



System i

Pliki i systemy plików

Zintegrowany system plików

*Wersja 6 wydanie 1*







System i

Pliki i systemy plików

Zintegrowany system plików

*Wersja 6 wydanie 1*

**Uwaga**

Przed skorzystaniem z tych informacji oraz z produktu, którego dotyczą, należy przeczytać informacje zawarte w sekcji “Uwagi”, na stronie 139.

To wydanie dotyczy systemu operacyjnego IBM i5/OS (numer produktu 5761–SS1) wersja 6 wydanie 1, modyfikacja 0, a także wszystkich kolejnych wydań i modyfikacji, chyba że w nowych wydaniach zostanie określone inaczej. Wersja ta nie działa na wszystkich modelach komputerów z procesorem RISC ani na modelach z procesorem CISC.

© Copyright International Business Machines Corporation 1999, 2008. Wszelkie prawa zastrzeżone.

# Spis treści

## Zintegrowany system plików . . . . . 1

Co nowego w wersji V6R1 . . . . .	1
Plik PDF z informacjami na temat zintegrowanego systemu plików . . . . .	2
Przegląd zintegrowanego systemu plików . . . . .	2
Czym jest zintegrowany system plików . . . . .	2
Zalety korzystania ze zintegrowanego systemu plików . . . . .	3
Pojęcia dotyczące zintegrowanego systemu plików . . . . .	4
Katalog . . . . .	4
Katalog bieżący . . . . .	6
Katalog osobisty . . . . .	7
Dostarczane katalogi . . . . .	7
Katalogi *TYPE2 . . . . .	10
Dowiązanie . . . . .	11
Dowiązanie stałe . . . . .	12
Dowiązanie symboliczne . . . . .	13
Nazwa ścieżki . . . . .	14
Plik strumieniowy . . . . .	15
Niezmienność nazw . . . . .	16
Atrybuty rozszerzone . . . . .	17
Obsługa skanowania . . . . .	18
Przykłady: Skanowanie w poszukiwaniu wirusów i skanowanie otwieranych plików . . . . .	18
Pokrewne wartości systemowe . . . . .	19
Operacje skanowania . . . . .	20
Zmiana obiektu . . . . .	21
Zmiana sygnatury . . . . .	21
Inny CCSID . . . . .	22
Podczas operacji składowania . . . . .	22
Sprawdzanie integralności obiektu . . . . .	22
Systemy plików . . . . .	23
Porównanie systemów plików . . . . .	24
System plików "root" (/) . . . . .	28
Rozróżnianie wielkości znaków w systemie plików "root" (/) . . . . .	28
Nazwy ścieżek w systemie plików "root" (/) . . . . .	28
Dowiązania w systemie plików "root" (/) . . . . .	29
Używanie komend zintegrowanego systemu plików w systemie plików "root" (/) . . . . .	29
Używanie funkcji API zintegrowanego systemu plików w głównym systemie plików (/) . . . . .	29
Kronikowanie zmian obiektów w systemie plików "root" (/) . . . . .	30
Urządzenia UDP i TCP w systemie plików "root" (/) . . . . .	30
System plików systemów otwartych (QOpenSys) . . . . .	30
Rozróżnianie wielkości znaków w systemie plików QOpenSys . . . . .	31
Nazwy ścieżek w systemie plików QOpenSys . . . . .	31
Dowiązania w systemie plików QOpenSys . . . . .	31
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QOpenSys . . . . .	31
Używanie funkcji API zintegrowanego systemu plików w systemie plików QOpenSys . . . . .	32
Kronikowanie zmian obiektów w systemie plików QOpenSys . . . . .	32
Systemy plików użytkownika (UDFS) . . . . .	32

Rozróżnianie wielkości liter w systemie plików użytkownika w ramach zintegrowanego systemu plików . . . . .	34
Nazwy ścieżek w systemie plików użytkownika w ramach zintegrowanego systemu plików . . . . .	34
Dowiązania w systemie plików użytkownika zintegrowanego systemu plików . . . . .	35
Korzystanie z komend zintegrowanego systemu plików w systemie plików użytkownika . . . . .	35
Korzystanie z funkcji API zintegrowanego systemu plików w systemie plików użytkownika . . . . .	36
Graficzny interfejs dostępu do systemu plików użytkownika . . . . .	36
Tworzenie systemu plików użytkownika (UDFS) zintegrowanego systemu plików . . . . .	36
Usuwanie zintegrowanego systemu plików użytkownika . . . . .	37
Wyświetlanie zintegrowanego systemu plików użytkownika . . . . .	37
Podłączanie systemu plików użytkownika (UDFS) zintegrowanego systemu plików . . . . .	37
Odcinanie systemu plików użytkownika (UDFS) w zintegrowanym systemie plików . . . . .	37
Składowanie i odtwarzanie systemu plików użytkownika w ramach zintegrowanego systemu plików . . . . .	38
Kronikowanie zmian obiektów w systemie plików użytkownika . . . . .	38
System plików użytkownika i niezależne pule pamięci dyskowej . . . . .	38
System plików bibliotek (QSYS.LIB) . . . . .	39
Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB . . . . .	39
Ograniczenia w obsłudze plików w systemie plików QSYS.LIB . . . . .	40
Obsługa przestrzeni użytkowników w systemie plików QSYS.LIB . . . . .	40
Obsługa zbiorów składowania w systemie plików QSYS.LIB . . . . .	40
Rozróżnianie wielkości znaków w systemie plików QSYS.LIB . . . . .	40
Nazwy ścieżek w systemie plików QSYS.LIB . . . . .	40
Dowiązania w systemie plików QSYS.LIB . . . . .	41
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QSYS.LIB . . . . .	41
Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB . . . . .	42
Niezależne pule ASP systemu QSYS.LIB . . . . .	42
Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	42
Ograniczenia w obsłudze plików w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	42
Obsługa przestrzeni użytkowników w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	43
Obsługa zbiorów składowania w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	43

Rozróżnianie wielkości znaków w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	43	Używanie funkcji API zintegrowanego systemu plików w systemie plików QFileSvr.400 . . . . .	58
Nazwy ścieżek w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	43	Sieciowy system plików (Network File System - NFS) . . . . .	58
Dowiązania w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	44	Charakterystyka systemu plików NFS . . . . .	58
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	44	Odmiany serwerów i klientów w sieciowym systemie plików . . . . .	59
Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP . . . . .	45	Dowiązania w systemie plików NFS . . . . .	59
System plików usług biblioteki dokumentów (QDLS) . . . . .	45	Używanie komend zintegrowanego systemu plików w systemie plików NFS . . . . .	60
Zintegrowany system plików i system HFS w systemie plików QDLS . . . . .	45	Używanie funkcji API zintegrowanego systemu plików w systemie plików NFS . . . . .	61
Rejestrowanie użytkowników w systemie plików QDLS . . . . .	46	Dostęp do zintegrowanego systemu plików . . . . .	61
Rozróżnianie wielkości znaków w systemie plików QDLS . . . . .	46	Dostęp przy użyciu pozycji menu i ekranów . . . . .	61
Nazwy ścieżek w systemie plików QDLS . . . . .	46	Dostęp przy użyciu komend języka CL . . . . .	63
Dowiązania w systemie plików QDLS . . . . .	46	Reguły dotyczące nazw ścieżek dla komend CL i ekranów . . . . .	66
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QDLS . . . . .	46	Praca z danymi wyjściowymi komend RTVDIRINF i PRTDIRINF . . . . .	68
Używanie funkcji API zintegrowanego systemu plików w systemie plików QDLS . . . . .	47	Dostęp do danych RTVDIRINF . . . . .	79
System plików nośników optycznych (QOPT) . . . . .	47	Używanie danych zwracanych przez komendę RTVDIRINF . . . . .	80
Zintegrowany system plików i system HFS w systemie plików QOPT . . . . .	48	Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator . . . . .	80
Rozróżnianie wielkości znaków w systemie plików QOPT . . . . .	48	Dostęp przy użyciu funkcji API . . . . .	82
Nazwy ścieżek w systemie plików QOPT . . . . .	48	Dostęp przy użyciu komputera PC . . . . .	82
Dowiązania w systemie plików QOPT . . . . .	49	Uzyskiwanie dostępu za pomocą produktu System i Navigator . . . . .	82
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QOPT . . . . .	49	Uzyskiwanie dostępu za pomocą produktu i5/OS NetServer . . . . .	83
Używanie funkcji API zintegrowanego systemu plików w systemie plików QOPT . . . . .	49	Dostęp przy użyciu protokołu FTP . . . . .	84
System plików i5/OS NetClient (QNTC) . . . . .	50	Konwersja zintegrowanego systemu plików . . . . .	85
Uprawnienia i prawa własności w systemie plików QNTC . . . . .	50	Przekształcanie katalogów z formatu *TYPE1 na *TYPE2 . . . . .	85
Rozróżnianie wielkości znaków w systemie plików QNTC . . . . .	50	Przegląd konwersji formatu *TYPE1 do *TYPE2 . . . . .	85
Nazwy ścieżek w systemie plików QNTC . . . . .	50	Zagadnienia związane z konwersją katalogów . . . . .	86
Dowiązania w systemie plików QNTC . . . . .	51	Określenie statusu konwersji . . . . .	86
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QNTC . . . . .	51	Tworzenie profili użytkowników . . . . .	86
Używanie funkcji API zintegrowanego systemu plików w systemie plików QNTC . . . . .	52	Zmiana nazw obiektów . . . . .	87
Zmienne środowiskowe systemu plików QNTC . . . . .	52	Uwagi na temat profili użytkowników . . . . .	87
Tworzenie katalogów w systemie plików QNTC . . . . .	53	Wymagania pamięci dyskowej . . . . .	88
Aktywowanie usługi uwierzytelniania sieciowego w systemie plików QNTC . . . . .	54	Wskazówki: dowiązanie symboliczne . . . . .	88
System plików serwera plików i5/OS (QFileSvr.400) . . . . .	54	Wskazówki: niezależne ASP . . . . .	89
Rozróżnianie wielkości znaków w systemie plików QFileSvr.400 . . . . .	55	Wskazówki: składowanie i odtwarzanie . . . . .	89
Nazwy ścieżek w systemie plików QFileSvr.400 . . . . .	55	Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików . . . . .	89
Komunikacja w systemie plików QFileSvr.400 . . . . .	55	Skanowanie zintegrowanego systemu plików . . . . .	89
Bezpieczeństwo i uprawnienia do obiektów w systemie plików QFileSvr.400 . . . . .	56	Konwersja nazw w celu obsługi dodatkowych znaków . . . . .	90
Dowiązania w systemie plików QFileSvr.400 . . . . .	57	Przegląd automatycznej konwersji nazw . . . . .	90
Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QFileSvr.400 . . . . .	57	Zagadnienia związane z konwersją nazw . . . . .	91
		Określenie statusu konwersji . . . . .	91
		Zmiana nazw obiektów . . . . .	91
		Uwagi na temat profili użytkowników . . . . .	91
		Wskazówki: dowiązanie symboliczne . . . . .	92
		Wskazówki: niezależne ASP . . . . .	92
		Wskazówki: składowanie i odtwarzanie . . . . .	92
		Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików . . . . .	92
		Kronikowanie obiektów . . . . .	92
		Przegląd kronikowania . . . . .	93
		Zarządzanie kronikami . . . . .	93
		Jakie obiekty należy uwzględnić w kronikowaniu . . . . .	93

Kronikowane obiekty zintegrowanego systemu plików . . . . .	93	Obsługa dużych plików . . . . .	113
Kronikowane operacje . . . . .	95	Reguły dotyczące ścieżek dla funkcji API . . . . .	114
Uwagi specjalne dotyczące pozycji kronik . . . . .	96	Deskryptor pliku . . . . .	115
Zagadnienia dotyczące używania wielu napędów dysków i kronikowania . . . . .	97	Bezpieczeństwo . . . . .	115
Uruchamianie kronikowania . . . . .	97	Obsługa gniazd . . . . .	116
Zmiana kronikowania . . . . .	97	Nazewnictwo i obsługa różnych języków . . . . .	116
Przerywanie kronikowania . . . . .	98	Konwersja danych . . . . .	117
Operacja odzyskiwania systemu plików "root" (/), systemu plików QOpenSys i systemu plików UDFS . . . . .	98	Przykład: funkcje C zintegrowanego systemu plików . . . . .	118
Porównanie komend Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) . . . . .	98	Praca z plikami i folderami w programie System i Navigator . . . . .	123
Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) . . . . .	100	Tworzenie folderu . . . . .	123
Ponowne tworzenie obiektów zintegrowanego systemu plików . . . . .	100	Usuwanie pliku lub folderu . . . . .	123
Przykłady: komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) . . . . .	101	Przenoszenie plików lub folderów do innego systemu plików . . . . .	124
Przykład: usuwanie problemów z obiektem . . . . .	101	Ustawianie uprawnień . . . . .	125
Przykład: usuwanie problemów występujących w strukturze katalogów . . . . .	101	Konfigurowanie konwersji plików tekstowych . . . . .	125
Przykład: usunięcie wszystkich uszkodzonych obiektów w systemach plików "root" (/), QOpenSys i UDFS . . . . .	101	Wysyłanie pliku lub folderu do innego systemu . . . . .	126
Przykład: usunięcie wszystkich uszkodzonych obiektów w systemach plików "root" (/), QOpenSys i UDFS . . . . .	101	Zmiana opcji dla wysyłania pliku lub folderu . . . . .	126
Przykład: uruchomienie wielu instancji komendy RCLLNK w celu szybkiego odzyskania wszystkich obiektów w systemach plików "root" (/), QOpenSys i UDFS . . . . .	102	Tworzenie współużytkowanego zasobu plikowego . . . . .	126
Obsługa programistyczna . . . . .	102	Zmiana współużytkowanego zasobu plikowego . . . . .	127
Kopiowanie danych między plikami strumieniowymi a zbiorami baz danych . . . . .	102	Usuwanie współużytkowanego zasobu plikowego . . . . .	127
Kopiowanie danych za pomocą komend CL . . . . .	102	Tworzenie nowego systemu plików użytkownika . . . . .	127
Kopiowanie danych za pomocą funkcji API . . . . .	104	Podłączanie systemu plików utworzonego przez użytkownika . . . . .	127
Kopiowanie danych za pomocą funkcji przesyłania danych . . . . .	104	Odłączanie systemu plików użytkownika . . . . .	128
Przesyłanie danych ze zbioru bazy danych do pliku strumieniowego . . . . .	104	Praca z dynamicznie podłączanymi systemami plików . . . . .	128
Przesyłanie danych z pliku strumieniowego do zbioru bazy danych . . . . .	104	Określanie, czy obiekty mają być skanowane . . . . .	129
Przesyłanie danych do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru . . . . .	105	Zwracanie obiektów . . . . .	130
Tworzenie pliku opisu formatu . . . . .	105	Pobieranie obiektów . . . . .	130
Kopiowanie danych między plikami strumieniowymi a zbiorami składowania . . . . .	106	Zdalne wywołanie procedury niezależne od transportów . . . . .	131
Wykonywanie operacji za pomocą funkcji API . . . . .	106	Funkcje API wyboru sieci . . . . .	131
Funkcje ILE C . . . . .	112	Funkcje API konwersji nazwy na adres . . . . .	132
		Zewnętrzna reprezentacja danych (XDR) . . . . .	132
		Uwierzytelnianie funkcji API . . . . .	134
		Funkcje API wywołań RPC niezależnych od transportu (TI-RPC) . . . . .	134
		Uprozczone funkcje API wywołań TI-RPC . . . . .	134
		Funkcje API najwyższego poziomu TI-RPC . . . . .	134
		Funkcje API poziomu pośredniego aplikacji TI-RPC . . . . .	135
		Funkcje API poziomu najwyższego aplikacji TI-RPC . . . . .	135
		Inne funkcje API wywołań TI-RPC . . . . .	135
		Informacje pokrewne dotyczące zintegrowanego systemu plików . . . . .	136
		<b>Dodatek. Uwagi . . . . .</b>	<b>139</b>
		Informacje dotyczące interfejsu programistycznego . . . . .	141
		Znaki towarowe . . . . .	141
		Warunki . . . . .	141





---

## Zintegrowany system plików

Zintegrowany system plików jest częścią systemu operacyjnego i5/OS, która umożliwia obsługę strumieniowych danych we/wy i zarządzanie pamięcią podobne do komputerów osobistych i systemów operacyjnych UNIX, zapewniając strukturę integrującą wszystkie informacje przechowywane w systemie.

**Uwaga:** Korzystając z przykładów kodu, użytkownik wyraża zgodę na warunki zapisane w sekcji “Licencja na kod oraz Informacje dotyczące kodu” na stronie 136.

---

### Co nowego w wersji V6R1

Poniżej omówiono nowe lub znacznie zmienione informacje zawarte w kolekcji tematów Zintegrowany system plików.

#### Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator

Jako alternatywę dla komend Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF) i Drukuj informacje katalogowe (Print Directory Information - PRDIRINF), program System i Navigator udostępnia funkcję zbierania i analizowania atrybutów obiektów w zintegrowanym systemie plików.

Szczegółowe informacje można znaleźć w sekcji “Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator” na stronie 80.

#### Zaawansowane funkcje wprowadzania i pobierania obiektów

Poza możliwością wprowadzania i pobierania pojedynczych plików, obecnie możliwe jest również wprowadzanie i pobieranie dowolnych kwalifikujących się do takich operacji obiektów w folderze.

Szczegółowe informacje można znaleźć w sekcjach “Zwracanie obiektów” na stronie 130 i “Pobieranie obiektów” na stronie 130.

#### Automatyczna konwersja w celu obsługi dodatkowych znaków w nazwach

Wkrótce po zainstalowaniu systemu operacyjnego i5/OS w wersji V6R1, system rozpoczyna konwersję katalogów w systemach plików nierozróżniających wielkości liter w celu zapewnienia obsługi nowych znaków Unicode i zasad wyboru wielkości liter w nazwach.

Szczegółowe informacje można znaleźć w sekcji “Konwersja nazw w celu obsługi dodatkowych znaków” na stronie 90.

#### Obsługa protokołu IPv6

Następujące systemy plików, które w wersjach systemu i5/OS poprzedzających V6R1 obsługiwały wyłącznie adresy IPv4, obecnie obsługują zarówno adresy IPv4, jak i IPv6:

- “System plików i5/OS NetClient (QNTC)” na stronie 50
- “System plików serwera plików i5/OS (QFileSvr.400)” na stronie 54
- “Sieciowy system plików (Network File System - NFS)” na stronie 58

#### Obsługa kodowania Unicode w wybranych komendach CL

Wybrane komendy CL obsługują w obecnej wersji kodowanie Unicode. Więcej informacji można znaleźć w tabeli w sekcji “Dostęp przy użyciu komend języka CL” na stronie 63.

## Porzucenie obsługi systemu plików QNetWare

Od wersji V6R1 systemu i5/OS w zintegrowanym systemie plików nie jest już obsługiwany system plików QNetWare ani funkcje z nim związane. Z tego powodu z niniejszej publikacji usunięto wszelkie informacje na temat systemu plików QNetWare.

## Znajdowanie nowych lub zmienionych informacji

Aby ułatwić określenie obszarów, w których zostały wprowadzone zmiany techniczne, w Centrum informacyjnym zastosowano:

- symbol » służący do zaznaczania początku nowego lub zmienionego fragmentu;
- symbol « służący do zaznaczania końca nowego lub zmienionego fragmentu.

Nowe i zmienione informacje w plikach PDF mogą być oznaczone symbolem | na lewym marginesie.

Więcej informacji na temat zmian i nowości w bieżącej wersji zawiera Wiadomość dla użytkowników.

---

## Plik PDF z informacjami na temat zintegrowanego systemu plików

Informacje zawarte w tym temacie są także dostępne w postaci pliku PDF, który można wyświetlić i wydrukować.

Aby wyświetlić lub pobrać niniejszy dokument w formacie PDF, kliknij odsyłacz Zintegrowany system plików (około 1845 KB).

## Zapisywanie plików PDF

Aby zapisać plik PDF na stacji roboczej w celu jego wyświetlenia lub wydrukowania, wykonaj następujące czynności:

1. Kliknij prawym przyciskiem myszy odsyłacz do pliku PDF w przeglądarce.
2. Kliknij opcję zapisania pliku PDF lokalnie.
3. Przejdź do katalogu, w którym ma zostać zapisany plik PDF.
4. Kliknij opcję **Zapisz**.

## Pobieranie programu Adobe Reader

Do przeglądania i drukowania plików PDF potrzebny jest program Adobe Reader. Bezpłatną kopię tego programu można pobrać z serwisu WWW firmy Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

## Przegląd zintegrowanego systemu plików

Sekcja zawiera podstawowe informacje na temat zintegrowanego systemu plików w systemie operacyjnym i5/OS oraz możliwości jego wykorzystania.

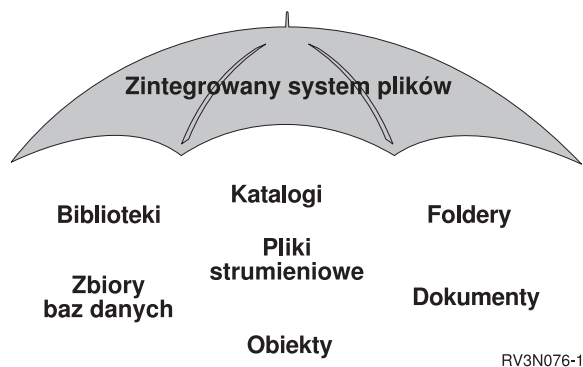
## Czym jest zintegrowany system plików

*Zintegrowany system plików* jest częścią systemu operacyjnego i5/OS, która umożliwia obsługę strumieniowych operacji wejścia/wyjścia i zarządzanie pamięcią podobne do komputerów PC i systemów operacyjnych UNIX, zapewniając strukturę integrującą wszystkie informacje przechowywane w systemie.

Zintegrowany system plików obejmuje 10 systemów plików, z których każdy posiada własny zestaw struktur logicznych i reguł przetwarzania danych składowanych w pamięci.

Oto najważniejsze cechy zintegrowanego systemu plików:

- Obsługa informacji przechowywanych w plikach strumieniowych; mogą to być długie, nieprzerwane ciągi danych. Takim ciągiem może być, na przykład, tekst dokumentu lub elementy obrazu. Obsługa plików strumieniowych służy do lepszego wykorzystania aplikacji klient/serwer.
- Hierarchiczna struktura katalogów, która pozwala organizować obiekty podobnie, jak owoce na gałęziach drzewa. Dostęp do obiektu można uzyskać poprzez podanie ścieżki do niego, zawierającej katalog obiektu.
- Wspólny interfejs, dający użytkownikom i aplikacjom dostęp nie tylko do plików strumieniowych, ale także do zbiorów baz danych, dokumentów i innych obiektów przechowywanych na serwerze.
- Jednolity widok plików strumieniowych składowanych lokalnie w systemie, na serwerze Integrated xSeries Server (IXS) lub na zdalnym serwerze Windows NT. Pliki strumieniowe można też przechowywać zdalnie na serwerze w sieci LAN, na innym zdalnym systemie System i lub na serwerze systemu plików NFS.



Rysunek 1. Struktura obejmująca całość informacji składowanych w systemie operacyjnym i5/OS

#### Pojęcia pokrewne

“Systemy plików” na stronie 23

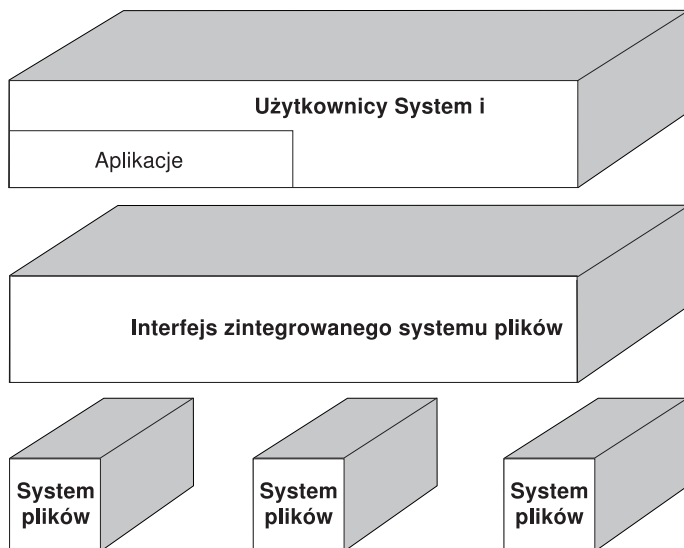
*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

## Zalety korzystania ze zintegrowanego systemu plików

Zintegrowany system plików powiększa, i tak szerokie, możliwości zarządzania danymi przez system i5/OS o lepszą obsługę istniejących i przyszłych sposobów przetwarzania informacji, takich jak klient/serwer, systemy otwarte i multimedia.

Zintegrowany system plików umożliwia:

- szybki dostęp do danych systemu i5/OS, zwłaszcza w przypadku takich aplikacji, jak System i Access, które korzystają z serwera plików systemu i5/OS;
- efektywniejszą obsługę danych strumieniowych, takich jak obrazy, dźwięk i wideo;
- udostępnienie platformy systemu plików i katalogów dla obsługi standardu systemów otwartych bazujących na systemie UNIX, takich jak POSIX i X/Open Portability Guide (XPG). Taka struktura plików i katalogów zapewnia też znane środowisko pracy dla użytkowników systemów operacyjnych PC, takich jak DOS i Microsoft Windows;
- obsługę plików o szczególnych cechach i możliwościach (takich jak oparte na rekordach zbiory baz danych, pliki strumieniowe podobne jak w systemie UNIX oraz usługi serwera plików) jako osobnych systemów plików, przy jednoczesnym zarządzaniu nimi za pomocą wspólnego interfejsu;



Rysunek 2. Wspólny interfejs dla różnych systemów plików.

- lepsze wykorzystanie graficznego interfejsu użytkownika przez użytkowników komputerów osobistych; na przykład użytkownicy systemów Windows mogą za pomocą narzędzi graficznego interfejsu użytkownika Windows wykonywać operacje na plikach strumieniowych i innych obiektach systemu i5/OS w analogiczny sposób, jak na plikach składanych lokalnie na ich komputerach PC;
- zapewnienie spójności nazw obiektów i związanych z nimi informacji niezależnie od języka narodowego; na przykład przy przejściu ze strony kodowej jednego języka na stronę kodową innego języka znaki specyficzne dla danego języka nie zostaną zmienione.

#### Pojęcia pokrewne

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

## Pojęcia dotyczące zintegrowanego systemu plików

Poniższy temat wprowadza podstawowe pojęcia dotyczące zintegrowanego systemu plików, takie jak: katalog, dowiązanie, nazwa ścieżki, plik strumieniowy, ciągłość nazwy, atrybut rozszerzony oraz obsługa skanowania.

### Katalog

*Katalog* jest specjalnym obiektem używanym do wyszukiwania obiektów według nazw podanych przez użytkownika. Każdy katalog zawiera listę dołączonych do niego obiektów. Lista ta może zawierać inne katalogi.

Zintegrowany system plików tworzy hierarchiczną strukturę katalogów umożliwiającą dostęp do wszystkich obiektów w systemie. Struktura ta przypomina odwrócone drzewo, gdzie korzeń jest na szczycie, a gałęzie na dole. W strukturze katalogów katalogi reprezentują gałęzie. Gałęzie katalogów mają podrzędne gałęzie zwane podkatalogami. Do różnych gałęzi katalogów i podkatalogów dołączone są obiekty, takie jak pliki. Znalezienie obiektu wymaga podania ścieżki wiodącej poprzez katalogi do podkatalogu, do którego został dołączony ten obiekt. Obiekty, które są dołączone do danego katalogu, są zwykle określane jako będące w tym katalogu.

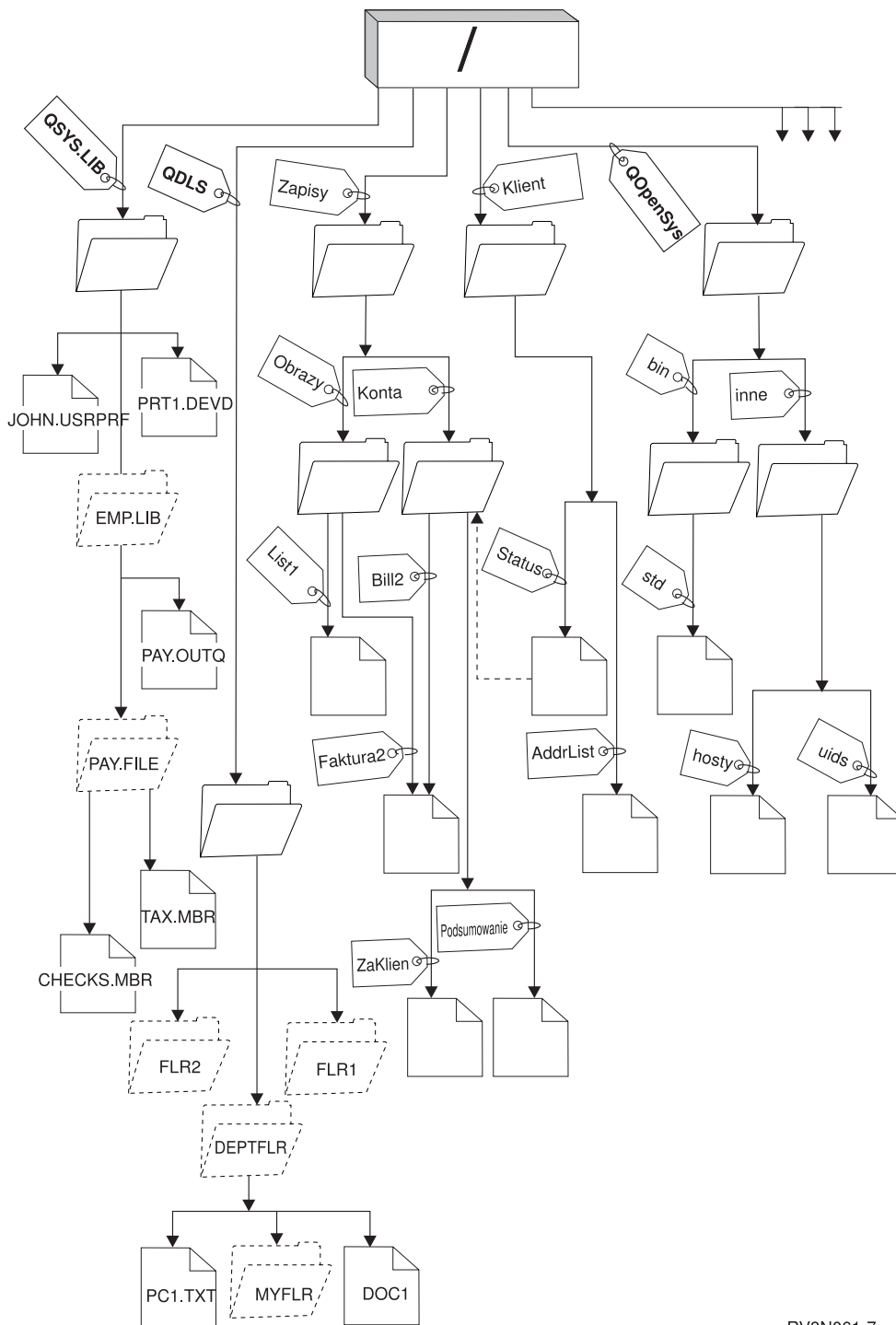
Pojedyncza gałąź katalogu, razem ze wszystkimi swoimi podrzędnymi gałęziami (podkatalogami) i ze wszystkimi obiektami dołączonymi do tych gałęzi, jest nazywana *poddrzewem*. Każdy system plików jest głównym poddrzewem w strukturze katalogów zintegrowanego systemu plików. W bibliotece QSYS.LIB oraz w poddrzewach systemu plików niezależnej puli ASP biblioteka jest obsługiwana w ten sam sposób co podkatalog. Obiekty z biblioteki obsługiwane są podobnie jak obiekty z podkatalogu. Ponieważ zbiory baz danych zawierają obiekty (podzbiory zbiorów baz danych),

obsługuje się je podobnie jak podkatalogi, a nie jak obiekty. W systemie plików obsługującym biblioteki dokumentów (poddrewo) foldery obsługiwane są podobnie jak podkatalogi, a dokumenty w folderach podobnie jak obiekty w podkatalogu.

Ze względu na różnice w systemach plików pewne operacje można wykonywać tylko w niektórych poddrzewach struktury katalogów.

Obsługa katalogów zintegrowanego systemu plików jest podobna do obsługi katalogów w systemie plików DOS. Ponadto dostarcza ona funkcji typowych dla systemów UNIX, takich jak możliwość przechowywania jednego egzemplarza pliku z możliwością dostępu do niego poprzez wiele ścieżek za pomocą dowiązań.

Systemy plików i ich obiekty stanowią gałęzie drzewa katalogów zintegrowanego systemu plików. Przykład drzewa katalogów zintegrowanego systemu plików znaleźć można na następującym rysunku.



RV3N061-7

Rysunek 3. Przykładowe drzewo katalogów zintegrowanego systemu plików

## Katalog bieżący

Katalog bieżący pełni podobną rolę jak biblioteka bieżąca. Nazywa go się także *bieżącym katalogiem roboczym* lub po prostu *katalogiem roboczym*.

*Katalog bieżący* jest pierwszym katalogiem, w którym system operacyjny szuka programów oraz plików i w którym przechowuje tymczasowe pliki i wydruki. Gdy użytkownik żąda wykonania czynności na obiekcie takim jak zbiór, system szuka obiektu w bieżącym katalogu użytkownika, chyba że użytkownik poda inną ścieżkę katalogów.

## Katalog osobisty

*Katalog osobisty* staje się bieżącym katalogiem użytkownika bezpośrednio po jego wpisaniu się do systemu. Nazwa katalogu osobistego podana jest w profilu użytkownika.

Gdy uruchamia się zadanie dla użytkownika, system szuka w profilu użytkownika nazwy katalogu osobistego tego użytkownika. Jeśli katalogu o danej nazwie nie ma w systemie, jako katalog osobisty jest używany katalog główny "root" oznaczany przez ukośnik (/).

Administrator systemu, który tworzy profil dla użytkownika, tworzy także katalog osobisty użytkownika. Zalecane jest tworzenie katalogów osobistych użytkowników w katalogu /home. Katalog ten jest podkatalogiem katalogu "root" (/). Domyślnie system nadaje katalogowi osobistemu użytkownika taką samą nazwę, jaką ma profil użytkownika.

Na przykład komenda `CRTUSRPRF USRPRF(Jasio) HOMEDIR(*USRPRF)` przypisze użytkownikowi o identyfikatorze Jasio katalog osobisty /home/JASIO. Jeśli katalog /home/JASIO nie istnieje, katalogiem osobistym użytkownika stanie się katalog "root" (/).

W dowolnym momencie po wpisaniu się do systemu użytkownicy mogą zmienić swój katalog bieżący z katalogu osobistego, używając komendy języka CL Zmiana bieżącego katalogu (Change Current Directory - CHGCURDIR), funkcji API `chdir()` lub funkcji API `fhdir()`.

Katalog osobisty, wybrany podczas inicjowania procesu, pozostanie domyślnie katalogiem osobistym każdego wątku, niezależnie od tego, czy aktywny profil użytkownika dla danego wątku został zmieniony po zainicjowaniu procesu. Jednak za pomocą funkcji API Zmiana zadania Change Job (QWTCHGJB) można zmienić katalog osobisty, który był używany dla wątku, na katalog osobisty profilu użytkownika bieżącego wątku (lub katalog "root" (/), jeśli taki katalog osobisty nie istnieje). Wątki wtórne zawsze dziedziczą katalog osobisty tego wątku, który je utworzył. Należy zauważyć, że katalog osobisty procesu nie zmienia się, gdy do zmiany katalogu osobistego wątku jest używana funkcja QWTCHGJB QWTCHGJB. Katalog bieżący jest ograniczony do poziomu procesu, a katalog osobisty do poziomu wątku. Zmiana bieżącego katalogu roboczego w dowolnym wątku powoduje zmianę tego katalogu dla całego wątku. Zmiana katalogu osobistego dla wątku nie zmienia jego bieżącego katalogu roboczego.

### Informacje pokrewne

Komenda Zmiana bieżącego katalogu (Change current directory - CHGCURDIR)

`chdir()`--funkcja API Change Current Directory

`fhdir()`--funkcja API Change Current Directory by Descriptor

Aplikacyjne interfejsy programistyczne (API)

## Dostarczane katalogi

- | Po restarcie systemu zintegrowany system plików tworzy niżej wymienione katalogi, jeśli jeszcze nie istnieją.
- | utworzeniu katalogów przez system nie należy ich przenosić ani zmieniać ich nazw.

**Uwaga:** Poniższych katalogów utworzonych przez system nie należy zastępować dowiązaniem symbolicznymi do innych obiektów. Na przykład nie należy zastępować katalogu /home dowiązaniem symbolicznym do katalogu lub niezależnej puli ASP. W przeciwnym wypadku mogą wystąpić problemy z niezależną pulą ASP oraz z tworzeniem nowych profili użytkowników.

