UploadTranscriptionFile	Wgrywa plik z transkrypcją.

468.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

469 Metoda

IDocumentsTranscriptionCommunicationService.AcceptImportRequest

Akceptuje żądanie importu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

469.1 Składnia

C#	
<pre>bool Document ReCourtUser)</pre>	<pre>AcceptImportRequest (document, user</pre>

469.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

469.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

469.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

470Metoda IDocumentsTranscriptionCommunicationService.AcceptWorkflow

Akceptuje obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

470.1 Składnia

C#	
<pre>bool Document ReCourtUser)</pre>	<pre>AcceptWorkflow(document, user</pre>

470.1.1.1 Parametry

document

Typ: **Document**

Dokument

user

Typ: **ReCourtUser**

Użytkownik

470.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

470.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

471Metoda IDocumentsTranscriptionCommunicationService.CreateWorkflow

Tworzy obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

471.1 Składnia

```
C#
bool CreateWorkflow(
    Document document,
    string
    previewTranscriptionDocumentTaskListItemController,
    string manageTranscriptionDocumentTaskListItemController
)
```

471.1.1.1 Parametry

document

Typ: **Document**

Dokument

previewTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler podglądu

manageTranscriptionDocumentTaskListItemController

Typ: System.String

Kontroler zarządzania

471.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

471.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

472Metoda

IDocumentsTranscriptionCommunicationService.RejectImportRequest

Odrzuca żądanie importu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

472.1 Składnia

C#	
<pre>bool Document string ReCourtUser)</pre>	<pre>RejectImportRequest(document, reason, user</pre>

472.1.1.1 Parametry

document

Typ: **Document**

reason

Typ: System.String

user

Typ: **ReCourtUser**

472.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

472.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

473Metoda IDocumentsTranscriptionCommunicationService.RejectWorkflow

Odrzuca obieg dokumentu.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w

ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

473.1 Składnia

C#	
<pre>bool Document string ReCourtUser)</pre>	<pre>RejectWorkflow(document, reason, user</pre>

473.1.1.1 Parametry

document

Typ: **Document**

Dokument

reason

Typ: System.String

Powód

user

Typ: **ReCourtUser**

Użytkownik

473.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

473.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

474 Metoda

IDocumentsTranscriptionCommunicationService.UploadTranscriptionFile

Wgrywa plik z transkrypcją.

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

474.1 Składnia

```
C#  
  
void UploadTranscriptionFile(  
    UploadFileRequest request  
)
```

474.1.1.1 Parametry

request

Typ: [ReCourt.Services.Communication.Workflow.Transcription.Base.UploadFileRequest](#)

Plik

474.2 Zobacz również

[Interfejs IDocumentsTranscriptionCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

475 Klasa UploadFileRequest

Dane pliku

475.1 Hierarchia

System.Object

ReCourt.Services.Communication.Workflow.Transcription.Base.UploadFileRequest

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)


475.2 Składnia

C#

```
public class UploadFileRequest
```

UploadFileRequest prezentuje składowe.

475.3 Konstruktory

	Nazwa	Opis
	UploadFileRequest	Inicjalizuje nową instancję klasy UploadFileRequest

475.4 Pola

	Nazwa	Opis
	File	Zawartość
	FileInfo	Dane pliku

475.5 Zobacz również

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)



476 Konstruktor UploadFileRequest

Inicjalizuje nową instancję klasy [UploadFileRequest](#)

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)

Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w

ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

476.1 Składnia

C#

```
public UploadFileRequest ()
```

476.2 Zobacz również

[Klasa UploadFileRequest](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)



477Pola UploadFileRequest.UploadFileRequest

[UploadFileRequest](#) prezentuje składowe.

477.1 Pola

	Nazwa	Opis
	File	Zawartość
	FileInfo	Dane pliku

477.2 Zobacz również

[Klasa UploadFileRequest](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)



478 Pole UploadFileRequest.File

Zawartość

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

478.1 Składnia

C#

```
public Stream File
```

478.1.1.1 Zwraca

Typ: **Stream**

478.2 Zobacz również

[Klasa UploadFileRequest](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

479 Pole UploadFileRequest.FileInfo

Dane pliku

Przestrzeń nazw: [ReCourt.Services.Communication.Workflow.Transcription.Base](#)
 Moduł: ReCourt.Services.Communication.Workflow.Transcription.Base (w
 ReCourt.Services.Communication.Workflow.Transcription.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

479.1 Składnia

```
C#
public FileTransferInfo FileInfo
```

479.1.1.1 Zwraca

Typ: [FileTransferInfo](#)

479.2 Zobacz również

[Klasa UploadFileRequest](#)

[Przestrzeń nazw ReCourt.Services.Communication.Workflow.Transcription.Base](#)

480 Interfejsy RCS dla agenta CAPE

Interfejsy dla systemów zewnętrznych realizowane są interfejsami JSON:

[IJsonCommunicationService](#), [IJsonFileTransferCommunicationService](#)

Dla każdego z poszczególnych interfejsów zostały wystawione odpowiednie action performery.

W skład interfejsu komunikacji wchodzi Action Performery:

Nazwa	Rodzaj zadania
CheckCaseTrialsIntegrityActionPerformer	Sprawdza integralność danych dla sprawy
GetCaseParticipantsActionPerformer	Pobiera uczestników postępowania dla sprawy
GetDiscriminantsStoragesActionPerformer	Pobiera listę zasobów dla wyróżników
GetDiscriminantStorageActionPerformer	Pobiera zasób dla wyróżnika
GetRcsConfigurationActionPerformer	Pobiera dane konfiguracyjne
GetTrialByldActionPerformer	Pobiera dane sprawy



GetTrialByTrialIdActionPerformer	Pobiera dane sprawy
GetTrialFilesActionPerformer	Pobiera informacje o plikach sprawy
GetTrialsActionPerformer	Pobiera dane spraw
GetTrialsBySignatureActionPerformer	Pobiera dane sprawy
GetTrialsGroupedByCaseActionPerformer	Pobiera dane spraw
GetTrialStorageActionPerformer	Pobieraz zasób dla sprawy
RegisterAgentServerActionPerformer	Rejestruje agenta w RCS
SetCaseCapeStatusActionPerformer	Ustawienie statusu CAPE
SetCaseCapeStatusTrialIdActionPerformer	Ustawienie statusu CAPE

Parametry dla poszczególnych AP:

CheckCaseTrialsIntegrityActionPerformer	
Kod	TRIAL_CheckCaseTrialsIntegrity
Wejściowy:	List<int>
Wyjściowy	bool

GetCaseParticipantsActionPerformer	
Kod	TRIAL_GetCaseParticipants
Wejściowy:	{ CaseSignature = string.Empty, DiscriminantCode = string.Empty }
Wyjściowy	List<TrialParticipant>

GetDiscriminantsStoragesActionPerformer	
Kod	COURT_GetDiscriminantsStorages
Wejściowy:	-
Wyjściowy	List<ReCourtDiscriminant>



GetDiscriminantStorageActionPerformer

Kod	COURT_GetDiscriminantStorage
Wejściowy:	string
Wyjściowy	ReCourtStorage

GetRcsConfigurationActionPerformer

Kod	GetRcsConfiguration
Wejściowy:	-
Wyjściowy	{ Guid = RCS.MachineIdentifier, Name = RCS.InstanceName }

GetTrialByIdActionPerformer

Kod	TRIAL_GetById
Wejściowy:	Int32
Wyjściowy	TrialInfo

GetTrialByTrialIdActionPerformer

Kod	TRIAL_GetByTrialId
Wejściowy:	string
Wyjściowy	TrialInfo

GetTrialFilesActionPerformer

Kod	TRIAL_GetTrialFiles
Wejściowy:	Int32
Wyjściowy	List<TrialFile>

GetTrialsActionPerformer	
Kod	TRIAL_GetTrials
Wejściowy:	<pre>{ PageNum = (int?)null, RecordsPerPage = (int?)null, DiscriminantCode = string.Empty, CaseSignature = string.Empty, DateFrom = (DateTime?)null, DateTo = (DateTime?)null }</pre>
Wyjściowy	PagedList<TrialInfoVO>

GetTrialsBySignatureActionPerformer	
Kod	TRIAL_GetBySignature
Wejściowy:	<pre>{ DiscriminantCode = string.Empty, CaseSignature = string.Empty, }</pre>
Wyjściowy	PagedList<TrialInfoVO>

GetTrialsGroupedByCaseActionPerformer	
Kod	TRIAL_GetTrialsGroupedByCase
Wejściowy:	<pre>{ PageNum = (int?)null, RecordsPerPage = (int?)null, DiscriminantCode = string.Empty, CaseSignatures = (List<string>)null, }</pre>



	<pre>DateFrom = (DateTime?)null, DateTo = (DateTime?)null }</pre>
Wyjściowy	PagedList<TrialInfoVO>