**/tmp** W katalogu /tmp aplikacje mogą przechowywać pliki tymczasowe. Katalog ten jest podkatalogiem katalogu "root" (/), więc nazwą ścieżki jest /tmp.

Gdy obiekt zostanie umieszczony w katalogu /tmp, pozostaje tam, dopóki aplikacja lub użytkownik aplikacji go nie usunie. System nie usuwa automatycznie obiektów z katalogu /tmp ani nie wykonuje na nich żadnych specjalnych działań.

Katalogiem /tmp i zawartymi w nim obiektami można zarządzać używając ekranów użytkownika i komend obsługujących zintegrowany system plików. Do kopiowania, usuwania lub zmiany nazwy katalogu /tmp lub plików w tym katalogu można na przykład użyć ekranu Praca z dowiązaniem obiektów (Work with Object Links) lub komendy WRKLNK. Wszyscy użytkownicy dostają uprawnienia \*ALL do tego katalogu, co oznacza, że mogą wykonywać większość działań dostępnych na tym katalogu.

Aplikacje mogą używać do zarządzania katalogiem /tmp i zawartymi w nim obiektami funkcji API, które obsługują zintegrowany system plików. Na przykład program użytkowy może usunąć obiekt z katalogu /tmp używając funkcji API unlink().

Jeśli katalog /tmp zostanie usunięty, system utworzy go ponownie podczas następnego restartu systemu.

Dla celów ochrony i ogólnodostępności systemu można podać wartość Yes atrybutu ograniczenie zmiany nazwy i usuwania dowiązań katalogu /tmp.

**Uwaga:** Atrybut ograniczenie zmiany nazwy i usuwania dowiązań odpowiada bitowi S\_ISVTX.

Jeśli atrybut ograniczenie zmiany nazwy i usuwania dowiązań ma wartość Yes, zmiana nazwy katalogu /tmp lub usuwanie dowiązań do obiektów w tym katalogu jest niemożliwe o ile nie jest spełniony jeden z poniższych warunków:

- użytkownik jest właścicielem obiektu,
- użytkownik jest właścicielem katalogu,
- użytkownik ma specjalne uprawnienia do wszystkich obiektów (\*ALLOBJ).

Jeśli atrybut ma wartość Yes, a użytkownik nie posiada odpowiednich uprawnień, wyświetlony zostanie błąd o numerze 3027 (EPERM) lub komunikat MSGCPFA0B1 (Żądana operacja jest niedozwolona. Problem dostępu) dla błędów podczas zmiany nazwy lub usuwania dowiązań przy użyciu następujących komend i funkcji API:

- komendy Usunięcie dowiązania (Remove Link - RMVLNK, DEL i ERASE),
- komendy Usunięcie katalogu (Remove Directory - RMVDIR, RD i RMDIR),
- komendy Zmiana nazwy obiektu (Rename Object - RNM i REN),
- komendy Przeniesienie obiektu (Move Object - MOV i MOVE),
- funkcji API Zmiana nazwy pliku lub katalogu (Rename File or Directory - rename()),
- funkcji API Zmiana nazwy pliku lub katalogu, zachowanie "nowego", jeśli istnieje (Rename File or Directory, Keep "new" If It Exists - Qp0IRenameKeep()),
- funkcji API Zmiana nazwy pliku lub katalogu, usunięcie dowiązania "nowego", jeśli istnieje (Rename File or Directory, Unlink "new" If It Exists - Qp0IRenameUnlink()),
- funkcji API Zmiana nazwy katalogu (Remove Directory - rmdir()),
- funkcji API Usunięcie dowiązania do pliku (Remove Link to File - unlink()).

Atrybut ograniczenie zmiany nazwy i usuwania dowiązań i bit S\_ISVTX mogą zostać zmienione za pomocą komendy Zmiana atrybutu (Change Attribute - CHGATR), funkcji API Określenie atrybutów (Set Attributes - Qp0ISetAttr()) lub Zmiana autoryzacji pliku (Change File Authorizations - chmod), jeśli użytkownik jest właścicielem obiektu lub jeśli posiada uprawnienie specjalne do wszystkich obiektów (\*ALL). Jeśli atrybut zostanie zmieniony na No, utraczona zostanie ogólnodostępność i bezpieczeństwo systemu zapewniane przez wartość Yes.

Wartością tego atrybutu katalogu /tmp tworzono podczas restartu systemu jest Yes. Jeśli katalog /tmp istnieje w momencie restartu systemu, wartość atrybutu nie jest zmieniana.

- /home** Administratorzy systemu tworzą w katalogu /home katalogi osobiste wszystkich użytkowników. Administrator systemu zwykle zakłada użytkownikowi katalog osobisty o nazwie zgodnej z profilem użytkownika, na przykład /home/jasio.
- /etc** W katalogu /etc przechowywane są pliki używane do administrowania systemem, pliki konfiguracyjne i inne pliki systemowe.



**/usr** Katalog /usr zawiera podkatalogi z informacjami używanymi przez system. Zazwyczaj pliki w katalogu /usr są rzadko zmieniane.

**/usr/bin**  
Katalog /usr/bin zawiera standardowe programy narzędziowe.

**/QIBM**  
Katalog /QIBM jest katalogiem systemowym i jest oferowany z systemem.

**/QIBM/ProdData**  
Katalog /QIBM/ProdData jest katalogiem systemowym służącym do przechowywania danych programów licencjonowanych.

**/QIBM/UserData**  
Katalog /QIBM/UserData jest katalogiem systemowym wykorzystywanym do przechowywania danych dotyczących użytkowników programów licencjonowanych, takich jak pliki konfiguracyjne.

**/QOpenSys/QIBM**  
Katalog /QOpenSys/QIBM jest katalogiem systemowym dla systemu plików QOpenSys.

**/QOpenSys/QIBM/ProdData**  
Katalog /QOpenSys/QIBM/ProdData jest katalogiem systemowym systemu plików QOpenSys i służy do przechowywania danych programów licencjonowanych.

**/QOpenSys/QIBM/UserData**  
Katalog /QOpenSys/QIBM/UserData jest katalogiem systemowym dla systemu plików QOpenSys i jest wykorzystywany do przechowywania danych dotyczących użytkowników programów licencjonowanych, takich jak pliki konfiguracyjne.

**/nazwa\_asp/QIBM**  
Katalogi /nazwa\_asp/QIBM są katalogami systemowymi dla niezależnych pul ASP, istniejących w systemie, a nazwa\_asp jest nazwą niezależnej puli ASP.

**/nazwa\_asp/QIBM/UserData**  
Katalog /nazwa\_asp/QIBM/UserData jest katalogiem systemowym wykorzystywanym dla danych o użytkownikach programów licencjonowanych, takich jak pliki konfiguracyjne dla niezależnych pul ASP, istniejących w systemie, a nazwa\_asp to nazwa niezależnej puli ASP.

**/dev** Katalog /dev zawiera różne pliki i katalogi systemowe.

**/dev/xti**  
Katalog /dev/xti zawiera sterowniki urządzeń UDP i TCP.

#### **Pojęcia pokrewne**

“Katalog osobisty” na stronie 7

*Katalog osobisty* staje się bieżącym katalogiem użytkownika bezpośrednio po jego wpisaniu się do systemu. Nazwa katalogu osobistego podana jest w profilu użytkownika.

#### **Odsyłacze pokrewne**

“Urządzenia UDP i TCP w systemie plików "root" (/)" na stronie 30

System plików "root" (/) w katalogu /dev/xti zawiera teraz dwa sterowniki urządzeń o nazwie udp i tcp.

“System plików systemów otwartych (QOpenSys)” na stronie 30

System plików QOpenSys jest zgodny ze standardami systemów otwartych bazującymi na rodzinie systemów UNIX, takimi jak POSIX i XPG (X/Open Portability Guide). Podobnie jak główny system plików "root" (/), ten system plików wykorzystuje oferowaną przez zintegrowany system plików obsługę plików strumieniowych i katalogów.

#### **Informacje pokrewne**

Komenda Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK)

## Katalogi \*TYPE2

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

**Uwaga:** Pojęcia plików strumieniowych typu \*TYPE1 i \*TYPE2 oraz formatów katalogów \*TYPE1 i \*TYPE2 nie są równoznaczne. Nie są one powiązane.

Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1.

Zalety katalogów \*TYPE2 są następujące:

- zwiększona wydajność,
- zwiększona niezawodność,
- zwiększona funkcjonalność,
- w wielu przypadkach, mniejszy obszar pamięci dyskowej.

Katalogi \*TYPE2 w porównaniu z katalogami \*TYPE1 zwiększają wydajność systemu plików, szczególnie podczas tworzenia i usuwania katalogów.

Katalogi \*TYPE2 są bardziej niezawodne niż katalogi \*TYPE1. Po nieprawidłowym zamknięciu systemu katalogi \*TYPE2 są całkowicie odzyskiwane, chyba że wystąpiła awaria pamięci dyskowej. Katalogi \*TYPE1 mogą wymagać użycia komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) w celu całkowitego odzyskania.

Katalogi \*TYPE2 oferują dodatkowo następujące funkcje:

- Katalogi \*TYPE2 obsługują zmianę wielkości liter w nazwie zapisanej w systemie plików stosującym wielkie litery (na przykład zmianę A na a).
- Obiekt w katalogu \*TYPE2 może mieć do miliona dowiązań, natomiast w katalogach \*TYPE1 liczba ta wynosi 32 767 dowiązań. Oznacza to, że dla pliku strumieniowego można zdefiniować do 1 miliona dowiązań stałych, a katalog \*TYPE2 może zawierać do 999 998 podkatalogów.
- Podczas korzystania z programu System i Navigator, podczas otwierania katalogu w formacie \*TYPE2 lista pozycji jest automatycznie sortowana w porządku binarnym.
- Niektóre nowe funkcje, takie jak obsługa skanowania zintegrowanego systemu plików, dostępne są tylko dla obiektów w katalogach \*TYPE2.

Zazwyczaj katalogi \*TYPE2 zawierające mniej niż 350 obiektów wymagają mniej pamięci dyskowej niż katalogi \*TYPE1 zawierające tę samą liczbę obiektów. Katalogi \*TYPE2 zawierające więcej niż 350 obiektów są o 10 procent większe (średnio) niż katalogi \*TYPE1.

Istnieje kilka sposobów tworzenia katalogów \*TYPE2 w systemie:

- Nowe platformy System i, na których są fabrycznie instalowane systemy OS/400 w wersji V5R2, i5/OS w wersji V5R3 lub nowsze, mają katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.
- W przypadku nowej instalacji systemu OS/400 V5R2 lub systemu i5/OS w wersji V5R3 lub późniejszej na platformie System i, w systemie będą używane katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.
- Do konwersji systemów plików służy narzędzie konwersji V5R2. Więcej informacji na temat narzędzia konwersji można znaleźć w sekcji Konwersja na katalog \*TYPE2 w Centrum informacyjnym iSeries V5R2.
- Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2, i5/OS w wersji V5R3 lub nowszy.
- Pozostałe obsługiwane systemy plików z wyjątkiem UDFS w niezależnej puli ASP, które używają katalogów \*TYPE1, zostaną automatycznie poddane konwersji. Konwersja rozpoczyna się po zainstalowaniu systemu i5/OS w wersji V5R3 lub późniejszej. Nie powinna ona znacząco wpłynąć na działanie systemu.

Aby określić format katalogu dla systemu plików w systemie, należy użyć komendy Konwersja katalogu (Convert Directory - CVTDIR):

CVTDIR OPTION(\*CHECK)

**Uwaga:** Katalogi \*TYPE2 są obsługiwane w systemie OS/400 w wersji V5R2, i5/OS w wersji V5R3 lub nowszego, ale zawierają pewne różnice względem normalnej obsługi katalogów \*TYPE2.

#### Odsyłacze pokrewne

“Przekształcanie katalogów z formatu \*TYPE1 na \*TYPE2” na stronie 85

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2.

#### Informacje pokrewne

Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komenda Konwersja katalogu (Convert Directory - CVTDIR)

## Dowiązanie

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

Użytkownikom systemów plików o strukturze katalogowej wygodnie jest traktować obiekt (taki jak plik) jak coś, co ma nazwę służącą do identyfikacji przez system. W rzeczywistości obiekt identyfikuje się przy pomocy ścieżki katalogów. Czasami można mieć dostęp do obiektu poprzez podanie tylko jego *nazwy*. Jest to możliwe dlatego, że system pod pewnymi warunkami zakłada postać części ścieżki opisującej katalogi. Pomysł dowiązania wykorzystuje fakt, że to ścieżka katalogów identyfikuje obiekt. Nazwa odnosi się do dowiązania, a nie do obiektów.

Przyzwyczajenie się do tego, że nazwę ma dowiązanie, a nie obiekt, pozwala zobaczyć ukryte wcześniej możliwości. Może istnieć wiele dowiązań do tego samego obiektu. Na przykład, dwóch użytkowników może współużytkować plik przez przypisanie dowiązania z katalogów osobistych każdego użytkownika do tego pliku (patrz temat “Katalog osobisty” na stronie 7). Pewne typy dowiązań mogą przekraczać granice systemów plików i mogą istnieć bez obiektów.

Istnieją dwa typy dowiązań: dowiązanie stałe i powiązanie symboliczne. Używając nazw ścieżek w programach można wybrać używanie dowiązań stałych lub symbolicznych. Obydwa typy dowiązań mają swoje zalety i wady. Przypadki, w których jeden z typów dowiązań ma przewagę nad drugim, przedstawia poniższa tabela.

Tabela 1. Porównanie dowiązań stałych i symbolicznych

Element	Dowiązanie stałe	Dowiązanie symboliczne
Rozwijanie nazwy	Szybsze. Dowiązanie stałe zawiera bezpośrednie wskazanie do obiektu.	Wolniejsze. Dowiązanie symboliczne zawiera ścieżkę do obiektu, która musi zostać rozwinięta, aby znalezienie obiektu było możliwe.
Istnienie obiektu	Wymagane. Obiekt musi istnieć, by można było utworzyć do niego dowiązanie stałe.	Niekonieczne. Można utworzyć dowiązanie symboliczne, gdy obiekt, na który ono wskazuje, nie istnieje.
Usuwanie obiektu	Ograniczone. Aby można było usunąć obiekt, wszystkie dowiązania stałe do niego muszą zostać usunięte.	Nieograniczone. Obiekt można usunąć nawet wtedy, gdy istnieją wskazujące na niego dowiązania symboliczne.
Obiekty statyczne (atributy nie zmieniają się)	Szybsze. W przypadku obiektów statycznych najistotniejsze dla wydajności jest rozwijanie nazwy. Rozwijanie nazwy jest szybsze w przypadku dowiązania stałego.	Wolniejsze. Rozwijanie nazwy jest wolniejsze w przypadku dowiązania symbolicznego.

Tabela 1. Porównanie dowiązań stałych i symbolicznych (kontynuacja)

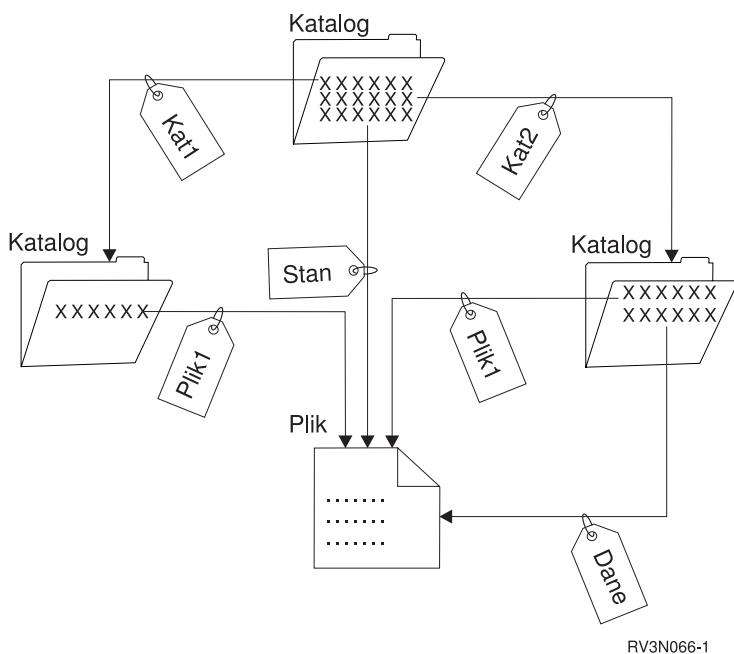
Element	Dowiązanie stałe	Dowiązanie symboliczne
Zasięg	Ograniczone. Dowiązania stałe nie mogą przekraczać granic systemów plików.	Nieograniczone. Dowiązania symboliczne mogą przekraczać granice systemów plików.

## Dowiązanie stałe

*Dowiązanie stałe*, zwane po prostu dowiązaniem, nie może istnieć bez przyłączenia do rzeczywistego obiektu.

Gdy obiekt zostanie utworzony w katalogu (na przykład poprzez skopiowanie pliku do katalogu), zostanie ustalone pierwsze dowiązanie stałe między katalogiem a obiektem. Użytkownicy i programy użytkowe mogą dodawać inne dowiązania stałe. Każde dowiązanie stałe jest wskazywane przez osobną pozycję w katalogu. Dowiązania z tego samego katalogu nie mogą mieć tej samej nazwy, ale dowiązania z różnych katalogów - mogą.

Może istnieć wiele dowiązań stałych do obiektów z tego samego lub z różnych katalogów, o ile są one obsługiwane przez dany system plików. Jedyny wyjątek stanowi obiekt będący innym katalogiem. Może istnieć tylko jedno dowiązanie stałe z katalogu do katalogu.



Rysunek 4. Każde dowiązanie stałe definiuje pozycję katalogu

Dowiązania stałe można usuwać bez wpływu na istnienie obiektu, dopóki pozostaje przynajmniej jedno dowiązanie do tego obiektu. Po usunięciu ostatniego dowiązania stałego, obiekt zostanie usunięty z systemu, chyba że jakaś aplikacja ma ten obiekt otwarty. Każda aplikacja mająca otwarty obiekt może używać go dalej aż do zamknięcia tego obiektu. Po zamknięciu obiektu przez ostatnią używającą go aplikację obiekt zostaje usunięty z systemu. Obiektu nie można otworzyć po usunięciu ostatniego dowiązania stałego.

Pojęcie dowiązania stałego, z pewnymi ograniczeniami, może się także odnosić do systemu plików QSYS.LIB lub systemu QSYS.LIB niezależnej puli ASP i do systemu plików obsługującego obiekty biblioteki dokumentów (QDLS). Tak więc biblioteka ma jedno dowiązanie stałe do każdego zawartego w niej obiektu. Podobnie folder ma jedno dowiązanie stałe do każdego dokumentu. Wiele stałych dowiązań do *tego samego obiektu* nie jest jednak dozwolonych w systemach plików QSYS.LIB, systemach QSYS.LIB niezależnej puli ASP i w systemach QDLS.

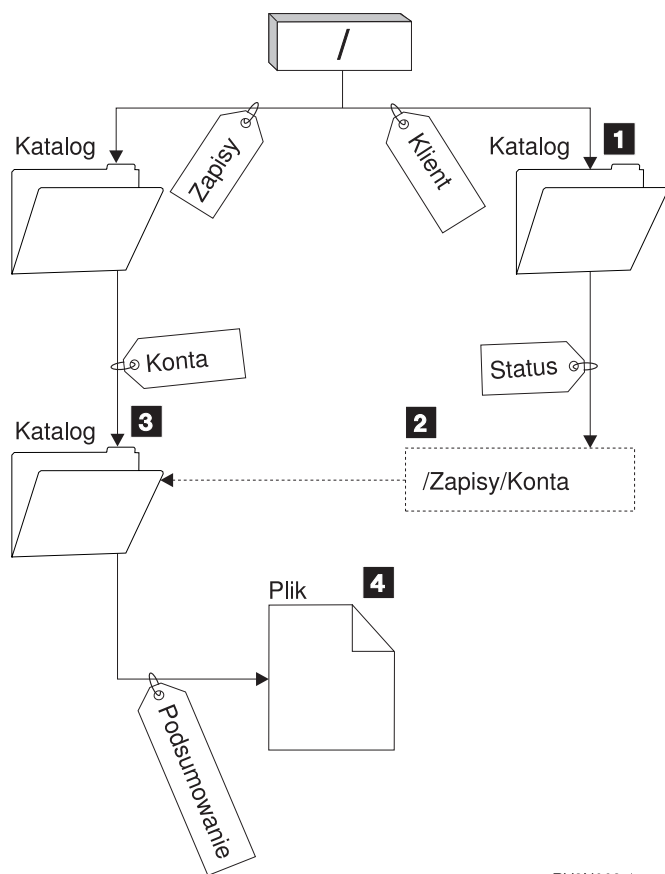
Dowiązania stałego nie można zakładać między systemami plików. Na przykład katalog w systemie plików QOpenSys nie może mieć dowiązania stałego do obiektu w systemie plików QSYS.LIB lub w QSYS.LIB niezależnej puli ASP lub do dokumentu w systemie plików QDLS.

## Dowiązanie symboliczne

*Dowiązanie symboliczne* (symbolic link), zwane także dowiązaniem miękkim (soft link), jest ścieżką zawartą w pliku.

Gdy system napotka dowiązanie symboliczne, "idzie" ścieżką występującą w tym dowiązaniu symbolicznym, a następnie pozostałą częścią ścieżki występującą po dowiązaniu symbolicznym. Jeśli ścieżka ta zaczyna się od /, system powraca do katalogu / ("root") i idzie ścieżką, począwszy od tego punktu. Jeśli ścieżka nie zaczyna się znakiem /, system powraca do poprzedniego katalogu i "idzie" ścieżką zawartą w dowiązaniu symbolicznym, rozpoczynając od tego katalogu.

Rozważmy poniższy przykład sposobu użycia dowiązania symbolicznego:



RV3N068-1

Rysunek 5. Przykład zastosowania dowiązania symbolicznego

Użytkownik wybiera opcję menu, aby wyświetlić stan konta klienta. Program wyświetlający menu używa poniższej ścieżki:

```
/Klient/Stan/Podsumowanie
```

System "idzie" za dowiązaniem *Klient*, które prowadzi do katalogu 1, następnie "idzie" za dowiązaniem *Stan*. Dowiązanie *Stan* jest dowiązaniem symbolicznym, zawierającym nazwę ścieżki 2. Ponieważ nazwa ścieżki zaczyna się od znaku /, system powraca do katalogu / ("root") i podąża po kolei poprzez dowiązania *Zapisy* i *Konta*. Ta ścieżka prowadzi do innego katalogu (3). Teraz system kończy ścieżkę dostarczoną przez program. Następnie system podąża za dowiązaniem *Podsumowanie*, które prowadzi do pliku 4 zawierającego dane potrzebne użytkownikowi.

W przeciwieństwie do dowiązania stałego, dowiązanie symboliczne jest obiektem (typu \*SYMLNK); może ono istnieć bez wskazywania na istniejący obiekt. Dowiązania można użyć na przykład do utworzenia ścieżki do pliku, który zostanie dodany lub zastąpiony w późniejszym czasie.

Ponadto, inaczej niż dowiązanie stałe, dowiązanie symboliczne może przekraczać granice systemów plików. Na przykład, przy pracy z jednym systemem plików można użyć dowiązania symbolicznego do udostępnienia pliku znajdującego się w innym systemie plików. Mimo, że system plików QSYS.LIB, system QSYS.LIB niezależnej puli ASP oraz system QDLS nie obsługują tworzenia i przechowywania dowiązań symbolicznych, można utworzyć dowiązanie symboliczne w systemie plików "root" (/) lub QOpenSYS, co umożliwia:

- dostęp do podzbiorów zbiorów bazy danych w systemie plików QSYS.LIB lub QSYS.LIB niezależnej puli ASP,
- dostęp do dokumentów w systemie plików QSYS.LIB.

## Nazwa ścieżki

Nazwa ścieżki informuje system, jak odnaleźć dany obiekt.

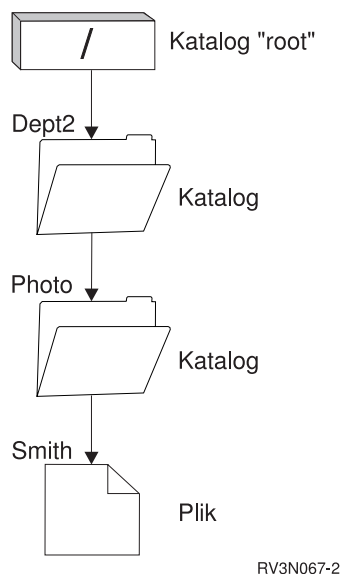
Jest to sekwencja nazw katalogów zakończona nazwą obiektu. Poszczególne nazwy katalogów i obiektów rozdziela znak ukośnika (/); na przykład:

katalog1/katalog2/plik

Dla ułatwienia w komendach systemowych zamiast ukośnika można używać ukośnika odwrotnego (\).

Ścieżkę określa się na dwa sposoby:

- *Ścieżka bezwzględna* zaczyna się od najwyższego poziomu, czyli katalogu "root" (określanego przez znak /). Jako przykład rozważmy poniższą ścieżkę z katalogu / do pliku o nazwie Smith.



Rysunek 6. Składniki ścieżki

Ścieżka bezwzględna do pliku Smith wygląda następująco:

/Dept2/Photo/Smith

Bezwzględną nazwę ścieżki nazywa się również *pełną nazwą ścieżki*.

- Jeśli ścieżka nie zaczyna się znakiem /, system przyjmuje, że ścieżka zaczyna się w bieżącym katalogu użytkownika. Taką ścieżkę nazywamy *ścieżką względną*. Na przykład, jeśli bieżący katalog użytkownika o nazwie Dept2 zawiera podkatalog Photo, w którym znajduje się plik Smith, to względną ścieżką do tego pliku jest:

Photo/Smith

Zauważ, że ścieżka nie zawiera nazwy bieżącego katalogu. Pierwszą pozycję nazwy stanowi katalog lub obiekt znajdujący się na *kolejnym niższym poziomie* względem katalogu bieżącego.

### Odsyłacze pokrewne

“Reguły dotyczące ścieżek dla funkcji API” na stronie 114

Gdy do wykonywania działań na obiekcie używane są zintegrowany system plików lub funkcja API ILE C, obiekt identyfikowany jest przez podanie ścieżki do katalogu. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API.

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 66

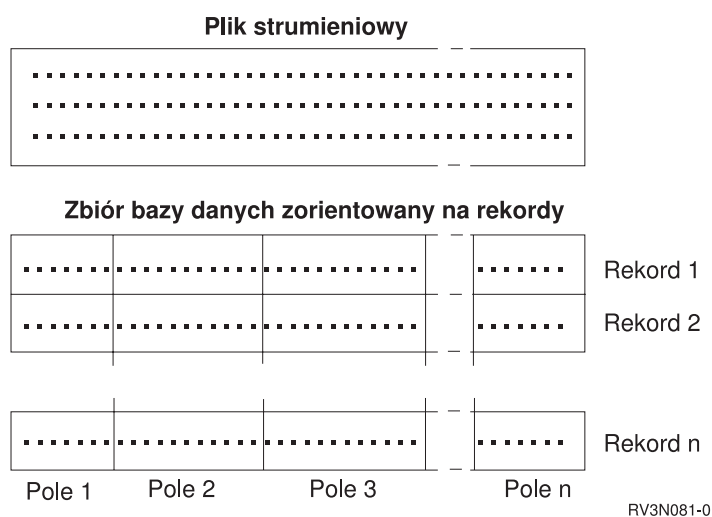
Gdy do wykonywania działań na obiekcie używane są komendy lub ekrany zintegrowanego systemu plików, obiekt identyfikowany jest przez podanie jego właściwej ścieżki.

## Plik strumieniowy

*Plik strumieniowy* jest dostępną losowo sekwencją bajtów, pozbawioną struktury narzucanej przez system.

Zintegrowany system plików służy do przechowywania informacji w formie plików strumieniowych (stream files) i ich przetwarzania. Dokumenty przechowywane w folderach systemu są plikami strumieniowymi. Innymi przykładami plików strumieniowych są pliki komputerów PC i pliki w systemach UNIX. Plik strumieniowy zintegrowanego systemu plików jest obiektem systemowym typu \*STMF.

Aby lepiej zrozumieć pliki strumieniowe, najlepiej porównać je ze zbiorami baz danych systemu i5/OS. Zbiór bazy danych zawiera wstępnie zdefiniowane części składowe złożone z jednego lub kilku pól o konkretnych cechach, takich jak długość i typ danych.



Rysunek 7. Porównanie pliku strumieniowego z plikiem opartym na rekordach.

Różna struktura plików strumieniowych i plików opartych na rekordach wpływa na ich wykorzystanie, na sposób pisania aplikacji, które mają z nimi współpracować, oraz na wybór typu pliku do użycia w aplikacji. Przykładowo plik oparty na rekordach nadaje się do przechowywania danych statystycznych klienta, takich jak nazwisko, adres i bilans rachunku. Za pomocą dostępnego w systemie bogatego oprogramowania możliwy jest dostęp do poszczególnych pól zbioru zawierającego rekordy oraz manipulacja tymi polami. Z kolei plik strumieniowy bardziej nadaje się do przechowywania takich danych, jak na przykład zdjęcia klientów, które składają się z nieprzerwanych ciągów bitów reprezentujących różne kolory. Pliki strumieniowe bardzo dobrze nadają się do przechowywania ciągów danych, takich jak tekst dokumentu, obrazy, dźwięk i wideo.

Plik ma ustawioną jedną z dwóch opcji formatu: plik strumieniowy \*TYPE1 lub plik strumieniowy \*TYPE2. Format pliku zależy od wydania systemu, w którym został utworzony, i jeśli utworzono go w systemie plików użytkownika, od wartości podanej dla tego systemu plików.

**Uwaga:** Pojęcia plików strumieniowych typu \*TYPE1 i \*TYPE2 oraz formatów katalogów \*TYPE1 i \*TYPE2 nie są równoznaczne. Nie są one powiązane.

## Pliki strumieniowe \*TYPE1

Plik strumieniowy \*TYPE1 ma ten sam format, co pliki strumieniowe utworzone w wydaniach poprzedzających wersję V4R4 systemu OS/400.

Plik strumieniowy \*TYPE1 może mieć wielkość minimalnie 4096 bajtów, a maksymalnie około 128 GB (1 GB to w przybliżeniu 1 073 741 824 bajtów).

## Pliki strumieniowe \*TYPE2

Pliki strumieniowe typu \*TYPE2 zapewniają wydajny dostęp do plików.

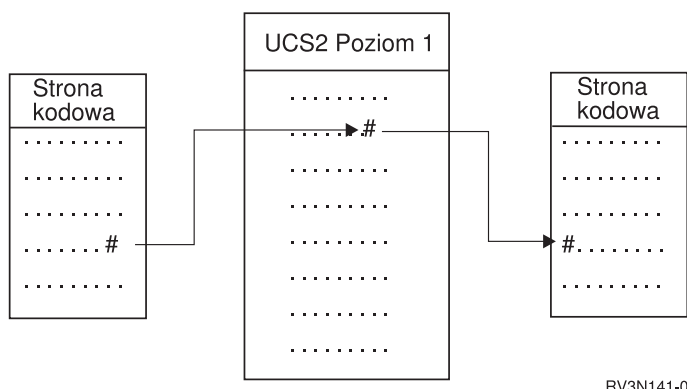
Maksymalna wielkość obiektu wynosi dla nich około 1 TB (1 TB jest równy około 1 099 511 627 776 bajtom) w systemach plików "root" (/), QOpenSys i użytkownika. W pozostałych przypadkach wielkość maksymalna wynosi około 256 GB. Możliwe jest także odwzorowywanie pamięci i określanie atrybutów optymalizujących przydzielanie pamięci głównej. Wszystkie pliki utworzone w systemach w wersji V4R4 i nowszych systemu OS/400 są plikami strumieniowymi \*TYPE2, chyba że zostały utworzone w systemie plików użytkownika, który zdefiniowano w formacie \*TYPE1.

**Uwaga:** Plików większych niż 256 gigabajtów nie można zeszkładować ani odtworzyć w systemach i5/OS starszych niż V5R3.

## Niezmienność nazw

Użytkownicy systemu plików "root" (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmienność znaków w nazwach obiektów.

W szczególności dotyczy to sytuacji, gdy te systemy plików obejmują wiele systemów i podłączonych urządzeń, na których używane są różne schematy kodowania znaków (strony kodowe). System przechowuje znaki w nazwach w formacie 16-bitowym znanym jako UCS2 Poziom 1 (nazywanym także *Unicode*) dla katalogów \*TYPE1 i UTF-16 dla katalogów \*TYPE2. UCS2 Poziom 1 i UTF-16 są podzbiórmi standardu ISO 10646. Gdy nazwa ma być użyta, system dokonuje konwersji formatu znaków na reprezentację znaków odpowiednią dla aktualnie używanej strony kodowej. Nazwy rozszerzonych atrybutów związanych z każdym obiektem traktowane są w taki sam sposób.



Rysunek 8. Utrzymywanie niezmienności znaków w różnych schematach kodowania

Obsługa ta ułatwia współdziałanie systemu z urządzeniami używającymi innych stron kodowych. Na przykład użytkownicy komputerów PC mogą uzyskać dostęp do pliku systemu i5/OS używając tej samej nazwy pliku, pomimo



że ich komputery PC nie używają tej samej strony kodowej, co system. Konwersja stron kodowych jest wykonywana automatycznie przez system. Oczywiście urządzenie musi używać strony kodowej zawierającej znaki wykorzystywane w danej nazwie.

### **Pojęcia pokrewne**

“Katalogi \*TYPE2” na stronie 10

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

“Przegląd automatycznej konwersji nazw” na stronie 90

Niektóre systemy plików, które nie rozróżniają wielkości liter, na przykład system "root" (/) i UDFS utworzone przy użyciu funkcji CASE(\*MONO), obsługują nazwy zapisane w standardzie Unicode 4.0. System wykonuje automatyczną konwersję nazw, aby obsłużyć dodatkowe znaki w tych nazwach.

## **Atrybuty rozszerzone**

Atrybut rozszerzony jest informacją związaną z obiektem, która zawiera dodatkowe szczegóły na temat tego obiektu. Atrybut rozszerzony składa się z nazwy, będącej odnośnikiem do atrybutu, i wartości. Wartością może być tekst, dane binarne lub dane innego typu.

Atrybuty rozszerzone dla obiektu istnieją tylko tak długo, jak długo istnieje dany obiekt.

Atrybuty rozszerzone (EA) występują w wielu odmianach i mogą być używane do przechowywania różnych informacji. Należy znać w szczególności następujące trzy atrybuty rozszerzone:

### **.SUBJECT**

Krótki opis zawartości i przeznaczenia obiektu.

**.TYPE** Typ danych w obiekcie. Typem tym może być tekst, dane binarne, kod źródłowy programu, skompilowany program lub inne informacje.

### **.CODEPAGE**

Strona kodowa, która ma zostać użyta przy pracy z obiektem. Strona kodowa używana dla obiektu jest wykorzystywana również dla atrybutów rozszerzonych związanych z obiektem.

Kropka (.) jako pierwszy znak nazwy oznacza, że atrybuty te są standardowymi systemowymi atrybutami rozszerzonymi zarezerwowanymi dla systemu.

Różne obiekty w różnych systemach plików mogą mieć lub nie atrybuty rozszerzone. Systemy plików QSYS.LIB i QSYS.LIB niezależnej puli ASP obsługują trzy predefiniowane atrybuty rozszerzone: .SUBJECT, .TYPE i .CODEPAGE. W systemie plików obsługującym obiekty biblioteki dokumentów (QDLS) foldery i dokumenty mogą mieć atrybuty rozszerzone dowolnego typu. Jedne foldery i dokumenty mogą mieć atrybuty rozszerzone, a inne nie. W systemach plików "root" (/), QOpenSys i systemach plików użytkownika wszystkie katalogi, pliki strumieniowe i dowiązania symboliczne mogą mieć dowolne atrybuty rozszerzone. Inne obiekty natomiast mogą wcale nie posiadać atrybutów rozszerzonych.

Do wyświetlenia atrybutu rozszerzonego .SUBJECT dla danego obiektu można użyć komend Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK) oraz Wyświetlenie dowiązań obiektu (DSPLNK). W zintegrowanym systemie plików nie istnieje żadna inna metoda dająca aplikacjom i użytkownikom dostęp do atrybutów rozszerzonych. Jedynym odstępstwem od tej reguły są następujące komendy CL: Wyświetlenie danych podłączonego FS (Display a UDFS - DSPUDFS) i Wyświetlenie informacji o dołączonym systemie plików (Display Mounted File System Information - DSPMFSINF), które wyświetlają atrybuty rozszerzone.

Atrybuty rozszerzone związane z pewnymi obiektami w systemie plików QDLS mogą być zmieniane poprzez interfejsy dostarczane przez hierarchiczny system plików (HFS).

Jeśli komputer PC klienta jest połączony z platformą System i poprzez system operacyjny OS/2 lub Windows, interfejsy programistyczne jego systemu operacyjnego (jak DosQueryFileInfo i DosSetFileInfo) mogą posłużyć do

wydania zapytania i skonfigurowania atrybutów rozszerzonych dowolnego obiektu pliku. Użytkownicy systemu OS/2 mogą także zmienić atrybuty rozszerzone obiektu na pulpicie używając notatnika ustawień, to znaczy wybierając opcję **Ustawienia** menu obiektu.

Przy definiowaniu atrybutów rozszerzonych należy uwzględniać następujące zasady dotyczące nazewnictwa:

- nazwa atrybutu rozszerzonego może mieć do 255 znaków;
- nie należy używać kropki (.) jako pierwszego znaku nazwy; atrybut rozszerzony o nazwie rozpoczynającej się od kropki jest interpretowany jako standardowy systemowy atrybut rozszerzony;
- aby zmniejszyć prawdopodobieństwo wystąpienia konfliktu nazw, należy używać spójnej struktury nazewnictwa atrybutów rozszerzonych; zalecana jest następująca forma:

NazwaFirmyNazwaProduktu.Nazwa\_Atrybutu

## Obsługa skanowania

System operacyjny i5/OS umożliwia skanowanie obiektów zintegrowanego systemu plików.

Ułatwia to pracę użytkownikom dzięki możliwości skanowania różnego rodzaju pozycji. Użytkownicy decydują też, kiedy skanowanie zostanie przeprowadzone oraz jakie działanie zostanie podjęte w zależności od wyników skanowania.

Dwoma nowymi punktami wyjścia związanymi z obsługą są:

- QIBM\_QP0L\_SCAN\_OPEN - Skanowanie zintegrowanego systemu plików po otwarciu programu obsługi wyjścia.

Dla tego punktu wyjścia wywoływane jest skanowanie zintegrowanego systemu plików dla otwartych programów obsługi wyjścia, które wykonują przetwarzanie skanowania po otwarciu obiektu zintegrowanego systemu plików w określonych warunkach.

- QIBM\_QP0L\_SCAN\_CLOSE - Skanowanie zintegrowanego systemu plików po zamknięciu programu obsługi wyjścia.

Dla tego punktu wyjścia wywoływane jest skanowanie zintegrowanego systemu plików dla zamkniętych programów obsługi wyjścia, które wykonują przetwarzanie skanowania po zamknięciu obiektu zintegrowanego systemu plików w określonych warunkach.

**Uwaga:** Skanowane mogą być tylko obiekty w systemach plików, które zostały w pełni poddane konwersji katalogów do formatu \*TYPE2.

### Zadania pokrewne

“Określanie, czy obiekty mają być skanowane” na stronie 129

Określanie obiektów, które mają być skanowane, jest możliwe w systemach plików "root" (/), QOpenSys i użytkownika. Wykonaj następujące czynności, aby ustawić opcje skanowania.

### Informacje pokrewne

Skanowanie zintegrowanego systemu plików przy otwieraniu programu obsługi wyjścia

Skanowanie zintegrowanego systemu plików przy zamykaniu programu obsługi wyjścia

## Przykłady: Skanowanie w poszukiwaniu wirusów i skanowanie otwieranych plików

Poniższe przykłady pokazują, w poszukiwaniu jakich obiektów można wykonywać skanowanie.

- Wirusy

Program obsługi wyjścia może wykonać skanowanie w poszukiwaniu wirusów. Jeśli w pliku zostanie znaleziony wirus, program antywirusowy może naprawić problem lub spróbować poddać wirusa kwarantannie. Ponieważ sama platforma System i nie może zostać zainfekowana znalezionym wirusem, działanie to ma na celu ograniczenie ryzyka przenoszenia wirusa na inne systemy.

- Wywołania w celu sprawdzenia, czy plik został otwarty

Można także wykonać skanowanie, aby sprawdzić, kiedy plik został otwarty. Uruchamiając skanowanie można sprawdzić datę i godzinę dostępu do pewnych plików. Jest to przydatne, jeśli potrzebna jest możliwość śledzenia działań wybranych użytkowników.

Skanowanie może być uruchomione w dwóch różnych momentach, zależnie od konfiguracji wartości systemowych i środowiska skanowania. Poniższa lista przedstawia różne rodzaje skanowania, zależnie od momentu, w którym jest wykonywane.

#### 1. Skanowanie w czasie wykonywania

Skanowanie w czasie wykonywania to skanowanie pliku lub plików w trakcie normalnych codziennych czynności. Zapewnia integralność plików przy każdym dostępie. Skanowanie w trakcie normalnych czynności umożliwia zapewnienie aktualności danych w plikach bez względu na powód skanowania.

##### **Przykład skanowania pod kątem wirusów w czasie wykonywania**

Użytkownik uzyskuje dostęp do pliku w zintegrowanym systemie plików z komputera PC. Podczas otwierania pliku z komputera PC następuje skanowanie. Zarejestrowany został otwarty program obsługi wyjścia, a wartość systemowa QSCANFS wskazuje na skanowanie plików w systemach "root" (/), QOpenSys i UDFS. Skanowanie wykrywa jeden wirus, a program antywirusowy przystępuje do naprawy problemu. Po naprawieniu pliku przez program obsługi wyjścia nie jest on już zainfekowany. Dostęp z komputera PC nie wiąże się więc z infekcją.

Następnie można założyć, że zamiast skanowania w poszukiwaniu wirusów zdecydowano nie przeprowadzać skanowania w czasie wykonywania. Po uzyskaniu dostępu do zainfekowanego pliku z komputera PC wirus zostaje przeniesiony do tego komputera. Uruchomienie skanowania w czasie wykonywania umożliwia wykrycie wirusa, zanim zainfekuje on komputer PC.

Główną wadą tej metody jest zwiększenie czasu oczekiwania na zasoby o czas wykonywania skanowania. Użytkownicy, którzy usiłują uzyskać dostęp do pliku, muszą czekać, aż skanowanie zostanie zakończone. System zapewnia, że skanowanie jest przeprowadzane tylko gdy jest to wymagane, a nie podczas każdego dostępu.

#### 2. Skanowanie masowe lub uruchamiane ręcznie

Opcji tej można użyć, aby jednocześnie skanować wiele pozycji. W takim przypadku można skonfigurować uruchamianie skanowania, gdy system nie jest obciążony pracą, na przykład podczas dni wolnych. Spowoduje to minimalny wpływ skanowania na czas oczekiwania na dostęp do pliku w trakcie normalnej codziennej pracy. Skanowanie jest przeprowadzane bez łączenia się z siecią. Pozwala to zatem skrócić czas oczekiwania na przeskanowanie pliku podczas jego wykonywania, gdyż ponowne skanowanie pliku nie jest konieczne, jeśli nie uległ on zmianie od czasu ostatniego skanowania masowego.

##### **Pojęcia pokrewne**

"Pokrewne wartości systemowe"

Wartości systemowe QSCANFS i QSCANFSCTL służą do zdefiniowania środowiska skanowania w systemie.

##### **Informacje pokrewne**

Skanowanie zintegrowanego systemu plików przy otwieraniu programu obsługi wyjścia

Skanowanie zintegrowanego systemu plików przy zamykaniu programu obsługi wyjścia

## **Pokrewne wartości systemowe**

Wartości systemowe QSCANFS i QSCANFSCTL służą do zdefiniowania środowiska skanowania w systemie.

Poniżej podano nazwy i opisy wartości systemowych związanych ze skanowaniem. Podane wartości i ich opcje sterujące dotyczą programu System i Navigator. Odpowiadające im wartości dostępne z interfejsu znakowego są podane w nawiasach po nazwach System i Navigator. Na przykład wyniki generowane dla wartości systemowej QSCANFSCTL, gdy zostanie wybrana w programie System i Navigator opcja sterowania Skanowanie wyłącznie dostępu przez serwery plików, są takie same, jak wyniki podania opcji sterowania \*FSVROONLY w interfejsie znakowym.

Nazwy i opisy wartości systemowych:

1. Do skanowania systemów plików "root" (/), QOpenSys i użytkownika należy używać zarejestrowanego programu obsługi wyjścia.

Ta wartość systemowa może posłużyć do określenia, czy system plików ma być skanowany. Jeśli system plików został w pełni poddany konwersji, skanowane będą tylko obiekty w systemie plików "root" (/), QOpenSys i użytkownika. Wartość ta określa, czy obiekt ma być skanowany przez programy obsługi wyjścia zarejestrowane dla dowolnego punktu wyjścia zintegrowanego systemu plików.

Wartość domyślna to skanowanie obiektów dla każdego zarejestrowanego programu obsługi wyjścia.

## 2. Sterowanie skanowaniem (QSCANFCTL)

Dla tej wartości systemowej można użyć domyślnych opcji sterowania lub użyć zdefiniowanych opcji. Poniżej podano krótkie opisy poszczególnych opcji sterowania dla wartości systemowych dostępnych w programie System i Navigator:

- Skanowanie wyłącznie dostępu przez serwery plików (z opcją \*FSVROONLY)  
Skanowanie będzie wykonywane tylko wtedy, gdy dostęp do platformy System i uzyskano z serwera plików. Jeśli opcja ta nie zostanie wybrana, skanowane są wszystkie próby dostępu.
- Zgłoś niepowodzenie żądania, jeśli wykonanie programu obsługi wyjścia nie powiedzie się (z opcją \*ERRFAIL)  
Jeśli występują błędy podczas wywoływania programu obsługi wyjścia, zgłoszenie lub operacja, która wyzwołała wywołanie programu obsługi wyjścia, zakończy się niepowodzeniem. Gdy opcja ta nie jest wybrana, system pominię błędny program obsługi wyjścia i będzie traktował obiekt, jakby nie był skanowany.
- Wykonaj aktualizację dostępu z prawami zapisu (bez opcji \*NOWRTUPG)  
Aktualizacja dostępu zostanie wykonana dla deskryptora skanowania przekazanego do programu obsługi wyjścia w celu uwzględnienia dostępu z prawami zapisu. Jeśli nie wybrano opcji \*NOWRTUPG, system nie będzie próbował przeprowadzić aktualizacji dostępu z prawami zapisu.  
Jeśli podano opcję \*NOWRTUPG, system **nie** będzie próbował zaktualizować dostępu do deskryptora skanowania przekazanego do programu obsługi wyjścia w celu uwzględnienia dostępu z prawami zapisu. Jeśli nie podano opcji \*NOWRTUPG, system będzie próbował zaktualizować dostęp z prawami zapisu.
- Do sterowania skanowaniem służy atrybut 'Tylko gdy obiekty zostały zmienione' (z opcją \*USEOCOATR)  
Użyty zostanie atrybut 'tylko zmiana obiektu' (skanowanie obiektu tylko, gdy został zmieniony). Jeśli opcja ta nie została wybrana, atrybut nie zostanie użyty, a obiekt zostanie przeskanowany po wprowadzeniu zmian i gdy oprogramowanie skanowania wykaże aktualizację.
- Zgłoś niepowodzenie żądania zamknięcia, jeśli skanowanie nie powiedzie się podczas zamykania (bez opcji \*NOFAILCLO)  
Jeśli skanowanie obiektu w trakcie przetwarzania zamknięcia zakończy się niepowodzeniem, zgłoszenie zamknięcia zwróci błąd. Jeśli opcja ta nie zostanie wybrana, zgłoszenie zamknięcia nie zwróci błędu. Nie wybranie tej wartości powoduje przesłonięcie podanej wartości 'niepowodzenie zgłoszenia w przypadku niepowodzenia programu obsługi wyjścia'.  
Jeśli podano opcję \*NOFAILCLO, system **nie** zwróci błędu dla zgłoszenia zamknięcia ze wskazaniem przyczyny niepowodzenia nawet wtedy, gdy skanowanie obiektu w ramach przetwarzania zamknięcia nie powiedzie się.
- Skanuj przy kolejnym dostępie po odtworzeniu obiektu (bez opcji \*NOPOSTRST)  
Obiekty będą skanowane po odtworzeniu. Jeśli podano atrybut 'obiekt nie będzie skanowany', obiekt zostanie przeskanowany raz, po odtworzeniu. Jeśli podano atrybut 'tylko zmiana obiektu', obiekt zostanie przeskanowany po odtworzeniu.  
Jeśli podano opcję \*NOPOSTRST, gdy obiekt jest odtwarzany, nie zostanie przeskanowany, ponieważ został właśnie odtworzony. Jeśli atrybut obiektu ma wartość 'obiekt nie będzie skanowany', obiekt nie będzie skanowany ani razu. Jeśli atrybut obiektu ma wartość 'tylko zmiana obiektu', obiekt będzie skanowany, gdy zostanie zmieniony po odtworzeniu.

### Informacje pokrewne

Wartości systemowe ochrony: kontrola skanowania

## Operacje skanowania

Skanowanie może zostać uruchomione z wielu powodów. Sekcja zawiera informacje na temat możliwych okoliczności i przyczyn uruchomienia skanowania.

Aby przejrzeć aktualny status skanowania, można użyć komendy Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK), komendy Wyświetlenie dowiązań obiektu (Display Object Links - DSPLNK), funkcji API Pobranie atrybutów (Qp0lGetAttr()) lub strony Właściwości (Properties) w programie System i Navigator.

### Informacje pokrewne

Komenda Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK)

Komenda Wyświetlenie dowiązań obiektu (Display Object Links - DSPLNK)

Qp0lGetAttr()--funkcja API Get Attributes

### Zmiana obiektu:

Skanowanie zostanie uruchomione, gdy użytkownik uzyska dostęp do obiektu po wprowadzeniu zmian w obiekcie.

Zazwyczaj modyfikowane są dane obiektu. Przykładami zmian w obiekcie jest bezpośrednio zapisanie danych do obiektu albo odzworowanie pamięci, obcięcie lub usunięcie zawartości obiektu. Jeśli atrybut CCSID obiektu zostanie zmieniony, spowoduje to także skanowanie obiektu przed ponownym dostępem.

### Zmiana sygnatury:

Skanowanie jest dokonywane w chwili dostępu do obiektu, jeśli globalna sygnatura jest inna niż sygnatura obiektu.

Globalne lub niezależne sygnatury puli ASP oznaczają poziom oprogramowania przypisanego do programów obsługi wyjścia związanych ze skanowaniem. Gdy obiekt był ostatnio skanowany, sygnatura obiektu odzwierciedla sygnaturę globalną lub niezależnej puli ASP. Gdy obiekt nie jest w grupie niezależnych pul ASP, sygnaturę obiektu porównuje się z globalną sygnaturą skanowania. Jeśli obiekt jest w niezależnej puli ASP, sygnatura obiektu porównywana jest z powiązaną sygnaturą skanowania grupy niezależnych pul ASP.

**Uwaga:** W poniższym przykładzie użyto klucza skanowania fraz i sygnatury klucza skanowania. Klucz skanowania to metoda identyfikowania zestawu oprogramowania skanującego. Przykład przygotowano dla konkretnej firmy. Sygnatura klucza skanowania umożliwi zestawowi oprogramowania skanującego wskazanie poziomu prowadzonej obsługi. Przykładem jest zestaw definicji wirusów.

Poniżej znajduje się przykład sytuacji, w której obiekt nie znajduje się w grupie niezależnych ASP i następuje skanowanie:

1. Program obsługi wyjścia jest rejestrowany w punkcie wyjścia QIBM\_QP0L\_SCAN\_OPEN. Klucz skanowania i sygnatura klucza skanowania zostały podane następująco:

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 0000000000

Globalna sygnatura skanowania to: 0000 i nie jest aktualizowana.

2. Program obsługi wyjścia jest następnie rejestrowany dla punktu wyjścia QIBM\_QP0L\_SCAN\_CLOSE. Klucz skanowania i sygnatura klucza skanowania zostały podane następująco:

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 1111111111

Globalna sygnatura skanowania jest aktualizowana do wartości: 0001.

3. Następnie otwierany jest plik, który aktualnie ma sygnaturę 0000. Istnienie programów obsługi wyjścia wraz z różnicą globalnych sygnatur skanowania (od 0000 do 0001) powoduje zainicjowanie operacji skanowania. Po pomyślnym zakończeniu skanowania sygnatura pliku jest aktualizowana do wartości 0001.
4. Jeśli tabela zostanie otwarta przez innego użytkownika, skanowanie nie zostanie powtórzone, ponieważ sygnatura obiektu i globalna sygnatura są takie same.

Poniższy przykład ilustruje sytuację, w której program skanujący usiłuje ponownie skanowanie:

1. W systemie dodano obsługę skanowania w poszukiwaniu nowych typów wirusów. Funkcja API Zmiana sygnatury skanowania (Change Scan Signature - QP0LCHSG) została wywołana w celu zaktualizowania sygnatury klucza skanowania dla klucza skanowania. Klucz skanowania i jego sygnatura mają następujące wartości:

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 2222222222

Globalna sygnatura klucza skanowania jest aktualizowana do wartości: 0002.

2. Jeśli uprzednio skanowany plik zostanie teraz otwarty, różnica sygnatur spowoduje ponowne skanowanie pliku.

Przykład ten ilustruje dalej sytuację, w której obiekt należy do grupy niezależnych ASP:

1. Niezależna pula ASP jest udostępniana po raz pierwszy i plik w tej puli zostaje otwarty. Gdy otwarty zostanie pierwszy plik, klucz skanowania niezależnej ASP porównywany jest z listą kluczy skanowania w systemie. Wartości różnią się, ponieważ nie istnieje lista skanowania dla niezależnej ASP. Wówczas lista kluczy skanowania niezależnej ASP pobiera globalną listę kluczy skanowania. Lista kluczy skanowania niezależnej ASP otrzyma wówczas klucz skanowania XXXXXX i sygnaturę klucza 222222222. Wynikiem będzie zmiana sygnatury skanowania niezależnej ASP do wartości 0001. Gdy w niezależnej ASP zostanie otwarty plik z sygnaturą obiektu 0000, porównywana jest ona z sygnaturą skanowania niezależnej ASP 0001 i z powodu różnicy wartości plik jest skanowany. Po pomyślnym zakończeniu skanowania sygnatura pliku jest aktualizowana do wartości 0001.

**Uwaga:** Zmiana sygnatury spowoduje wyzwolenie skanowania, chyba że obiekt ma przypisany atrybut 'tylko zmiana obiektu' i podaną wartość systemową \*USEOCOATR.

#### **Informacje pokrewne**

Skanowanie zintegrowanego systemu plików przy otwieraniu programu obsługi wyjścia

Skanowanie zintegrowanego systemu plików przy zamykaniu programu obsługi wyjścia

Funkcja API Zmiana sygnatury skanowania (Change Scan Signature - QP0LCHSG)

#### **Inny CCSID:**

Dostęp do obiektu z innym identyfikatorem kodowanego zestawu znaków (CCSID), niż poprzednio skanowany, spowoduje uruchomienie skanowania.

Przykładem takiej sytuacji jest otwarcie pliku z danymi w CCSID 819 w stronie kodowej CCSID 1200 i pomyślne przeprowadzenie operacji skanowania. Tak długo, jak dane pliku nie zostaną zmienione, jak i po każdym otwarciu pliku w CCSID 1200 skanowanie nie będzie uruchamiane. Jeśli jednak plik zostanie otwarty w innym CCSID, na przykład 37, skanowanie zostanie uruchomione dla tego identyfikatora CCSID 37. Gdy skanowanie to także będzie pomyślne, kolejne próby dostępu z CCSID 1200 i 37 nie spowodują uruchomienia skanowania.

Przechowywane są tylko dwa identyfikatory CCSID i jeden wskaźnik binarny w celu zminimalizowania ilości danych w systemie. Jeśli często uzyskiwany jest dostęp do tego samego obiektu w wielu różnych trybach kodowania CCSID, skanowanie to może wywołać znaczące zwiększenie dodatkowych operacji skanowania.

#### **Podczas operacji składowania:**

Jest to kolejny przykład sytuacji, w której może wystąpić skanowanie. System może zażądać skanowania w trakcie składowania obiektu.

Komenda SAV zawiera teraz parametr SCAN, który umożliwia podanie, czy pliki powinny być skanowane, gdy są składowane. Można także zażądać wyłączenia składowania, gdy skanowanie nie powiodło się dla pliku w trakcie poprzedniej operacji składowania. Spowoduje to, że pliki, które nie przeszły pomyślnie skanowania, nie zostaną umieszczone na nośniku i przeniesione do innego systemu.

**Uwaga:** Nie oznacza to, że gdy obiekt jest odtwarzany, zostanie oznaczony jako przeskanowany. Za każdym razem, gdy obiekt jest odtwarzany, kasowana jest cała historia statusu skanowania.

#### **Informacje pokrewne**

Komenda Składowanie obiektu (Save Object - SAV)

#### **Sprawdzanie integralności obiektu:**

Skanowanie może być także uruchomione, gdy w komendzie Sprawdzenie integralności obiektu (Check Object Integrity - CHKOBJITG) podany zostanie parametr SCANFS z wartością \*YES.

Opcja ta przydaje się do sprawdzenia poprawności pliku bez jego otwierania. Jeśli podany zostanie parametr SCANFS (\*STATUS), wszystkie obiekty, które nie przeszły pomyślnie poprzedniego skanowania otrzymają w protokole zapis o niepowodzeniu skanowania.

### Informacje pokrewne

Komenda Zmiana integralności obiektu (Change Object Integrity - CHGOBJITG)

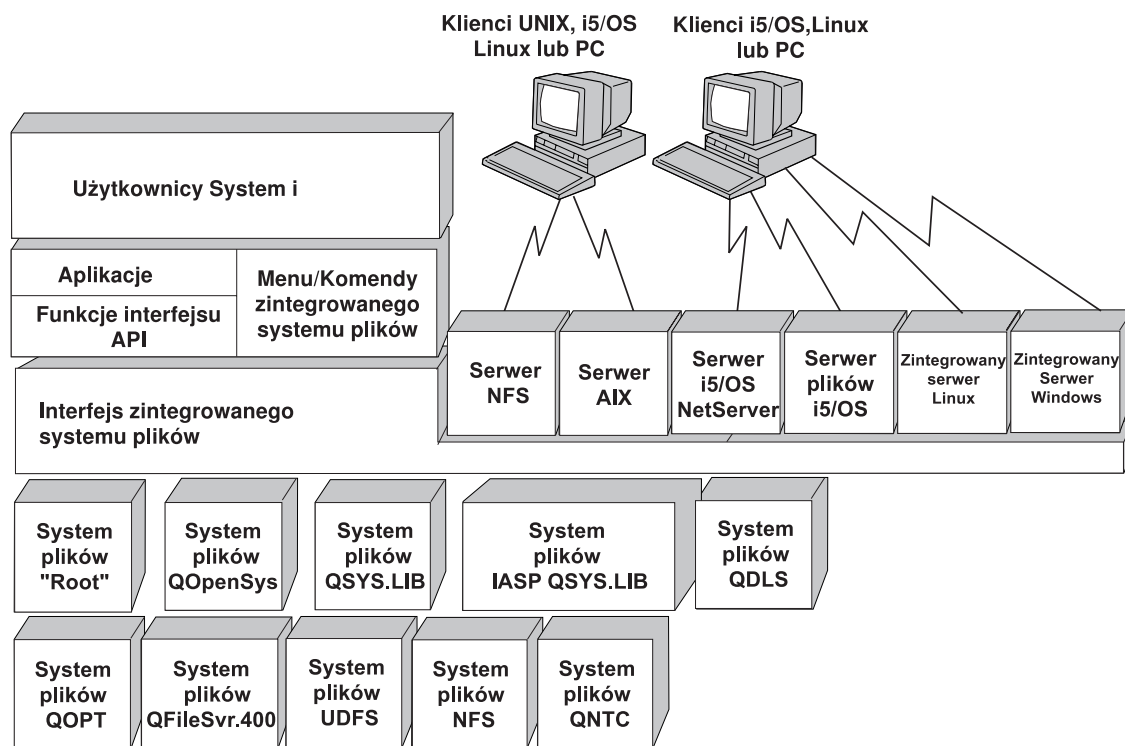
## Systemy plików

System plików umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

Każdy system plików ma zestaw struktur logicznych i reguł wykorzystywanych do pracy z informacjami zawartymi w pamięci. Te struktury i reguły mogą być różne dla różnych systemów plików. W rzeczywistości, z punktu widzenia struktur i reguł, obsługiwane przez i5/OS dostępu do zbiorów baz danych i obiektów innych typów poprzez biblioteki może być traktowane jako system plików. Podobnie obsługiwane przez i5/OS dostępu do dokumentów (będących w rzeczywistości plikami strumieniowymi) poprzez strukturę folderów może być traktowane jak oddzielny system plików.

W rzeczywistości zintegrowany system plików traktuje obsługę bibliotek i folderów jak oddzielne systemy plików. Inne rodzaje obsługi zarządzania plikami, mające inne cechy i możliwości, są także traktowane jak oddzielne systemy plików.

Użytkownik może współdziałać z dowolnym systemem plików poprzez wspólny interfejs. Interfejs ten jest zoptymalizowany do wprowadzania i pobierania danych strumieniowych, w odróżnieniu od wprowadzania i pobierania rekordów obsługiwanego przez interfejsy zarządzania danymi. Do współdziałania z systemami plików poprzez ten wspólny interfejs służą komendy, menu i ekrany oraz interfejsy API.



Rysunek 9. Systemy plików, serwery plików i interfejs zintegrowanego systemu plików

## Używanie sieciowego systemu plików poprzez interfejs zintegrowanego systemu plików

Dostęp do sieciowego systemu plików (Network File System - NFS) możliwy jest przez interfejs zintegrowanego systemu plików. Korzystając z sieciowego systemu plików należy pamiętać o powyższych uwagach i ograniczeniach.

### Informacje pokrewne

Pamięć optyczna



Obsługa sieciowego systemu plików i5/OS - plik PDF

Planowanie ochrony zintegrowanego systemu plików

Planowanie ochrony zintegrowanego systemu plików

## Porównanie systemów plików

Poniższe tabele zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

Tabela 2. Podsumowanie systemów plików (część 1 z 2)

Cecha	"root" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
Standardowa część i5/OS	Tak	Tak	Tak	Tak	Tak
Typ zbioru	Strumieniowy	Strumieniowy	Rekord <sup>12</sup>	Strumieniowy	Strumieniowy
Limit wielkości zbioru	T2=1 TB; T1=128 GB	T2=1 TB; T1=128 GB	Rozmiary zbiorów baz danych	4 GB	Różne <sup>17</sup>
Dostęp poprzez serwer plików i5/OS	Tak	Tak	Tak	Tak	Tak
Bezpośredni dostęp poprzez procesor we/wy serwera plików <sup>1</sup>	Nie	Nie	Nie	Nie	Tak
Względna szybkość operacji otwarcia/zamknięcia	Średnia <sup>2</sup>	Średnia <sup>2</sup>	Niska <sup>2</sup>	Niska <sup>2</sup>	Średnia <sup>2</sup>
Wyszukiwanie nazw z rozróżnianiem małych i wielkich liter	Nie	Tak	Nie <sup>4</sup>	Nie <sup>5</sup>	Nie
Maksymalna długość każdego składnika ścieżki	255 znaków <sup>3</sup>	255 znaków <sup>3</sup>	10/6 znaków <sup>6</sup>	8/3 znaki <sup>7</sup>	255 znaków <sup>3</sup>
Maksymalna długość ścieżki <sup>8</sup>	16 MB	16 MB	55–66 znaków <sup>4</sup>	82 znaki	255 znaków
Maksymalna długość atrybutów rozszerzonych obiektu	2 GB	2 GB	Zmienia się <sup>9</sup>	32 kB	0 <sup>18</sup>
Maksymalna liczba poziomów hierarchii katalogów w systemie plików	Nieograniczona <sup>10</sup>	Nieograniczona <sup>10</sup>	3	32	127
Maksymalna liczba dowiązań do jednego obiektu <sup>11</sup>	Zmienia się <sup>15</sup>	Zmienia się <sup>15</sup>	1	1	1
Obsługa dowiązań symbolicznych	Tak	Tak	Nie	Nie	Nie
Obiekt lub zbiór może mieć właściciela	Tak	Tak	Tak	Tak	Nie
Obsługa komend zintegrowanego systemu plików	Tak	Tak	Tak	Tak	Tak
Obsługa funkcji API zintegrowanego systemu plików	Tak	Tak	Tak	Tak	Tak
Obsługa funkcji API hierarchicznego systemu plików (HFS)	Nie	Nie	Nie	Tak	Nie
Ochrona wątków <sup>13</sup>	Tak	Tak	Tak	Nie	Tak
Obsługa kronikowania obiektów	Tak	Tak	Yes <sup>14</sup>	Nie	Nie



Tabela 2. Podsumowanie systemów plików (część 1 z 2) (kontynuacja)

Cecha	"root" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
<b>Uwagi:</b>					
1. Sprzęt procesor we/wy serwera plików używany jest przez Serwer LAN.					
2. Parametr Szybkość dotyczy sytuacji, gdy dostęp do systemu plików jest realizowany za pośrednictwem serwera plików i5/OS.					
3. W przypadku niektórych wartości CCSID maksymalna długość może wynosić mniej niż 255 znaków.					
4. Maksymalna długość ścieżki w systemie plików QSYS.LIB jest równa 55 znaków. Maksymalna długość ścieżki w systemie niezależnych plików QSYS.LIB jest równa 66 znaków.					
5. Szczegóły zawiera temat "System plików usług biblioteki dokumentów (QDLS)" na stronie 45.					
6. Ta wartość może mieć maksymalnie 10 znaków nazwy obiektu i 6 znaków typu obiektu.					
7. Ta wartość może mieć maksymalnie 8 znaków nazwy i od 1 do 3 znaków rozszerzenia określającego typ pliku (jeśli ma występować).					
8. Wartości opierają się na założeniu, że ścieżka bezwzględna rozpoczyna się od znaku /, po którym następuje nazwa systemu plików (taka jak /QDLS...).					
9. Systemy plików QSYS.LIB i QSYS.LIB niezależnej puli ASP obsługują trzy predefiniowane atrybuty rozszerzone: .SUBJECT, TYPE i CODEPAGE. Maksymalna długość jest określana przez złożenie długości tych trzech atrybutów rozszerzonych.					
10. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu.					
11. Wyjątkiem od tej reguły jest katalog, który może zawierać tylko jedno dowiązanie do innego katalogu.					
12. Obszary użytkowników w systemie plików SYS.LIB i niezależnej puli ASP obsługują operacje wejścia/wyjścia plików strumieniowych.					
13. W przypadku, gdy operacja jest kierowana do obiektu w systemie plików z ochroną wątków, funkcje API zintegrowanego systemu plików zapewniają taką ochronę. Jeśli działają one na obiektach w systemach plików bez ochrony wątków, po uruchomieniu kilku wątków w jednym zadaniu funkcje te kończą się niepomyślnie.					
14. Systemy plików QSYS.LIB i niezależnej puli ASP obsługują kronikowanie innych typów obiektów niż obiekty systemów plików "root" (/), UDFS, i QOpenSys.					
15. Katalogi w formacie *TYPE2 mają ograniczenie miliona dowiązań dla jednego obiektu i 999 998 podkatalogów. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt.					
16. Dane w tej kolumnie odnoszą się zarówno do systemu plików QSYS.LIB jak i do systemu plików QSYS.LIB niezależnej puli ASP.					
17. Ten limit zależy od systemu, do którego uzyskiwany jest dostęp.					
18. QNTC nie obsługuje atrybutów rozszerzonych.					
<b>Skróty</b>					
• T1 = *TYPE1 *STMF					
• T2 = *TYPE2 *STMF					
• B = bajty KB = kilobajty MB = megabajty GB = gigabajty TB = terabajty					

Tabela 3. Podsumowanie systemów plików (część 2 z 2)

Cecha	QOPT	QFileSvr.400	UDFS	NFS
Standardowa część i5/OS	Tak	Tak	Tak	Tak
Typ zbioru	Strumieniowy	Strumieniowy	Strumieniowy	Strumieniowy
Limit wielkości zbioru	4 GB	Zmienny <sup>3</sup>	T2 = 1 TB; T1=128 GB	Zmienny <sup>14</sup>
Dostęp poprzez serwer plików i5/OS	Tak	Tak	Tak	Tak
Bezpośredni dostęp poprzez procesor we/wy serwera plików <sup>1</sup>	Nie	Nie	Nie	Nie
Względna szybkość operacji otwarcia/zamknięcia	Niska	Niska <sup>2</sup>	Średnia <sup>2</sup>	Średnia <sup>2</sup>

Tabela 3. Podsumowanie systemów plików (część 2 z 2) (kontynuacja)

Cecha	QOPT	QFileSvr.400	UDFS	NFS
Wyszukiwanie nazw z rozróżnieniem małych i wielkich liter	Nie	Nie <sup>2</sup>	Tak <sup>11</sup>	Zmienia się <sup>2</sup>
Maksymalna długość każdego składnika ścieżki	Zmienia się <sup>4</sup>	Zmienia się <sup>2</sup>	255 znaków <sup>15</sup>	Zmienia się <sup>2</sup>
Maksymalna długość nazwy ścieżki <sup>5</sup>	294 znaki	Nieograniczona <sup>2</sup>	16 MB	Nieograniczona <sup>2</sup>
Maksymalna długość atrybutów rozszerzonych obiektu	8 MB	0 <sup>6</sup>	2 GB <sup>10</sup>	0 <sup>6</sup>
Maksymalna liczba poziomów hierarchii katalogów w systemie plików	Nieograniczona <sup>7</sup>	Nieograniczona <sup>2</sup>	Nieograniczona <sup>7</sup>	Nieograniczona <sup>2</sup>
Maksymalna liczba dowiązań do jednego obiektu <sup>8</sup>	1	1	Zmienny <sup>13</sup>	Zmienia się <sup>2</sup>
Obsługa dowiązań symbolicznych	Nie	Nie	Tak	Tak <sup>2</sup>
Obiekt lub zbiór może mieć właściciela	Nie	Nie <sup>9</sup>	Tak	Tak <sup>2</sup>
Obsługa komend zintegrowanego systemu plików	Tak	Tak	Tak	Tak
Obsługa funkcji API zintegrowanego systemu plików	Tak	Tak	Tak	Tak
Obsługa funkcji API hierarchicznego systemu plików (HFS)	Tak	Nie	Nie	Nie <sup>2</sup>
Wątkowo bezpieczne <sup>12</sup>	Tak	Tak	Tak	Tak
Obsługa kronikowania obiektów	Nie	Nie	Tak	Nie

Tabela 3. Podsumowanie systemów plików (część 2 z 2) (kontynuacja)

Cecha	QOPT	QFileSvr.400	UDFS	NFS
<b>Uwagi:</b>				
1. Sprzęt procesor we/wy serwera plików używany jest przez Serwer LAN.				
2. Wartość zależy od tego, który zdalny system plików jest dostępny.				
3. Przy połączeniu z systemem wcześniejszym niż V6R1 limitem wielkości zbioru jest 2 GB-1. W przeciwnym razie limit wielkości zbioru zależy od systemu plików, do którego uzyskiwany jest dostęp.				
4. Szczegóły zawiera temat "System plików nośników optycznych (QOPT)" na stronie 47.				
5. Wartości opierają się na założeniu, że bezwzględna nazwa ścieżki rozpoczyna się od znaku /, po którym następuje nazwa systemu plików.				
6. System plików QFileSvr.400 nie zwraca atrybutów rozszerzonych, nawet jeśli używany właśnie system plików je obsługuje.				
7. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu.				
8. Wyjątkiem od tej reguły jest katalog, który może zawierać tylko jedno dowiązanie do innego katalogu.				
9. Dostępny system plików może obsługiwać prawa własności do obiektów.				
10. Maksymalna długość atrybutów rozszerzonych całego UDFS nie może przekroczyć 40 B.				
11. Rozróżnianie wielkości znaków może zostać określone przy tworzeniu systemu UDFS. Jeśli w trakcie tworzenia UDFS zostanie użyty parametr *MIXED, spowoduje to rozróżnianie wielkości liter.				
12. Funkcje API zintegrowanego systemu plików zapewniają ochronę wątków wtedy, gdy są używane z poziomu procesu z możliwością obsługi wielowątkowości. System plików nie zezwala na dostęp do systemów plików bez ochrony wątków.				
13. Katalogi *TYPE2 posiadają limit miliona dowiązań na obiekt. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt.				
14. Ten limit zależy od systemu, do którego uzyskiwany jest dostęp.				
15. W przypadku niektórych wartości CCSID maksymalna długość może wynosić mniej niż 255 znaków.				
<b>Skróty</b>				
• T1 = *TYPE1 *STMF				
• T2 = *TYPE2 *STMF				
• B = bajty KB = kilobajty MB = megabajty GB = gigabajty TB = terabajty				

### Odsyłacze pokrewne

"System plików "root" (/)" na stronie 28

Bazowy system plików ("root", /) wykorzystuje w pełni możliwość obsługi plików strumieniowych i hierarchiczną strukturę katalogów zintegrowanego systemu plików. Ma cechy systemów plików DOS i OS/2.