GetTrialStorageActionPerformer

Kod	TRIAL_GetTrialStorage
Wejściowy:	Int32
Wyjściowy	ReCourtStorage

RegisterAgentServerActionPerformer

Kod	RegisterAgentServer
Wejściowy:	string
Wyjściowy	-

SetCaseCapeStatusActionPerformer

Kod	TRIAL_SetCaseCapeStatus
Wejściowy:	<pre>{ Id = default(int), StatusCode = default(string) }</pre>
Wyjściowy	<pre>{ Result = (bool) false }</pre>

SetCaseCapeStatusTrialIdActionPerformer



Kod	TRIAL_SetCaseCapeStatus
Wejściowy:	<pre>{ TrialId = default(string), StatusCode = default(string) }</pre>
Wyjściowy	<pre>{ Result = (bool) false }</pre>

481 Interfejsy RCS dla systemu Portali Informacyjnych

W skład interfejsu komunikacji wchodzi Action Performery:

Nazwa	Rodzaj zadania
CreateWebPreviewRecordingActionPerformer	Tworzy pliki nagrania na potrzeby odtwarzania w portalach informacyjnych
GetTrialsToUpdateActionPerformer	Zwraca listę sygnatur spraw, które należy ponownie zaimportować do PI (po modyfikacji)
RegisterAgentManagerActionPerformer	Rejestruje import managera w RCS
GetTrialsActionPerformer	Pobiera sygnatury spraw

Parametry dla poszczególnych AP:

CreateWebPreviewRecordingActionPerformer	
Kod	TRIAL_CreateWebPreviewRecording
Wejściowy:	<pre>{ CaseSignature = string.Empty, DiscriminantCode = string.Empty, DestinationResource = {Path = string.Empty, Domain = string.Empty, User = string.Empty, Password = string.Empty }, ExportVideo = bool }</pre>
Wyjściowy	bool

GetTrialsToUpdateActionPerformer



Kod	GetTrialsToUpdate
Wejściowy:	{ DiscriminantCode = string.Empty }
Wyjściowy	List<string>

RegisterAgentManagerActionPerformer	
Kod	RegisterAgentManager
Wejściowy:	{ Id = int, Description = string.Empty, Location = string.Empty }
Wyjściowy	-

GetTrialsActionPerformer	
Kod	TRIAL_GetTrials
Wejściowy:	<pre>{ PageNum = (int?)null, RecordsPerPage = (int?)null, DiscriminantCode = string.Empty, CaseSignature = string.Empty, DateFrom = (DateTime?)null, DateTo = (DateTime?)null }</pre>
Wyjściowy	List<string>

481.1 Mechanizm tworzenia pliku webm dla Portali Informacyjnych

Za pomocą metody `CreateWebPreviewRecordingActionPerformer` w interfejsie zewnętrznym opisanym w tym rozdziale, można zlecić przygotowanie zestawu plików dla sprawy, które będą importowane do systemów PI.

Zlecenie musi wskazać sygnaturę oraz kod wydziału nagrania, z którego mają być przygotowane dane do eksportu. Zlecenie to, jest zapisywane w systemach ReCourt, a następnie przetwarzane przez jednostkę CJR i działający na nim MediaServer w zdefiniowanym oknie czasowym (w którym nie są prowadzone rejestracje rozpraw). Wynikiem wywołania tej funkcji jest informacja o przyjęciu zlecenia lub nie.

Zadanie zostanie wykonane przez system ReCourt, a następnie wynik zostanie umieszczony we wskazanym w parametrze zasobie. W wyniku operacji zostaną utworzone pliki nagrania:

- *_info.xml (z danymi sprawy)
- *_clerk.xml (z adnotacjami)
- *.webm (z nagraniem audio lub audio/wideo wg zadanego parametru)
- *.xml (z opisem operacji)

Plik *.xml opisu operacji będzie miał następującą strukturę:

```
<Export>
<Result>true/false</Result> <!-- informacja o powodzeniu -->
<ErrorCode>[kod błędu]</ErrorCode> <!-- informacja o błędzie jeśli niepowodzenie -->
<Message>[komunikat przetwarzania]</Message> <!-- przyczyna błędu jeśli niepowodzenie -
->
</Export>
```

Plik opisu operacji jest plikiem informacyjnym przetwarzania zleconego zadania I nie jest importowany do PI.

Pliki eksportu po przetworzeniu w systemach PI powinny zostać usunięte przez mechanizm zasilenia PI.

Przebieg działania:

1. Zlecenie konwersji w interfejsie zewnętrznym dla PI
2. Przyjęcie zlecenia przez moduł RCS systemu ReCourt i zapisanie go do kolejki
3. Moduł ReCourt na CJR w zdefiniowanym oknie czasowym pobiera zadania do wykonania
4. Moduł ReCourt na CJR po wykonaniu zadania powiadamia moduł RCS o wykonanej konwersji
5. Moduł RCS systemu ReCourt przygotowuje paczkę z plikami nagrania we wskazanym w żądaniu miejscu (zasobie)
6. Moduł RCS systemu ReCourt tworzy plik opisu operacji informujący o zakończeniu przetwarzania żądania

Przyczyny błędów (kody) przetwarzania:

1. Brak wskazanej sprawy (E01)
2. Wskazana sprawa nie została oznaczona do publikacji (E02)

Błędy przetwarzania nagrania (nagranie uszkodzone lub inny problem z plikiem audio/wideo) (E03)



1 Ekstrakcja ścieżek audio w RCS (na potrzeby transkrypcji)

Opisana w dokumentacji interfejsów i serwisów wewnętrznych – DPNAiR ReCourt cz. III – Załącznik nr 1- Interfejsy wewnętrzne ReCourt , opis narzędzia AVTool.

V. Interfejsy CRCS dla systemów lub programów zewnętrznych

- 1 Interfejsy zewnętrzne CRCS
- 2 Przestrzeń nazw ReCourt.Services.Central.Communication.CrcsSettings.Base

2.1 Interfejsy

	Interfejs	Opis
→	ICrcsSettingsCommunicationService	Interfejs odpowiada za obsługę zadań pobrania wydziałów sądów, sądów oraz komunikację.



3 Interfejs ICrcsSettingsCommunicationService

Interfejs odpowiada za obsługę zadań pobrania wydziałów sądów, sądów oraz komunikację.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.CrcsSettings.Base](#)

Moduł: ReCourt.Services.Central.Communication.CrcsSettings.Base (w

ReCourt.Services.Central.Communication.CrcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

3.1 Składnia

C#

```
public interface ICrcsSettingsCommunicationService
```

ICrcsSettingsCommunicationService prezentuje składowe.

3.2 Metody

	Nazwa	Opis
	NeededSetting	Sprawdzenie ustawień

3.3 Zobacz również


[Przestrzeń nazw ReCourt.Services.Central.Communication.CrcsSettings.Base](#)

4 Metody

ICrcsSettingsCommunicationService.ICrcsSettingsCommunicationService

[ICrcsSettingsCommunicationService](#) prezentuje składowe.

4.1 Metody

	Nazwa	Opis
	NeededSetting	Sprawdzenie ustawień

4.2 Zobacz również

[Interfejs ICrcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.CrcsSettings.Base](#)



5 Metoda ICrcsSettingsCommunicationService.NeededSetting

Sprawdzenie ustawień

Przestrzeń nazw: [ReCourt.Services.Central.Communication.CrcsSettings.Base](#)
Moduł: ReCourt.Services.Central.Communication.CrcsSettings.Base (w
ReCourt.Services.Central.Communication.CrcsSettings.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

5.1 Składnia

C#

```
void NeededSetting ()
```


5.2 Zobacz również

[Interfejs ICrcsSettingsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.CrcsSettings.Base](#)

6 Przestrzeń nazw
ReCourt.Services.Central.Communication.DictionaryRepository.Base

6.1 Interfejsy

	Interfejs	Opis
	IDictionaryRepositoryCommunicationService	Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelni akt.

7 Interfejs IDictionaryRepositoryCommunicationService

Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelnia akt.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
 Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

7.1 Składnia





















```
C#
public interface IDictionaryRepositoryCommunicationService
```




IDictionaryRepositoryCommunicationService prezentuje składowe.

7.2 Metody

Nazwa	Opis
 AddAppeal	Dodaje nową apelację do słownika
 AddBenchRole	Dodaje rolę ław sędziowskich.
 AddCaseType	Dodaje typ sprawy.
 AddCourtDepartments	Dodaje wydział.
 AddCourts	Dodaje sąd.
 AddDistrict	Dodaje nowy okręg do słownika
 AddParticipantFunction	Dodaje funkcję uczestnika postępowania.
 GetAppeals	Zwraca listę apelacji
 GetAppealsPager	Zwraca stronicowaną listę apelacji
 GetBenchRoles	Zwraca listę ról ław sędziowskich.
 GetBenchRolesPager	Zwraca stronicowaną listę ról ław sędziowskich.
 GetCaseTypes	Zwraca typy spraw.



 GetCaseTypesPager	Zwraca stronicowaną listę typów spraw.
 GetCourtDepartments	Zwraca listę wydziałów.
 GetCourtDepartmentsPager	Zwraca stronicowaną listę wydziałów.
 GetCourts	Zwraca listę sądów.
 GetCourtsPager	Zwraca stronicowaną listę sądów.
 GetCourtsWithAppealDistrict	Zwraca listę sądów - apelacje, okręgi (widok).
 GetDistricts	Zwraca listę okręgów
 GetDistrictsPager	Zwraca stronicowaną listę okręgów
 GetMachineDiscriminants	Zwraca stronicowaną listę wydziałów.
 GetParticipantFunctions	Zwraca listę funkcji uczestnika postępowania;
 GetParticipantFunctionsPager	Zwraca stronicowaną listę funkcji uczestnika postępowania.
 RemoveAppeal	Usuwa wybraną apelację
 RemoveBenchRole	Usuwa określoną rolę ław sędziowskich.
 RemoveCaseType	Usuwa określony typ spraw.
 RemoveDistrict	Usuwa apelację o wybranym id
 RemoveParticipantFunction	Usuwa funkcję uczestnika postępowania.
 UpdateAppeal	Aktualizuje apelację
 UpdateBenchRole	Aktualizuje rolę ław sędziowskich.
 UpdateCaseType	Aktualizuje typ spraw.
 UpdateCourtDepartments	Aktualizuje wydział.

 UpdateCourts	Aktualizuje sąd.
 UpdateDistrict	Aktualizuje okręg
 UpdateParticipantFunction	Aktualizuje funkcję uczestnika postępowania.

7.3 Zobacz również

[Przeźren nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

8 Metody









IDictionaryRepositoryCommunicationService.IDictionaryRepositoryCommunicationService

[IDictionaryRepositoryCommunicationService](#) prezentuje składowe.

8.1 Metody

	Nazwa	Opis
⇒	AddAppeal	Dodaje nową apelację do słownika
⇒	AddBenchRole	Dodaje rolę ław sędziowskich.
⇒	AddCaseType	Dodaje typ sprawy.
⇒	AddCourtDepartments	Dodaje wydział.
⇒	AddCourts	Dodaje sąd.
⇒	AddDistrict	Dodaje nowy okręg do słownika
⇒	AddParticipantFunction	Dodaje funkcję uczestnika postępowania.
⇒	GetAppeals	Zwraca listę apelacji
⇒	GetAppealsPager	Zwraca stronicowaną listę apelacji
⇒	GetBenchRoles	Zwraca listę ról ław sędziowskich.
⇒	GetBenchRolesPager	Zwraca stronicowaną listę ról ław sędziowskich.
⇒	GetCaseTypes	Zwraca typy spraw.
⇒	GetCaseTypesPager	Zwraca stronicowaną listę typów spraw.
⇒	GetCourtDepartments	Zwraca listę wydziałów.
⇒	GetCourtDepartmentsPager	Zwraca stronicowaną listę wydziałów.
⇒	GetCourts	Zwraca listę sądów.
⇒	GetCourtsPager	Zwraca stronicowaną listę sądów.



 GetCourtsWithAppealDistrict	Zwraca liste sądów - apelacje, okręgi (widok).
 GetDistricts	Zwraca listę okręgów
 GetDistrictsPager	Zwraca stronicowaną listę okręgów
 GetMachineDiscriminants	Zwraca stronicowaną listę wydziałów.
 GetParticipantFunctions	Zwraca listę funkcji uczestnika postępowania;
 GetParticipantFunctionsPager	Zwraca stronicowaną listę funkcji uczestnika postępowania.
 RemoveAppeal	Usuwa wybraną apelację
 RemoveBenchRole	Usuwa określoną rolę ław sędziowskich.
 RemoveCaseType	Usuwa określony typ spraw.
 RemoveDistrict	Usuwa apelację o wybranym id
 RemoveParticipantFunction	Usuwa funkcję uczestnika postępowania.
 UpdateAppeal	Aktualizuje apelację
 UpdateBenchRole	Aktualizuje rolę ław sędziowskich.
 UpdateCaseType	Aktualizuje typ spraw.
 UpdateCourtDepartments	Aktualizuje wydział.
 UpdateCourts	Aktualizuje sąd.
 UpdateDistrict	Aktualizuje okręg
 UpdateParticipantFunction	Aktualizuje funkcję uczestnika postępowania.

8.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI



**MINISTERSTWO
SPRAWIEDLIWOŚCI**

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO





9 Metoda IDictionaryRepositoryCommunicationService.AddAppeal

Dodaje nową apelację do słownika

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

9.1 Składnia

```
C#  
  
int AddAppeal (  
    ReCourtDictionaryItem item  
)
```

9.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

9.1.1.2 Zwraca

Typ:

Int32

id nowej apelacji

9.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

10 Metoda IDictionaryRepositoryCommunicationService.AddBenchRole

Dodaje rolę ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

10.1 Składnia

C#	
<pre>int ReCourtDictionaryItem)</pre>	<pre>AddBenchRole(item</pre>

10.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

10.1.1.2 Zwraca

Typ:

Int32

Identyfikator nowej roli ław sędziowskich.

10.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

11 Metoda IDictionaryRepositoryCommunicationService.AddCaseType

Dodaje typ sprawy.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

11.1 Składnia

```
C#  
  
int AddCaseType (  
    ReCourtDictionaryItem item  
)
```

11.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

11.1.1.2 Zwraca

Typ:

Int32

Identyfikator typu sprawy.

11.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



12 Metoda IDictionaryRepositoryCommunicationService.AddCourtDepartments

Dodaje wydział.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
 Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
 ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
 (1.0.0.0)

12.1 Składnia

C#	
int	AddCourtDepartments (
ReCourtCourtDepartment	item
)	

12.1.1.1 Parametry

item

Typ: **ReCourtCourtDepartment**

Dane słownika

12.1.1.2 Zwraca

Typ:

Int32

Informacja o powodzeniu operacji.

12.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

13 Metoda IDictionaryRepositoryCommunicationService.AddCourts

Dodaje sąd.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

13.1 Składnia

C#	
<pre>int ReCourtCourt)</pre>	<pre>AddCourts (item</pre>

13.1.1.1 Parametry

item

Typ: **ReCourtCourt**

Dane słownika

13.1.1.2 Zwraca

Typ:

Int32

Informacja o powodzeniu operacji.

13.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



14 Metoda IDictionaryRepositoryCommunicationService.AddDistrict

Dodaje nowy okręg do słownika

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

14.1 Składnia

C#	
<pre>int AddDistrict(ReCourtDictionaryItem item)</pre>	

14.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

14.1.1.2 Zwraca

Typ:

Int32

id nowego okręgu

14.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



15 Metoda IDictionaryRepositoryCommunicationService.AddParticipantFunction

Dodaje funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

15.1 Składnia

```
C#  
  
int AddParticipantFunction(  
    ReCourtDictionaryItem item  
)
```

15.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

15.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanej funkcji.

15.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



16 Metoda IDictionaryRepositoryCommunicationService.GetAppeals

Zwraca listę apelacji

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

16.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetAppeals ()
```

16.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

lista apelacji

16.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

17 Metoda IDictionaryRepositoryCommunicationService.GetAppealsPager

Zwraca stronicowaną listę apelacji

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

17.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetAppealsPager (pageNum, recordsPerPage, sortCriteria</pre>

17.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

17.1.1.2 Zwraca

Typ:
stronicowana lista apelacji

PagedList(ReCourtDictionaryItem)

17.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



18 Metoda IDictionaryRepositoryCommunicationService.GetBenchRoles

Zwraca listę ról ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

18.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetBenchRoles ()
```

18.1.1.1 Zwraca

Typ: **List(ReCourtDictionaryItem)**
Lista roli ław sędziowskich.

18.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

19 Metoda IDictionaryRepositoryCommunicationService.GetBenchRolesPager

Zwraca stronicowaną listę ról ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

19.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetBenchRolesPager (pageNum, recordsPerPage, sortCriteria</pre>

19.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

19.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Lista stronicowanych ról ław sędziowskich.