"System plików systemów otwartych (QOpenSys)" na stronie 30

System plików QOpenSys jest zgodny ze standardami systemów otwartych bazującymi na rodzinie systemów UNIX, takimi jak POSIX i XPG (X/Open Portability Guide). Podobnie jak główny system plików "root" (/), ten system plików wykorzystuje oferowaną przez zintegrowany system plików obsługę plików strumieniowych i katalogów.

"Systemy plików użytkownika (UDFS)" na stronie 32

System plików użytkownika (UDFS) znajduje się w puli pamięci dyskowej (ASP) lub w puli niezależnej (IASP) wybranej przez użytkownika. Systemy plików tego typu są tworzone i zarządzane przez użytkownika.

"System plików bibliotek (QSYS.LIB)" na stronie 39

System plików QSYS.LIB obsługuje strukturę bibliotek i5/OS.

"Niezależne pule ASP systemu QSYS.LIB" na stronie 42

System plików QSYS.LIB niezależnej ASP obsługuje strukturę biblioteki i5/OS w niezależnej puli pamięci dyskowej ASP utworzonej i zdefiniowanej przez użytkownika. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów i5/OS obsługiwanych przez biblioteki w systemowej puli ASP i w niezależnej puli ASP.

"System plików usług biblioteki dokumentów (QDLS)" na stronie 45

System plików QDLS obsługuje strukturę folderów. Umożliwia on dostęp do dokumentów i folderów.

"System plików nośników optycznych (QOPT)" na stronie 47

System plików QOPT umożliwia dostęp do danych strumieniowych przechowywanych na nośnikach optycznych.

“System plików i5/OS NetClient (QNTC)” na stronie 50

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych na serwerze Integrated xSeries Server (IXS) z systemem operacyjnym Windows NT 4.0 lub późniejszym bądź systemem operacyjnym Linux. System plików QNTC zapewnia również dostęp do danych i obiektów przechowywanych na serwerach zdalnych z systemem Windows NT 4.0 lub późniejszym, systemem Linux z oprogramowaniem Samba 3.0 lub późniejszym, bądź obsługiwanymi wersjami serwera i5/OS NetServer.

“System plików serwera plików i5/OS (QFileSvr.400)” na stronie 54

System plików QFileSvr.400 zapewnia przezroczysty dostęp do innych systemów plików, znajdujących się na zdalnych platformach System i. Dostęp odbywa się poprzez hierarchiczną strukturę katalogów.

“Sieciowy system plików (Network File System - NFS)” na stronie 58

Sieciowy system plików (Network File System - NFS) daje użytkownikom dostęp do danych i obiektów przechowywanych na zdalnym serwerze obsługującym NFS.

### **Informacje pokrewne**

Zarządzanie kronikami

## **System plików "root" (/)**

Bazowy system plików ("root", /) wykorzystuje w pełni możliwość obsługi plików strumieniowych i hierarchiczną strukturę katalogów zintegrowanego systemu plików. Ma cechy systemów plików DOS i OS/2.

Dodatkowo, system ten:

- jest zoptymalizowany do obsługi danych wejściowych i wyjściowych plików strumieniowych,
- obsługuje wiele dowiązań stałych i symbolicznych,
- obsługuje gniazda lokalne,
- obsługuje bezpieczne dla wątków funkcje API,
- obsługuje obiekty \*FIFO,
- obsługuje obiekty /dev/null i /dev/zero \*CHRSF a także obiekty \*CHRSF,
- obsługuje kronikowanie zmian obiektów,
- obsługuje skanowanie obiektów z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików.

Bazowy system plików "root" (/) obsługuje znakowe pliki specjalne (\*CHRSF) o nazwach /dev/null oraz /dev/zero. Znakowe pliki specjalne są powiązane z urządzeniami lub zasobami systemu komputerowego. Mają one nazwy ścieżek wyświetlane w katalogach i taką samą ochronę dostępu, jak zwykłe pliki. Specjalne pliki znakowe /dev/null i /dev/zero są zawsze puste, a wszelkie dane zapisywane w plikach /dev/null i /dev/zero są usuwane. Pliki /dev/null i /dev/zero są obiektami typu \*CHRSF i mogą być wykorzystywane jak zwykajne pliki, z wyjątkiem tego, że dane z pliku /dev/null nie są nigdy odczytywane, a plik /dev/zero zawsze zwraca wartość pomyślną z danymi ustawionymi na zero.

## **Dostęp do systemu plików "root" (/)**

Do systemu plików "root" (/) można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

## **Rozróżnianie wielkości znaków w systemie plików "root" (/)**

System plików zachowuje małe i wielkie litery we wprowadzanych nazwach obiektów, lecz nie rozróżnia wielkości liter podczas wyszukiwania nazwy.

## **Nazwy ścieżek w systemie plików "root" (/)**

Nazwy ścieżek w systemie plików "root" (/) mają określoną postać.

/Katalog/Katalog . . . /Obiekt

- Każdy komponent nazwy ścieżki może zawierać do 255 znaków, znacznie więcej niż w systemach plików QSYS.LIB lub QDLS. Pełna nazwa ścieżki może być bardzo długa i zawierać do 16 megabajtów znaków.

- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia przestrzeni programów i systemu.
- Podczas zapisywania nazwy składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów \*TYPE1) i UTF-16 (dla katalogów \*TYPE2).

#### Pojęcia pokrewne

“Niezmienność nazw” na stronie 16

Użytkownicy systemu plików “root” (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmiennosc znaków w nazwach obiektów.

“Katalogi \*TYPE2” na stronie 10

Bazowy system plików “root” (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

“Nazwa ścieżki” na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

### Dowiązania w systemie plików “root” (/)

W systemie plików “root” (/) dozwolonych jest wiele dowiązań stałych do jednego obiektu. W pełni obsługiwane są dowiązania symboliczne.

Można używać dowiązań symbolicznych z bazowego systemu plików “root” (/) do obiektów w innych systemach, takich jak QSYS.LIB, QSYS.LIB niezależnej puli ASP lub QDLS.

#### Pojęcia pokrewne

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

### Używanie komend zintegrowanego systemu plików w systemie plików “root” (/)

Wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL oraz wszystkie ekrany opisane w temacie Dostęp za pomocą menu i ekranów mogą być uruchamiane w systemie plików “root” (/). Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

#### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 61

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych w systemie.

#### Odsyłacze pokrewne

“Dostęp przy użyciu komend języka CL” na stronie 63

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także przeprowadzić, wpisując komendy języka CL. Przy ich użyciu można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

### Używanie funkcji API zintegrowanego systemu plików w głównym systemie plików (/)

Wszystkie funkcje API wymienione w temacie Używanie interfejsów API mogą być uruchamiane w głównym systemie plików (“root” - /).

#### Odsyłacze pokrewne

“Wykonywanie operacji za pomocą funkcji API” na stronie 106

Wiele interfejsów API, które wykonują operacje na obiektach zintegrowanego systemu plików, ma postać funkcji napisanych w języku C.

#### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Kronikowanie zmian obiektów w systemie plików "root" (/)

- l Obiekty niektórych typów mogą być w systemie plików "root" (/) kronikowane. Umożliwia to odtworzenie zmian
- l obiektu dokonanych od czasu ostatniego składowania tego obiektu.

### Pojęcia pokrewne

"Kronikowanie obiektów" na stronie 92

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Dodatkowo, główną funkcją kronikowania jest asysta w powielaniu zmian obiektów w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia.

## Urządzenia UDP i TCP w systemie plików "root" (/)

System plików "root" (/) w katalogu /dev/xti zawiera teraz dwa sterowniki urządzeń o nazwie udp i tcp.

Oba sterowniki są znakowymi plikami specjalnymi (\*CHRFS) i są tworzone w trakcie ładowania programu początkowego (IPL). Sterowniki protokołów UDP (User Datagram Protocol) i TCP (Transmission Control Protocol) służą do otwierania połączenia z dostawcami transportów UDP i TCP. Oba sterowniki będą urządzeniami użytkownika i otrzymają nowy numer główny urządzenia. Będą też posiadać operacje otwarcia z klonowaniem, co oznacza, że każda operacja otwarcia otrzyma unikalną instancję urządzenia. Korzystanie z takich urządzeń jest możliwe tylko w środowisku PASE systemu i5/OS. Tabela wyświetlona poniżej zawiera obiekty, które będą utworzone i ich właściwości.

Tabela 4. Obiekty sterowników urządzeń i ich właściwości

Nazwa ścieżki	Typ	Główny	Poboczny	Właściciel	Uprawnienia do danych właściciela	Grupa	Grupowe uprawnienia do danych	Publiczne uprawnienia do danych
/dev/xti	*DIR	N/A	N/A	QSYS	*RWX	Brak	*RX	*RX
/dev/xti/tcp	*CHRFS	Klon	TCP	QSYS	*RW	Brak	*RW	*RW
/dev/xti/udp	*CHRFS	Klon	UDP	QSYS	*RW	Brak	*RW	*RW

### Informacje pokrewne

i5/OS PASE

## System plików systemów otwartych (QOpenSys)

System plików QOpenSys jest zgodny ze standardami systemów otwartych bazującymi na rodzinie systemów UNIX, takimi jak POSIX i XPG (X/Open Portability Guide). Podobnie jak główny system plików "root" (/), ten system plików wykorzystuje oferowaną przez zintegrowany system plików obsługę plików strumieniowych i katalogów.

Dodatkowo, system ten:

- jest dostępny poprzez hierarchiczną strukturę katalogów, podobną do struktury systemów UNIX,
- jest zoptymalizowany do obsługi danych wejściowych i wyjściowych plików strumieniowych,
- obsługuje wiele dowiązań stałych i symbolicznych,
- obsługuje nazwy z rozróżnieniem małych i wielkich liter,
- obsługuje gniazda lokalne,
- obsługuje bezpieczne dla wątków funkcje API,
- obsługuje obiekty \*FIFO,
- obsługuje kronikowanie zmian obiektów,
- obsługuje skanowanie obiektów z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików.

System plików QOpenSys ma te same cechy, co system plików "root" (/), z tym tylko, że rozróżnia on wielkość liter w celu umożliwienia obsługi standardów systemów otwartych opartych na standardach systemów UNIX.

## Dostęp do systemu plików QOpenSys

Do systemu plików QOpenSys można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

## Rozróżnianie wielkości znaków w systemie plików QOpenSys

W odróżnieniu od systemu plików "root" (/), system plików QOpenSys rozróżnia małe i wielkie litery podczas wyszukiwania nazw obiektów.

Na przykład ciąg znaków podany wielkimi literami nie zostanie uznany za identyczny z tym samym ciągiem, w którym jakkolwiek litera jest mała.

Rozróżnianie małych i wielkich liter pozwala używać powtarzających się nazw, o ile poszczególne nazwy różnią się wielkością liter. W systemie plików QOpenSys w tym samym katalogu można przechowywać obiekt o nazwie Payroll, obiekt o nazwie PayRoll i obiekt o nazwie PAYROLL.

## Nazwy ścieżek w systemie plików QOpenSys

Nazwy ścieżek w systemie plików QOpenSys mają określoną postać.

/QOpenSys/Katalog/Katalog/ . . . /Obiekt

- Każdy składnik ścieżki może mieć do 255 znaków. Pełna ścieżka może mieć do 16 MB długości.
- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia przestrzeni programów i systemu.
- Podczas zapisywania nazwy składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów \*TYPE1) i UTF-16 (dla katalogów \*TYPE2).

### Pojęcia pokrewne

"Niezmienność nazw" na stronie 16

Użytkownicy systemu plików "root" (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmienną znaków w nazwach obiektów.

"Katalogi \*TYPE2" na stronie 10

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

"Nazwa ścieżki" na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

## Dowiązania w systemie plików QOpenSys

W systemie plików QOpenSys dozwolonych jest wiele dowiązań stałych do jednego obiektu. W pełni obsługiwane są dowiązania symboliczne.

Można używać dowiązań symbolicznych z systemu plików QOpenSys do obiektów w innych systemach plików.

### Pojęcia pokrewne

"Dowiązanie" na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QOpenSys

Wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL oraz wszystkie ekrany opisane w temacie Dostęp za pomocą menu i ekranów mogą być używane w systemie plików QOpenSys. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 61

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych w systemie.

#### **Odsyłacze pokrewne**

“Dostęp przy użyciu komend języka CL” na stronie 63

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także przeprowadzić, wpisując komendy języka CL. Przy ich użyciu można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

## **Używanie funkcji API zintegrowanego systemu plików w systemie plików QOpenSys**

Wszystkie funkcje API wymienione w temacie Wykonywanie operacji przy użyciu funkcji API mogą być uruchamiane w systemie plików QOpenSys.

#### **Odsyłacze pokrewne**

“Wykonywanie operacji za pomocą funkcji API” na stronie 106

Wiele interfejsów API, które wykonują operacje na obiektach zintegrowanego systemu plików, ma postać funkcji napisanych w języku C.

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **Kronikowanie zmian obiektów w systemie plików QOpenSys**

- | Obiekty niektórych typów mogą być w systemie plików QOpenSys kronikowane. Umożliwia to odtworzenie zmian
- | obiektu dokonanych od czasu ostatniego składowania tego obiektu.

#### **Pojęcia pokrewne**

“Kronikowanie obiektów” na stronie 92

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Dodatkowo, główną funkcją kronikowania jest asysta w powielaniu zmian obiektów w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia.

## **Systemy plików użytkownika (UDFS)**

System plików użytkownika (UDFS) znajduje się w puli pamięci dyskowej (ASP) lub w puli niezależnej (IASP) wybranej przez użytkownika. Systemy plików tego typu są tworzone i zarządzane przez użytkownika.

Dodatkowo system ten:

- zawiera hierarchiczną strukturę katalogów podobną do struktury systemów operacyjnych komputerów PC, takich jak DOS i OS/2;
- jest zoptymalizowany do obsługi danych wejściowych i wyjściowych plików strumieniowych;
- obsługuje wiele dowiązań stałych i symbolicznych;
- obsługuje gniazda lokalne;
- obsługuje funkcje API z ochroną wątków;
- obsługuje obiekty \*FIFO;
- obsługuje kronikowanie zmian obiektów;
- obsługuje skanowanie obiektów z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików.

Użytkownik może tworzyć różne systemy plików UDFS nadając każdemu z nich inną nazwę. Użytkownik może podczas tworzenia UDFS określić także inne atrybuty, takie jak:

- numer puli ASP lub nazwę niezależnej puli ASP, w których przechowywane są obiekty systemu plików UDFS,
- rozróżnianie małych i wielkich liter w nazwach obiektów umiejscowionych w obrębie systemu plików UDFS.



Można w ten sposób określić, czy w trakcie wyszukiwania obiektu w obrębie systemu UDFS w nazwach obiektów będą rozróżniane małe i wielkie litery.

- atrybut skanowania podczas tworzenia obiektu, który definiuje atrybut skanowania dla obiektów utworzonych w systemie plików UDFS;
- wartości atrybutów ograniczone (restricted), zmiana nazwy (rename) i usunięcie dowiązania (unlink);
- | • wartość kontrolna dla obiektu tworzonego w systemie plików UDFS;
- | • różne formaty (\*TYPE1 i \*TYPE2) plików strumieniowych tworzonych w systemie plików UDFS;
- | • opcja pamięci dyskowej dla plików strumieniowych tworzonych w systemie plików UDFS;
- | • opcja pamięci głównej dla plików strumieniowych tworzonych w systemie plików UDFS.

## System plików użytkownika - pojęcia

W systemie plików użytkownika (UDFS), tak jak i w systemie “root” (/) oraz QOpenSys można tworzyć katalogi, pliki strumieniowe, dowiązania symboliczne, gniazda lokalne i obiekty \*FIFO.

UDFS jest reprezentowany przez pojedynczy obiekt: specjalny plik blokowy (\*BLKSF). W momencie utworzenia UDFS użytkownik również automatycznie tworzy specjalne pliki blokowe. Specjalny plik blokowy jest dostępny dla użytkowników poprzez komendy zintegrowanego systemu plików, funkcje API i przez interfejs QFileSvr.400.

- | **Uwaga:** Zmiana atrybutów lub uprawnień blokowego pliku specjalnego powoduje wprowadzenie równoważnych
- | zmian w katalogu głównym systemu plików UDFS. Zależność ta zachodzi również w drugim kierunku.
- | Zmiany nie mają wpływu na inne obiekty w systemie plików UDFS.

System UDFS może znajdować się w dwóch stanach: **podłączonym (mounted)** lub **odłączonym (unmounted)**. Jeśli użytkownik podłączy (mount) UDFS, obiekty znajdujące się w nim będą dostępne. Jeśli użytkownik odłączy (unmount) UDFS, obiekty znajdujące się w nim będą niedostępne.

- | Aby uzyskać dostęp do obiektów w UDFS, należy podłączyć UDFS do katalogu (na przykład /home/JON). Po
- | podłączeniu UDFS do katalogu dotychczasowa zawartość tego katalogu, łącznie z podkatalogami i obiektami, staje się
- | niedostępna. Po podłączeniu UDFS jego zawartość staje się dostępna przez ścieżkę, która powstała po jego
- | podłączeniu. Na przykład katalog /home/JON zawiera plik /home/JON/payroll. UDFS zawiera 3 katalogi mail,
- | action i outgoing. Po podłączeniu UDFS do katalogu /home/JON, plik /home/JON/payroll staje się niedostępny,
- | natomiast 3 katalogi UDFS staną się dostępne jako /home/JON/mail, /home/JON/action i /home/JON/outgoing.
- | Odłączenie UDFS spowoduje, że plik /home/JON/payroll stanie się znowu dostępny, natomiast trzy katalogi w
- | systemie UDFS staną się niedostępne.

Wykonanie ładowania programu początkowego (IPL) systemu lub wykonanie na katalogach operacji Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) spowoduje odłączenie wszystkich systemów plików UDFS. Dlatego muszą zostać ponownie podłączone po IPL lub uruchomieniu komendy RCLSTG na katalogach.

**Uwaga:** System UDFS znajdujący się w niezależnej puli ASP nie może być podłączony.

## Uzyskiwanie dostępu do systemów plików UDFS poprzez interfejs zintegrowanego systemu plików

Do systemu plików UDFS można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

### Pojęcia pokrewne

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

“Plik strumieniowy” na stronie 15

*Plik strumieniowy* jest dostępną losowo sekwencją bajtów, pozbawioną struktury narzucanej przez system.

### **Informacje pokrewne**

Komenda Tworzenie systemu plików użytkownika (Create User-Defined FS - CRTUDFS)

## **Rozróżnianie wielkości liter w systemie plików użytkownika w ramach zintegrowanego systemu plików**

Podczas tworzenia systemu plików użytkownika (UDFS) można określić, czy ma on rozróżniać wielkość liter w nazwach obiektów, czy też nie.

Jeśli zostanie wybrana opcja rozróżniania wielkości liter, system będzie podczas wyszukiwania nazw obiektów traktował wielkie i małe litery jako różne znaki. Na przykład nazwa podana wielkimi literami nie zostanie uznana za identyczną z tą samą nazwą, w której jakkolwiek litera jest mała. Dlatego /home/MAREK/ i /home/marek/ rozpoznane zostaną jako inne katalogi. Aby system UDFS rozróżniał wielkość liter, należy podczas użycia komendy CRTUDFS podać wartość \*MIXED dla parametru CASE.

Jeśli zostanie wybrany brak rozróżniania wielkości liter, system nie będzie rozróżniał wielkości liter podczas wyszukiwania nazw. System będzie zatem traktował /home/JUREK i /HOME/jurek jako nazwy tego samego katalogu, nie zaś dwóch różnych katalogów. Aby system plików UDFS nie rozróżniał wielkości liter, należy podczas wykonywania komendy CRTUDFS podać wartość \*MONO parametru CASE.

W obu przypadkach system plików zachowuje nazwy obiektów w takiej postaci, w jakiej zostały one podane przez użytkownika. Ustawienie rozróżniania wielkości liter dotyczy jedynie wyszukiwania przez użytkownika nazw w systemie.

### **Informacje pokrewne**

Komenda Tworzenie systemu plików użytkownika (Create User-Defined FS - CRTUDFS)

## **Nazwy ścieżek w systemie plików użytkownika w ramach zintegrowanego systemu plików**

Gdy konieczne jest operowanie całym systemem plików użytkownika (UDFS) wraz ze wszystkimi obiektami w nim zawartymi, system plików UDFS jest reprezentowany przez blokowy plik specjalny (\*BLKSF).

Jeśli system UDFS znajduje się w podstawowej puli ASP, nazwa specjalnego pliku blokowego musi mieć format:  
/dev/QASPXX/nazwa\_udfs.udfs

gdzie XX jest numerem puli ASP, w której użytkownik przechowuje UDFS, natomiast nazwa\_udfs jest unikalną nazwą UDFS zapisaną w tej ASP. Należy zwrócić uwagę, aby nazwa UDFS została podana z rozszerzeniem .udfs.

Jeśli system UDFS znajduje się w niezależnej puli ASP, nazwa specjalnego pliku blokowego musi mieć format:  
/dev/nazwa\_asp/nazwa\_udfs.udfs

gdzie nazwa\_asp to nazwa puli independent, w której znajduje się system plików UDFS a nazwa\_udfs to unikalna nazwa systemu UDFS w tej niezależnej puli ASP. Należy zwrócić uwagę, aby nazwa UDFS została podana z rozszerzeniem .udfs.

Nazwa ścieżki do obiektów wewnątrz UDFS zależy od katalogu, do którego został dołączony UDFS. Na przykład, jeśli dołączymy UDFS /dev/qasp01/wysocki.udfs do /home/dennis, to nazwy ścieżek do wszystkich obiektów wewnątrz UDFS będą się rozpoczynać od /home/dennis.

Dodatkowe reguły nazwa ścieżek:

- Każdy składnik ścieżki może mieć do 255 znaków. Pełna ścieżka może mieć do 16 MB długości.
- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu lub pamięci serwera.

- Podczas zapisywania nazwy składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów \*TYPE1) i UTF-16 (dla katalogów \*TYPE2).

#### Pojęcia pokrewne

“Niezmienność nazw” na stronie 16

Użytkownicy systemu plików “root” (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmiennosc znaków w nazwach obiektów.

“Katalogi \*TYPE2” na stronie 10

Bazowy system plików “root” (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

“Nazwa ścieżki” na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

## Dowiązania w systemie plików użytkownika zintegrowanego systemu plików

System plików użytkownika (UDFS) umożliwia tworzenie wielu dowiązań stałych do tego samego obiektu i w pełni obsługuje dowiązania symboliczne.

Można utworzyć dowiązanie symboliczne z UDFS do obiektu w innym systemie plików.

#### Pojęcia pokrewne

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Korzystanie z komend zintegrowanego systemu plików w systemie plików użytkownika

W systemie plików użytkownika mogą być używane wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL oraz wszystkie ekrany opisane w temacie Dostęp za pomocą menu i ekranów.

Istnieją jednak komendy CL przeznaczone specjalnie dla systemu plików UDFS i innych dołączanych systemów plików. Są one opisane w poniższej tabeli.

Tabela 5. Komendy CL systemu plików użytkownika

Komenda	Opis
ADDMFS	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.
CRTUDFS	Tworzenie systemu plików UDFS (Create UDFS). Tworzy system plików użytkownika.
DLTUDFS	Usunięcie systemu plików UDFS (Delete UDFS). Usuwa system plików użytkownika.
DSPMFSINF	Wyświetlenie danych podłączonego systemu plików (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.
DSPUDFS	Wyświetlenie systemu plików UDFS (Display UDFS). Wyświetla informacje o systemie plików użytkownika.
MOUNT	Podłączenie systemu plików. Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. Jest to alternatywna postać komendy ADDMFS.
RMVMFS	Usuwanie podłączonego FSS (Remove Mounted File System). Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego.
UNMOUNT	Odlączenie systemu plików. Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego. Jest to alternatywna postać komendy RMVMFS.

**Uwaga:** Należy podłączyć UDFS, aby funkcje API systemu plików mogły wykonywać operacje na obiektach przechowywanych w wybranym systemie plików UDFS.

#### **Zadania pokrewne**

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 61

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych w systemie.

#### **Odsyłacze pokrewne**

“Dostęp przy użyciu komend języka CL” na stronie 63

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także przeprowadzić, wpisując komendy języka CL. Przy ich użyciu można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

## **Korzystanie z funkcji API zintegrowanego systemu plików w systemie plików użytkownika**

Wszystkie funkcje API wymienione w temacie Wykonywanie operacji przy użyciu funkcji API mogą być uruchamiane w systemie plików użytkownika.

**Uwaga:** Należy podłączyć UDFS, aby funkcje API systemu plików mogły wykonywać operacje na obiektach przechowywanych w wybranym systemie plików UDFS.

#### **Odsyłacze pokrewne**

“Wykonywanie operacji za pomocą funkcji API” na stronie 106

Wiele interfejsów API, które wykonują operacje na obiektach zintegrowanego systemu plików, ma postać funkcji napisanych w języku C.

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **Graficzny interfejs dostępu do systemu plików użytkownika**

Uruchamiany na komputerze PC program System i Navigator oferuje graficzny interfejs użytkownika, zapewniający łatwy i wygodny dostęp do systemów plików UDFS.

Interfejs ten umożliwi tworzenie, usuwanie, wyświetlanie, podłączenia i odłączanie systemu plików UDFS z klienta w systemie Windows.

Program System i Navigator umożliwi wykonywanie operacji na systemie plików UDFS. Podstawowe zadania opisuje temat:

- “Tworzenie nowego systemu plików użytkownika” na stronie 127
- “Podłączanie systemu plików utworzonego przez użytkownika” na stronie 127
- “Odłączanie systemu plików użytkownika” na stronie 128

## **Tworzenie systemu plików użytkownika (UDFS) zintegrowanego systemu plików**

Komenda Utworzenie systemu plików użytkownika ( Create User-Defined File System - CRTUDFS) tworzy system plików, który może być udostępniony przy użyciu przestrzeni nazw zintegrowanego systemu plików, funkcji API i komend języka CL.

Komendy ADDMFS lub MOUNT umieszczają UDFS ponad istniejącym lokalnie katalogiem. Użytkownik może stworzyć UDFS w dowolnej puli ASP lub niezależnej puli ASP.

Dla systemu plików UDFS można także określić następujące pozycje:

- numer puli ASP lub nazwę niezależnej puli ASP, w których przechowywane są obiekty systemu plików UDFS;
- rozróżnianie małych i wielkich liter w nazwach obiektów umiejscowionych w obrębie systemu plików UDFS;

Można w ten sposób określić, czy w trakcie wyszukiwania obiektu w obrębie systemu UDFS w nazwach obiektów będą rozróżniane małe i wielkie litery.

- atrybut skanowania podczas tworzenia obiektu, który definiuje atrybut skanowania dla obiektów utworzonych w systemie plików UDFS;
- wartości atrybutu ograniczone (restricted), zmiana nazwy (rename) i usunięcie dowiązania (unlink);
- | • wartość kontrolna dla obiektu tworzonych w systemie plików UDFS;
- | • różne formaty (\*TYPE1 i \*TYPE2) plików strumieniowych tworzonych w systemie plików UDFS;
- | • opcja pamięci dyskowej dla plików strumieniowych tworzonych w systemie plików UDFS;
- | • opcja pamięci głównej dla plików strumieniowych tworzonych w systemie plików UDFS.

#### **Informacje pokrewne**

Komenda Tworzenie systemu plików użytkownika (Create User-Defined FS - CRTUDFS)

Komenda Dodanie podłączonego FS (Add Mounted FS - ADDMFS)

### **Usuwanie zintegrowanego systemu plików użytkownika**

Komenda Usunięcie systemu plików użytkownika (Delete User-Defined File System - DLTUDFS) powoduje usunięcie istniejącego, niepodłączonego systemu UDFS i wszystkich obiektów w tym systemie plików.

Komenda ta nie zadziała, jeśli UDFS jest podłączony. Usunięcie UDFS spowoduje usunięcie wszystkich zawartych w nim obiektów. Jeśli użytkownik nie ma specjalnych uprawnień do usunięcia wszystkich obiektów w obrębie UDFS, to żaden z obiektów nie zostanie usunięty.

#### **Informacje pokrewne**

Komenda Usunięcie systemu plików UDFS (Delete User-Defined FS - DLTUDFS)

### **Wyświetlanie zintegrowanego systemu plików użytkownika**

Komenda Wyświetlenie systemu plików użytkownika (Display User-Defined File System - DSPUDFS) powoduje wyświetlenie atrybutów istniejącego systemu UDFS, bez względu na to, czy jest on podłączony, czy nie.

Komenda Wyświetlenie danych podłączonego FS (Display Mounted File System Information - DSPMFSINF) podaje również informacje o podłączonym systemie UDFS, a także o wszystkich podłączonych systemach plików.

#### **Informacje pokrewne**

Komenda Wyświetlenie systemu plików UDFS (Display User-Defined FS - DSPUDFS)

Komenda Wyświetlenie danych podłączonego FS (Display Mounted FS Information - DSPMFSINF)

### **Podłączanie systemu plików użytkownika (UDFS) zintegrowanego systemu plików**

Komendy Dodanie podłączonego systemu plików (Add Mounted File System - ADDMFS) i MOUNT powodują udostępnienie obiektów znajdujących się w systemie plików w przestrzeni nazw zintegrowanego systemu plików.

Aby dołączyć system plików (UDFS), należy w komendzie ADDMFS dla parametru TYPE podać wartość \*UDFS.

- | Wykonanie ładowania programu początkowego (IPL) systemu lub wykonanie na katalogach operacji Odzyskiwanie
- | pamięci (Reclaim Storage - RCLSTG) spowoduje odłączenie wszystkich systemów plików UDFS. Dlatego muszą
- | zostać ponownie podłączone po IPL lub uruchomieniu komendy RCLSTG na katalogach.

**Uwaga:** System UDFS znajdujący się w niezależnej puli ASP nie może być podłączony.

#### **Informacje pokrewne**

Komenda Dodanie podłączonego FS (Add Mounted FS - ADDMFS)

### **Odłączanie systemu plików użytkownika (UDFS) w zintegrowanym systemie plików**

Po wykonaniu komendy odłączenia (unmount), zawartość systemu plików użytkownika (UDFS) staje się niedostępna dla interfejsów zintegrowanego systemu plików.

- | **Uwaga:** Również wykonanie ładowania programu początkowego (IPL) systemu lub wykonanie na katalogach operacji
- | Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) spowoduje odłączenie wszystkich systemów plików
- | UDFS.

Obiekty znajdujące się w UDFS stają się niedostępne w momencie odłączenia UDFS. Komenda Usuwanie podłączonego systemu plików (Remove Mounted File System - RMVMFS) lub komenda UNMOUNT spowodują, że podłączone systemy plików będą niedostępne dla przestrzeni nazw zintegrowanego systemu plików. Jeśli któryś z obiektów w systemie plików jest używany (na przykład plik jest otwarty) w trakcie wykonywania komendy, to użytkownik otrzyma komunikat o błędzie. System UDFS pozostanie wówczas podłączony. Jeśli użytkownik podłączył inny UDFS, przesyłając którąkolwiek część starego UDFS, to starego UDFS nie można będzie odłączyć, dopóki nie zostanie on odsłonięty.

Na przykład użytkownik podłączył najpierw UDFS /dev/qasp02/jenn.udfs do /home/judy w przestrzeni nazw zintegrowanego systemu plików. Jeśli następnie użytkownik podłączył inny system plików /pubs do /home/judy, to zawartość jenn.udfs staje się niedostępna. Co więcej, nie można odłączyć jenn.udfs, dopóki nie zostanie odłączony drugi system plików z /home/judy.

**Uwaga:** System UDFS znajdujący się w niezależnej puli ASP nie może być podłączony.

#### **Informacje pokrewne**

Komenda Usuwanie podłączonego FS (Remove Mounted FS - RMVMFS)

## **Składowanie i odtwarzanie systemu plików użytkownika w ramach zintegrowanego systemu plików**

Użytkownik ma możliwość składowania i odtwarzania wszystkich obiektów systemu plików użytkownika (UDFS) wraz ze związanymi z nimi uprawnieniami.

Komenda Składowanie obiektu (Save Object - SAV) umożliwia składowanie obiektów systemu plików UDFS, natomiast komenda Odtworzenie obiektu (Restore Object - RST) umożliwia użytkownikowi odtworzenie obiektów systemu plików UDFS. Obydwie komendy funkcjonują bez względu na to, czy system plików UDFS jest dołączony, czy odłączony. Jednak aby poprawnie składować atrybuty UDFS, a nie tylko obiekty znajdujące się w UDFS, system plików powinien zostać odłączony.

#### **Informacje pokrewne**

Komenda Składowanie obiektu (Save Object - SAV)

Komenda Odtworzenie obiektu (Restore Object - RST)

## **Kronikowanie zmian obiektów w systemie plików użytkownika**

- | Możliwe jest kronikowanie niektórych typów obiektów w systemach plików użytkownika (UDFS). Umożliwia to
- | odtworzenie zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu.

#### **Pojęcia pokrewne**

“Kronikowanie obiektów” na stronie 92

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Dodatkowo, główną funkcją kronikowania jest asysta w powielaniu zmian obiektów w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia.

## **System plików użytkownika i niezależne pule pamięci dyskowej**

W chwili udostępnienia niezależnej puli pamięci dyskowej (ASP), w systemie plików "root" (/) wprowadzanych jest kilka zmian.

Są one następujące:

- W katalogu /dev tworzony jest katalog dla niezależnej ASP. Nazwa tego katalogu odpowiada nazwie opisu urządzenia powiązanego z ASP. Jeśli katalog ten istnieje przed zgłoszeniem podłączenia i nie jest pusty, operacja podłączenia będzie kontynuowana, ale praca z systemami plików UDFS w ASP nie będzie możliwa. Należy wówczas zablokować niezależną ASP i zmienić nazwę katalogu lub usunąć jego zawartość, a następnie ponowić zgłoszenie podłączenia.
- W katalogu /dev/nazwa\_asp powstanie obiekt specjalnego pliku blokowego przypisany do wszystkich UDFS, które znajdują się w niezależnej puli ASP. System zawsze utworzy domyślny system plików UDFS. Ścieżka do domyślnego specjalnego pliku blokowego UDFS: /dev/nazwa\_asp/QDEFAULT.UDFS

- Domyślny UDFS jest podłączany do katalogu /nazwa\_asp. Katalog /nazwa\_asp nie musi istnieć przed zgłoszeniem podłączenia. Jednak jeśli istnieje, musi być pusty. W przeciwnym przypadku pula ASP będzie udostępniona, ale domyślny system plików UDFS nie zostanie podłączony. Należy wówczas zmienić nazwę katalogu lub usunąć jego zawartość i zablokować, a następnie udostępnić pulę ASP lub użyć komendy MOUNT do podłączenia domyślnego systemu plików UDFS.
- Jeśli niezależna pula ASP jest podstawową lub dodatkową ASP, a domyślny system UDFS został pomyślnie podłączony, podłączony zostanie dodatkowy system plików. System plików QSYS.LIB niezależnej ASP zostanie podłączony w ścieżce /nazwa\_asp/QSYS.LIB.

**Uwaga:** Nie można podłączyć lub odłączyć tego systemu plików niezależnie od domyślnego systemu UDFS. Zostanie on zawsze podłączony lub odłączony automatycznie.

#### **Odsyłacze pokrewne**

“Niezależne pule ASP systemu QSYS.LIB” na stronie 42

System plików QSYS.LIB niezależnej ASP obsługuje strukturę biblioteki i5/OS w niezależnej puli pamięci dyskowej ASP utworzonej i zdefiniowanej przez użytkownika. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów i5/OS obsługiwanych przez biblioteki w systemowej puli ASP i w niezależnej puli ASP.

## **System plików bibliotek (QSYS.LIB)**

System plików QSYS.LIB obsługuje strukturę bibliotek i5/OS.

Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów i5/OS obsługiwanych przez biblioteki, które znajdują się w systemie i w podstawowych pulach ASP użytkowników.

Dodatkowo, system ten:

- obsługuje wszystkie interfejsy użytkownika i interfejsy programistyczne wykonujące działania na bibliotekach i5/OS i na obiektach w tych bibliotekach,
- obsługuje wszystkie języki programowania i wszystkie narzędzia wykonujące działania na zbiorach baz danych,
- oferuje wiele funkcji administracyjnych przeznaczonych do zarządzania obiektami i5/OS,
- obsługuje operacje strumienia wejścia/wyjścia na podzbiorach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

Przed wprowadzeniem zintegrowanego systemu plików w wersji 3 systemu OS/400, system plików QSYS.LIB był jedynym dostępnym systemem plików. Programiści używający do pisania aplikacji języków programowania takich jak RPG lub COBOL i narzędzi takich jak DDS korzystali z systemu plików QSYS.LIB. Operatorzy systemu używający komend, menu i ekranów do pracy z kolejkami wydruków korzystali z systemu plików QSYS.LIB, podobnie jak administratorzy systemu tworzący i zmieniający profile użytkowników.

Wszystkie te elementy i oparte na nich aplikacje działają tak, jak działały zanim wprowadzono zintegrowany system plików. Nie mają one jednak dostępu do systemu plików QSYS.LIB poprzez zintegrowany system plików.

## **Używanie systemu plików QSYS.LIB poprzez interfejs zintegrowanego system plików**

Do systemu plików QSYS.LIB można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend, ekranów użytkownika i funkcji API zintegrowanego systemu plików.

### **Lista autoryzacji QPWFSEVER w systemie plików QSYS.LIB**

Lista QPWFSEVER jest listą autoryzacji (typ obiektu \*AUTL). Określa ona dodatkowe wymagania dotyczące dostępu dla wszystkich obiektów w systemie plików QSYS.LIB, do których istnieje dostęp poprzez klientów zdalnych.

Uprawnienia podane w tej liście odnoszą się do wszystkich obiektów znajdujących się w systemie plików QSYS.LIB.

Domyślnym uprawnieniem do danego obiektu jest PUBLIC \*USE. Administrator może za pomocą komendy Edycja listy autoryzacji (Edit Authorization List - EDTAUTL) lub Praca z listami autoryzacji (Work With Authorization List - WRKAUTL) zmieniać wartość uprawnienia. Administrator może przypisać uprawnienie PUBLIC \*EXCLUDE do listy autoryzacji, aby zablokować ogólny dostęp do obiektów systemu plików QSYS.LIB ze zdalnych klientów.

## Ograniczenia w obsłudze plików w systemie plików QSYS.LIB

Poniżej opisane zostały niektóre ograniczenia w obsłudze plików w systemie plików QSYS.LIB.

- Zbiory logiczne nie są obsługiwane.
- Zbiorami fizycznymi obsługiwanymi w trybie tekstowym są zbiory fizyczne opisywane przez programy i zawierające tylko jedno pole oraz źródłowe zbiory fizyczne zawierające tylko jedno pole tekstowe. Zbiorami fizycznymi obsługiwanymi w trybie binarnym są, oprócz zbiorów fizycznych obsługiwanych w trybie tekstowym, opisane zewnętrznie zbiory fizyczne.
- Blokada na poziomie bajtów nie jest obsługiwana. Więcej informacji na temat blokowania zakresów bajtów znaleźć można w temacie fcntl() - Perform File Control Command.
- Jeśli jakiegokolwiek zadanie ma otwarty podzbiór bazy danych, tylko jedno zadanie może w danym momencie uzyskać dostęp do tego podzbioru, aby go zapisać. Inne zadania mogą uzyskać dostęp jedynie do odczytu.

## Obsługa przestrzeni użytkowników w systemie plików QSYS.LIB

System plików QSYS.LIB niezależnej ASP obsługuje operacje danych wejściowych i wyjściowych pliku strumieniowego dla obiektów przestrzeni użytkownika.

Na przykład program może zapisać i odczytać dane strumieniowe w przestrzeni użytkownika. Obszar użytkownika może mieć maksymalnie 16 776 704 bajtów.

Należy wziąć pod uwagę, że do obszarów użytkowników nie jest przyporządkowany identyfikator CCSID (coded character set identifier). Dlatego zwracany jest domyślny identyfikator CCSID.

## Obsługa zbiorów składowania w systemie plików QSYS.LIB

System plików QSYS.LIB obsługuje operacje strumieniowego wejścia/wyjścia dla obiektów zbioru składowania.

Na przykład, w zbiorze składowania mogą znajdować się dane, które można odczytać lub skopiować do innego zbioru, jeśli trzeba je umieścić w istniejącym, pustym obiekcie zbioru składowania. Kiedy zbiór składowania jest otwarty, nie są dozwolone żadne inne otwarte instancje tego pliku. Zbiór składowania **może** mieć wiele instancji otwartych do odczytu, pod warunkiem, że każde zadanie ma tylko jedną taką instancję. Zbioru składowania nie można otwierać w celu wykonania jednoczesnych operacji zapisu i odczytu. Operacje strumieniowego wejścia/wyjścia dla pliku składowania nie są dozwolone, jeśli w ramach zadania wykonywanych jest wiele wątków.

Operacje strumieniowego wejścia/wyjścia dla zbioru składowania nie są obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez system plików NFS. Są one jednak dostępne z klientów PC oraz poprzez system plików QFileSvr.400.

## Rozróżnianie wielkości znaków w systemie plików QSYS.LIB

System plików QSYS.LIB nie rozróżnia małych i wielkich liter w nazwach obiektów.

Wyszukiwanie nazw obiektów daje ten sam wynik niezależnie od tego, czy litery w nazwie obiektu są wielkie, czy małe.

Jeśli jednak nazwa ujęta jest w cudzysłów, wielkość wszystkich liter w nazwie zostaje zachowana. Wyszukiwanie z użyciem cytowanych nazw rozróżnia więc wielkość liter w cytowanej nazwie.

## Nazwy ścieżek w systemie plików QSYS.LIB

Każdy składnik ścieżki musi zawierać nazwę obiektu i następujący po niej typ obiektu.

- Na przykład:



/QSYS.LIB/QGPL.LIB/PRT1.OUTQ

/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR

Nazwa i typ obiektu oddzielane są kropką (.). Obiekty w bibliotece mogą mieć tę samą nazwę, jeśli są różnych typów, więc typ obiektu musi zostać podany w celu jednoznacznego określenia obiektu.

- Nazwa obiektu w każdym składniku może mieć do 10 znaków, a typ obiektu - do 6 znaków.
- Hierarchia katalogów w ramach systemu plików QSYS.LIB może mieć dwa lub trzy poziomy (w ścieżce mogą występować dwa lub trzy składniki), zależnie od typu danego obiektu. Jeśli obiekt jest zbiorem bazy danych, hierarchia może obejmować trzy poziomy (bibliotekę, zbiór, podzbiór); w innym przypadku mogą wystąpić jedynie dwa poziomy (biblioteka, obiekt). Maksymalna długość ścieżki określona jest przez kombinację długości składników i liczby poziomów katalogów.

Jeśli jako dwa pierwsze poziomy podaje się "root" (/) i QSYS.LIB, hierarchia katalogów w systemie plików QSYS.LIB może mieć pięć poziomów.

- Podczas zapisywania nazw występujące w nich znaki są poddawane konwersji do identyfikatora CCSID 37. Identyfikatory z ogranicznikami są jednak przechowywane z identyfikatorem CCSID zadania.

Więcej informacji dotyczących identyfikatorów CCSID można znaleźć w temacie Globalizacja i5/OS.

#### Pojęcia pokrewne

"Nazwa ścieżki" na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

## Dowiązania w systemie plików QSYS.LIB

W systemie plików QSYS.LIB nie można tworzyć ani przechowywać dowiązań symbolicznych.

Związek między biblioteką a obiektem w tej bibliotece jest równoważny stałemu dowiązaniu między biblioteką a każdym zawartym w niej obiektem. Zintegrowany system plików obsługuje związek biblioteka-obiekt jako dowiązanie. Jest więc możliwe tworzenie dowiązań z systemu plików obsługującego dowiązania symboliczne do obiektu w systemie plików QSYS.LIB.

#### Pojęcia pokrewne

"Dowiązanie" na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QSYS.LIB

Wiele z komend i ekranów zintegrowanego systemu plików działa w systemie plików QSYS.LIB.

Komendy wymienione w temacie "Dostęp przy użyciu komend języka CL" na stronie 63, z wyjątkiem następujących, działają w systemie plików QSYS.LIB:

- komenda Dodanie dowiązania (Add Link - ADDLNK) może być używana tylko do tworzenia dowiązania symbolicznego *do* obiektu w systemie plików QSYS.LIB,
- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- komendy Uruchomienie kronikowania (Start Journal - STRJRN) i Zakończenie kronikowania (End Journal - ENDJRN) nie mogą być używane na zbiorach fizycznych bazy danych ani bibliotekach,
- następujące komendy nie są obsługiwane:
  - Zwrócenie obiektu (Check In Object - CHKIN),
  - Pobranie obiektu (Check Out Object - CHKOUT),
  - Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK).

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie "Dostęp przy użyciu pozycji menu i ekranów" na stronie 61.

## Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB

Wiele funkcji API zintegrowanego systemu plików jest dostępnych w systemie plików QSYS.LIB.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106, z wyjątkiem następujących, działają w systemie plików QSYS.LIB:

- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- funkcji `symlink()` można użyć tylko do tworzenia dowiązania *do* obiektu w systemie plików QSYS z innego systemu plików, który obsługuje dowiązania symboliczne,
- funkcje API `QjoStartJournal()` i `QjoEndJournal()` nie mogą być użyte na fizycznych zbiorach bazy danych ani bibliotekach.

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Niezależne pule ASP systemu QSYS.LIB

System plików QSYS.LIB niezależnej ASP obsługuje strukturę biblioteki i5/OS w niezależnej puli pamięci dyskowej ASP utworzonej i zdefiniowanej przez użytkownika. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów i5/OS obsługiwanych przez biblioteki w systemowej puli ASP i w niezależnej puli ASP.

Dodatkowo, system ten:

- obsługuje wszystkie interfejsy użytkownika i interfejsy programistyczne wykonujące działania na bibliotekach i5/OS i na obiektach w tych bibliotekach w niezależnych pulach ASP,
- obsługuje wszystkie języki programowania i wszystkie narzędzia wykonujące działania na zbiorach baz danych,
- oferuje wiele funkcji administracyjnych przeznaczonych do zarządzania obiektami i5/OS,
- obsługuje operacje strumienia wejścia/wyjścia na podzbiorach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

## Uzyskanie dostępu do systemu plików QSYS.LIB w niezależnej puli ASP poprzez interfejs zintegrowanego systemu plików

Do systemu plików QSYS.LIB niezależnej puli ASP można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

Korzystając z interfejsów zintegrowanego systemu plików, należy uwzględnić uwagi i ograniczenia.

### Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB niezależnej puli ASP

QPWFSERVER jest listą autoryzacji (typ obiektu \*AUTL). Określa ona dodatkowe wymagania dotyczące dostępu dla wszystkich obiektów w systemie plików QSYS.LIB niezależnej puli ASP, do których istnieje dostęp poprzez klientów zdalnych.

Uprawnienia podane w tej liście odnoszą się do wszystkich obiektów znajdujących się w systemie plików QSYS.LIB niezależnej puli ASP.

Domyślnym uprawnieniem do danego obiektu jest PUBLIC \*USE. Administrator może za pomocą komendy Edycja listy autoryzacji (Edit Authorization List - EDTAUTL) lub Praca z listami autoryzacji (Work With Authorization List - WRKAUTL) zmieniać wartość uprawnienia. Administrator może przypisać uprawnienie PUBLIC \*EXCLUDE do listy autoryzacji, aby zablokować ogólny dostęp do obiektów QSYS.LIB niezależnej puli ASP ze zdalnych klientów.

### Ograniczenia w obsłudze plików w systemie plików QSYS.LIB niezależnej puli ASP

Poniżej znajdują się ograniczenia dotyczące obsługi plików w systemie plików QSYS.LIB niezależnej puli ASP

- Zbiory logiczne nie są obsługiwane.

- Zbiorami fizycznymi obsługiwanymi w trybie tekstowym są zbiory fizyczne opisywane przez programy i zawierające tylko jedno pole oraz źródłowe zbiory fizyczne zawierające tylko jedno pole tekstowe. Zbiorami fizycznymi obsługiwanymi w trybie binarnym są, oprócz zbiorów fizycznych obsługiwanym w trybie tekstowym, opisane zewnętrznie zbiory fizyczne.
- Blokada na poziomie bajtów nie jest obsługiwana. Więcej informacji na temat blokowania zakresów bajtów znaleźć można w temacie `fentl()` -- Perform File Control Command.
- Jeśli jakiegokolwiek zadanie ma otwarty podzbiór bazy danych, tylko jedno zadanie może w danym momencie uzyskać dostęp do tego podzbioru, aby go zapisać. Inne zadania mogą uzyskać dostęp jedynie do odczytu.

## Obsługa przestrzeni użytkowników w systemie plików QSYS.LIB niezależnej puli ASP

System plików QSYS.LIB niezależnej ASP obsługuje operacje danych wejściowych i wyjściowych pliku strumieniowego dla obiektów przestrzeni użytkownika.

Na przykład program może zapisać i odczytać dane strumieniowe w przestrzeni użytkownika. Obszar użytkownika może mieć maksymalnie 16 776 704 bajtów.

Należy wziąć pod uwagę, że do obszarów użytkowników nie jest przyporządkowany identyfikator CCSID (coded character set identifier). Dlatego zwracany jest domyślny identyfikator CCSID.

## Obsługa zbiorów składowania w systemie plików QSYS.LIB niezależnej puli ASP

System plików QSYS.LIB niezależnej puli obsługuje operacje strumieniowego wejścia/wyjścia dla obiektów zbioru składowania.

Na przykład, w zbiorze składowania mogą znajdować się dane, które można odczytać lub skopiować do innego zbioru, jeśli trzeba je umieścić w istniejącym, pustym obiekcie zbioru składowania. Kiedy zbiór składowania jest otwarty, nie są dozwolone żadne inne otwarte instancje tego pliku. Zbiór składowania **może** mieć wiele instancji otwartych do odczytu, pod warunkiem, że każde zadanie ma tylko jedną taką instancję. Zbioru składowania nie można otwierać w celu wykonania jednoczesnych operacji zapisu i odczytu. Operacje strumieniowego wejścia/wyjścia dla pliku składowania nie są dozwolone, jeśli w ramach zadania wykonywanych jest wiele wątków.

Operacje strumieniowego wejścia/wyjścia dla zbioru składowania nie są obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez system plików NFS. Są one jednak dostępne z klientów PC oraz poprzez system plików QFileSvr.400.

## Rozróżnianie wielkości znaków w systemie plików QSYS.LIB niezależnej puli ASP

System plików QSYS.LIB niezależnej puli ASP nie rozróżnia małych i wielkich liter w nazwach obiektów.

Wyszukiwanie nazw obiektów daje ten sam wynik niezależnie od tego, czy litery w nazwie obiektu są wielkie, czy małe.

Jeśli jednak nazwa ujęta jest w cudzysłów, wielkość wszystkich liter w nazwie zostaje zachowana. Wyszukiwanie z użyciem cytowanych nazw rozróżnia więc wielkość liter w cytowanej nazwie.

## Nazwy ścieżek w systemie plików QSYS.LIB niezależnej puli ASP

Każdy składnik ścieżki musi zawierać nazwę obiektu i następujący po niej typ obiektu.

- Na przykład:

```
/nazwa_asp/QSYS.LIB/QGPL.LIB/PRT1.OUTQ
```

```
/nazwa_asp/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR
```

gdzie `nazwa_asp` to nazwa niezależnej puli ASP. Nazwa i typ obiektu oddzielane są kropką (.). Obiekty w bibliotece mogą mieć tę samą nazwę, jeśli są różnych typów, więc typ obiektu musi zostać podany w celu jednoznacznego określenia obiektu.

- Nazwa obiektu w każdym składniku może mieć do 10 znaków, a typ obiektu - do 6 znaków.

- Hierarchia katalogów w ramach systemu plików QSYS.LIB niezależnej puli ASP może mieć dwa lub trzy poziomy (w ścieżce mogą występować dwa lub trzy składniki), zależnie od typu danego obiektu. Jeśli obiekt jest zbiorem bazy danych, hierarchia może obejmować trzy poziomy (bibliotekę, zbiór, podzbiór); w innym przypadku mogą wystąpić jedynie dwa poziomy (biblioteka, obiekt). Maksymalna długość ścieżki określona jest przez kombinację długości składników i liczby poziomów katalogów.

Jeśli jako trzy pierwsze poziomy podaje się /, nazwa\_asp i system plików QSYS.LIB, hierarchia katalogów w systemie plików niezależnej puli ASP może mieć sześć poziomów.

- Podczas zapisywania nazw występujące w nich znaki są poddawane konwersji do identyfikatora kodowanego zestawu znaków (CCSID) 37. Identyfikatory z ogranicznikami są jednak przechowywane z identyfikatorem CCSID zadania.

Więcej informacji o identyfikatorach CCSID zawiera temat Globalizacja i5/OS w Centrum informacyjnym i5/OS.

#### Pojęcia pokrewne

“Nazwa ścieżki” na stronie 14

Nazwa ścieżki informuje system, jak odnaleźć dany obiekt.

## Dowiązania w systemie plików QSYS.LIB niezależnej puli ASP

W systemie plików QSYS.LIB niezależnej puli ASP nie można tworzyć ani przechowywać dowiązań symbolicznych.

Związek między biblioteką a obiektem w tej bibliotece jest równoważny stałemu dowiązaniu między biblioteką a każdym zawartym w niej obiektem. Zintegrowany system plików obsługuje związek biblioteka-obiekt jako dowiązanie. Jest więc możliwe tworzenie dowiązań z systemu plików obsługującego dowiązania symboliczne do obiektu w systemie plików QSYS.LIB niezależnej puli ASP.

#### Pojęcia pokrewne

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP

Wiele komend zintegrowanego systemu plików i ekranów jest dozwolonych w systemie plików QSYS.LIB niezależnej puli ASP.

Większość komend wymienionych w temacie “Dostęp przy użyciu komend języka CL” na stronie 63 działa w systemie plików QSYS.LIB niezależnej puli ASP. Istnieje jednak kilka wyjątków:

- komenda Dodanie dowiązania (Add Link - ADDLNK) może być używana tylko do tworzenia dowiązania symbolicznego do obiektu w systemie plików QSYS.LIB niezależnej puli ASP,
- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- komendy Uruchomienie kronikowania (Start Journal - STRJRN) i Zakończenie kronikowania (End Journal - ENDJRN) nie mogą być używane na zbiorach fizycznych bazy danych ani bibliotekach,
- komendą Przenieś obiekt (Move Object - MOV) nie można przenosić bibliotek w systemie plików QSYS.LIB niezależnej ASP do podstawowych ASP; można jednak przenosić biblioteki z systemu plików QSYS.LIB niezależnej puli ASP do systemowej ASP lub innych niezależnych pul ASP,
- jeśli do składowania i odtwarzania obiektów biblioteki w niezależnej puli ASP wykorzystywane są komendy Składowanie obiektu (Save Object - SAV) lub Odtworzenie obiektu (Restore Object - RST), to niezależna pula ASP musi być skojarzona z zadaniem wykonującym komendę SAV lub RST lub musi być określona niezależna pula ASP w parametrze ASPDEV; konwencja nazw ścieżek /nazwa\_asp/QSYS.LIB/obiekt.typ nie jest obsługiwana w SAV i RST,
- następujące komendy nie są obsługiwane:
  - Zwrócenie obiektu (Check In Object - CHKIN),
  - Pobranie obiektu (Check Out Object - CHKOUT),

- Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK).

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 61.

## **Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP**

Wiele funkcji API zintegrowanego systemu plików jest dozwolonych w systemie plików QSYS.LIB niezależnej puli ASP.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106 działają w systemie plików QSYS.LIB niezależnej puli ASP z wyjątkiem następujących sytuacji:

- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- funkcji `symlink()` można użyć tylko do tworzenia dowiązania do obiektu w systemie plików QSYS.LIB niezależnej puli ASP z innego systemu plików, który obsługuje dowiązania symboliczne.
- funkcje API `QjoStartJournal()` i `QjoEndJournal()` nie mogą być użyte na fizycznych zbiorach bazy danych ani bibliotekach.
- jeśli do składowania i odtwarzania obiektów biblioteki w niezależnej puli ASP wykorzystywane są funkcje API `QsrSave()` lub `QsrRestore()`, to niezależna pula ASP musi być skojarzona z zadaniem wykonującym czynności składowania lub odtwarzania lub musi być określona niezależna pula ASP w kluczu `ASPDEV`. Funkcje API `QsrSave()` i `QsrRestore()` nie obsługują konwencji nazewnictwa nazwy ścieżki (`/asp_name/QSYS.LIB/object.type`).

### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **System plików usług biblioteki dokumentów (QDLS)**

System plików QDLS obsługuje strukturę folderów. Umożliwia on dostęp do dokumentów i folderów.

Dodatkowo, system ten:

- Obsługuje foldery i obiekty biblioteki dokumentów (DLO) systemu i5/OS.
- Obsługuje dane przechowywane w plikach strumieniowych.

## **Uzyskiwanie dostępu do systemu plików QDLS poprzez interfejs zintegrowanego systemu plików**

Do systemu plików QDLS można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

Korzystając z interfejsów zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

## **Zintegrowany system plików i system HFS w systemie plików QDLS**

Działania na obiektach w systemie plików QDLS można wykonywać nie tylko korzystając z komend CL dotyczących obiektów biblioteki dokumentów (DLO), ale także przy użyciu interfejsu zintegrowanego systemu plików lub funkcji API udostępnianych przez hierarchiczny system plików (HFS).

O ile zintegrowany system plików oparty jest na modelu programu środowiska Integrated Language Environment (ILE), system plików HFS bazuje na oryginalnym modelu programu System i.

Funkcje API HFS umożliwiają dodatkowe działania, których nie obsługuje zintegrowany system plików. W szczególności dzięki funkcjom API HFS można uzyskać dostęp do atrybutów rozszerzonych katalogów (zwanymi także *atrybutami pozycji katalogu*) i zmieniać je. Należy wziąć pod uwagę, że reguły nazewnictwa przy pracy z funkcjami API HFS są inne niż reguły nazewnictwa dla funkcji API wykorzystujących interfejs zintegrowanego systemu plików.

### **Informacje pokrewne**

Funkcje API hierarchicznego systemu plików

## Rejestrowanie użytkowników w systemie plików QDLS

Użytkownik pracujący z obiektami w systemie plików QDLS musi zostać wpisany do katalogu dystrybucyjnego systemu.

## Rozróżnianie wielkości znaków w systemie plików QDLS

System plików QDLS zamienia małe litery alfabetu angielskiego (od **a** do **z**) na wielkie, gdy są one używane w nazwach obiektów. Oznacza to, że przy wyszukiwaniu nazw obiektów zawierających jedynie te znaki nie jest rozróżniana wielkość liter.

W przypadku wszystkich innych znaków w systemie plików QDLS wielkość liter jest rozróżniana.

### Informacje pokrewne

Nazwa dokumentu i folderu

## Nazwy ścieżek w systemie plików QDLS

Każdy składnik ścieżki może być po prostu nazwą.

- Na przykład:

`/QDLS/FLR1/DOC1`

lub nazwą z rozszerzeniem (podobnym do rozszerzenia pliku DOS), np:

`/QDLS/FLR1/DOC1.TXT`

- Nazwa w każdym ze składników może mieć do 8 znaków, a rozszerzenie (o ile występuje) może mieć do 3 znaków. Maksymalna długość ścieżki bezwzględnej zaczynającej się od `/QDLS` wynosi 82 znaki.
- Hierarchia katalogów w systemie plików QDLS (usług bibliotek dokumentów) może mieć głębokość 32 poziomów. Jeśli pierwszymi dwoma poziomami są `/` i `QDLS`, hierarchia katalogów może liczyć 34 poziomy.
- Dopóki w bibliotece QUSRSYS nie zostanie utworzony obszar danych Q0DEC500, przed zapamiętaniem nazw występujące w nich znaki są przekształcane do strony kodowej zadania. Jeśli obszar danych zostanie utworzony, to przed zachowaniem nazw występujące w nich znaki są przekształcane do strony kodowej 500. Dzięki tej funkcji zapewniona jest zgodność z systemem plików QDLS z poprzednich wersji. Nazwa może zostać odrzucona, jeśli nie można jej poddać konwersji do właściwej strony kodowej.

Więcej informacji o stronach kodowych można znaleźć w sekcji Globalizacja w systemie i5/OS w Centrum informacyjnym i5/OS.

### Pojęcia pokrewne

“Nazwa ścieżki” na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

## Dowiązania w systemie plików QDLS

W systemie plików QDLS nie można tworzyć ani przechowywać dowiązań symbolicznych.

Zintegrowany system plików obsługuje związek folderu z zawartymi w nim obiektami biblioteki dokumentów jako równoważny dowiązaniu między folderem a każdym spośród zawartych w nim obiektów. Jest więc możliwe tworzenie dowiązań z systemu plików obsługującego dowiązania symboliczne do systemu plików QDLS.

### Pojęcia pokrewne

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QDLS

Wiele z komend i ekranów zintegrowanego systemu plików działa w systemie plików QDLS.

Komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 63, z następującymi wyjątkami, działają w systemie plików QDLS:

- komendy ADDLNK można używać jedynie do tworzenia dowiązań *do* obiektu w systemie plików QDLS z innego systemu plików, który obsługuje dowiązania symboliczne;
- komendy CHKIN i CHKOUT są obsługiwane jedynie w przypadku dokumentów, natomiast nie w przypadku folderów;
- następujące komendy nie są obsługiwane:
  - APYJRNCHG
  - CHGJRNOBJ
  - DSPJRN
  - ENDJRN
  - RCLLNK
  - RCVJRNE
  - RTVJRNE
  - SNDJRNE
  - STRJRN

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 61.

## Używanie funkcji API zintegrowanego systemu plików w systemie plików QDLS

Wiele funkcji API zintegrowanego systemu plików jest dostępnych w systemie plików QDLS.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106, z wyjątkiem następujących, działają w systemie plików QDLS:

- funkcji `symlink()` można użyć tylko do tworzenia dowiązania do obiektu w systemie plików QDLS z innego systemu plików, który obsługuje dowiązania symboliczne,
- poniższe funkcje nie są obsługiwane:
  - `givedescriptor()`
  - `ioctl()`
  - `link()`
  - `QjoEndJournal()`
  - `QjoRetrieveJournalEntries()`
  - `QjoRetrieveJournalInformation()`
  - `QJORJIDI()`
  - `QJOSJRNE()`
  - `QjoStartJournal()`
  - `readlink()`
  - `takedescriptor()`

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## System plików nośników optycznych (QOPT)

System plików QOPT umożliwia dostęp do danych strumieniowych przechowywanych na nośnikach optycznych.

Dodatkowo, system ten:

- zawiera hierarchiczną strukturę katalogów podobną do struktury systemów operacyjnych komputerów PC, takich jak DOS i OS/2,

- jest zoptymalizowany do obsługi danych wejściowych i wyjściowych plików strumieniowych,
- obsługuje dane przechowywane w plikach strumieniowych.

## Dostęp do systemu plików QOPT poprzez zintegrowany system plików

Dostęp do systemu plików QOPT jest możliwy poprzez zintegrowany system plików po użyciu serwera komend, ekranów użytkownika i funkcji API zintegrowanego systemu plików lub serwera PC.

Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

### Informacje pokrewne

Pamięć optyczna

## Zintegrowany system plików i system HFS w systemie plików QOPT

Działania na obiektach w systemie plików QOPT można wykonywać zarówno przez interfejs zintegrowanego systemu plików, jak i przez funkcje API udostępniane przez hierarchiczny system plików (HFS).

O ile zintegrowany system plików oparty jest na modelu programu środowiska Integrated Language Environment (ILE), system plików HFS bazuje na oryginalnym modelu programu System i.

Funkcje API HFS umożliwiają dodatkowe działania, których nie obsługuje zintegrowany system plików. W szczególności, dzięki funkcjom API HFS można uzyskać dostęp do atrybutów rozszerzonych katalogów (zwanymi także *atrybutami pozycji katalogu*) i zmieniać je oraz pracować z plikami na nośnikach optycznych. Należy wziąć pod uwagę, że reguły nazewnictwa przy pracy z funkcjami API HFS są inne niż reguły nazewnictwa dla funkcji API wykorzystujących interfejs zintegrowanego systemu plików.

Więcej informacji na temat funkcji API HFS znaleźć można w kolekcji tematów Programowanie urządzeń optycznych.

### Informacje pokrewne

Funkcje API hierarchicznego systemu plików

## Rozróżnianie wielkości znaków w systemie plików QOPT

Zależnie od formatu nośnika optycznego, wielkość liter może, lecz nie musi być zachowywana podczas tworzenia plików w systemie plików QOPT. Niezależnie jednak od formatu nośnika optycznego, wyszukiwanie plików i katalogów odbywa się bez rozróżniania wielkości liter.

## Nazwy ścieżek w systemie plików QOPT

Nazwa ścieżki musi zaczynać się od ukośnika (/). Ścieżka dostępu złożona jest z nazwy systemu plików, nazwy woluminu, nazwy katalogu, nazwy podkatalogu i nazwy obiektu.

- Na przykład:  
/QOPT/WOLUMIN/KATALOG/PODKATALOG/OBIEKT
- Wymagane jest podanie nazwy systemu plików (QOPT).
- Wolumin i długość nazwy ścieżki mogą zależeć od formatu nośnika optycznego.
- Aby włączyć jeden lub kilka katalogów lub podkatalogów do nazwy ścieżki, w nazwie ścieżki można podać wartość /QOPT. Nazwy katalogów i plików mogą zawierać dowolne znaki oprócz znaków o kodach od X'00' do X'3F' i X'FF'. Dodatkowe ograniczenia mogą wynikać z formatu nośnika optycznego.
- Ostatnim elementem ścieżki jest nazwa obiektu. Liczba znaków w nazwie obiektu jest ograniczona przez liczbę znaków w nazwie katalogu.

Więcej informacji o regułach nazw ścieżek w systemie plików QOPT można znaleźć w części "Reguły nazw ścieżek" w sekcji Nazwy ścieżek.

### Pojęcia pokrewne

"Nazwa ścieżki" na stronie 14

Nazwa ścieżki informuje system, jak odnaleźć dany obiekt.



## Dowiązania w systemie plików QOPT

- | System plików QOPT dopuszcza tylko jedno dowiązanie do obiektu. Nie można w nim tworzyć ani przechowywać
- | dowiązań symbolicznych. Dostęp do plików w systemie plików QOPT można jednak uzyskać poprzez dowiązanie
- | symboliczne z systemów plików "root" (/), QOpenSys lub użytkownika.

### Pojęcia pokrewne

"Dowiązanie" na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QOPT

Wiele z komend i ekranów zintegrowanego systemu plików działa w systemie plików QOPT.

Większość komend wymienionych w temacie "Dostęp przy użyciu komend języka CL" na stronie 63 działa w systemie plików QOPT. Istnieje jednak kilka wyjątków. Należy pamiętać, że korzystanie z tych komend CL w przypadku procesu wielowątkowego może nie być bezpieczne. Zależnie od formatu nośnika optycznego mogą obowiązywać pewne ograniczenia. Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie "Dostęp przy użyciu pozycji menu i ekranów" na stronie 61.

Następujące komendy zintegrowanego systemu plików nie są obsługiwane w systemie plików QOPT:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHKIN
- CHKOUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

## Używanie funkcji API zintegrowanego systemu plików w systemie plików QOPT

Wiele funkcji API zintegrowanego systemu plików jest dostępnych w systemie plików QOPT.

Wszystkie funkcje API wymienione w sekcji "Wykonywanie operacji za pomocą funkcji API" na stronie 106 mogą być używane do operacji na systemie plików QOPT z bezpieczną obsługą wielowątkowości, z wyjątkiem następujących funkcji:

- QjoEndJournal()
- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()
- | • Qp0lGetPathFromFileID()

## Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## System plików i5/OS NetClient (QNTC)

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych na serwerze Integrated xSeries Server (IXS) z systemem operacyjnym Windows NT 4.0 lub późniejszym bądź systemem operacyjnym Linux. System plików QNTC zapewnia również dostęp do danych i obiektów przechowywanych na serwerach zdalnych z systemem Windows NT 4.0 lub późniejszym, systemem Linux z oprogramowaniem Samba 3.0 lub późniejszym, bądź obsługiwanymi wersjami serwera i5/OS NetServer.

System plików QNTC jest częścią podstawowego systemu operacyjnego i5/OS. Instalacja opcji 29, Integrated Sever Support, systemu operacyjnego nie jest wymagana do uzyskania dostępu do systemu plików /QNTC.

## Dostęp do systemu plików QNTC poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QNTC można uzyskać za pośrednictwem interfejsu zintegrowanego systemu plików, korzystając z serwera i5/OS NetServer, programu System i Navigator, komend zintegrowanego systemu plików, ekranów użytkownika lub funkcji API.

Należy pamiętać o poniższych uwagach i ograniczeniach.

## Uprawnienia i prawa własności w systemie plików QNTC

System plików QNTC nie obsługuje pojęcia własności pliku lub katalogu.

Wszelkie próby użycia komendy lub funkcji API w celu zmiany prawa własności plików przechowywanych w systemie QNTC zakończą się niepowodzeniem. Właścicielem wszystkich plików i katalogów w systemie QNTC jest systemowy profil użytkownika QDFTOWN.

Uprawnieniami do plików i katalogów serwera NT można administrować z poziomu serwera Windows NT. System plików QNTC nie obsługuje komend WRKAUT i CHGAUT.

## Rozróżnianie wielkości znaków w systemie plików QNTC

System plików QNTC zachowuje małe i wielkie litery we wprowadzanych nazwach obiektów, jednak sam nie rozróżnia wielkich i małych liter w nazwach. Oznacza to, że w przypadku rozróżniania wielkości liter przez system plików serwera należy zadbać o używanie odpowiednich wielkości liter podczas wprowadzania nazwy ścieżki w systemie plików QNTC.

Wyszukiwanie nazw obiektów daje ten sam wynik niezależnie od tego, czy litery w nazwie obiektu są wielkie, czy małe.

## Nazwy ścieżek w systemie plików QNTC

Ścieżka składa się z nazwy systemu plików, nazwy serwera, nazwy zasobu współużytkowanego, nazw katalogu i podkatalogów oraz nazwy obiektu.

Wymagania dotyczące nazwy ścieżki są następujące:

- Ścieżka musi zaczynać się od ukośnika (/) i może zawierać co najwyżej 255 znaków. Ścieżki dostępu mają następującą postać:

```
/QNTC/NazwaSerwera/NazwaZasobuWspółużytkowanego/Katalog/ . . . /Obiekt  
(QNTC stanowi wymaganą część nazwy ścieżki.)
```

- Nazwa serwera jest w nazwie ścieżki systemu plików QNTC częścią wymaganą. Nazwą serwera może być nazwa hosta TCP/IP, nazwa NetBIOS lub adres TCP/IP. Od wersji IPv6 poza adresami IPv4 obsługiwane są również adresy IPv6.
- Nazwa zasobu współużytkowanego może mieć do 12 znaków.
- Każdy składnik ścieżki (występujący po nazwie zasobu współużytkowanego) może mieć do 255 znaków.

- W ramach QNTC dostępnych jest w zasadzie 130 poziomów hierarchii. Jeśli wszystkie składniki ścieżki zostały dodane do ścieżki jako poziomy hierarchii, hierarchia katalogów może mieć aż 132 poziomy.
- Nazwy są przechowywane w identyfikatorze CCSID Unicode.
- Domyślnie każdy obsługiwany serwer działający w lokalnej podsieci będzie automatycznie wyświetlany jako katalog w systemie plików /QNTC. Aby dodać dostępne systemy znajdujące się poza lokalną podsiecią, należy użyć komendy Tworzenie katalogu (Create Directory - CRTDIR) lub funkcji API mkdir().

#### **Pojęcia pokrewne**

“Nazwa ścieżki” na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

#### **Informacje pokrewne**

Komenda Tworzenie katalogu (Create Directory - MKDIR)

mkdir()--funkcja API Make Directory

Glosariusz systemu i5/OS

## **Dowiązania w systemie plików QNTC**

System plików QNTC dopuszcza tylko jedno dowiązanie do obiektu. W systemie plików QNTC nie można tworzyć ani składować dowiązań symbolicznych.

Można uzyskać dostęp do plików w QNTC, używając dowiązania symbolicznego z systemu plików “root” (/) lub QOpenSys.

#### **Pojęcia pokrewne**

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## **Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QNTC**

Wiele z komend i ekranów zintegrowanego systemu plików działa w systemie plików QNTC

Komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 63, z następującymi wyjątkami, działają w systemie plików QNTC:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHGOWN
- CHGAUT
- CHGPGP
- CHKIN
- CHKOUT
- DSPAUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- RST (dostępna w serwerach Integrated xSeries Servers)
- SAV (dostępna w serwerach Integrated xSeries Servers)
- SNDJRNE

- STRJRN
- WRKAUT
- WRKOBJOWN
- WRKOBJPGP

Te same ograniczenia dotyczą ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 61.