19.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

20 Metoda IDictionaryRepositoryCommunicationService.GetCaseTypes

Zwraca typy spraw.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

20.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetCaseTypes()
```

20.1.1.1 Zwraca

Typ: **List(ReCourtDictionaryItem)**
Lista typów spraw.

20.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

21 Metoda IDictionaryRepositoryCommunicationService.GetCaseTypesPager

Zwraca stronicowaną listę typów spraw.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

21.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetCaseTypesPager (pageNum, recordsPerPage, sortCriteria</pre>

21.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Filtr

21.1.1.2 Zwraca

Typ:

Lista stronicowanych typów spraw.

PagedList(ReCourtDictionaryItem)

21.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



22 Metoda IDictionaryRepositoryCommunicationService.GetCourtDepartments

Zwraca listę wydziałów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

22.1 Składnia

C#

```
List<ReCourtCourtDepartment> GetCourtDepartments ()
```

22.1.1.1 Zwraca

Typ: **List(ReCourtCourtDepartment)**
Lista wydziałów.

22.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

23 Metoda

IDictionaryRepositoryCommunicationService.GetCourtDepartmentsPager

Zwraca stronicowaną listę wydziałów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

23.1 Składnia

C#	
<pre>PagedList<ReCourtCourtDepartment> int int SortCriteria)</pre>	<pre>GetCourtDepartmentsPager (pageNum, recordsPerPage, sortCriteria</pre>

23.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

23.1.1.2 Zwraca

Typ:

Lista sądów

PagedList(ReCourtCourtDepartment)

23.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

24 Metoda IDictionaryRepositoryCommunicationService.GetCourts

Zwraca liste sądów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

24.1 Składnia

C#

```
List<ReCourtCourt> GetCourts ()
```

24.1.1.1 Zwraca

Typ:

List(ReCourtCourt)

Lista sądów

24.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

25 Metoda IDictionaryRepositoryCommunicationService.GetCourtsPager

Zwraca stronicowaną listę sądów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

25.1 Składnia

C#	
<pre>PagedList<ReCourtCourt> int int SortCriteria)</pre>	<pre>GetCourtsPager (pageNum, recordsPerPage, sortCriteria</pre>

25.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

25.1.1.2 Zwraca

Typ:

Lista sądów

PagedList(ReCourtCourt)

25.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

26 Metoda

IDictionaryRepositoryCommunicationService.GetCourtsWithAppealDistrict

Zwraca liste sądów - apelacje, okręgi (widok).

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

26.1 Składnia

C#

```
List<CentralCourtAppealDistrict> GetCourtsWithAppealDistrict()
```

26.1.1.1 Zwraca

Typ:

List(CentralCourtAppealDistrict)

Lista sądów - apelacje, okręgi

26.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



27 Metoda IDictionaryRepositoryCommunicationService.GetDistricts

Zwraca listę okręgów

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

27.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetDistricts()
```

27.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

lista okręgów

27.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

28 Metoda IDictionaryRepositoryCommunicationService.GetDistrictsPager

Zwraca stronicowaną listę okręgów

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

28.1 Składnia

C#	
<pre>PagedList<ReCourtDictionaryItem> int int SortCriteria)</pre>	<pre>GetDistrictsPager(pageNum, recordsPerPage, sortCriteria</pre>

28.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

28.1.1.2 Zwraca

Typ:
stronicowana lista okręgów

PagedList(ReCourtDictionaryItem)

28.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

29 Metoda

IDictionaryRepositoryCommunicationService.GetMachineDiscriminants

Zwraca stronicowaną listę wydziałów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

29.1 Składnia

C#	
List<CentralDiscriminantMachine> Guid)	GetMachineDiscriminants (<i>identifier</i>)

29.1.1.1 Parametry

identifier

Typ: System.Guid

Identyfikator

29.1.1.2 Zwraca

Typ:

Lista sądów

List(CentralDiscriminantMachine)

29.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



30 Metoda IDictionaryRepositoryCommunicationService.GetParticipantFunctions

Zwraca listę funkcji uczestnika postępowania;

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

30.1 Składnia

C#

```
List<ReCourtDictionaryItem> GetParticipantFunctions()
```

30.1.1.1 Zwraca

Typ:

List(ReCourtDictionaryItem)

Lista funkcji uczestnika postępowania.

30.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

31 Metoda

IDictionaryRepositoryCommunicationService.GetParticipantFunctionsPager

Zwraca stronicowaną listę funkcji uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

31.1 Składnia

```
C#  
  
PagedList<ReCourtDictionaryItem> GetParticipantFunctionsPager (  
    int                pageNum,  
    int                recordsPerPage,  
    SortCriteria       sortCriteria  
)
```

31.1.1.1 Parametry

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronę

sortCriteria

Typ: **SortCriteria**

Sortowanie

31.1.1.2 Zwraca

Typ:

PagedList(ReCourtDictionaryItem)

Lista stronicowanych funkcji uczestnika postępowania.

31.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

32 Metoda IDictionaryRepositoryCommunicationService.RemoveAppeal

Usuwa wybraną apelację

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

32.1 Składnia

```
C#  
  
bool RemoveAppeal (  
    int itemId  
)
```

32.1.1.1 Parametry

itemId

Typ: System.Int32

32.1.1.2 Zwraca

Typ:

Boolean

informacja o powodzeniu operacji

32.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

33 Metoda IDictionaryRepositoryCommunicationService.RemoveBenchRole

Usuwa określoną rolę ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

33.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveBenchRole(itemId</pre>

33.1.1.1 Parametry

itemId

Typ: System.Int32

Id pozycji słownika

33.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

33.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



34 Metoda IDictionaryRepositoryCommunicationService.RemoveCaseType

Usuwa określony typ spraw.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

34.1 Składnia

C#	
<pre>bool int)</pre>	<pre>RemoveCaseType (itemId</pre>

34.1.1.1 Parametry

itemId

Typ: System.Int32

Id pozycji słownika

34.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

34.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

35 Metoda IDictionaryRepositoryCommunicationService.RemoveDistrict

Usuwa apelację o wybranym id

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

35.1 Składnia

```
C#  
  
bool RemoveDistrict(  
    int itemId  
)
```

35.1.1.1 Parametry

itemId

Typ: System.Int32

Id pozycji słownika

35.1.1.2 Zwraca

Typ:

Boolean

informacja o powodzeniu operacji

35.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

36 Metoda

IDictionaryRepositoryCommunicationService.RemoveParticipantFunction

Usuwa funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

36.1 Składnia

```
C#  
  
bool RemoveParticipantFunction(  
    int itemId  
)
```

36.1.1.1 Parametry

itemId

Typ: System.Int32

Id pozycji słownika

36.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

36.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

37 Metoda IDictionaryRepositoryCommunicationService.UpdateAppeal

Aktualizuje apelację

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

37.1 Składnia

C#	
<pre>bool ReCourtDictionaryItem)</pre>	<pre>UpdateAppeal (item</pre>

37.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

37.1.1.2 Zwraca

Typ: **Boolean**

37.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



38 Metoda IDictionaryRepositoryCommunicationService.UpdateBenchRole

Aktualizuje rolę ław sędziowskich.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

38.1 Składnia

C#	
bool	UpdateBenchRole (
ReCourtDictionaryItem	item
))

38.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

38.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

38.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

39 Metoda IDictionaryRepositoryCommunicationService.UpdateCaseType

Aktualizuje typ spraw.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

39.1 Składnia

```
C#  
  
bool UpdateCaseType (  
    ReCourtDictionaryItem item  
)
```

39.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

39.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

39.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

40 Metoda

IDictionaryRepositoryCommunicationService.UpdateCourtDepartments

Aktualizuje wydział.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

40.1 Składnia

C#	
bool	UpdateCourtDepartments (
ReCourtCourtDepartment	item
))

40.1.1.1 Parametry

item

Typ: **ReCourtCourtDepartment**

Dane słownika

40.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

40.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

41 Metoda IDictionaryRepositoryCommunicationService.UpdateCourts

Aktualizuje sąd.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

41.1 Składnia

C#	
<pre>bool ReCourtCourt)</pre>	<pre>UpdateCourts (item</pre>

41.1.1.1 Parametry

item

Typ: **ReCourtCourt**

Dane słownika

41.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

41.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



42 Metoda IDictionaryRepositoryCommunicationService.UpdateDistrict

Aktualizuje okręg

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
 Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
 ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
 (1.0.0.0)

42.1 Składnia

C#	
bool	UpdateDistrict(ReCourtDictionaryItem item)

42.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

42.1.1.2 Zwraca

Typ:

Boolean

informacja o powodzeniu operacji

42.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)

43 Metoda

IDictionaryRepositoryCommunicationService.UpdateParticipantFunction

Aktualizuje funkcję uczestnika postępowania.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.DictionaryRepository.Base (w
ReCourt.Services.Central.Communication.DictionaryRepository.Base.dll) Wersja: 1.0.0.0
(1.0.0.0)

43.1 Składnia

C#	
<pre>bool ReCourtDictionaryItem)</pre>	<pre>UpdateParticipantFunction(item</pre>

43.1.1.1 Parametry

item

Typ: **ReCourtDictionaryItem**

Dane słownika

43.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

43.2 Zobacz również

[Interfejs IDictionaryRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.DictionaryRepository.Base](#)



44 Przestrzeń nazw ReCourt.Services.Central.Communication.Statistics.Base

44.1 Interfejsy

	Interfejs	Opis
	IStatisticsCommunicationService	Serwis statystyk danych

45 Interfejs IStatisticsCommunicationService

Serwis statystyk danych

Przestrzeń nazw: [ReCourt.Services.Central.Communication.Statistics.Base](#)
Moduł: ReCourt.Services.Central.Communication.Statistics.Base (w
ReCourt.Services.Central.Communication.Statistics.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)



45.1 Składnia

C#

```
public interface IStatisticsCommunicationService
```

IStatisticsCommunicationService prezentuje składowe.

45.2 Metody

	Nazwa	Opis
	GetCaseStatistics	Pobiera statystyki spraw
	GetTransferStatistics	Pobiera statystyki transferów



45.3 Zobacz również

[Przestrzeń nazw ReCourt.Services.Central.Communication.Statistics.Base](#)

46 Metody IStatisticsCommunicationService.IStatisticsCommunicationService

[IStatisticsCommunicationService](#) prezentuje składowe.