## Używanie funkcji API zintegrowanego systemu plików w systemie plików QNTC

Wiele funkcji API zintegrowanego systemu plików jest dostępnych w systemie plików QNTC.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106, z wyjątkiem następujących, działają w systemie plików QNTC:

- użycie funkcji `chmod()`, `fchmod()`, `utime()` lub `umask()` do obiektów w QNTC nie będzie miało żadnego skutku, ale próba ich użycia nie spowoduje błędu;
- system plików QNTC nie obsługuje poniższych funkcji:
  - `chown()`
  - `fchown()`
  - `fclear()`
  - `fclear64()`
  - `givedescriptor()`
  - `link()`
  - `QjoEndJournal()`
  - `QjoRetrieveJournalEntries()`
  - `QjoRetrieveJournalInformation()`
  - `QJORJIDI()`
  - `QJOSJRNE()`
  - `QjoStartJournal()`
  - `Qp0lGetPathFromFileID()`
  - `readlink()`
  - `symlink()`
  - `takedescriptor()`
- podczas wykonywania następujących funkcji system plików QNTC ignoruje wszelkie limity zasobów ustawione za pomocą funkcji API `setrlimit()`:
  - `write()`
  - `writev()`
  - `pwrite()`
  - `pwrite64()`

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Zmienne środowiskowe systemu plików QNTC

Funkcjonowanie systemu plików QNTC podczas przeglądania sieci jest zależne od dwóch zmiennych środowiskowych. Obsługa tych zmiennych została wprowadzona w wersji V5R4 systemu i5/OS. Można je utworzyć za pomocą komendy CL `ADDENVVAR`.

## QZLC\_SERVERLIST

Jeśli wartość tej zmiennej wynosi "2", wszystkie serwery w katalogu /QNTC zintegrowanego systemu plików są dostępne poprzez system plików QNTC. W wersjach poprzedzających V5R4 był to stan domyślny. Jeśli wartość zmiennej jest różna od "2", niektóre z serwerów w katalogu /QNTC mogą być niedostępne.

## QIBM\_ZLC\_NO\_BROWSE

Jeśli wartość tej zmiennej środowiskowej wynosi "1", katalog /QNTC zawiera tylko serwery utworzone za pomocą komendy CL CRTDIR lub funkcji API mkdir(). Określenie wartości tej zmiennej zwiększa wydajność wielu operacji na systemie plików QNTC. Wszystkie katalogi /QNTC należy jednak utworzyć za pomocą komendy CL.

## Tworzenie katalogów w systemie plików QNTC

| Aby dodać katalog serwera do katalogu /QNTC, należy użyć komendy Tworzenie katalogu (Create Directory - CRTDIR) lub funkcji API mkdir().

| Domyślnie katalog QNTC tworzony jest automatycznie dla wszystkich działających serwerów w domenie serwera i5/OS NetServer i lokalnej podsięci. Serwery znajdujące się poza lokalną podsięcią lub domeną serwera i5/OS NetServer należy dodać używając komendy CRTDIR lub funkcji API mkdir(). Na przykład:

| CRTDIR '/QNTC/NTSRV1'

| dodaje serwer NTSRV1 do struktury katalogów systemu plików QNTC, aby umożliwić dostęp do plików i katalogów w serwerze.

| Można także dodać nowy serwer do struktury katalogów za pomocą adresu TCP/IP. Nazwa serwera może być podana w postaci adresu IPv4 lub IPv6. Na przykład:

| CRTDIR '/QNTC/9.130.67.24'

| lub:

| CRTDIR '/QNTC/2001:0db8:3c4d:0015:0000:0000:abcd:ef12'

dodaje serwer do struktury katalogów systemu plików QNTC.

### Uwagi:

- Skonfigurowanie obsługi protokołu WINS na serwerze i5/OS NetServer pozwala automatycznie tworzyć katalogi dla serwerów znajdujących się poza lokalną podsięcią.
- Użycie komendy CL CRTDIR lub funkcji API mkdir() w celu dodania katalogów do struktury katalogów spowoduje, że katalogi te przestaną być widoczne po ponownym wykonaniu IPL systemu lub wykonaniu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG). Po wykonaniu IPL systemu lub wykonaniu na katalogach komendy RCLSTG konieczne jest ponowne wykonanie komendy CRTDIR lub funkcji API mkdir().

Wydajność komendy CL i funkcji API używanych do dodawania katalogu można zwiększyć dodając zmienną środowiskową QIBM\_ZLC\_NO\_BROWSE, jak w poniższym przykładzie:

```
ADDENVVAR ENVVAR(QIBM_ZLC_NO_BROWSE) VALUE(1) LEVEL(*SYS)
```

Zmienna środowiskowa powoduje, że system plików pomija przeglądanie sieci podczas wykonywania operacji na plikach.

### Informacje pokrewne

Komenda Tworzenie katalogu (Create Directory - MKDIR)

mkdir()--funkcja API Make Directory

## Aktywowanie usługi uwierzytelniania sieciowego w systemie plików QNTC

System plików QNTC umożliwia dostęp platformy System i do serwerów systemu plików CIFS, które obsługują protokół uwierzytelniania Kerberos V5.

Zamiast używać hasła zarządzania siecią LAN do uwierzytelniania na każdym serwerze, odpowiednio skonfigurowana platforma System i może teraz uzyskać dostęp do obsługiwanych serwerów CIFS z wykorzystaniem pojedynczej transakcji logowania.

Aby udostępnić usługę uwierzytelniania sieciowego (NAS) w systemie plików QNTC, należy skonfigurować następujące elementy:

- Usługę uwierzytelniania sieciowego (NAS)
- Odwzorowywanie tożsamości dla przedsiębiorstwa (Enterprise Identity Mapping - EIM)

Gdy powyższe pozycje zostaną skonfigurowane, można włączyć obsługę usługi uwierzytelniania NAS w systemie plików QNTC. Aby umożliwić użytkownikom korzystanie z obsługi uwierzytelniania NAS w systemie plików QNTC, należy wykonać poniższe czynności.

- Profil danego użytkownika w systemie i5/OS musi mieć ustawioną wartość \*NO parametru lokalnego zarządzania hasłami (LCLPDMGT). Wartość \*NO oznacza, że użytkownik nie posiada hasła dostępu do serwera i nie ma możliwości zalogowania się do sesji terminalu 5250. Dostęp do serwera jest możliwy jedynie za pośrednictwem aplikacji obsługujących uwierzytelnianie sieciowe, takich jak System i Navigator lub emulator terminalu 5250 w programie System i Access.

Jeśli użytkownik poda wartość \*YES, hasło będzie zarządzane na serwerze, a użytkownik będzie uwierzytelniany bez wykorzystania usługi NAS.

- Użytkownik musi posiadać bilet protokołu Kerberos i połączenie z programem System i Navigator.
- Używany bilet protokołu Kerberos dla platformy System i musi być przekazywalny. Aby bilet mógł być przekazywany, należy wykonać następujące kroki:
  1. Uzyskaj dostęp do narzędzia **Aktywni użytkownicy i komputery** (Active Directory Users and Computers) na kontrolerze domeny Kerberos (KDC) dla dziedziny usługi NAS.
  2. Wybierz użytkowników.
  3. Wybierz nazwę odpowiadającą nazwie usługi.
  4. Wybierz opcję **Właściwości** (Properties).
  5. Wybierz kartę **Konto** (Account).
  6. Wybierz opcję **Konto zaufane do delegowania** (Account is trusted for delegation).

### Informacje pokrewne

Usługa uwierzytelniania sieciowego

Odwzorowywanie tożsamości dla przedsiębiorstwa (Enterprise Identity Mapping - EIM)

## System plików serwera plików i5/OS (QFileSvr.400)

System plików QFileSvr.400 zapewnia przezroczysty dostęp do innych systemów plików, znajdujących się na zdalnych platformach System i. Dostęp odbywa się poprzez hierarchiczną strukturę katalogów.

System plików QFileSvr.400 może być traktowany jako klient, który realizuje żądania użytkowników dotyczące obsługi zbiorów. System plików QFileSvr.400 współdziała z serwerem plików i5/OS w systemie docelowym w celu wykonywania bieżących operacji na zbiorach.

## Dostęp do systemu plików QFileSvr.400 poprzez interfejs zintegrowanego systemu plików

Do systemu plików QFileSvr.400 można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików i5/OS lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API.

Korzystając z interfejsów zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

**Uwaga:** Charakterystyki systemu plików QFileSvr.400 są określone przez charakterystyki systemów plików dostępnych na serwerze docelowym.

## Rozróżnianie wielkości znaków w systemie plików QFileSvr.400

System plików QFileSvr.400 zachowuje małe i wielkie litery we wprowadzanej nazwie obiektu dla pierwszego poziomu katalogów, który w zasadzie jest katalogiem “root” (/) systemu docelowego.

System plików QFileSvr.400 nie rozróżnia jednak wielkości liter podczas wyszukiwania nazwy.

Dla wszystkich innych katalogów rozróżnianie wielkości liter jest uzależnione od własności używanego systemu plików. System plików QFileSvr.400 zachowuje małe i wielkie litery we wprowadzanej nazwie obiektu podczas przesyłania żądania pliku do serwera plików i5/OS.

## Nazwy ścieżek w systemie plików QFileSvr.400

Nazwy ścieżek w systemie plików QFileSvr.400 mają określoną postać.

- Poprawną postacią jest:

/QFileSvr.400/NazwaPołożeniaZdalnego/Katalog/Katalog . . . /Obiekt

Katalog pierwszego poziomu (w powyższym przykładzie będzie to **NazwaPołożeniaZdalnego**) reprezentuje jednocześnie dwa atrybuty:

- Nazwę systemu docelowego, który zostanie użyty do nawiązania połączenia komunikacyjnego. Nazwą systemu docelowego może być:
  - nazwa hosta TCP/IP (na przykład beowulf.newyork.corp.com)

**Uwaga:** Nazwa hosta może być rozstrzygana zarówno na adres IPv4, jak i IPv6, pod warunkiem, że system docelowy również jest w wersji V6R1. W przypadku wersji wcześniejszych, niż V6R1, obsługiwane są wyłącznie adresy IPv4.

- nazwa SNA LU 6.2 (na przykład appn.newyork)

- Katalog “root” (/) serwera docelowego.

Ze względu na tę reprezentację, wszelkie atrybuty określone podczas tworzenia katalogu pierwszego poziomu są ignorowane.

Korzystanie z tego systemu plików wymaga uprzedniego utworzenia katalogu pierwszego poziomu. Można to zrobić za pomocą dowolnego interfejsu zintegrowanego systemu plików, który umożliwia tworzenie katalogów.

**Uwaga:** Katalogi pierwszego poziomu istnieją tylko do najbliższego IPL. Dlatego po każdym IPL muszą być tworzone od nowa.

- Każdy składnik ścieżki może mieć do 255 znaków. Pełna ścieżka może mieć do 16 megabajtów długości.

**Uwaga:** System plików, w którym znajduje się obiekt, może ograniczać długość składników i długość ścieżki do wartości mniejszej niż określona przez system plików QFileSvr.400.

- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu i systemu oraz przez ograniczenia nałożone przez używany system plików.
- Podczas zapisywania nazw występujące w nich znaki są poddawane konwersji na kodowanie Unicode.

### Pojęcia pokrewne

“Niezmienność nazw” na stronie 16


Użytkownicy systemu plików “root” (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmienną nazwę obiektów.

“Nazwa ścieżki” na stronie 14

*Nazwa ścieżki* informuje system, jak odnaleźć dany obiekt.

## Komunikacja w systemie plików QFileSvr.400

Poniżej przedstawiono metody komunikacji w systemie plików QFileSvr.400.

- Połączenia TCP z serwerem plików serwera docelowego mogą zostać ustanowione tylko wtedy, gdy jest w nim aktywny podsystem QSERVER.
- Próby połączenia SNA LU 6.2 wykonywane są tylko w przypadku, gdy istnieje nieużywana sesja obsługiwana lokalnie (na przykład sesja ustanowiona specjalnie do użycia przez połączenie LU 6.2). Podczas nawiązywania połączeń LU 6.2 system plików QFileSvr.400 używa trybu BLANK. W systemie docelowym zadanie QPWFSERV jest wpisywane do kolejki w podsystemie QSERVER. Profil użytkownika tego zadania jest zdefiniowany przez pozycję dotyczącą komunikacji dla trybu BLANK. Więcej informacji na temat komunikacji LU 6.2 można znaleźć w publikacji APPC Programming .
- Żądania serwera plików używające TCP jako protokołu komunikacji są obsługiwane zgodnie z kontekstem zadania, które zgłasza żądanie. Żądania serwera plików używające SNA jako protokołu komunikacyjnego są obsługiwane przez zadanie Q400FILSVR systemu i5/OS.
- Jeśli nie zostało ustanowione połączenie z serwerem docelowym, system plików QFileSvr400 zakłada, że katalog pierwszego poziomu reprezentuje nazwę hosta TCP/IP. Aby ustanowić połączenie z serwerem docelowym, system plików QFileSvr.400 wykonuje następujące czynności:
  1. Zamienia nazwę jednostki zdalnej na adres IP.

**Uwaga:** Nazwa zdalnego miejsca może być rozstrzygana zarówno na adres IPv4, jak i IPv6, pod warunkiem, że system docelowy również jest w wersji V6R1. W przypadku wersji wcześniejszych niż V6R1 obsługiwane są wyłącznie adresy IPv4.

2. Łączy się z programem odwzorowującym serwera poprzez port 449 za pomocą otrzymanego adresu IP. Następnie wysyła zapytanie do programu odwzorowującego serwera o usługę "as-file". Odpowiedź na zadane pytanie może być dwojaka, w zależności od sytuacji:
  - Jeśli usługa "as-file" znajduje się w tabeli usług serwera docelowego, program odwzorowujący serwera zwraca numer portu, na którym nasłuchuje demon serwera plików i5/OS.
  - Jeśli program odwzorowujący serwera docelowego nie jest aktywny, używany jest domyślny numer portu dla usługi "as-file" (8473).

Następnie system plików QFileSvr.400 próbuje ustanowić połączenie TCP z demonem serwera plików i5/OS na serwerze docelowym. Po ustanowieniu połączenia system plików QFileSvr.400 wymienia żądania i odpowiedzi z serwerem plików. W obrębie podsystemu QSERVER żądania prestartu QPWFSERVS0 przejmują kontrolę nad połączeniem. Każde zadanie prestartu działa z własnym profilem użytkownika.

3. Jeśli nazwa jednostki zdalnej nie zostanie zamieniona na adres IP, to zakłada się, że katalog pierwszego poziomu ma nazwę taką, jak SNA LU 6.2. Dlatego też ustanawia się połączenie między APPC i serwerem plików i5/OS.
- Następnie system plików QFileSvr.400 okresowo (co dwie godziny) sprawdza, czy istnieją jakieś nieużywane połączenia (na przykład z połączeniem nie są powiązane żadne otwarte pliki), dla których nie zanotowano żadnej aktywności w ciągu ostatnich dwóch godzin. Jeśli system znajdzie takie połączenie, to zostanie ono zakończone.
  - System plików QFileSvr.400 nie wykrywa zapętleń. Przedstawiona poniżej ścieżka jest przykładem zapętlenia:
 

```
/QFileSvr.400/Remote2/QFileSvr.400/Remote1/QFileSvr.400/Remote2/...
```

Remote1 jest w tym przypadku systemem lokalnym. Gdy ścieżka zawiera zapętlenie, po krótkim czasie system plików QFileSvr.400 zgłosi błąd przekroczenia czasu oczekiwania (time-out).
  - Przy komunikacji poprzez SNA, system plików QFileSvr.400 użyje istniejącej wolnej sesji. Aby prawidłowo podłączyć się do zdalnego systemu komunikacyjnego, konieczne jest uruchomienie trybu i nawiązanie sesji dla QFileSvr.400.

## Bezpieczeństwo i uprawnienia do obiektów w systemie plików QFileSvr.400

Jeśli w obu systemach skonfigurowane są usługi uwierzytelniania sieciowego (NAS) i odwzorowywania tożsamości (EMI), a użytkownik został uwierzytelniony za pomocą protokołu Kerberos, możliwe jest wykorzystanie protokołu Kerberos do uwierzytelniania dostępu do systemu plików znajdującego się na docelowej platformie System i.

- Jeśli uwierzytelnianie Kerberos nie powiedzie się, do potwierdzenia dostępu może być użyty identyfikator użytkownika i hasło.



**Uwaga:** Jeśli bilet nadania biletu lub bilet systemu System i wygaśnie już po potwierdzeniu przez system docelowy uprawnień dostępu, wygaśnięcie biletu będzie miało moc dopiero po zakończeniu połączenia z systemem docelowym.

- Jeśli do uwierzytelniania nie jest wykorzystywany protokół Kerberos, to aby użytkownik mógł uzyskać dostęp do systemu plików znajdującego się na docelowej platformie System i, musi on posiadać na systemie docelowym identyfikator użytkownika i hasło odpowiadające identyfikatorowi i hasłu na systemie lokalnym.

**Uwaga:** Jeśli hasło użytkownika na systemie lokalnym lub docelowym zostanie zmienione po tym, jak system docelowy zatwierdził dostęp użytkownika, zmiana ta zostanie uwzględniona dopiero po zakończeniu połączenia użytkownika z systemem docelowym. Opóźnienie to nie będzie jednak miało miejsca, jeśli profil użytkownika na systemie docelowym zostanie usunięty, a w jego miejsce zostanie utworzony inny profil o tym samym identyfikatorze użytkownika. W tym przypadku uprawnienia dostępu do systemu docelowego zostaną zweryfikowane przez system plików QFileSvr.400.

- Uprawnienia do obiektu są określone na podstawie profilu użytkownika znajdującego się na systemie docelowym. Oznacza to, że użytkownik uzyska dostęp do obiektu znajdującego się w systemie plików systemu docelowego tylko wtedy, gdy jego profil użytkownika w systemie docelowym ma odpowiednie uprawnienia do tego obiektu.

#### **Informacje pokrewne**

Usługa uwierzytelniania sieciowego

Odwzorowywanie tożsamości dla przedsiębiorstwa (Enterprise Identity Mapping - EIM)

## **Dowiązania w systemie plików QFileSvr.400**

System plików QFileSvr.400 dopuszcza tylko jedno dowiązanie do obiektu.

W systemie plików QFileSvr.400 nie można tworzyć ani przechowywać dowiązań symbolicznych. Można jednak uzyskać dostęp do danych, używając dowiązania symbolicznego z systemu plików “root” (/), QOpenSys lub systemu plików użytkownika.

#### **Pojęcia pokrewne**

“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## **Używanie komend i ekranów zintegrowanego systemu plików w systemie plików QFileSvr.400**

Wiele z komend i ekranów zintegrowanego systemu plików działa w systemie plików QFileSvr.400.

Komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 63, z wyjątkiem następujących, działają w systemie plików QFileSvr.400:

- ADDLNK
- APYJRNCHG
- CHGAUT
- CHGJRNOBJ
- CHGOWN
- DSPAUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RST
- RTVJRNE
- SAV

- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 61.

## **Używanie funkcji API zintegrowanego systemu plików w systemie plików QFileSvr.400**

Wiele funkcji API zintegrowanego systemu plików jest dostępnych w systemie plików QFileSvr.400.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106, z wyjątkiem następujących, działają w systemie plików QFileSvr.400:

- chown()
- fchown()
- givedescriptor()
- link()
- mkfifo()
- QjoEndJournal()
- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE
- QjoStartJournal
- Qp0lGetPathFromFileID()
- symlink()
- takedescriptor()

### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **Sieciowy system plików (Network File System - NFS)**

Sieciowy system plików (Network File System - NFS) daje użytkownikom dostęp do danych i obiektów przechowywanych na zdalnym serwerze obsługującym NFS.

Sieciowy system plików może być eksportowany przez serwer NFS, a następnie dynamicznie dołączany przez klientów NFS.

Dodatkowo, każdy system plików dołączony lokalnie przez NFS będzie miał cechy i ograniczenia katalogu lub systemu plików, z którego został dołączony z serwera zdalnego. Operacje na przyłączonych systemach plików nie są wykonywane lokalnie. Żądania przechodzą przez połączenie do serwera i muszą być dostosowane do wymagań i ograniczeń systemu plików znajdującego się na serwerze.

## **Dostęp do systemów plików NFS poprzez interfejs zintegrowanego systemu plików**

System plików NFS jest dostępny poprzez interfejs zintegrowanego systemu plików. Należy pamiętać o poniższych uwagach i ograniczeniach.

### **Charakterystyka systemu plików NFS**

Charakterystyka każdego systemu plików dołączonego przez NFS zależy od typu systemu plików, który został dołączony z serwera.

Należy uzmysłowić sobie, że żądania wykonywane na tym, co wygląda jak lokalny katalog lub system plików, są w rzeczywistości realizowane na serwerze przez połączenie NFS.

Zależność klient/serwer może być myląca. Załóżmy, że użytkownik dołączył system plików QDLS z serwera "ponad" jednym z podkatalogów katalogu "root" (/) klienta. Chociaż dołączony system plików wygląda, jakby był rozszerzeniem lokalnego katalogu, w rzeczywistości funkcjonuje on jak system plików QDLS.

Uzmysłowienie sobie relacji, jakie występują w systemie w związku z dołączaniem systemów plików przez NFS, jest ważne z punktu widzenia przetwarzania żądań lokalnie i przez połączenie z serwerem. To, że komenda działa poprawnie na poziomie lokalnym, nie znaczy, że będzie ona również działać na katalogu dołączonym z serwera. Każdy katalog dołączony do klienta będzie miał cechy i charakterystykę systemu plików znajdującego się na serwerze.

## Odmiany serwerów i klientów w sieciowym systemie plików

Istnieją trzy główne możliwości połączeń klient/serwer, przy każdym z nich sieciowy system plików (NFS) funkcjonuje inaczej i inna jest jego charakterystyka.

Możliwościami tymi są:

- Użytkownik podłącza system plików z platformy System i po stronie klienta.
- Użytkownik podłącza system plików z platformy UNIX po stronie klienta.
- Użytkownik podłącza system plików po stronie klienta korzystając z systemu niebędącego ani platformą System i, ani platformą UNIX.

W pierwszym przypadku podłączony system plików zachowuje się na kliencie podobnie, jak zachowywałby się na platformie System i. Jednak obie charakterystyki systemu plików NFS i oferowanego systemu plików muszą być uwzględnione. Na przykład, jeśli użytkownik podłącza system plików QDLS z serwera do klienta, będzie on miał charakterystyki i ograniczenia systemu plików QDLS. Na przykład w systemie plików QDLS komponenty nazwy ścieżki mają ograniczenie długości do 8 znaków z 3-znakowym rozszerzeniem. Jednak podłączony system plików będzie miał także ograniczenia i charakterystykę systemu plików NFS. Na przykład, nie można użyć komendy CHGAUD do zmiany wartości kontroli obiektu systemu plików NFS.

W drugim scenariuszu należy pamiętać, że każdy system plików podłączony z serwera UNIX będzie się zachowywał podobnie do systemu plików i5/OS QOpenSys.

W trzecim przypadku należy przejrzeć dokumentację dotyczącą systemu plików związanego z systemem operacyjnym.

### Odsyłacze pokrewne

"System plików systemów otwartych (QOpenSys)" na stronie 30

System plików QOpenSys jest zgodny ze standardami systemów otwartych bazującymi na rodzinie systemów UNIX, takimi jak POSIX i XPG (X/Open Portability Guide). Podobnie jak główny system plików "root" (/), ten system plików wykorzystuje oferowaną przez zintegrowany system plików obsługę plików strumieniowych i katalogów.

## Dowiązania w systemie plików NFS

W systemie plików NFS może istnieć wiele dowiązań stałych do jednego obiektu.

W pełni obsługiwane są dowiązania symboliczne. Można używać dowiązań symbolicznych z systemu plików NFS do obiektów w innych systemach. Możliwość stosowania wielu dowiązań stałych i dowiązań symbolicznych jest całkowicie uzależniona od systemu plików, który został dołączony z NFS.

### Pojęcia pokrewne

"Dowiązanie" na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

## Używanie komend zintegrowanego systemu plików w systemie plików NFS

Wiele komend zintegrowanego systemu plików jest dostępnych również w systemie plików NFS.

Wszystkie komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 63 i ekrany opisane w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 61, z wyjątkiem następujących, działają również w sieciowym systemie plików (NFS):

- APYJRNCHG
- CHGJRNOBJ
- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP
- CHKIN
- CHKOUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- SNDJRNE
- STRJRN

Istnieją jednak komendy odnoszące się tylko do pewnych podłączonych systemów plików, w tym do NFS. Jednakże używanie tych komend może nie być bezpieczne w przypadku procesu wielowątkowego. Poniższa tabela opisuje te komendy.

Tabela 6. Komendy CL sieciowego systemu plików

Komenda	Opis
ADDMFS	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.
CHGNFSEXP	Zmiana systemu plików NFS. Operacja eksportu dodaje lub usuwa drzewa katalogów z tabeli eksportu systemów plików eksportowanych do klientów systemu NFS.
DSPMFSINF	Wyświetlenie danych podłączonego systemu plików (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.
ENDNFSSVR	Zakończenie pracy serwera NFS (End Network File System Server). Kończy działanie jednego lub wszystkich demonów NFS w serwerze.
EXPORTFS	Eksportowanie systemu plików. Dodaje lub usuwa drzewa katalogów z tabeli eksportu systemów plików eksportowanych do klientów systemu NFS.
MOUNT	Podłączenie systemu plików. Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. Jest to alternatywna postać komendy ADDMFS.
RLSIFSLCK	Zwolnienie blokad zintegrowanego systemu plików (Release in Integrated File System Locks). Zwalnia wszystkie blokady zakresu bitów NFS nałożone przez klienta lub dla obiektu.
RMVMFS	Usuwanie podłączonego FSS (Remove Mounted File System). Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego.
STRNFSSVR	Uruchomienie serwera NFS (Start Network File System Server). Rozpoczyna działanie jednego lub wszystkich demonów NFS w serwerze.
UNMOUNT	Odłączenie systemu plików. Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego. Jest to alternatywna postać komendy RMVMFS.

**Uwaga:** Aby komendy zintegrowanego systemu plików mogły wykonywać działania na systemie plików NFS, muszą być podłączone.

#### Informacje pokrewne



Obsługa sieciowego systemu plików i5/OS - plik PDF

## Używanie funkcji API zintegrowanego systemu plików w systemie plików NFS

Wiele funkcji API zintegrowanego systemu plików jest dostępnych również w systemie plików NFS.

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 106, z wyjątkiem następujących, działają w systemie plików NFS:

- mkfifo()
- QjoEndJournal()
- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Pełny opis funkcji języka C dotyczących konkretnie systemu plików NFS zawiera publikacja i5/OS Network File

System Support .

**Uwaga:** Aby wykonywać działania za pomocą funkcji API, system NFS musi być podłączony.

#### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

---

## Dostęp do zintegrowanego systemu plików

Wszystkie interfejsy użytkownika, takie jak menu, komendy i ekrany, używane do pracy z bibliotekami, obiektami, bazami danych, folderami i dokumentami w systemie, nadal działają tak, jak przed wprowadzeniem zintegrowanego systemu plików.

Interfejsów tych jednak nie można używać do pracy z plikami strumieniowymi, katalogami i innymi obiektami obsługiwanymi przez zintegrowany system plików.

Dla zintegrowanego systemu plików dostarczono osobny zestaw interfejsów. Można ich używać jako obiektów w każdym systemie plików, do którego można uzyskać dostęp poprzez katalogi zintegrowanego systemu plików.

Z poziomu systemu można pracować z katalogami i obiektami zintegrowanego systemu plików, używając do tego celu menu i ekranów lub komend języka CL (control language). Dodatkowo, aby korzystać z plików strumieniowych, katalogów i innych usług zintegrowanego systemu plików, można użyć zestawu funkcji API.

Zintegrowany system plików można także obsługiwać przy użyciu narzędzia System i Navigator, graficznego interfejsu użytkownika, służącego do zarządzania i administrowania systemem z pulpitu systemu Windows.

## Dostęp przy użyciu pozycji menu i ekranów

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych w systemie.

Aby wyświetlić menu zintegrowanego systemu plików:

1. Wpisz się do systemu.
2. Naciśnij klawisz Enter, aby kontynuować.
3. Z głównego menu wybierz opcję **Zbiory, biblioteki i foldery** (Files, Libraries, and Folders).
4. Z menu Zbiory, biblioteki i foldery (Files, Libraries, and Folders) wybierz opcję **zintegrowany system plików**.

Z poziomu tego ekranu można, zależnie od potrzeb, używać komend katalogów, komend obiektów lub komend ochrony w zintegrowanym systemie plików. Jeśli jednak zna się komendę CL, można ją wpisać w wierszu komend u dołu ekranu i nacisnąć **Enter**, pomijając opcje menu.

Zintegrowany system plików stanie się również dostępny z dowolnego menu systemu po wykonaniu następujących czynności:

1. W dowolnym wierszu komend wpisz **GO DATA**, aby wyświetlić menu Zbiory, biblioteki i foldery (Files, Libraries, and Folders).
2. Wybierz opcję **zintegrowany system plików**.

Aby wyświetlić menu komend Network File System, należy wpisać **GO CMDNFS** w dowolnym wierszu komend. Aby wyświetlić menu komend dotyczących systemu plików definiowanego przez użytkownika (UDFS), należy wpisać **GO CMDUDFS**.

Z menu zintegrowanego systemu plików możliwe jest wywoływanie ekranów lub komend, przy pomocy których można wykonywać następujące operacje:

- tworzyć, konwertować i usuwać katalogi;
- wyświetlać i zmieniać nazwę bieżącego katalogu;
- dodawać, wyświetlać, zmieniać i usuwać dowiązania do obiektu;
- kopiować, przenosić i usuwać obiekty;
- blokować (check out) i odblokowywać (check in) obiekty;
- składować (back up) i odtwarzać obiekty;
- wyświetlać i zmieniać właścicieli obiektów i uprawnienia użytkowników do obiektów;
- wyświetlać i zmieniać atrybuty obiektów;
- kopiować dane między plikami strumieniowymi i podzbiorami baz danych;
- tworzyć i usuwać systemy plików definiowanych przez użytkownika oraz wyświetlać ich status;
- eksportować systemy plików z serwera;
- podłączać (mount) i odłączać (unmount) systemy plików od komputera typu klient.

Niektóre systemy plików nie obsługują wszystkich powyższych operacji.

#### **Pojęcia pokrewne**

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

#### **Odsyłacze pokrewne**

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 66

Gdy do wykonywania działań na obiekcie używane są komendy lub ekrany zintegrowanego systemu plików, obiekt identyfikowany jest przez podanie jego właściwej ścieżki.

“Dostęp przy użyciu komend języka CL” na stronie 63

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także przeprowadzić, wpisując komendy języka CL. Przy ich użyciu można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

## Dostęp przy użyciu komend języka CL

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także przeprowadzić, wpisując komendy języka CL. Przy ich użyciu można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

Tabela 1 zawiera spis komend zintegrowanego systemu plików. Więcej informacji o komendach CL związanych z systemami plików użytkownika, systemem Network File System i podłączonym systemem plików zawierają tematy “Systemy plików użytkownika (UDFS)” na stronie 32 i “Sieciowy system plików (Network File System - NFS)” na stronie 58. Jeśli komenda wykonuje tę samą operację, co komenda systemu OS/2 lub DOS, utworzono alias (alternatywną nazwę komendy) dla wygody użytkowników systemów OS/2 i DOS.

Tabela 7. Komendy zintegrowanego systemu plików

Komenda	Opis	Alias
ADDLNK <sup>3</sup>	Dodanie dowiązania (Add Link). Dodaje dowiązanie między katalogiem i obiektem.	
ADDMFS <sup>3</sup>	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.	MOUNT
APYJRNCHG <sup>2 3</sup>	Zastosowanie kronikowanych zmian (Apply Journalled Changes). Korzysta z pozycji kronik do wprowadzenia zmian, które wystąpiły od ostatniego składowania obiektu kronikowanego lub do wprowadzenia zmian do określonego punktu.	
CHGATR <sup>3</sup>	Zmiana atrybutu (Change Attribute). Zmienia atrybut obiektu, grupy obiektów lub drzewa katalogów, przy czym atrybut ten zostaje zmieniony dla katalogu oraz zawartości jego i wszystkich podkatalogów tego katalogu.	
CHGAUD <sup>3</sup>	Zmiana wartości kontroli (Change Auditing Value). Włącza lub wyłącza kontrolę dla obiektu.	
CHGAUT <sup>3</sup>	Zmiana uprawnień (Change Authority). Nadaje konkretne uprawnienia do obiektu użytkownikowi lub grupie użytkowników.	
CHGCURDIR <sup>3</sup>	Zmiana bieżącego katalogu (Change Current Directory). Zmienia katalog, który ma być używany jako bieżący.	CD, CHDIR
CHGJRNOBJ <sup>2 3</sup>	Zmiana obiektów kronikowanych. Zmienia atrybuty kronikowania dla obiektu lub listy obiektów bez konieczności zakończenia i ponownego uruchomienia kronikowania dla obiektu.	
CHGNFSEXP	Zmiana eksportu NFS (Change Network File System Export). Dodaje drzewa katalogów lub usuwa je z tabeli eksportu systemów plików, które są eksportowane do klientów NFS.	EXPORTFS
CHGOWN <sup>3</sup>	Zmiana właściciela (Change Owner). Zmienia właściciela obiektu.	
CHGPGP <sup>3</sup>	Zmiana grupy podstawowej (Change Primary Group). Zmienia grupę podstawową z jednego użytkownika na innego.	
CHKIN <sup>3</sup>	Odblokowanie (Check In). Odblokowuje obiekt, który był uprzednio zablokowany.	
CHKOBJITG <sup>3</sup>	Sprawdzanie integralności obiektu (Check Object Integrity). Sprawdza, czy nie występują naruszenia integralności w obiektach.	
CHKOUT <sup>3</sup>	Zablokowanie (Check Out). Blokuję obiekt, co uniemożliwia innym użytkownikom dokonywanie w nim zmian.	
CPY <sup>3</sup>	Kopiowanie (Copy). Kopiuje pojedynczy obiekt lub grupę obiektów.	COPY
CPYFRMSTMF <sup>3</sup>	Kopiowanie z pliku strumieniowego (Copy from Stream File). Kopiuje dane z pliku strumieniowego do podzbioru bazy danych.	

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
CPYTOSTMF <sup>3</sup>	Kopiowanie do pliku strumieniowego (Copy to Stream File). Kopiuje dane z podzbioru bazy danych do pliku strumieniowego.	
CRTDIR <sup>3</sup>	Tworzenie katalogu (Create Directory). Dodaje nowy katalog do systemu.	MD, MKDIR
CRTUDFS <sup>3</sup>	Tworzenie systemu plików UDFS (Create UDFS). Tworzy system plików definiowany przez użytkownika (UDFS).	
CVTDIR	Konwersja katalogu (Convert directory). Dostarcza informacji o konwersji katalogów zintegrowanego systemu plików z formatu *TYPE1 do formatu *TYPE2.	
CVTRPCSRC	Konwersja kodu źródłowego RPC (Convert RPC Source). Generuje kod w języku C na podstawie pliku wejściowego napisanego w języku RPC.	RPCGEN
DLTUDFS <sup>3</sup>	Usunięcie systemu plików UDFS (Delete UDFS). Usuwa system plików definiowany przez użytkownika (UDFS).	
DSPAUT	Wyświetlenie uprawnień (Display Authority). Wyświetla listę użytkowników mających uprawnienia do obiektu i ich uprawnienia do tego obiektu.	
DSPCURDIR	Wyświetlenie bieżącego katalogu (Display Current Directory). Wyświetla nazwę bieżącego katalogu.	
DSPJRN <sup>2 3</sup>	Wyświetlenie kroniki (Display Journal). Konwertuje pozycje kroniki (zawarte w jednym lub wielu dziennikach) na postać umożliwiającą zewnętrzną reprezentację.	
DSPLNK	Wyświetlenie dowiązań obiektu (Display Object Links). Wyświetla listę obiektów w katalogu, zawiera opcje umożliwiające wyświetlenie informacji o obiektach.	
DSPF	Wyświetlenie pliku strumieniowego (Display Stream File). Wyświetla plik strumieniowy lub zbiór bazy danych.	
DSPMFSINF	Wyświetlenie danych podłączonego systemu plików (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.	STATFS
DSPUDFS	Wyświetlenie systemu plików UDFS (Display UDFS). Wyświetla system plików definiowany przez użytkownika (UDFS).	
EDTF	Edycja zbioru strumieniowego (Edit Stream File). Przeprowadza edycję pliku strumieniowego lub zbioru bazy danych.	
ENDJRN <sup>2 3</sup>	Zakończenie kronikowania (End Journal). Kończy kronikowanie zmian dla obiektu lub listy obiektów.	
ENDNFSSVR	Zakończenie pracy serwera NFS (End Network File System Server). Kończy działanie jednego lub wszystkich demonów NFS w serwerze.	
ENDRPCBIND	Zakończenie demona konsolidacji RPC (End RPC Binder Daemon). Kończy działanie demona RPCBind.	
MOV <sup>3</sup>	Przeniesienie (Move). Przenosi obiekt do innego katalogu.	MOVE
PRTDIRINF	Drukuj informacje katalogowe (Print Directory Information). Dla obiektów w zintegrowanym systemie plików służy do drukowania informacji katalogu, zebranych przez komendę Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF).	
RCLLNK <sup>3</sup>	Odzyskanie dowiązań obiektu (Reclaim Object Links). Identyfikuje i, jeśli jest to możliwe, usuwa problemy z używanymi podłączonymi systemami plików.	



Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
RCVJRNE <sup>2 3</sup>	Pobranie pozycji kroniki (Receive Journal Entry). Umożliwia określonemu programowi obsługi wyjścia użytkownika otrzymywanie pozycji kroniki w sposób ciągły.	
RLSIFSLCK <sup>3</sup>	Zwolnienie blokad zintegrowanego systemu plików (Release in Integrated File System Locks). Zwalnia wszystkie blokady zakresu bitów nałożone na klienta NFS lub na obiekt.	
RMVDIR <sup>3</sup>	Usuwanie katalogu (Remove Directory). Usuwa katalog z systemu.	RD, RMDIR
RMVLNK <sup>3</sup>	Usuwanie dowiązania (Remove Link). Usuwa dowiązanie do obiektu.	DEL, ERASE
RMVMFS <sup>3</sup>	Usuwanie podłączonego FSS (Remove Mounted File System). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnych katalogów klienta.	UNMOUNT
RNM <sup>3</sup>	Zmiana nazwy (Rename). Zmienia nazwę obiektu w katalogu.	REN
RPCBIND	Uruchomienie demona konsolidacji RPC (Start RPC Binder Daemon). Uruchamia demon RPCBind.	
RST <sup>3</sup>	Odtwarzanie (Restore). Kopiuje obiekt lub grupę obiektów z urządzenia składowania do systemu.	
RTVCURDIR	Odtworzenie bieżącego katalogu (Retrieve Current Directory). Odtwarza nazwę bieżącego katalogu i wstawia ją do określonej zmiennej (używanej w programach języka CL).	
RTVDIRINF	Pobranie informacji katalogu. Zbiera atrybuty obiektów w zintegrowanym systemie plików.	
RTVJRNE <sup>2 3</sup>	Odtworzenie pozycji kroniki (Retrieve Journal Entry). Pobiera określoną pozycję kroniki i umieszcza wyniki w zmiennych języka CL.	
SAV <sup>3</sup>	Składowanie (Save). Kopiuje obiekt lub grupę obiektów z systemu do urządzenia składowania.	
SNDJRNE <sup>2 3</sup>	Wysłanie pozycji do kroniki (Send Journal Entry). Dodaje do dziennika pozycje kroniki użytkownika opcjonalnie powiązane z kronikowanym obiektem.	
STRJRN <sup>2 3</sup>	Uruchomienie kronikowania (Start Journal). Uruchamia kronikowanie zmian (dotyczących obiektu lub listy obiektów w określonej kronice).	
STRNFSSVR	Uruchomienie serwera NFS (Start Network File System Server). Uruchamia jeden lub wszystkie demony NFS w serwerze.	
WRKAUT	Praca z uprawnieniami (Work with Authority). Wyświetla listę użytkowników i ich uprawnienia, a także zawiera opcje dodawania użytkowników, zmiany ich uprawnień i usuwania ich.	
WRKLNK	Praca z dowiązaniem obiektów (Work with Object Links). Wyświetla listę obiektów w katalogu i zawiera opcje przeprowadzania działań na obiektach.	
WRKOBJOWN <sup>1</sup>	Praca z obiektami według właścicieli (Work with Objects by Owner). Wyświetla listę obiektów należących do profilu użytkownika i zawiera opcje przeprowadzania działań na obiektach.	
WRKOBJPGP <sup>1</sup>	Praca z obiektami według grupy głównej (Work with Objects by Primary Group). Wyświetla listę obiektów kontrolowanych przez grupę podstawową i zawiera opcje przeprowadzania działań na obiektach.	

**Uwagi:**

1. Komendy WRKOBJOWN i WRKOBJPGP mogą wyświetlać wszystkie typy obiektów, ale jest możliwe, że nie będą w pełni funkcjonować we wszystkich systemach plików.

2. Więcej informacji na ten temat zawiera sekcja Zarządzanie kronikowaniem w Centrum informacyjnym i5/OS.
3. Komendy te obsługują format Unicode. Więcej informacji zawiera sekcja Obsługa kodu Unicode w języku control language w Centrum informacyjnym i5/OS.

### Pojęcia pokrewne

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 61

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych w systemie.

### Informacje pokrewne

Język CL

## Reguły dotyczące nazw ścieżek dla komend CL i ekranów

Gdy do wykonywania działań na obiekcie używane są komendy lub ekrany zintegrowanego systemu plików, obiekt identyfikowany jest przez podanie jego właściwej ścieżki.

Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając nazwy ścieżek. Według nich określenie *obiekt* odnosi się do dowolnego katalogu, pliku, dowiązania lub innego obiektu:

- Nazwy obiektów wewnątrz każdego katalogu muszą być unikalne.
- Nazwa ścieżki przekazana do komendy CL zintegrowanego systemu plików musi być przedstawiona za pomocą identyfikatora kodowanego zestawu znaków (CCSID) obowiązującego dla danego zadania. Jeśli identyfikatorem CCSID zadania jest liczba 65535, nazwa ścieżki musi zostać przedstawiona w domyślnym identyfikatorze CCSID zadania. Ponieważ łańcuchy tekstowe są zwykle kodowane za pomocą identyfikatora CCSID 37, przed przekazaniem nazwy ścieżki do komendy, należy ją poddać konwersji na identyfikator CCSID zadania.

**Uwaga:** W przypadku tych komend, które obsługują Unicode, ograniczenie niniejsze nie istnieje, pod warunkiem że komendy są wywołane tak, aby wykorzystać obsługę Unicode. Można na przykład wywołać funkcję API QCAPCMD za pomocą komendy i informacji dotyczących nazwy ścieżki w CCSID Unicode. Więcej informacji zawiera sekcja Obsługa kodu Unicode w języku control language w Centrum informacyjnym i5/OS.

- W wierszu komend ścieżki muszą się znajdować w znakach pojedynczego cudzysłowu ('). Gdy ścieżki wpisuje się na ekranach, apostrofy nie są obowiązkowe. Jeśli ścieżka zawiera ciągi znaków w cudzysłowie, należy ją ująć w znaki ' '.
- Ścieżkę należy wprowadzać od lewej do prawej strony, poczynając od najwyższego poziomu katalogu i kończąc na obiekcie, na którym ma być wykonana komenda. Nazwy składników ścieżki oddzielane są ukośnikiem (/).

**Uwaga:** W przypadku niektórych komend CL dozwolony jest również ukośnik odwrotny (\), który może służyć jako znak rozdzielający dokonujący automatycznie konwersji ukośnika odwrotnego (\) na ukośnik (/). Jednak dla niektórych komend CL ukośnik odwrotny (\) nie spełnia powyższej dodatkowej funkcji. Z tego względu powyższego znaku rozdzielającego należy używać z ostrożnością.

Na przykład:

'Kat1/Kat2/Kat3/ZbiórUż'

lub

'Dir1\Dir2\Dir3\UsrFile'

- Znaków ukośnika (/), ukośnika odwrotnego (\) oraz znaków pustych nie można używać w poszczególnych składnikach ścieżki, gdy znaki te pełnią funkcję znaków rozdzielających. Komendy nie zamieniają małych liter na wielkie. Nazwa może zostać zamieniona na pisaną wielkimi literami lub nie, w zależności od tego, czy system plików zawierający obiekt uwzględnia wielkość liter i czy obiekt jest tworzony, czy poszukiwany.

- Długość nazwy obiektu jest ograniczona przez system plików, w którym znajduje się obiekt, i przez maksymalną długość ciągu komendy. Komendy akceptują nazwy obiektów o maksymalnej długości 255 znaków i ścieżki o maksymalnej długości 5000 znaków.

- Znak rozdzielający (na przykład /) na początku ścieżki oznacza, że ścieżka rozpoczyna się od najwyższego katalogu, na przykład: "root" (/).

```
'/Kat1/Kat2/Kat3/ZbiórUż'
```

- Jeśli ścieżka nie rozpoczyna się znakiem rozdzielającym (na przykład /), to przyjmuje się, że rozpoczyna się w bieżącym katalogu użytkownika, który wprowadził komendę, na przykład:

```
'MójKat/MójZbiór'
```

gdzie MójKat jest podkatalogiem bieżącego katalogu użytkownika.

- Znak tyldy (~) ze znakiem separatora (na przykład /) na początku ścieżki oznacza, że ścieżka rozpoczyna się w katalogu osobistym użytkownika, który wprowadził komendę, na przykład:

```
'~/KatUżytk/ObiektUżytk'
```

- Znak tyldy (~), po którym występuje nazwa użytkownika, a następnie znak separatora (na przykład /) na początku nazwy ścieżki oznacza, że ścieżka rozpoczyna się w katalogu osobistym użytkownika określonego przez podaną nazwę, na przykład:

```
'~nazwa-użytkownika/KatUżytk/ObiektUżytk'
```

- W niektórych komendach, aby znaleźć nazwy odpowiadające wzorcowi, można w ostatnim składniku ścieżki użyć gwiazdki (\*) lub znaku zapytania (?). Znak \* oznacza, że system ma szukać nazw zawierających dowolną liczbę znaków w miejscu znaku \*. Znak ? oznacza, że system ma szukać nazw zawierających pojedynczy znak w miejscu znaku ?. Poniższy przykład wskazuje wszystkie obiekty, których nazwa zaczyna się znakiem *d* i kończy ciągiem *txt*:

```
'/Kat1/Kat2/Kat3/d*txt'
```

Poniższy przykład wskazuje wszystkie obiekty, których nazwa zaczyna się znakiem *d*, po którym występuje dowolny pojedynczy znak, i kończy ciągiem *txt*:

```
'/Kat1/Kat2/Kat3/d?txt'
```

- Aby uniknąć pomyłek związanych z wartościami specjalnymi systemu i5/OS, nazwy ścieżek nie mogą rozpoczynać się od pojedynczego znaku gwiazdki (\*). Do określania ogólnego wzorca nazwy ścieżki służą dwie gwiazdki (\*\*) na początku nazwy, na przykład:

```
'**,plik'
```

**Uwaga:** Dotyczy to tylko względnych nazw ścieżek, w których przed gwiazdką (\*) nie występują inne znaki.

- Podczas działań na obiektach w systemie plików QSYS.LIB nazwy składników muszą przybrać postać *nazwa.typ-obiektu*, na przykład:

```
'/QSYS.LIB/PAY.LIB/TAX.FILE'
```

- Podczas działań na obiektach w systemie plików QSYS.LIB niezależnej puli ASP nazwy składników muszą przybrać postać *nazwa.asp-nazwa.typ-obiektu*, na przykład:

```
'/nazwa_asp/QSYS.LIB/PAYDAVE.LIB/PAY.FILE'
```

- Ścieżka musi znajdować się w dodatkowym cudzysłowie pojedynczym (') lub w cudzysłowie ("), jeśli użyto jednego z poniższych znaków:

- gwiazdka (\*),

**Uwaga:** Aby uniknąć pomyłek związanych z wartościami specjalnymi systemu i5/OS, nazwy ścieżek nie mogą rozpoczynać się od pojedynczego znaku gwiazdki (\*).

- znak zapytania (?),
- cudzysłów pojedynczy ('),
- cudzysłów ("),
- tylda (~), jeśli została użyta jako pierwszy znak w pierwszym członie nazwy ścieżki (w każdym innym miejscu tylda jest interpretowana jak normalny znak).

Na przykład:

```
'"/Dir1/Dir/A*Smith"'
```

lub

```
'''/Dir1/Dir/A*Smith'''
```

Jednak nie jest to zalecane, ponieważ znaczenie znaku w łańcuchu komendy może być mylnie zinterpretowane, a niepoprawne wpisanie łańcucha znaków jest częstym błędem.

- W nazwach ścieżek nie wolno używać dwukropka (:). Ma on specjalne znaczenie w systemie.
- Podczas przetwarzania komend i związanych z nimi ekranów użytkownika system nie rozpoznaje znaków o kodach mniejszych niż 40 (szesnastkowo) jako znaków, których można użyć w komendach lub na ekranach. Gdy używa się takich znaków, muszą one zostać wprowadzone w postaci szesnastkowej, tak jak poniżej:

```
crtdir dir(x'02')
```

Dlatego nie zaleca się używania w nazwie ścieżki znaków o kodach mniejszych niż 40 (szesnastkowo).

Ograniczenie to odnosi się tylko do komend i związanych z nimi ekranów, a nie do funkcji API. Dodatkowo, wartość szesnastkowa 0 jest niedozwolona w nazwach ścieżek.

### Pojęcia pokrewne

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki.

Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

### Informacje pokrewne

Język CL

## Praca z danymi wyjściowymi komend RTVDIRINF i PRDIRINF

Komenda RTVDIRINF służy do zbierania atrybutów dla obiektów, które zawiera. Zebrane informacje przechowywane są w zbiorach baz danych (tabelach), których nazwy zawierają prefiks pliku informacji określony przez parametr INFFILEPFX. Tabele tworzone są w bibliotece określonej w parametrze INFLIB.

W wyniku użycia komendy RTVDIRINF tworzone są trzy tabele. Jedna zawiera atrybuty obiektów, druga katalogi, a trzecia określa pliki użyte do zapisania atrybutów obiektów.

- | W wersji V6R1 i późniejszych można te trzy tabele tworzyć również za pomocą programu System i Navigator. Więcej informacji znajduje się w temacie “Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator” na stronie 80.

Poniższy temat opisuje pola tabeli, w której przechowywane są atrybuty obiektów. Jeśli dla parametru określającego prefiks pliku z informacjami (INFFILEPFX) podano wartość \*GEN, zbiory bazy danych zostaną utworzone z unikalnym prefiksem wygenerowanym przez tę komendę. Prefiks ten będzie składał się ze znaków QAEZD i czterech cyfr. Dla plików utworzonych do zapisania zebranych informacji przydzielone zostaną nazwy zawierające przedrostek, a następnie literę D (dla pliku z informacjami o katalogach) lub O (dla pliku zawierającego informacje o obiektach w katalogach). Na przykład, gdy komenda zostanie wpisana po raz pierwszy z parametrem \*GEN, w bibliotece podanej w parametrze Biblioteka z informacjami (Information library - INFLIB) utworzone zostaną pliki QAEZD0001D i QAEZD0001O. Użytkownicy mogą podawać własne prefiksy nazw plików dla bazy danych o długości do dziewięciu znaków.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów)

Nazwa pola	Typ pola	Opis pola
QEZDIRIDX	INTEGER	Identyfikator relacyjny używany między tabelami katalogów i obiektów. Może on służyć do łączenia tabel w celu uzyskania danych pełnej nazwy ścieżki. Wartość w polu QEZDIRIDX tabeli obiektów odpowiada wartości QEZDIRIDX w tabeli katalogów dotyczącej katalogu macierzystego obiektu. <b>Uwaga:</b> Jeśli jako parametr wejściowy komendy RTVDIRINF zostanie podany katalog główny (/), pole QEZDIRIDX będzie mieć wartość 1, chociaż katalog główny (/) nie posiada katalogu nadrzędnego.
QEZOBJNAM <sup>1</sup>	VARGRAPHIC (1024)	Nazwa obiektu. <sup>2</sup>

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZOBJLEN	INTEGER	Liczba bajtów zawartych w nazwie obiektu (pole QEZOBJNAM).
QEZNMCCSID	INTEGER	Identyfikator CCSID oznaczający nazwę obiektu (pole QEZOBJNAM).
QEZREGION	GRAPHIC (2)	Dwuznakowy identyfikator kraju nazwy obiektu (pole QEZOBJNAM). Decyduje on o działaniach definiowanych według lokalizacji, takich jak kolejność zestawiania.
QEZLANGID	GRAPHIC (3)	Trzyznakowy identyfikator oznaczający język, w którym zapisana jest nazwa obiektu (pole QEZOBJNAM).
QEZMODE	INTEGER	Tryb i rodzaj dostępu do pliku. Więcej informacji o trybie można znaleźć w sekcji dotyczącej funkcji API otwierania plików open().
QEZOBJTYPE <sup>1</sup>	GRAPHIC (10)	Typ obiektu.
QEZCCSID	INTEGER	Identyfikator CCSID danych i atrybutów rozszerzonych obiektu.
QEZALCSIZE <sup>1</sup>	BIGINT	Liczba bajtów przydzielonych dla obiektu.
QEZDTASIZE	BIGINT	Wielkość danych w obiekcie podana w bajtach. Wielkość ta nie obejmuje nagłówków obiektów ani wielkości atrybutów rozszerzonych przypisanych do obiektu.
QEZEAS	BIGINT	Liczba atrybutów rozszerzonych przypisanych do obiektu.
QEZCEAS	BIGINT	Liczba krytycznych atrybutów rozszerzonych przypisanych do obiektu.
QEZEXTATRS	BIGINT	Całkowita liczba bajtów dla wszystkich danych atrybutów rozszerzonych.
QEZCRTTIM	TIMESTAMP	Data i godzina utworzenia obiektu.
QEZACCTIM	TIMESTAMP	Data i godzina ostatniego dostępu do danych obiektu.
QEZCHGTIMA <sup>1</sup>	TIMESTAMP	Data i godzina ostatniej zmiany atrybutów obiektu.
QEZCHGTIMD	TIMESTAMP	Data i godzina ostatniej zmiany danych obiektu.
QEZSTGFREE <sup>1</sup>	SMALLINT	Informacje o tym, czy dane obiektu zostały odłączone ze zwolnieniem pamięci. Poprawne wartości:  0 - Dane obiektu nie zostały odłączone.  1 - Dane obiektu zostały odłączone.
QEZCHKOUT <sup>1</sup>	SMALLINT	Wskaźnik sprawdzenia obiektu. Poprawne wartości:  0 - Obiekt nie jest sprawdzany.  1 - Obiekt jest sprawdzany.
QEZCHKOWN	GRAPHIC (10)	Użytkownik, który uruchomił sprawdzenie obiektu. Pole to jest puste, jeśli sprawdzenie nie zostało zaznaczone.
QEZCHKTIM	TIMESTAMP	Data i godzina ostatniego sprawdzenia obiektu. Jeśli obiekt nie został pobrany, pole będzie zawierać wartość NULL.
QEZLOCAL	SMALLINT	Określa, czy obiekt jest przechowywany lokalnie, czy w systemie zdalnym. Decyzja o miejscu przechowywania obiektu zależy od reguł konkretnego systemu plików. Obiekty w systemach plików, które nie zawierają indykatora lokalnego lub zdalnego, traktowane są jak obiekty zdalne. Poprawne wartości:  1 - Dane obiektu przechowywane są lokalnie.  2 - Dane obiektu przechowywane są w systemie zdalnym.
QEZOWN <sup>1</sup>	GRAPHIC (10)	Nazwa profilu użytkownika będącego właścicielem obiektu lub następująca wartość specjalna:  *NOUSRPRF - Ta wartość specjalna używana jest przez system plików Network File System do wskazania, że w lokalnym serwerze iSeries nie ma profilu użytkownika z ID użytkownika (UID) odpowiadającym UID obiektu zdalnego.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZUID	INTEGER	Każdy użytkownik w systemie musi mieć unikalny numer identyfikacyjny użytkownika (UID).
QEZOWNPGP	GRAPHIC (10)	Nazwa profilu użytkownika będącego grupą podstawową obiektu lub następujące wartości specjalne:  *NONE - Obiekt nie posiada skojarzonej z nim grupy podstawowej.  *NOUSRPRF - Ta wartość specjalna używana jest przez system plików Network File System do wskazania, że w lokalnym serwerze nie ma profilu użytkownika z ID grupy (GID) odpowiadającym GID obiektu zdalnego.
QEZGID	INTEGER	Profile grup identyfikowane są według unikalnych numerów identyfikacyjnych grup (GID).
QEZAUTLST	GRAPHIC (10)	Nazwa listy autoryzacji używanej do zabezpieczania wskazanego obiektu. Wartość *NONE oznacza, że do określenia uprawnień do obiektu nie jest używana żadna lista autoryzacji.
QEZASP	SMALLINT	Pula pamięci dyskowej, w której obiekt jest przechowywany.
QEZJRNSTS <sup>1</sup>	SMALLINT	Aktualny status kronikowania obiektu. W polu będzie jedna z następujących wartości:  0 (NOT_JOURNALED) - Obiekt nie jest aktualnie kronikowany.  1 (JOURNALED) - Obiekt jest aktualnie kronikowany.
QEZJSUBTRE	SMALLINT	Gdy flaga ta zostanie zwrócona, oznacza, że obiekt jest katalogiem z semantyką poddrzewa kronikowania zintegrowanego systemu plików.  0 - Obiekt nie jest kronikowany z semantyką poddrzewa.  1 - Obiekt jest kronikowany z semantyką poddrzewa. Nowe obiekty utworzone w poddrzewie tego katalogu odziedziczają atrybuty kronikowania i opcje katalogu.
QEZJOPTENT	SMALLINT	Gdy kronikowanie jest aktywne, kronikowane są pozycje opcjonalne. Lista pozycji opcjonalnych różni się zależnie od typu obiektu.  0 - Obiekt nie jest kronikowany z pozycjami opcjonalnymi.  1 - Obiekt jest kronikowany z pozycjami opcjonalnymi.
QEZJAFTERI	SMALLINT	Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu po wprowadzeniu zmiany.  0 - Obiekt nie jest kronikowany z obrazami po zmianach.  1 - Obiekt jest kronikowany z obrazami po zmianach.
QEZJBEFORI	SMALLINT	Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu przed wprowadzeniem zmian.  0 - Obiekt nie jest kronikowany z obrazami przed wprowadzeniem zmian.  0 - Obiekt jest kronikowany z obrazami przed wprowadzeniem zmian.
QEZJRNID	GRAPHIC (10)	Pole to przypisuje do kronikowanego obiektu identyfikator, którego można użyć w różnych komendach i funkcjach API związanych z kronikowaniem. Pole to jest puste, jeśli obiekt nigdy nie był kronikowany.
QEZJRNNAM	GRAPHIC (10)	Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę aktualnie używanej kroniki. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę kroniki ostatnio używanej dla obiektu. Jeśli obiekt nigdy nie był kronikowany, wszystkie bajty wartości tego pola zostaną wyzerowane binarnie. Pole to jest puste, jeśli obiekt nigdy nie był kronikowany.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZRNLB	GRAPHIC (10)	Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się aktualnie używana kronika. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się kronika ostatnio używana dla obiektu. Jeśli obiekt nigdy nie był kronikowany, wszystkie bajty wartości tego pola zostaną wyzerowane binarnie. Pole to jest puste, jeśli obiekt nigdy nie był kronikowany.
QEZRNRSTR	TIMESTAMP	Liczba sekund, które upłynęły od epoki odpowiadającej dacie i godzinie uruchomienia kronikowania obiektu. Jeśli obiekt nigdy nie był kronikowany, wartością tego pola będzie binarne zero. Pole ma wartość NULL, jeśli obiekt nigdy nie był kronikowany.
QEZAUDT	GRAPHIC (10)	Wartość kontroli przypisana do obiektu. Poprawne wartości:  *NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.  *USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.  *CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.  *ALL - Kontrola obiektu jest przeprowadzana dla każdego dostępu do obiektu, uzyskanego przez dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.  *NOTAVL - Użytkownik wykonujący operację nie jest uprawniony do pobrania aktualnej wartości kontrolowania obiektu.
QEZBLKSIZ	INTEGER	Wielkość bloku obiektu.
QEZNLNK	INTEGER	Liczba dowiązań stałych do obiektu.
QEZFILEID <sup>1</sup>	GRAPHIC (16)	ID zbioru dla obiektu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu.
QEZFILEIDS	INTEGER	4-bajtowy ID zbioru. Numer ten jednoznacznie identyfikuje obiekt w systemie plików. Nie może on identyfikować obiektu w całym systemie.
QEZGENID	BIGINT	ID generacji przypisany do ID zbioru.
QEZFSD	BIGINT	ID systemu plików, w którym znajduje się obiekt. Numer ten jednoznacznie identyfikuje system plików, w którym znajduje się obiekt.
QEZRDEV	BIGINT	Jeśli obiekt reprezentuje plik specjalny urządzenia, jest to rzeczywiste urządzenie reprezentowane przez ten obiekt.
QEZDOM	GRAPHIC (10)	Domena obiektu. Poprawne wartości:  *SYSTEM - Obiekt istnieje w domenie systemowej.  *USER - Obiekt istnieje w domenie użytkownika.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZCRTAUD	GRAPHIC (10)	<p>Wartość kontroli przypisana do obiektu utworzonego w katalogu. Poprawne wartości:</p> <p>*NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.</p> <p>*USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.</p> <p>*CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.</p> <p>*ALL - Kontrola obiektu jest przeprowadzana po każdym dostępie do obiektu dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.</p> <p>*NOTAVL - Użytkownik wykonujący operację nie jest uprawniony do pobrania aktualnej wartości kontroli tworzonego obiektu.</p> <p>*SYSVAL - Wartość kontroli dla obiektów tworzonych w katalogu jest określana na podstawie systemowej wartości kontroli (QCRTOBJAUD).</p>
QEZSCN	GRAPHIC (1)	<p>Określa, czy obiekt ma podlegać skanowaniu po zarejestrowaniu programów obsługi wyjścia w jednym ze zintegrowanych punktów wyjściowych systemu plików związanych ze skanowaniem.</p> <p>Poprawne wartości:</p> <p>x'00' (SCANNING_NO) - obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>x'01' - obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>x'02' (SCANNING_CHGONLY) - obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania tylko w przypadku modyfikacji od obiektu momentu ostatniego skanowania obiektu. Nie będzie podlegać skanowaniu w przypadku aktualizacji oprogramowania do skanowania. Atrybut ten obowiązuje tylko wtedy, gdy podano wartość *USECOATR wartości systemowej Sterowanie skanowaniem systemów plików (QSCANFCTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p>



Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZINHSCN	GRAPHIC (1)	<p>Określa, czy obiekty utworzone w katalogu mają podlegać skanowaniu po zarejestrowaniu programów obsługi wyjścia w jednym ze zintegrowanych punktów wyjścia systemu plików związanych ze skanowaniem.</p> <p>Poprawne wartości:</p> <p>x'00' - po utworzeniu w katalogu obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCNTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>x'01' - po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>x'02' - po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu od momentu ostatniego skanowania obiektu. Nie będzie podlegać skanowaniu w przypadku aktualizacji oprogramowania do skanowania. Atrybut ten obowiązuje tylko wtedy, gdy podano wartość *USEOCOATR wartości systemowej Sterowanie skanowaniem systemów plików (QSCANFSCNTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCNTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p>
QEZSTATUS	GRAPHIC (1)	<p>Status skanowania przypisany do obiektu. W polu będzie jedna z następujących wartości:</p> <p>x'00' (SCAN_REQUIRED) - Skanowanie jest wymagane dla obiektu, ponieważ nie podlegał on jeszcze skanowaniu przez programy obsługi wyjścia dotyczące skanowania lub dane obiektu bądź identyfikator CCSID uległ modyfikacji od ostatniego skanowania. Przykłady modyfikacji danych obiektu lub identyfikatora CCSID to: zapis do obiektu, bezpośrednio lub przez odwzorowanie pamięci, obcięcie obiektu, usunięcie obiektu i zmiana atrybutu identyfikatora CCSID obiektu.</p> <p>x'01' (SCAN_SUCCESS) - Obiekt podlegał skanowaniu przez program obsługi wyjścia dotyczący skanowania i dla ostatniego żądania skanowania operacja skanowania obiektu zakończyła się pomyślnie.</p> <p>x'02' (SCAN_FAILURE) - Obiekt podlegał skanowaniu przez program obsługi wyjścia dotyczący skanowania i dla ostatniego żądania skanowania operacja skanowania obiektu nie powiodła się i nie została zakończona. Po oznaczeniu obiektu jako błędny, nie będzie on podlegał kolejnemu skanowaniu do czasu zmiany sygnatury skanowania obiektu na inną niż sygnatura klucza globalnego skanowania lub sygnatura klucza skanowania grupowego niezależnej ASP. Dlatego kolejne żądania pracy z obiektem będą powodować błąd i wyświetlenie informacji o niepowodzeniu skanowania. Przykłady żądań, które spowodują błąd, to otwarcie obiektu, zmiana identyfikatora CCSID lub kopiowanie obiektu.</p> <p>x'05' (SCAN_PENDING_CVN) - Obiekt nie znajduje się w katalogu *TYPE2 i dlatego nie będzie poddany skanowaniu do czasu przeprowadzenia konwersji katalogu.</p> <p>x'06' (SCAN_NOT_REQUIRED) - Obiekt nie wymaga skanowania, ponieważ jest oznaczony jako nie do skanowania.</p>

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZSSIGDF	GRAPHIC (1)	<p>Sygnatury skanowania oznaczają poziom obsługi oprogramowania do skanowania.</p> <p>Gdy obiekt jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z powiązaną sygnaturą skanowania grupy niezależnych pul ASP. Gdy obiekt nie jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z wartością globalnej sygnatury skanowania. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Porównywane sygnatury nie różnią się.</p> <p>x'01' - Porównywane sygnatury różnią się.</p>
QEZSBINARY	GRAPHIC (1)	<p>Określa, czy podczas ostatniego skanowania obiekt podlegał skanowaniu w trybie binarnym. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Obiekt nie podlegał skanowaniu w trybie binarnym.</p> <p>x'01' - Obiekt podlegał skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_FAILURE, skanowanie obiektu w trybie binarnym nie powiodło się.</p>
QEZSCCSID1	INTEGER	<p>Określa, czy obiekt podlegał skanowaniu z wymienionym identyfikatorem CCSID podczas poprzedniego skanowania. Jeśli status skanowania obiektu wskazuje wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu z identyfikatorem CCSID. Jeśli status skanowania obiektu wskazuje wartość SCAN_FAILURE, skanowanie obiektu z identyfikatorem CCSID nie powiodło się. Wartość 0 oznacza, że pole to nie obowiązuje.</p>
QEZSCCSID2	INTEGER	<p>Określa, czy obiekt podlegał skanowaniu z wymienionym identyfikatorem CCSID podczas poprzedniego skanowania. Jeśli status skanowania obiektu wskazuje wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu z identyfikatorem CCSID. Jeśli status skanowania obiektu wskazuje wartość SCAN_FAILURE, w polu tym będzie wartość 0. Wartość 0 oznacza, że pole nie obowiązuje.</p>
QEZUDATE	TIMESTAMP	<p>Liczba sekund, które upłynęły od epoki odpowiadającej dacie i godzinie ostatniego użycia obiektu. Gdy obiekt zostanie utworzony, pole to jest zerowe. Jeśli informacje o użyciu obiektu nie są obsługiwane dla typu systemu i5/OS lub systemu plików, do którego należy obiekt, pole to jest zerowe.</p>
QEZUDCOUNT	INTEGER	<p>Liczba dni, przez które obiekt był używany. Użycie obiektu ma różne znaczenie, w zależności od systemu plików i typów obiektów obsługiwanych w systemach plików. Użycie może oznaczać otwarcie lub zamknięcie zbioru lub dodanie dowiązań, zmianę nazwy, odtworzenie lub sprawdzenie obiektu. Licznik ten jest zwiększany każdego dnia, w którym obiekt jest używany i zerowany po wywołaniu funkcji API Qp0lSetAttr().</p>
QEZURESET	INTEGER	<p>Liczba sekund, które upłynęły od epoki odpowiadającej dacie ostatniego zerowania licznika dni używania obiektu. Data ta otrzymuje wartość bieżącej daty, gdy wywołana zostanie funkcja API Qp0lSetAttr() do wyzerowania licznika dni używania.</p>
QEZPRMLNK	SMALLINT	<p>Gdy obiekt ma kilka nazw, pole to otrzyma wartość tylko dla pierwszej znalezionej nazwy.</p>
QEZALWCKPW	SMALLINT	<p>Określa, czy plik strumieniowy (*STMF) może być współużytkowany przez programy czytające i piszące w trakcie przetwarzania punktu kontrolnego składowania podczas użycia. Poprawne wartości:</p> <p>0 - Obiekt może być współużytkowany tylko przez programy czytające.</p> <p>1 - Obiekt może być współużytkowany przez programy czytające i piszące.</p>

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZSIG <sup>1</sup>	SMALLINT	Określa, czy obiekt zawiera podpis cyfrowy systemu i5/OS. Poprawne wartości:  0 - obiekt nie zawiera podpisu cyfrowego systemu i5/OS.  1 - obiekt zawiera podpis cyfrowego systemu i5/OS.
QEZYSSIG	SMALLINT	Określa, czy obiekt został podpisany przez źródło zaufane w systemie. Poprawne wartości:  0 - Żaden z podpisów nie pochodzi ze źródła zaufanego w systemie.  1 - Obiekt został podpisany przez źródło zaufane w systemie. Jeśli obiekt zawiera kilka podpisów, przynajmniej jeden podpis pochodzi ze źródła zaufanego w systemie.
QEZMLTSIG	SMALLINT	Określa, czy obiekt zawiera więcej niż jeden podpis cyfrowy systemu i5/OS. Poprawne wartości:  0 - Obiekt zawiera tylko jeden podpis cyfrowy.  1 - Obiekt zawiera więcej niż jeden podpis cyfrowy. Jeśli w polu QEZYSSIG znajduje się wartość 1, przynajmniej jeden z podpisów pochodzi ze źródła zaufanego w systemie.
QEZDSTGOPT	SMALLINT	Opcja ta służy do określenia sposobu przydzielenia pamięci dyskowej w systemie dla wskazanego obiektu. Opcję tę można podać tylko dla plików strumieniowych w systemach plików "root" (/), QOpenSys i użytkownika (UDFS). opcja ta zostanie zignorowana dla bajtowych plików strumieniowych *TYPE1. Poprawne wartości:  0 - Pamięć dyskowa zostanie przydzielona normalnie. Oznacza to, że gdy wymagana będzie dodatkowa pamięć dyskowa, zostanie przydzielona w obszarach o logicznych wielkościach odpowiednich do pomieszczenia bieżących i przewidywanych wymagań pamięci z jednoczesnym minimalizowaniem liczby operacji dyskowych we/wy.  1 - Pamięć dyskowa będzie przydzielana tak, aby zminimalizować przestrzeń zajmowaną przez obiekt. Oznacza to, że w miarę potrzeb dodatkowa pamięć dyskowa będzie przydzielana w niewielkich ilościach, aby zaspokoić bieżące zapotrzebowanie. Dostęp do obiektu o wielu małych przydziałach może zwiększyć liczbę dyskowych operacji wejścia/wyjścia dla danego obiektu.  2 - System dynamicznie określi optymalną wielkość pamięci dyskowej dla obiektu, usiłując zrównoważyć ilość pamięci i dyskowych operacji we/wy. Na przykład, jeśli plik zajmuje dużo małych rozszerzeń pamięci dyskowej, a jednocześnie często występują dla niego operacje odczytu i zapisu, przyszłe operacje przydziału pamięci dyskowej przypiszą większe przestrzenie pamięci w celu zminimalizowania liczby operacji we/wy. Jeśli plik jest często obcinany, przyszłe operacje przydziału pamięci dyskowej będą przypisywały małe jednostki pamięci, w celu zminimalizowania zajętej przestrzeni dyskowej. Dodatkowo, informacje dla tego systemu i jego działań będą obsługiwane według wielkości pliku strumieniowego. Informacje o wielkości tego pliku posłużą także do określenia optymalnej wielkości przydziału pamięci dyskowej dla obiektu względem wielkości innych obiektów.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZMSTGOPT	SMALLINT	<p>Opcja ta służy do określenia sposobu przydzielenia i użycia pamięci głównej w systemie dla wskazanego obiektu. Opcję tę można podać tylko dla plików strumieniowych w systemach plików "root" (/), QOpenSys i użytkownika (UDFS). Poprawne wartości:</p> <p>0 - Pamięć główna zostanie przydzielona normalnie. Przydzielone i użyte zostanie jak najwięcej pamięci głównej. Spowoduje to zminimalizowanie liczby operacji we/wy dla dysku, ponieważ informacje będą przechowywane w pamięci głównej.</p> <p>1 - Pamięć główna będzie przydzielana tak, aby zminimalizować przestrzeń zajmowaną przez obiekt. Przydzielone i użyte zostanie jak najmniej pamięci głównej. Spowoduje to zminimalizowanie zużycia pamięci głównej i zwiększenie liczby operacji we/wy dla dysku, ponieważ mniej informacji będzie przechowywane w pamięci głównej.</p> <p>2 - System dynamicznie określi optymalny przydział pamięci głównej dla obiektu w oparciu o pozostałe działania systemu i rywalizację o pamięć główną. Oznacza to, że gdy rywalizacja o pamięć główną jest niewielka, system przydzieli jak najwięcej pamięci w celu zminimalizowania liczby operacji dyskowych wejścia/wyjścia. Kiedy rywalizacja o pamięć główną jest znaczna, system przydzieli mniej pamięci głównej w celu zminimalizowania rywalizacji. Opcja ta ma znaczenie tylko wtedy, gdy opcja stronicowania puli pamięci ma wartość *CALC. Gdy opcja stronicowania puli pamięci ma wartość *FIXED, system działa tak samo, jak dla opcji STG_NORMAL. Gdy dostęp do obiektu uzyskiwany jest poprzez serwer plików, opcja ta nie ma znaczenia. Jej działanie jest takie samo, jak dla opcji STG_NORMAL.</p>
QEZDIRTYP2	SMALLINT	<p>Format podanego obiektu katalogu. Poprawne wartości:</p> <p>0 - Format katalogu to *TYPE1.</p> <p>1 - Format katalogu to *TYPE2.</p>
QEZFILTYP2 <sup>1</sup>	SMALLINT	<p>Format pliku strumieniowego (*STMF). Poprawne wartości:</p> <p>0 - Plik strumieniowy ma format *TYPE1.</p> <p>1 - Plik strumieniowy ma format *TYPE2.</p>
QEZUDFTYP2	SMALLINT	<p>Domyślny format pliku dla plików strumieniowych (*STMF) utworzonych w systemie plików użytkownika. Poprawne wartości:</p> <p>0 - Plik strumieniowy ma format *TYPE1.</p> <p>1 - Plik strumieniowy ma format *TYPE2.</p>
QEZNONSAV	SMALLINT	<p>Określa, czy obiekt może być zeskładowany, czy nie. Poprawne wartości:</p> <p>0 - Obiekt zostanie zeskładowany.</p> <p>1 - Obiekt nie zostanie zeskładowany. Dodatkowo, jeśli obiekt jest katalogiem, żaden z obiektów w poddrzewie katalogu nie zostanie zeskładowany, jeśli nie zostanie osobno określony jako obiekt do składowania. Poddrzewo obejmuje wszystkie podkatalogi i obiekty w tych podkatalogach.</p>
QEZCLSTRSP	SMALLINT	<p>Obiekt należy do pamięci dyskowej przydzielonej dla zintegrowanych serwerów xSeries jako wirtualne napędy dysków serwerów xSeries. Z perspektywy serwera iSeries wirtualne dyski widziane są jako pliki strumieniowe w zintegrowanym systemie plików.</p> <p>0 - Obiekt nie jest pamięcią dysku wirtualnego.</p> <p>1 - Obiekt jest pamięcią dysku wirtualnego.</p>

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZCASE	SMALLINT	Wskazuje rozróżnianie wielkości znaków systemu plików, w którym znajduje się obiekt.  0 - System plików nie rozróżnia wielkości znaków.  1 - System plików rozróżnia wielkość znaków.
QEZOFLOW	SMALLINT	Wskazuje, czy obiekt spowodował przepełnienie puli pamięci, w której się znajduje. Poprawne wartości:  0 - Pula pamięci dyskowej nie jest przepełniona.  0 - Pula pamięci dyskowej jest przepełniona.
QEZPCREAD	SMALLINT	Określa, czy obiekt może być zapisany lub usunięty, jeśli jego atrybuty rozszerzone zostały zapisane lub usunięte albo zmieniono jego wielkość. Poprawne wartości:  0 - Obiekt można zmienić.  1 - Obiektu nie można zmienić.
QEZPCHID <sup>1</sup>	SMALLINT	Określa, czy obiekt można wyświetlić przy użyciu zwykłego listingu katalogów.  0 - Obiekt nie jest ukryty.  1 - Obiekt jest ukryty.
QEZPCSYS	SMALLINT	Określa, czy obiekt jest plikiem systemowym i jest wyłączony z normalnego wyszukiwania katalogów.  0 - Obiekt nie jest plikiem systemowym.  1 - Obiekt jest plikiem systemowym.
QEZPCARC	SMALLINT	Określa, czy obiekt zmienił się od ostatniego sprawdzenia obiektu.  0 - Obiekt nie zmienił się.  1 - Obiekt zmienił się.
QEZSYSARC	SMALLINT	Określa, czy obiekt zmienił się i powinien być składowany. Gdy czas zmiany obiektu jest aktualizowany, atrybut ten jest włączany, a po składowaniu obiektu atrybut jest wyłączany.  0 - Obiekt nie zmienił się i nie musi być składowany.  1 - Obiekt zmienił się i musi być składowany.
QEZJRCVNAM	GRAPHIC(10)	Najstarszy dziennik wymagany do wprowadzenia zmian w dzienniku. Gdy pole informacji o wprowadzeniu zmian zawiera wartość PARTIAL_TRANSACTION, dziennik zawiera dane początku transakcji częściowej. W przeciwnym przypadku dziennik zawiera rozpoczęcie operacji składowania. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZJRCVLIB	GRAPHIC(10)	Nazwa biblioteki, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZJRCVASP	GRAPHIC(10)	Nazwa puli ASP, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. Poprawne wartości:  *SYSBAS - Dziennik znajduje się w puli ASP systemowej lub użytkownika.  Urządzenie ASP - Nazwa urządzenia ASP, które zawiera dziennik.