46.1 Metody

	Nazwa	Opis
	GetCaseStatistics	Pobiera statystyki spraw
	GetTransferStatistics	Pobiera statystyki transferów

46.2 Zobacz również

[Interfejs IStatisticsCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.Statistics.Base](#)

47 Metoda `IStatisticsCommunicationService.GetCaseStatistics`

Pobiera statystyki spraw

Przestrzeń nazw: [ReCourt.Services.Central.Communication.Statistics.Base](#)
Moduł: `ReCourt.Services.Central.Communication.Statistics.Base` (w
`ReCourt.Services.Central.Communication.Statistics.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

47.1 Składnia

```
C#  
  
List<TrialStatistic> GetCaseStatistics (  
    Nullable<DateTime> dateFrom,  
    Nullable<DateTime> dateTo,  
    Nullable<int> selectedCourtId  
)
```

47.1.1.1 Parametry

dateFrom

Typ: `System.Nullable(DateTime)`

Od

dateTo

Typ: `System.Nullable(DateTime)`

Do

selectedCourtId

Typ: `System.Nullable(Int32)`

Sąd

47.1.1.2 Zwraca

Typ: `List(TrialStatistic)`
Statystyki

47.2 Zobacz również

[Interfejs `IStatisticsCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Central.Communication.Statistics.Base`](#)

48 Metoda `IStatisticsCommunicationService.GetTransferStatistics`

Pobiera statystyki transferów

Przestrzeń nazw: [ReCourt.Services.Central.Communication.Statistics.Base](#)
Moduł: `ReCourt.Services.Central.Communication.Statistics.Base` (w
`ReCourt.Services.Central.Communication.Statistics.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

48.1 Składnia

```
C#  
  
List<TransfersStatistic> GetTransferStatistics (  
    Nullable<DateTime>     dateFrom,  
    Nullable<DateTime>     dateTo,  
    Nullable<int>           selectedCourtId  
)
```

48.1.1.1 Parametry

dateFrom

Typ: `System.Nullable(DateTime)`

Od

dateTo

Typ: `System.Nullable(DateTime)`

Do

selectedCourtId

Typ: `System.Nullable(Int32)`

Sąd

48.1.1.2 Zwraca

Typ: `List(TransfersStatistic)`
Statystyki

48.2 Zobacz również

[Interfejs `IStatisticsCommunicationService`](#)


[Przestrzeń nazw `ReCourt.Services.Central.Communication.Statistics.Base`](#)

49 Przestrzeń

nazw

ReCourt.Services.Central.Communication.TrialRepository.Base

49.1 Interfejsy

	Interfejs	Opis
	ITrialRepositoryCommunicationService	Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelnia akt.

50 Interfejs ITrialRepositoryCommunicationService

Interfejs odpowiada za zarządzanie adnotacjami oraz czytelniami akt. Wchodzi w niego skład usuwanie, ustawianie, pobieranie, aktualizowanie adnotacji i czytelnii akt.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.TrialRepository.Base](#)
 Moduł: ReCourt.Services.Central.Communication.TrialRepository.Base (w
 ReCourt.Services.Central.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)



50.1 Składnia

C#

```
public interface ITrialRepositoryCommunicationService
```

ITrialRepositoryCommunicationService prezentuje składowe.

50.2 Metody

	Nazwa	Opis
	GetTrial	Zwraca rozprawę.
	GetTrials	Zwraca stronicowane rozprawy.

50.3 Zobacz również



[Przestrzeń nazw ReCourt.Services.Central.Communication.TrialRepository.Base](#)

51 Metody

`ITrialRepositoryCommunicationService.ITrialRepositoryCommunicationService`

[ITrialRepositoryCommunicationService](#) prezentuje składowe.

51.1 Metody

	Nazwa	Opis
	GetTrial	Zwraca rozprawę.
	GetTrials	Zwraca stronicowane rozprawy.

51.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.TrialRepository.Base](#)



52 Metoda ITrialRepositoryCommunicationService.GetTrial

Zwraca rozprawę.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.TrialRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.TrialRepository.Base (w
ReCourt.Services.Central.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

52.1 Składnia

C#	
CentralTrial	GetTrial (
int	id
)	

52.1.1.1 Parametry

id

Typ: System.Int32

Id sprawy

52.1.1.2 Zwraca

Typ:

CentralTrial

Obiekt informacji o rozprawie.

52.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.TrialRepository.Base](#)

53 Metoda ITrialRepositoryCommunicationService.GetTrials

Zwraca stronicowane rozprawy.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.TrialRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.TrialRepository.Base (w
ReCourt.Services.Central.Communication.TrialRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

53.1 Składnia

C#	
<pre>PagedList<CentralTrialVO> TrialCriteria int int)</pre>	<pre>GetTrials(criteria, pageNum, recordsPerPage</pre>

53.1.1.1 Parametry

criteria

Typ: **TrialCriteria**

Filtr

pageNum

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

53.1.1.2 Zwraca

Typ:

Stronicowana lista rozpraw.

PagedList(CentralTrialVO)

53.2 Zobacz również

[Interfejs ITrialRepositoryCommunicationService](#)


[Przestrzeń nazw ReCourt.Services.Central.Communication.TrialRepository.Base](#)

54 Przestrzeń

nazw

ReCourt.Services.Central.Communication.UserRepository.Base

54.1 Interfejsy

	Interfejs	Opis
	IUserRepositoryCommunicationService	Interfejs odpowiada za zarządzanie użytkownikami. Umożliwia zmianę uprawnień oraz ról.

55 Interfejs IUserRepositoryCommunicationService

Interfejs odpowiada za zarządzanie użytkownikami. Umożliwia zmianę uprawnień oraz ról.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)














55.1 Składnia

C#

```
public interface IUserRepositoryCommunicationService
```

IUserRepositoryCommunicationService prezentuje składowe.

55.2 Metody

	Nazwa	Opis
	AddRole	Dodaje nową rolę.
	AddUser	Dodanie użytkownika.
	AuthorizeUser	Uwierzytelnianie użytkownika.
	EnableUser	Włączanie użytkownika lub wyłączenie.
	GetAction	Pobiera akcję na podstawie kodu.
	GetActions	Pobiera listę akcji.
	GetActiveDirectoryUserNames	Pobiera listę użytkowników w domenie.
	GetReportRights	Pobiera listę raportów.
	GetRole	Pobiera rolę.
	GetRoles	Pobiera role uwzględniając filtr.
	GetUserACL	Pobiera ACL użytkownika.
	GetUserById	Pobiera użytkownika o podanym identyfikatorze.
	GetUserByName	Pobiera użytkownika o podanej nazwie.

 GetUserNames	Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.
 GetUserPrivilegesVO	Pobiera uprawnienia użytkownika.
 GetUserRoles	Pobiera listę ról użytkownika.
 GetUsers	Pobiera użytkowników umożliwiając filtrację listy wynikowej.
 GetUsersForAction	Pobiera listę użytkowników uprawnionych do akcji.
 IsAuthorizedUser	Sprawdza uwierzytelnienie użytkownika.
 IsUserInDomain	Sprawdzenie czy użytkownik jest w domenie.
 UpdateRole	Aktualizuje dane roli.
 UpdateUser	Aktualizuje dane użytkownika.
 UpdateUserLastVisited	Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

55.3 Zobacz również









[Przeźrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

56 Metody




IUserRepositoryCommunicationService.IUserRepositoryCommunicationService

[IUserRepositoryCommunicationService](#) prezentuje składowe.

56.1 Metody

Nazwa	Opis
 AddRole	Dodaje nową rolę.
 AddUser	Dodanie użytkownika.
 AuthorizeUser	Uwierzytelnianie użytkownika.
 EnableUser	Włączanie użytkownika lub wyłączenie.
 GetAction	Pobiera akcję na podstawie kodu.
 GetActions	Pobiera listę akcji.
 GetActiveDirectoryUserNames	Pobiera listę użytkowników w domenie.
 GetReportRights	Pobiera listę raportów.
 GetRole	Pobiera rolę.
 GetRoles	Pobiera role uwzględniając filtr.
 GetUserACL	Pobiera ACL użytkownika.
 GetUserById	Pobiera użytkownika o podanym identyfikatorze.
 GetUserByName	Pobiera użytkownika o podanej nazwie.
 GetUserNames	Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.
 GetUserPrivilegesVO	Pobiera uprawnienia użytkownika.
 GetUserRoles	Pobiera listę ról użytkownika.



 GetUsers	Pobiera użytkowników umożliwiając filtrację listy wynikowej.
 GetUsersForAction	Pobiera listę użytkowników uprawnionych do akcji.
 IsAuthorizedUser	Sprawdza uwierzytelnienie użytkownika.
 IsUserInDomain	Sprawdzenie czy użytkownik jest w domenie.
 UpdateRole	Aktualizuje dane roli.
 UpdateUser	Aktualizuje dane użytkownika.
 UpdateUserLastVisited	Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

56.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

57 Metoda IUserRepositoryCommunicationService.AddRole

Dodaje nową rolę.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

57.1 Składnia

C#	
<pre>int ReCourtCentralRole)</pre>	<pre>AddRole(role</pre>

57.1.1.1 Parametry

role

Typ: **ReCourtCentralRole**

Rola

57.1.1.2 Zwraca

Typ:

Int32

Identyfikator dodanej roli.

57.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

58 Metoda IUserRepositoryCommunicationService.AddUser

Dodanie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

58.1 Składnia

```
C#  
  
int AddUser (  
    ReCourtUser user,  
    List<ReCourtCentralRole> roles,  
    List<ReCourtCentralUserACL> userAcl  
)
```

58.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

roles

Typ: System.Collections.Generic.List(**ReCourtCentralRole**)

Lista ról

userAcl

Typ: System.Collections.Generic.List(**ReCourtCentralUserACL**)

Uprawnienia

58.1.1.2 Zwraca

Typ:

Identyfikator dodanego użytkownika.

Int32

58.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

59 Metoda IUserRepositoryCommunicationService.AuthorizeUser

Uwierzytelnianie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

59.1 Składnia

C#	
<pre>bool string string)</pre>	<pre>AuthorizeUser(username, userPassword</pre>

59.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

userPassword

Typ: System.String

Hasło

59.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

59.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

60 Metoda IUserRepositoryCommunicationService.EnableUser

Włączanie użytkownika lub wyłączenie.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

60.1 Składnia

```
C#  
  
bool EnableUser (  
    int     userId,  
    bool    enable  
)
```

60.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

enable

Typ: System.Boolean

Czy włączony

60.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

60.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

61 Metoda IUserRepositoryCommunicationService.GetAction

Pobiera akcję na podstawie kodu.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

61.1 Składnia

C#	
ReCourtAction	GetAction (
string	code
)	

61.1.1.1 Parametry

code

Typ: System.String

Kod akcji

61.1.1.2 Zwraca

Typ:

ReCourtAction

Pobrana akcja.

61.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

62 Metoda IUserRepositoryCommunicationService.GetActions

Pobiera listę akcji.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

62.1 Składnia

C#

```
List<ReCourtActionVO> GetActions()
```

62.1.1.1 Zwraca

Typ:

List(ReCourtActionVO)

Lista akcji.

62.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

63 Metoda

IUserRepositoryCommunicationService.GetActiveDirectoryUserNames

Pobiera listę użytkowników w domenie.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

63.1 Składnia

```
C#  
  
PagedList<ReCourtDomainUser> GetActiveDirectoryUserNames (  
    string filter,  
    int pageNumber  
)
```

63.1.1.1 Parametry

filter

Typ: System.String

Filtr

pageNumber

Typ: System.Int32

Strona

63.1.1.2 Zwraca

Typ:

PagedList(ReCourtDomainUser)

Stronicowana lista użytkowników w domenie.

63.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

64 Metoda IUserRepositoryCommunicationService.GetReportRights

Pobiera listę raportów.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

64.1 Składnia

C#	
List<ReCourtReportRightVO> int)	GetReportRights (<i>userId</i>

64.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

64.1.1.2 Zwraca

Typ:

List(ReCourtReportRightVO)

Lista raportów.

64.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

65 Metoda IUserRepositoryCommunicationService.GetRole

Pobiera rolę.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

65.1 Składnia

C#	
ReCourtCentralRole	GetRole (
int	roleId
))

65.1.1.1 Parametry

roleId

Typ: System.Int32

Id roli

65.1.1.2 Zwraca

Typ:

ReCourtCentralRole

Znaleziona rola.

65.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

66 Metoda IUserRepositoryCommunicationService.GetRoles

Pobiera role uwzględniając filtr.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

66.1 Składnia

C#	
List<ReCourtRoleVO> ReCourtRoleCriteria)	GetRoles (<i>filterCriteria</i>)

66.1.1.1 Parametry

filterCriteria

Typ: **ReCourtRoleCriteria**

Filtr

66.1.1.2 Zwraca

Typ:

List(ReCourtRoleVO)

Lista ról.

66.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

67 Metoda IUserRepositoryCommunicationService.GetUserACL

Pobiera ACL użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

67.1 Składnia

```
C#  
  
List<ReCourtCentralUserACL>          GetUserACL (  
    int                                userId  
)
```

67.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

67.1.1.2 Zwraca

Typ:

List(ReCourtCentralUserACL)

Lista ACL użytkownika.

67.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

68 Metoda IUserRepositoryCommunicationService.GetById

Pobiera użytkownika o podanym identyfikatorze.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

68.1 Składnia

C#	
ReCourtUser	GetUserId (
int	<i>userId</i>
))

68.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

68.1.1.2 Zwraca

Typ:

ReCourtUser

Obiekt użytkownika.

68.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

69 Metoda IUserRepositoryCommunicationService.GetUserByName

Pobiera użytkownika o podanej nazwie.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

69.1 Składnia

C#	
ReCourtUser	GetUserByName (
string	<i>userName</i>
))

69.1.1.1 Parametry

userName

Typ: System.String

Nazwa użytkownika

69.1.1.2 Zwraca

Typ:

ReCourtUser

Obiekt użytkownika.

69.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)



70 Metoda IUserRepositoryCommunicationService.GetUserNames

Pobiera słownik nazw użytkowników wraz z ich identyfikatorami.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w

ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

70.1 Składnia

C#

```
Dictionary<int, string> GetUserNames()
```

70.1.1.1 Zwraca

Typ: **Dictionary(Int32, String)**

Słownik nazw użytkowników wraz z ich identyfikatorami.

70.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)



71 Metoda IUserRepositoryCommunicationService.GetUserPrivilegesVO

Pobiera uprawnienia użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

71.1 Składnia

C#	
List<ReCourtUserACLVO> int)	GetUserPrivilegesVO(<i>userId</i>

71.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

71.1.1.2 Zwraca

Typ:

List(ReCourtUserACLVO)

Lista uprawnień użytkownika.

71.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

72 Metoda IUserRepositoryCommunicationService.GetUserRoles

Pobiera listę ról użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

72.1 Składnia

C#	
List<ReCourtCentralRole> int)	GetUserRoles (<i>userId</i>)

72.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

72.1.1.2 Zwraca

Typ:

List(ReCourtCentralRole)

Lista ról.

72.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

73 Metoda IUserRepositoryCommunicationService.GetUsers

Pobiera użytkowników umożliwiając filtrację listy wynikowej.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

73.1 Składnia

```
C#  
  
PagedList<ReCourtUserVO> GetUsers (  
    ReCourtUserCriteria    filterCriteria,  
    int                    page,  
    int                    recordsPerPage,  
    SortCriteria           sortCriteria  
)
```

73.1.1.1 Parametry

filterCriteria

Typ: **ReCourtUserCriteria**

Filtr

page

Typ: System.Int32

Strona

recordsPerPage

Typ: System.Int32

Ilość na stronie

sortCriteria

Typ: **SortCriteria**

Sortowanie

73.1.1.2 Zwraca

Typ:

PagedList(ReCourtUserVO)

Stronicowana lista użytkowników.

73.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

74 Metoda IUserRepositoryCommunicationService.GetUsersForAction

Pobiera listę użytkowników uprawnionych do akcji.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

74.1 Składnia

```
C#  
  
List<ReCourtUser>          GetUsersForAction (  
    string                  actionName,  
    string                  discriminantCode  
)
```

74.1.1.1 Parametry

actionName

Typ: System.String

Nazwa akcji

discriminantCode

Typ: System.String

Wyróżnik

74.1.1.2 Zwraca

Typ:

Lista użytkowników.

List(ReCourtUser)

74.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

75 Metoda IUserRepositoryCommunicationService.IsAuthorizedUser

Sprawdza uwierzytelnienie użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

75.1 Składnia

C#	
<pre>bool string)</pre>	<pre>IsAuthorizedUser(username</pre>

75.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

75.1.1.2 Zwraca

Typ:

Boolean

True jeśli użytkownik jest uwierzytelniony, w przeciwnym razie false.

75.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)



76 Metoda IUserRepositoryCommunicationService.IsUserInDomain

Sprawdzenie czy użytkownik jest w domenie.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

76.1 Składnia

```
C#  
  
int IsUserInDomain(  
    string username  
)
```

76.1.1.1 Parametry

username

Typ: System.String

Nazwa użytkownika

76.1.1.2 Zwraca

Typ:

Int32

True jeśli użytkownik w domenie, w przeciwnym razie false.

76.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

77 Metoda IUserRepositoryCommunicationService.UpdateRole

Aktualizuje dane roli.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

77.1 Składnia

C#	
<pre>bool ReCourtCentralRole)</pre>	<pre>UpdateRole(role</pre>

77.1.1.1 Parametry

role

Typ: **ReCourtCentralRole**

Rola

77.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.

77.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

78 Metoda IUserRepositoryCommunicationService.UpdateUser

Aktualizuje dane użytkownika.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)
Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

78.1 Składnia

```
C#  
  
bool UpdateUser (  
    ReCourtUser user,  
    List<ReCourtCentralRole> roles,  
    List<ReCourtCentralUserACL> userAcl  
)
```

78.1.1.1 Parametry

user

Typ: **ReCourtUser**

Użytkownik

roles

Typ: System.Collections.Generic.List(**ReCourtCentralRole**)

Lista ról

userAcl

Typ: System.Collections.Generic.List(**ReCourtCentralUserACL**)

Uprawnienia

78.1.1.2 Zwraca

Typ:

Informacja o powodzeniu operacji.

Boolean

78.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)



79 Metoda IUserRepositoryCommunicationService.UpdateUserLastVisited

Oznacza określony typ danych jako odwiedzony przez użytkownika w bieżącej chwili.

Przestrzeń nazw: [ReCourt.Services.Central.Communication.UserRepository.Base](#)

Moduł: ReCourt.Services.Central.Communication.UserRepository.Base (w
ReCourt.Services.Central.Communication.UserRepository.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

79.1 Składnia

C#	
bool	UpdateUserLastVisited(
int	<i>userId</i> ,
DataTypeEnum	<i>dataType</i>
))

79.1.1.1 Parametry

userId

Typ: System.Int32

Id użytkownika

dataType

Typ: **DataTypeEnum**

Typ sprawdzenia

79.1.1.2 Zwraca

Typ:

Boolean

Informacja o powodzeniu operacji.


79.2 Zobacz również

[Interfejs IUserRepositoryCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Central.Communication.UserRepository.Base](#)

80 Przestrzeń nazw ReCourt.Services.Communication.Registration.Base

80.1 Interfejsy

	Interfejs	Opis
	RegistrationCommunicationService	Serwis rejestracji

81 Interfejs IRegistrationCommunicationService

Serwis rejestracji

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
 Moduł: ReCourt.Services.Communication.Registration.Base (w
 ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

81.1 Składnia

```
C#
public interface IRegistrationCommunicationService
```

IRegistrationCommunicationService prezentuje składowe.

81.2 Metody

	Nazwa	Opis
	GetMachineDiscriminantStatus	Pobieranie zestawienia wyóz
	GetMachines	Zwraca listę maszyn
	GetRcsGuid	Zwraca guid RCSa skojarzonego z agentem
	PingRcsDictionaryCorrectness	Przesyła informację o poprawności słowników
	SetMachineDictionaryCorrectness	Ustawia poprawność wyróżników RCS
	SetMachineRegistrationStatus	Ustawia status rejestracji
	ValidateDiscriminatDictionary	Sprawdzenie poprawności wyróżników

81.3 Zobacz również






[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

82 Metody

IRegistrationCommunicationService.IRegistrationCommunicationService

[IRegistrationCommunicationService](#) prezentuje składowe.

82.1 Metody

	Nazwa	Opis
	GetMachineDiscriminantStatus	Pobieranie zestawienia wyóz
	GetMachines	Zwraca listę maszyn
	GetRcsGuid	Zwraca guid RCSa skojarzonego z agentem
	PingRcsDictionaryCorrectness	Przesyła informację o poprawności słowników
	SetMachineDictionaryCorrectness	Ustawia poprawność wyróżników RCS
	SetMachineRegistrationStatus	Ustawia status rejestracji
	ValidateDiscriminatDictionaty	Sprawdzenie poprawności wyróżników

82.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

83 Metoda `IRegistrationCommunicationService.GetMachineDiscriminantStatus`

Pobieranie zestawienia wyóz

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: `ReCourt.Services.Communication.Registration.Base` (w
`ReCourt.Services.Communication.Registration.Base.dll`) Wersja: 1.0.0.0 (1.0.0.0)

83.1 Składnia

```
C#  
  
List<Tuple<Machine, DiscriminantStatus>>  
GetMachineDiscriminantStatus (  
    IEnumerable<Machine> source  
)
```

83.1.1.1 Parametry

source

Typ: `System.Collections.Generic.IEnumerable(Machine)`

Maszyny do sprawdzenia

83.1.1.2 Zwraca

Typ: `List(Tuple(Machine, DiscriminantStatus))`
Lista maszyn wraz z ich wyróżnikami

83.2 Zobacz również

[Interfejs `IRegistrationCommunicationService`](#)

[Przestrzeń nazw `ReCourt.Services.Communication.Registration.Base`](#)



84 Metoda IRegistrationCommunicationService.GetMachines

Zwraca listę maszyn

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: ReCourt.Services.Communication.Registration.Base (w
ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

84.1 Składnia

```
C#  
  
List<Machine> GetMachines (  
    IEnumerable<int> ids  
)
```

84.1.1.1 Parametry

ids

Typ: System.Collections.Generic.IEnumerable(**Int32**)

Id maszyn

84.1.1.2 Zwraca

Typ: **List(Machine)**
Lista maszyn

84.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)



85 Metoda IRegistrationCommunicationService.GetRcsGuid

Zwraca guid RCSa skojarzonego z agentem

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: ReCourt.Services.Communication.Registration.Base (w
ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

85.1 Składnia

```
C#  
  
string GetRcsGuid(  
    string agentGuid  
)
```

85.1.1.1 Parametry

agentGuid

Typ: System.String

Guid agenta

85.1.1.2 Zwraca

Typ:

String

Guid RCSa

85.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

86 Metoda IRegistrationCommunicationService.PingRcsDictionaryCorrectness

Przesyła informację o poprawności słowników

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: ReCourt.Services.Communication.Registration.Base (w
ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

86.1 Składnia

```
C#  
  
OperationResult PingRcsDictionaryCorrectness (  
    string ipAddress,  
    List<DictionaryCorrectness> status  
)
```

86.1.1.1 Parametry

ipAddress

Typ: System.String

Adres

status

Typ: System.Collections.Generic.List(**DictionaryCorrectness**)

Statusy

86.1.1.2 Zwraca

Typ:

Wynik

OperationResult

86.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

87 Metoda

IRegistrationCommunicationService.SetMachineDictionaryCorrectness

Ustawia poprawność wyróżników RCS

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: ReCourt.Services.Communication.Registration.Base (w
ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

87.1 Składnia

```
C#  
  
OperationResult SetMachineDictionaryCorrectness(  
    Guid identifier,  
    List<DictionaryCorrectness> status,  
    int userId  
)
```

87.1.1.1 Parametry

identifier

Typ: System.Guid

Identyfikator

status

Typ: System.Collections.Generic.List(**DictionaryCorrectness**)

Status

userId

Typ: System.Int32

Id użytkownika

87.1.1.2 Zwraca

Typ:

Wynik

OperationResult

87.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

88 Metoda IRegistrationCommunicationService.SetMachineRegistrationStatus

Ustawia status rejestracji

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
Moduł: ReCourt.Services.Communication.Registration.Base (w
ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

88.1 Składnia

```
C#  
  
OperationResult SetMachineRegistrationStatus (  
    Guid          identifier,  
    RegistrationStatus status  
)
```

88.1.1.1 Parametry

identifier

Typ: System.Guid

Identyfikator

status

Typ: **RegistrationStatus**

Status

88.1.1.2 Zwraca

Typ:
Wynik

OperationResult

88.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

89 Metoda IRegistrationCommunicationService.ValidateDiscriminatDictionaty

Sprawdzenie poprawności wyróżników

Przestrzeń nazw: [ReCourt.Services.Communication.Registration.Base](#)
 Moduł: ReCourt.Services.Communication.Registration.Base (w
 ReCourt.Services.Communication.Registration.Base.dll) Wersja: 1.0.0.0 (1.0.0.0)

89.1 Składnia

```
C#
IList<Tuple<DictionaryCorrectness, bool>>
ValidateDiscriminatDictionaty(
    Guid identifier,
    List<CentralDiscriminantMachine> correct
)
```

89.1.1.1 Parametry

identifier

Typ: System.Guid

Identyfikator

correct

Typ: System.Collections.Generic.List(**CentralDiscriminantMachine**)

Poprawność

89.1.1.2 Zwraca

Typ: **IList(Tuple(DictionaryCorrectness, Boolean))**

Wynik poprawności

89.2 Zobacz również

[Interfejs IRegistrationCommunicationService](#)

[Przestrzeń nazw ReCourt.Services.Communication.Registration.Base](#)

90 Interfejs CRCS dla agenta CAPE

Interfejsy dla systemów zewnętrznych realizowane są interfejsami JSON:

IJsonCommunicationService, IJsonFileTransferCommunicationService

Dla każdego z poszczególnych interfejsów zostały wystawione odpowiednie action performery.

W skład interfejsu komunikacji wchodzi Action Performery:



Nazwa	Rodzaj zadania
IcmpPingRcsActionPerformer	Sprawdzenie połączenia
PingRcsAsActionPerformer	Sprawdzenie połączenia
PullUpdateDownloadActionPerformer	Żądanie aktualizacji
SendLogsPackageActionPerformer	Wysłanie logów działania
SetUpdateStatusActionPerformer	Wysłanie aktualizacji stanu
SrbConfigurationFileActionPerformer	Wysłanie konfiguracji systemu zewnętrznego PI
SendMailActionPerformer	Wysłanie wiadomości mail
SetCapelImporterDetailedMessageActionPerformer	Zapisanie logów przetwarzania
SetCapelImporterMessageActionPerformer	Zapisanie logów przetwarzania
SetCapelImporterPowerStateActionPerformer	Zmiana stanu importera

Parametry dla poszczególnych AP:

IcmpPingRcsActionPerformer	
Kod	IcmpPingRcs
Wejściowy:	<pre>{ AgentInstanceGuid = default(string), PingSuccess = default(bool), }</pre>
Wyjściowy	-

PingRcsAsActionPerformer	
Kod	PingRcsAs
Wejściowy:	<pre>{ AgentInstanceGuid = default(string), AgentServerIpAddress = string.Empty,</pre>



	<pre>RcsGuid = default(string), IncomingDataPort = (int?)null, IncomingTransferPort = (int?)null }</pre>
Wyjściowy	Dictionary<string, string>

PullUpdateDownloadActionPerformer

Kod	PullUpdateDownload
Wejściowy:	Dictionary<string, object>
Wyjściowy	Dictionary<string, object>

SendLogsPackageActionPerformer

Kod	SendLogsPackage
Wejściowy:	Dictionary<string, object>
Wyjściowy	-

SetUpdateStatusActionPerformer

Kod	SetUpdateStatus
Wejściowy:	<pre>{ AgentInstanceGuid = string.Empty, UpdateStatus = UpdateStatus.None, UpdateInformation = string.Empty }</pre>
Wyjściowy	-

SrbConfigurationFileActionPerformer



Kod	SrbConfigurationFile
Wejściowy:	Dictionary<string, object>
Wyjściowy	-

SendMailActionPerformer

Kod	SendMail
Wejściowy:	{ AgentInstanceId = default(string), Subject = string.Empty, Message = string.Empty }
Wyjściowy	-

SetCapelImporterDetailedMessageActionPerformer

Kod	SetCapelImporterDetailedMessage
Wejściowy:	CapelImporterDetailedLog
Wyjściowy	-

SetCapelImporterMessageActionPerformer

Kod	SetCapelImporterMessage
Wejściowy:	CapelImporterLog
Wyjściowy	-

SetCapelImporterPowerStateActionPerformer

Kod	SetCapelImporterPowerState
------------	----------------------------



Wejściowy:	<pre>{ PowerOn = false, AgentInstanceIds = new List<string>(), }</pre>
Wyjściowy	-

1.1 Interfejs CRCS dla Managera agentów CAPE

W skład interfejsu komunikacji wchodzi Action Performery:

Nazwa	Rodzaj zadania
CancelCapelImporterExternalExportTaskActionPerformer	Przerwanie zadania
GetCapelImporterConfigurationActionPerformer	Pobranie konfiguracji
GetCapelImporterStateActionPerformer	Pobranie stanu importera
SetCapelImporterConfigurationActionPerformer	Ustawienie konfiguracji
SetCapelImporterExportSelectedCasesTaskActionPerformer	Wysłanie spraw na żądanie
SetCapelImporterExternalExportTaskActionPerformer	Wysłanie spraw na żądanie
ForceRcsAsPingActionPerformer	Wymuszenie aktualizacji stanu
GetAgentDiscriminantCodeForUserActionPerformer	Pobranie wyróżników
GetAgentDiscriminantsActionPerformer	Pobranie wyróżników
GetAgentConfigurationActionPerformer	Pobranie konfiguracji
GetCourtsByLevelActionPerformer	Pobranie GSW
GetRcsAsDetailedLogsActionPerformer	Pobranie logów przetwarzania
GetRcsAsLogsActionPerformer	Pobranie logów przetwarzania
GetRcsAsStatisticsActionPerformer	Pobranie statystyk przetwarzania
GetRcsAsStatusActionPerformer	Pobranie stanu agenta



PullAgentPackageLogsActionPerformer	Wymuszenie pobrania logów
RcsAsRestartActionPerformer	Restart agenta
RcsAsUpdateActionPerformer	Żądanie aktualizacji
SetAgentConfigurationActionPerformer	Ustawienie konfiguracji agenta
WebManagerUserAuthorizationActionPerformer	Autoryzacja użytkownika

Parametry dla poszczególnych AP:

CancelCapelImporterExternalExportTaskActionPerformer	
Kod	CancelCapelImporterExternalExportTask
Wejściowy:	{ AgentInstanceGuid = default(string), TaskId = default(TaskId) }
Wyjściowy	{ TimeOfExecution = (DateTime?)null }

GetCapelImporterConfigurationActionPerformer	
Kod	GetCapelImporterConfiguration
Wejściowy:	{ AgentInstanceGuid = default(string), }
Wyjściowy	-

GetCapelImporterStateActionPerformer	
Kod	GetCapelImporterState



Wejściowy:	{ AgentInstanceGuid = default(string) }
Wyjściowy	-

SetCapelImporterExternalExportTaskActionPerformer

Kod	SetCapelImporterExternalExportTask
Wejściowy:	{ AgentInstanceGuid = default(string), CapelImporterStartupConfiguration = (CapelImporterScheduledExportStartupConfiguration)null }
Wyjściowy	-

SetCapelImporterExportSelectedCasesTaskActionPerformer

Kod	SetCapelImporterExportSelectedCasesTask
Wejściowy:	{ AgentInstanceGuid = default(string), CapelImporterStartupConfiguration = (CapelImporterSelectedCasesExportStartupConfiguration)null }
Wyjściowy	-

SetCapelImporterConfigurationActionPerformer

Kod	CancelCapelImporterExternalExportTask
Wejściowy:	{ FullConfiguration = false, AgentInstanceIds = new List<string>(),



	<pre>CapelImporterConfiguration = (CapelImporterConfiguration)null }</pre>
Wyjściowy	-



ForceRcsAsPingActionPerformer

Kod	ForceRcsAsPing
Wejściowy:	{ AgentInstanceGuid = default(string), }
Wyjściowy	-

GetAgentDiscriminantCodeForUserActionPerformer

Kod	GetAgentDiscriminantCodeForUser
Wejściowy:	{ AgentInstanceId = default(int), UserId = default(int) }
Wyjściowy	ReCourtDiscriminant

GetAgentDiscriminantsActionPerformer

Kod	GetAgentDiscriminant
Wejściowy:	{ AgentInstanceId = default(int), }
Wyjściowy	List<ReCourtDiscriminant>

GetAgentConfigurationActionPerformer

Kod	GetAgentConfiguration
Wejściowy:	{ AgentInstanceGuid = default(string), }
Wyjściowy	string



GetCourtsByLevelActionPerformer

Kod	GetCourtsByLevel
Wejściowy:	-
Wyjściowy	List<ReCourtCourt>

GetRcsAsDetailedLogsActionPerformer

Kod	GetRcsAsDetailedLogs
Wejściowy:	<pre>{ Id = default(string), PageNumber = default(int), PageSize = default(int), DateFrom = default(DateTime?), DateTo = default(DateTime?), }</pre>
Wyjściowy	PagedList<CapelImporterDetailedLog>

GetRcsAsLogsActionPerformer

Kod	GetRcsAsLogs
Wejściowy:	<pre>{ Id = default(string), PageNumber = default(int), PageSize = default(int), TrialId = default(string), DateFrom = default(DateTime?), DateTo = default(DateTime?), }</pre>
Wyjściowy	PagedList<CapelImporterLog>



GetRcsAsStatisticsActionPerformer

Kod	GetRcsStatistics
Wejściowy:	{ DateFrom = default(DateTime?), DateTo = default(DateTime?), CourtId = default(int?), }
Wyjściowy	List<TrialStatistic>

GetRcsAsStatusActionPerformer

Kod	GetRcsAsStatus
Wejściowy:	-
Wyjściowy	PagedList<AgentInstance>

PullAgentPackageLogsActionPerformer

Kod	PullAgentPackageLogs
Wejściowy:	Dictionary<string, object>
Wyjściowy	FileData

RcsAsRestartActionPerformer

Kod	RcsAsRestart
Wejściowy:	{ AgentInstanceGuid = default(string), PullMode = default(bool) }
Wyjściowy	-

RcsAsUpdateActionPerformer	
Kod	RcsAsUpdate
Wejściowy:	Dictionary<string, object>
Wyjściowy	-

SetAgentConfigurationActionPerformer	
Kod	SetConfiguration
Wejściowy:	<pre>{ AgentInstanceGuid = default(string), Configuration = (XElement)null }</pre>
Wyjściowy	-

WebManagerUserAuthorizationActionPerformer	
Kod	WebManagerUserAuthorization
Wejściowy:	<pre>{ UserName = string.Empty, UserPassword = string.Empty, IsCas = false }</pre>
Wyjściowy	<pre>{ AuthenticationResult = default(bool), User = default(ReCourtuser), Priviligies = default(List<ReCourtUserACLVO>) }</pre>