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZJTRNI	GRAPHIC(1)	<p>Pole to zawiera informacje o aktualnym stanie obiektu i związane jest z granicami kontroli transakcji. Poprawne wartości:</p> <p>x'00' (NONE) - Brak transakcji częściowych.</p> <p>x'01' (PARTIAL_TRANSACTION) - Obiekt został odtworzony z transakcjami częściowymi. Nie można użyć obiektu do czasu, gdy użyta zostanie komenda Zastosowanie kronikowanych zmian (Apply Journalled Changes - APYJRNCHG) lub Usuwanie kronikowanych zmian (Remove Journalled Changes - RMVJRNCHG) w celu zastosowania lub wycofania zmian w transakcji częściowej.</p> <p>x'02' (ROLLBACK_ENDED) - Operacja wycofania dla obiektu została zakończona przy użyciu opcji "Zakończenie wycofania (End Rollback)" na ekranie Praca z definicjami transakcji (Work with Commitment Definition - WRKCMDFN). Zalecane jest odzyskanie obiektu, ponieważ nie można go użyć. Ostatecznie, można użyć komendy Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ) w celu udostępnienia obiektu do użycia. Wykonanie tej czynności może spowodować brak spójności obiektu.</p>
<b>Uwagi:</b>		
<ol style="list-style-type: none"> <li>1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRTDIRINF.</li> <li>2. W polu tym przechowywana jest tylko nazwa obiektu. Pozostałą część nazwy ścieżki zawiera pole QEZDIRNAM1, jeśli długość nazwy katalogu jest mniejsza niż 1 KB, lub pole QEZDIRNAM2, jeśli długości nazw przekraczają 1 KB.</li> </ol>		

Poniżej zamieszczona jest przykładowa tabela zawierająca katalogi przetworzone przez komendę RTVDIRINF.

Tabela 9. QAEZDxxxD (składowanie atrybutów katalogów)

Nazwa pola	Typ pola	Opis pola
QEZDIRIDX	INTEGER	Identyfikator relacyjny używany między tabelami katalogów i obiektów. Może on służyć do łączenia tabel w celu uzyskania danych pełnej nazwy ścieżki. Wartość w polu QEZDIRIDX tabeli obiektów odpowiada wartości QEZDIRIDX w tabeli katalogów dotyczącej katalogu macierzystego obiektu.
QEZDIRNAM1 <sup>1</sup>	VARGRAPHIC (1024)	Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki nie przekracza 1 KB (1024 bajtów).
QEZDIRNAM2 <sup>1</sup>	DBCLOB (16M)	Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki przekracza 1 KB (1024 bajty). Może pomieścić ścieżki do 16 MB długości.
QEZDRCCSID	INTEGER	Identyfikator CCSID katalogu.
QEZDREGION	GRAPHIC (2)	Identyfikator regionu ścieżki katalogu.
QEZLANGID	GRAPHIC (3)	Identyfikator języka ścieżki katalogu.
QEZDIRLEN <sup>1</sup>	INTEGER	Długość nazwy ścieżki katalogu.
QEZDIRFID	GRAPHIC (16)	Identyfikator zbioru katalogu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu.
QEZDFID	INTEGER	Identyfikator zbioru katalogu.
QEZDIRFSID	BIGINT	Systemowy identyfikator zbioru katalogu.
QEZDIRGID	BIGINT	Identyfikator pokolenia.
QEZPARDIR	INTEGER	Indeks katalogu nadrzędnego.
<b>Uwaga:</b>		
<ol style="list-style-type: none"> <li>1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRTDIRINF.</li> </ol>		

Poniższa tabela zawiera informacje przechowywane przez komendę RTVDIRINF, dotyczące plików utworzonych w wyniku jej działania. Jeśli plik zawierający te informacje nie istnieje, komenda RTVDIRINF tworzy go podczas kolejnych wywołań komendy dane są dopisywane do istniejącego pliku. Informacje te służą komendzie PRDIRINF do rozróżnienia, które zbiory baz danych posłużyły do zapisania informacji pobranych przez różne instancje komendy RTVDIRINF.

Tabela 10. QUSRSYS/QAEZDBFILE (składowanie utworzonych zbiorów)

Nazwa pola	Typ pola	Opis pola
QEZDIRSRC	VARGRAPHIC (5000)	Ścieżka podana w parametrze DIR (RTVDIRINF).
QEZPRCCSID	INTEGER	CCSID ścieżki.
QEZPREGION	GRAPHIC (2)	ID regionu ścieżki.
QEZPLANGID	GRAPHIC (3)	ID języka ścieżki.
QEZOBJFILE <sup>1</sup>	VARGRAPHIC (20)	Nazwa zbioru wygenerowanego do zapisania atrybutów obiektu.
QEZDIRFILE <sup>1</sup>	VARGRAPHIC (20)	Nazwa zbioru wygenerowanego do zapisania indeksów katalogu.
QEZLIB <sup>1</sup>	VARGRAPHIC (20)	Biblioteka, w której znajdują się oba wygenerowane zbiory.
QEZSTRTIME	TIMESTAMP	Data/godzina wywołania komendy RTVDIRINF.
QEZENDTIME	TIMESTAMP	Data/godzina zakończenia komendy RTVDIRINF.
<b>Uwaga:</b>		
1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRDIRINF.		

### Zadania pokrewne

“Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator” na stronie 80  
 Za pomocą programu System i Navigator można zbierać i analizować atrybuty obiektów w zintegrowanym systemie plików. Ten łatwy w użyciu interfejs graficzny oferuje te same funkcje, co komenda Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF). Można za jego pomocą badać dane zebrane przez interfejs oraz zadawać do nich zapytania. Dotyczy to także danych zebranych za pomocą komendy RTVDIRINF.

### Informacje pokrewne

Komenda Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF)

Qp0lGetPathFromFileID()--funkcja API Get Path Name of Object from Its File ID

Qp0lSetAttr()--funkcja API Set Attributes

Komenda Zastosowanie kronikowanych zmian (Apply Journaled Changed - APYJRNCHG)

Komenda Usuwanie kronikowanych zmian (Remove Journaled Changes - RMVJRNCHG)

Komenda Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ)

Komenda Drukuj informacje katalogowe (Print Directory Information - PRDIRINF)

### Dostęp do danych RTVDIRINF:

Dostęp do danych tabeli można uzyskać na kilka sposobów.

Poniżej znajdują się sposoby dostępu do danych utworzonych przez komendę RTVDIRINF:

- Korzystanie z komendy Drukuj informacje katalogowe (Print Directory Information - PRDIRINF)  
 Komenda ta służy do drukowania informacji katalogowych o obiektach i w zintegrowanym systemie plików. Drukowane informacje przechowywane są w zbiorze bazy danych wskazanym przez użytkownika w komendzie RTVDIRINF.
- Użycie programu lub komendy IBM, która może uruchamiać zapytania do tabeli bazy danych DB2 w systemie operacyjnym i5/OS.  
 Niektóre bardziej popularne narzędzia to komenda Uruch. interaktywnej sesji SQL (Start SQL Interactive Session - STRSQL) i narzędzie System i Navigator.

Na przykład, aby wybrać obiekty we wskazanej ścieżce (uprzednio wybranej przez komendę RTVDIRINF), która ma przydzieloną pamięć większą niż 10 kB, można uruchomić następujące zapytanie:

```
SELECT QEZOBJNAM, QEZALCSIZE
FROM nazwa_biblioteki/QAEZDxxxx0 WHERE
QEZALCSIZE > 10240
```

- Można także utworzyć własne programy i uzyskiwać dostęp do tabel bazy danych używając dowolnych poprawnych metod DB.
- Zamiast uruchamiać zapytania wydając komendy, można użyć programu System i Navigator do wczytywania, wyświetlania i analizowania danych informacyjnych katalogu (znanych jako dane atrybutów folderu w programie System i Navigator). Więcej informacji na ten temat zawiera sekcja poświęcona zbieraniu i analizowaniu atrybutów folderów w programie System i Navigator.

#### **Zadania pokrewne**

“Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator”

Za pomocą programu System i Navigator można zbierać i analizować atrybuty obiektów w zintegrowanym systemie plików. Ten łatwy w użyciu interfejs graficzny oferuje te same funkcje, co komenda Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF). Można za jego pomocą badać dane zebrane przez interfejs oraz zadawać do nich zapytania. Dotyczy to także danych zebranych za pomocą komendy RTVDIRINF.

#### **Informacje pokrewne**

Komenda Drukuj informacje katalogowe (Print Directory Information - PRDIRINF)

Komenda Uruch. interaktywnej sesji SQL (Start SQL Interactive Session - STRSQL)

Programowanie we wbudowanym języku SQL

Interfejs poziomu wywołania SQL

#### **Używanie danych zwracanych przez komendę RTVDIRINF:**

Poniżej znajdują się przykłady ilustrujące, dlaczego dane te są istotne i w jaki sposób można wykorzystać dane generowane na podstawie każdej z trzech tabel.

- Tabela 8 na stronie 68 dopuszcza tworzenie zapytań w celu utworzenia raportów statystyk w oparciu o dowolne pozycje tej tabeli. Komenda PRDIRINF nie obejmuje raportów opartych na wszystkich polach tabeli. Użyty zostanie podzestaw pól.
- Tabela 9 na stronie 78 zawiera dane, które dotyczą wszystkich katalogów w ścieżce podanej w parametrze DIR komendy RTVDIRINF. Dane te przydają się do poznania konkretnych atrybutów nazwy ścieżki, jak CCSID, ID języka lub długość. Każdy katalog przechowywany w tej tabeli zawiera unikalną wartość lub indeks, który go identyfikuje. Tabela 8 na stronie 68 umożliwia wyszukanie tego samego pola, QEZDIRIDX, zawierającego informacje, które obiekty należą do którego katalogu. Aby sprawdzić, który obiekt należy do którego katalogu, można użyć zapytań z połączeniami. Na przykład, następująca instrukcja wybiera nazwy wszystkich obiektów istniejących w katalogu "/MOJKAT":

```
SELECT QAEZDxxxx0.QEZOBJNAM FROM nazwa_biblioteki/QAEZDxxxxD,
nazwa_biblioteki/QAEZDxxxx0 WHERE QAEZDxxxxD.QEZDIRNAM1 = '/MOJKAT' AND
QAEZDxxxxD.QEZDIRIDX = QAEZDxxxx0.QEZDIRIDX
```

- Tabela 10 na stronie 79 używana jest głównie przez komendę PRDIRINF do pobrania konkretnych danych o uruchomieniach komendy RTVDIRINF. Przykładami są: nazwy utworzonych tabel, biblioteka, w której znajduje się tabela, i czas rozpoczęcia oraz zakończenia przetwarzania. Tabeli tej można użyć do sprawdzenia, kiedy uruchomiono komendę RTVDIRINF lub które tabele należy wyszukać, aby wydać zapytanie.

#### **Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator:**

- Za pomocą programu System i Navigator można zbierać i analizować atrybuty obiektów w zintegrowanym systemie plików. Ten łatwy w użyciu interfejs graficzny oferuje te same funkcje, co komenda Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF). Można za jego pomocą badać dane zebrane przez interfejs oraz zadawać do nich zapytania. Dotyczy to także danych zebranych za pomocą komendy RTVDIRINF.



| **Uwaga:** Ta metoda jest dostępna tylko w wersji V6R1 lub nowszej programu System i Navigator.

| Aby wygenerować raport atrybutów niektórych obiektów w zintegrowanym systemie plików, wykonaj następujące czynności:

| 1. W programie System i Navigator rozwiń gałąź **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).

| 2. Kliknij prawym przyciskiem myszy folder zawierający wybrane obiekty, a następnie wybierz kolejno opcje **Informacje o atrybutach folderów**(Folder Attribute Information) **Zbierz atrybuty** (Collect Attributes) .

| 3. W oknie Zbieranie atrybutów (Collect Attributes) określ preferencje. Wybierz opcję **Uwzględnij zawartość podfolderów zawartych w tym folderze** (Include the contents of subfolders contained in this folder), jeśli mają być zebrane również atrybuty podfolderów. Opcjonalnie można określić **Przedrostek pliku** (File Prefix) i **Bibliotekę** (Library). Kliknij przycisk **OK**, aby zacząć zbieranie atrybutów folderów.

| Proces zbierania danych może trochę potrwać. Należy poczekać kilka sekund, zanim wyświetlone zostanie okno Wyświetlenie zebranych atrybutów (Display Collected Attributes). Kliknij przycisk **Odśwież** (Refresh), aby wyświetlić najnowszy status procesu zbierania danych. Statusem tym może być: **Zakończony** (Completed), **Nieudany** (Failed) lub **Trwa** (In Progress). Przed przejściem do następnego etapu zaczekaj, aż żadnym ze statusów nie będzie **Trwa** (In Progress).

| 4. W oknie Wyświetlanie zebranych atrybutów (Display Collected Attributes) kliknij prawym przyciskiem myszy wpis, który ma zostać przeanalizowany, a następnie wybierz opcję **Analizuj informacje** (Analyze Information).

| **Uwaga:** Jeśli operacja zbierania atrybutów (Collect Attributes) została wykonana z poziomu programu System i Navigator, lub przed nią została wykonana komenda Odtwarzanie informacji katalogowych (Retrieve Directory Information - RTVDIRINF), można przejść bezpośrednio do okna Wyświetlenie zebranych atrybutów (Display Collected Attributes) klikając folder prawym przyciskiem myszy i wybierając kolejno opcje **Informacje o atrybutach folderu**(Folder Attribute Information) **Wyświetl zebrane atrybuty** (Display Collect Attributes).

| 5. W oknie Analiza informacji o folderze (Analyze Folder Information) dostosuj atrybuty do wyświetlenia na kartach **Kolumny** (Columns), **Filtr** (Filter) i **Kolejność** (Order). Następnie kliknij przycisk **OK**, aby wygenerować raport Informacje o atrybutach folderu (Folder Attribute Information).

| Poniżej przedstawiono przykład. Załóżmy, że chcemy wyświetlić pliki które mają rozmiar większy niż 10 MB, wraz z informacjami o ich właścicielach, zaś wyniki powinny być posortowane w pierwszej kolejności wg rozmiaru, a w drugiej wg właściciela.

| • Na karcie **Kolumny** (Columns) wybierz opcję *Właściciel* (Owner) i kliknij przycisk **Dodaj** (Add). Wybierz opcję *Ścieżka do folderu nadrzędnego* (Parent folder path) i kliknij przycisk **Dodaj** (Add). Wybierz opcję *Nazwa obiektu* (Object name) i kliknij przycisk **Dodaj** (Add). Wybierz opcję *Przydzielony rozmiar* (Allocated size) i kliknij przycisk **Dodaj** (Add).

| • Na karcie **Filtr** (Filter) wybierz opcję *Przydzielony rozmiar* (Allocated size) w polu **Pole** (Field) i wybierz opcję *Rozmiary większe niż* (Sizes greater than) w polu **Warunek** (Condition). Następnie wpisz wartość 10 w polu **Rozmiar** (Size) i wybierz opcję *Megabajty* (Megabytes) w polu **Jednostka** (Unit). Kliknij przycisk **Dodaj** (Add), aby utworzyć filtr.

| • Na karcie **Kolejność** (Order) wybierz opcję *Przydzielony rozmiar* (Allocated size) i *Malejąco* (Descending) w polu **Pierwszy warunek sortowania** (First sort). Następnie wybierz opcje *Właściciel* (Owner) i *Malejąco* (Descending) w polu **Drugi warunek sortowania** (Second sort).

| • Kliknij przycisk **OK**. Wyświetlony zostanie raport Informacje o atrybutach folderu (Folder Attribute Information) z dostosowanymi informacjami.

#### | **Odsyłacze pokrewne**

| “Praca z danymi wyjściowymi komend RTVDIRINF i PRDIRINF” na stronie 68

| Komenda RTVDIRINF służy do zbierania atrybutów dla obiektów, które zawiera. Zebrane informacje przechowywane są w zbiorach baz danych (tabelach), których nazwy zawierają prefiks pliku informacji określony przez parametr INFFILEPFX. Tabele tworzone są w bibliotece określonej w parametrze INFLIB.

| “Dostęp do danych RTVDIRINF” na stronie 79

| Dostęp do danych tabeli można uzyskać na kilka sposobów.

## Dostęp przy użyciu funkcji API

Funkcji API można użyć, aby uzyskać dostęp do zintegrowanego systemu plików.

### Odsyłacze pokrewne

“Wykonywanie operacji za pomocą funkcji API” na stronie 106

Wiele interfejsów API, które wykonują operacje na obiektach zintegrowanego systemu plików, ma postać funkcji napisanych w języku C.

## Dostęp przy użyciu komputera PC

Jeśli używany komputer PC jest podłączony do systemu System i, może on pracować z katalogami i obiektami zintegrowanego systemu plików w taki sposób, jakby były one składowane na tym komputerze PC.

Obiekty można kopiować między katalogami używając funkcji przeciągnij i upuść narzędzia Windows Explorer. Można nawet skopiować obiekt z systemu na komputer PC poprzez zaznaczenie obiektu na dysku systemu i przeciągnięcie go na dysk komputera PC.

Wszystkie obiekty kopiowane między produktem System i a komputerami PC za pośrednictwem interfejsu Windows mogą być poddawane automatycznej konwersji między kodowaniem EBCDIC a ASCII. Program rodzina produktów System i Access może zostać tak skonfigurowany, aby wykonywał konwersję automatycznie, lub też aby wykonywał konwersję na plikach o określonych rozszerzeniach.

W zależności od typu obiektu, użytkownik może używać do pracy z nim interfejsów i aplikacji komputera PC. Na przykład, plik strumieniowy zawierający tekst można poddać edycji z użyciem edytora komputera PC.

Po podłączeniu klienta PC do produktu System i, zintegrowany system plików udostępnia katalogi i obiekty systemu na kliencie PC. Komputery PC mogą pracować z plikami, w zintegrowanym systemie plików z użyciem klientów współużytkowania zasobów plikowych wbudowanych w systemy operacyjne Windows, klientów FTP lub System i Navigator (część rodzina produktów System i Access). Komputer PC korzysta z klientów udostępniania plików systemu Windows w celu uzyskania dostępu do uruchomionego w systemie serwera i5/OS NetServer.

### Pojęcia pokrewne

“Uzyskiwanie dostępu za pomocą produktu System i Navigator”

System i Navigator oferuje interfejs graficzny do zarządzania i administrowania systemami z komputera z systemem Windows. System i Navigator ułatwia eksploatację systemu i zarządzanie nim oraz przyczynia się do zwiększenia produktywności.

### Zadania pokrewne

“Uzyskiwanie dostępu za pomocą produktu i5/OS NetServer” na stronie 83

Funkcja obsługi Otoczenia sieciowego systemów Windows w systemie i5/OS (i5/OS NetServer) pozwala klientom z systemami Windows korzystać ze współużytkowanych katalogów i kolejek wyjściowych systemu i5/OS. i5/OS NetServer umożliwia komputerom PC z oprogramowaniem Windows bezproblemowy dostęp do danych i drukarek zarządzanych przez platformę System i.

### Odsyłacze pokrewne

“Dostęp przy użyciu protokołu FTP” na stronie 84

Klient protokołu FTP umożliwia przesyłanie plików znajdujących się na platformie System i.

## Uzyskiwanie dostępu za pomocą produktu System i Navigator

System i Navigator oferuje interfejs graficzny do zarządzania i administrowania systemami z komputera z systemem Windows. System i Navigator ułatwia eksploatację systemu i zarządzanie nim oraz przyczynia się do zwiększenia produktywności.

Można na przykład skopiować profil użytkownika z jednego systemu do innego po prostu przeciągając do myszy. Konfigurowanie ochrony i usług TCP/IP w aplikacjach odbywa się pod nadzorem kreatorów.

- | Wiele z tych zadań można wykonać za pomocą programu System i Navigator. Dla ułatwienia poniżej wymieniono
- | typowe zadania związane ze zintegrowanym systemem plików:

### **Praca z plikami i folderami**

- “Tworzenie folderu” na stronie 123
- “Usuwanie pliku lub folderu” na stronie 123
- “Ustawianie uprawnień” na stronie 125
- “Konfigurowanie konwersji plików tekstowych” na stronie 125
- “Wysyłanie pliku lub folderu do innego systemu” na stronie 126
- “Zmiana opcji dla wysyłania pliku lub folderu” na stronie 126
- “Określanie, czy obiekty mają być skanowane” na stronie 129
- | • “Zwracanie obiektów” na stronie 130
- | • “Pobieranie obiektów” na stronie 130
- | • “Zbieranie i analizowanie atrybutów folderów za pomocą programu System i Navigator” na stronie 80

### **Praca ze współużytkowanymi zasobami plikowymi**

- “Tworzenie współużytkowanego zasobu plikowego” na stronie 126
- “Zmiana współużytkowanego zasobu plikowego” na stronie 127

### **Praca z systemami plików użytkownika**

- “Tworzenie nowego systemu plików użytkownika” na stronie 127
- “Podłączanie systemu plików utworzonego przez użytkownika” na stronie 127
- “Odłączanie systemu plików użytkownika” na stronie 128
- | • “Praca z dynamicznie podłączanymi systemami plików” na stronie 128

### **Kronikowanie obiektów**

- “Uruchamianie kronikowania” na stronie 97
- “Przerywanie kronikowania” na stronie 98

## **Uzyskiwanie dostępu za pomocą produktu i5/OS NetServer**

Funkcja obsługi Otoczenia sieciowego systemów Windows w systemie i5/OS (i5/OS NetServer) pozwala klientom z systemami Windows korzystać ze współużytkowanych katalogów i kolejek wyjściowych systemu i5/OS. i5/OS NetServer umożliwia komputerom PC z oprogramowaniem Windows bezproblemowy dostęp do danych i drukarek zarządzanych przez platformę System i.

Klienci PC w sieci używają funkcji współużytkowania plików i drukarek wbudowanych we własne systemy operacyjne. Oznacza to, że korzystanie z funkcji i5/OS NetServer nie wymaga instalowania na komputerze PC żadnego dodatkowego oprogramowania.

Również klienci z systemem Linux i zainstalowanym oprogramowaniem klienckim Samba mogą bezproblemowo korzystać z danych i drukarek udostępnianych poprzez i5/OS NetServer. Współużytkowane katalogi serwera i5/OS NetServer mogą być podłączane przez klienty z systemem Linux jako systemy plików Samba. Podłączanie odbywa się w podobny sposób, jak w przypadku systemów plików NFS eksportowanych z systemu i5/OS.

Współużytkowany zasób plikowy i5/OS NetServer jest ścieżką do katalogu, który i5/OS NetServer udostępnia klientom PC w sieci System i. Zasób taki może wskazywać dowolny katalog zintegrowanego systemu plików w systemie. Zanim będzie możliwe korzystanie z funkcji współużytkowania plików z użyciem serwera i5/OS NetServer, należy utworzyć współużytkowany zasób plikowy serwera i5/OS NetServer i w razie potrzeby zmodyfikować współużytkowany zasób plikowy serwera i5/OS NetServer za pomocą programu System i Navigator.

Aby uzyskać dostęp do współużytkowanych zasobów plikowych zintegrowanego systemu plików za pomocą produktu i5/OS NetServer:

1. Kliknij prawym przyciskiem myszy pozycję **Start**, i wybierz opcję **Eksploruj**, aby otworzyć okno programu Windows Explorer w komputerze PC z systemem Windows.
2. Otwórz menu Narzędzia i wybierz **Mapuj dysk sieciowy**.
3. Wybierz dla współużytkowanego zasobu literę oznaczającą wolny napęd (na przykład napęd I:\).
4. Podaj nazwę współużytkowanego zasobu plikowego i5/OS NetServer. Można na przykład użyć następującej składni: `\\QSYSTEM1\NazwaZasobu`

**Uwaga:** QSYSTEM1 jest nazwą systemu serwera i5/OS NetServer, a *NazwaZasobu* jest nazwą współużytkowanego zasobu plikowego, który ma być używany.

5. Kliknij przycisk **OK**.

**Uwaga:** Nazwa systemu używana podczas nawiązywania połączenia za pomocą serwera i5/OS NetServer może być inna od nazwy używanej przez oprogramowanie rodzina produktów System i Access. Na przykład nazwą serwera i5/OS NetServer może być QAS400X, a ścieżka dostępu do plików może mieć postać `\\QAS400X\QDLS\MOJFOLDER.FLR\MOJPLIK.DOC`. Jednak nazwą rodzina produktów System i Access może być AS400X, a ścieżka dostępu do plików może mieć postać `\\AS400X\QDLS\MOJFOLDER.FLR\MOJPLIK.DOC`.

Serwer i5/OS NetServer umożliwia określenie, które katalogi mają być udostępniane w sieci. Katalogi te są wyświetlane jako pierwszy poziom pod nazwą systemu. Jeśli na przykład katalog `/home/franek` zostanie udostępniony pod nazwą `katalogfranka`, użytkownicy będą mogli z komputera PC uzyskać dostęp do tego katalogu poprzez nazwę `\\QAS400X\KATALOGFRANKA`, a w przypadku klientów z systemem Linux - poprzez nazwę `//qas400x/katalogfranka`.

Znacznie większą wydajność udostępniania plików dla klienta PC w porównaniu z innymi systemami plików systemu i5/OS zapewnia bazowy system plików "root" (/). W określonych sytuacjach może być pożądanym przeniesienie plików do systemu plików "root" (/). Więcej informacji znajduje się w temacie "Przenoszenie plików lub folderów do innego systemu plików" na stronie 124.

#### **Informacje pokrewne**

i5/OS NetServer

Współużytkowane zasoby plikowe systemu i5/OS

## **Dostęp przy użyciu protokołu FTP**

Klient protokołu FTP umożliwia przesyłanie plików znajdujących się na platformie System i.

Pozwala także na przesyłanie folderów i dokumentów w systemie plików usługi biblioteki dokumentów (QDLS). Klienta FTP można uruchomić interaktywnie w nienadzorowanym trybie wsadowym, w którym komendy są czytane z pliku, a odpowiedzi na nie zapisywane do pliku. Zapewnia on także inne funkcje umożliwiające przetwarzanie plików w systemie, na którym działa.

Obsługę protokołu FTP można wykorzystywać przy przesyłaniu plików do i z jednego z poniższych systemów plików:

- system plików "root" (/),
- system plików systemów otwartych (QOpenSys),
- system plików bibliotek (QSYS.LIB),
- system plików QSYS.LIB niezależnej puli ASP,
- system plików usług biblioteki dokumentów (QDLS),
- system plików nośników optycznych (QOPT),
- sieciowy system plików (Network File System - NFS),
- system plików i5/OS NetClient (QNTC),

- system plików QFileSvr.400.

Należy jednak pamiętać o następujących ograniczeniach:

- Zintegrowany system plików zawęży użycie FTP do przesyłania danych z plików. Nie można użyć FTP do przesyłania atrybutów.
- System plików QSYS.LIB i system plików QSYS.LIB niezależnej puli ASP zawęży użycie FTP do podzbiorów zbiorów fizycznych, podzbiorów źródłowych zbiorów fizycznych i zbiorów składowania. Nie można używać FTP do przesyłania obiektów innych typów, takich jak programy (\*PGM). Można jednak zeszkładować obiekty innych typów do zbioru składowania, przesłać zbiór składowania, a następnie odtworzyć obiekty.

#### **Informacje pokrewne**

protokół FTP

Przesyłanie plików za pomocą protokołu FTP

---

## **Konwersja zintegrowanego systemu plików**

- System operacyjny i5/OS wykonuje automatyczną konwersję systemów plików w zintegrowanym systemie plików w celu obsługi nowego formatu katalogów lub standardu Unicode.

### **Przekształcanie katalogów z formatu \*TYPE1 na \*TYPE2**

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2.

Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1. Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1.

Bezpośrednio po zainstalowaniu systemu w wersji i5/OS V5R3 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Nie powinna ona poważnie wpłynąć na działanie systemu.

### **Przegląd konwersji formatu \*TYPE1 do \*TYPE2**

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2.

Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1. Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1. Oprócz lepszej wydajności i niezawodności, niektóre nowe funkcje, takie jak obsługa skanowania zintegrowanego systemu plików, dostępne są tylko dla obiektów w katalogach \*TYPE2. Więcej informacji znajduje się w temacie "Obsługa skanowania" na stronie 18.

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Konwersja ta nie powinna znacząco wpłynąć na działanie systemu, ponieważ będzie działać w zadaniu w tle o niskim priorytecie.

Jeśli konwersja nie została jeszcze zakończona, a system wykona normalny lub nieprawidłowy IPL, funkcja konwersji podejmie działanie po zakończeniu IPL. Konwersja będzie ponownie uruchamiana po każdym IPL do czasu, gdy wszystkie odpowiednie systemy plików zostaną w pełni przekształcone.

Systemy plików, które można poddać automatycznej konwersji, to "root" (/), QOpenSys i system plików użytkownika dla pul ASP od 1 do 32.

**Uwaga:** Można uniknąć automatycznej konwersji do formatu katalogów \*TYPE2, jeśli konwersja ta zostanie przeprowadzona ręcznie przed zainstalowaniem systemu operacyjnego w wersji V5R3 lub nowszej.

#### **Pojęcia pokrewne**

“Katalogi \*TYPE2” na stronie 10

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w obsługują format katalogów \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1.

### **Odsyłacze pokrewne**

“Określenie statusu konwersji”

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. To przetwarzanie konwersji nastąpi w dodatkowym wątku zadania systemowego QFILESYS1.

“Wskazówki: niezależne ASP” na stronie 89

Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2 lub nowszej.

## **Zagadnienia związane z konwersją katalogów**

Poniżej przedstawiono kilka zagadnień, które należy wziąć pod uwagę podczas przeprowadzania konwersji katalogów.

### **Określenie statusu konwersji:**

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. To przetwarzanie konwersji nastąpi w dodatkowym wątku zadania systemowego QFILESYS1.

Aby określić status konwersji, można użyć komendy Konwersja katalogu (Convert Directory - CVTDIR) w następujący sposób:

```
CVTDIR OPTION(*CHECK)
```

Ta składnia komendy CVTDIR powoduje wyświetlenie listy aktualnego formatu katalogów systemów plików "root" (/), QOpenSys i UDFS oraz informacje o wykonywanej konwersji systemu plików. Dodatkowo, zawiera bieżący priorytet funkcji konwersji, aktualnie konwertowany system plików w systemie, liczbę dowiązań dotychczas przetworzonych dla tego systemu plików i procent katalogów, które zostały już przetworzone dla tego systemu plików. System uruchamia funkcję konwersji z bardzo niskim priorytetem (99), aby funkcja nie wpływała znacząco na działanie systemu. Można jednak zmienić ten priorytet używając wartości \*CHGPTY parametru OPTION w komendzie CVTDIR. Dodatkowe informacje o specyfikacji tego parametru zawiera temat CVTDIR.

Ponieważ zadanie QFILESYS1 przetwarza konwersję, można wyświetlić protokół zadania QFILESYS1 i przejrzeć ewentualne komunikaty o błędach w procesie konwersji. Dodatkowo, do powyższego protokołu wysyłane będą różne komunikaty informujące o postępie operacji konwersji. Komunikaty te zawierają następujące informacje: przetwarzany system plików, liczbę dowiązań dotychczas przetworzonych dla tego systemu plików, procent katalogów, które zostały już przetworzone dla tego systemu plików itp. Wszystkie komunikaty o błędach i wiele komunikatów o postępie operacji wysyłane jest do kolejki komunikatów QSYSOPR. Dlatego, na przyszłość warto sprawdzić, czy protokoły QHST lub protokoły zadania QFILESYS1 są zapisywane, ponieważ zawierają one te komunikaty. Gdy systemy plików zostaną w pełni przekształcone, a zintegrowany system plików działa poprawnie, można usunąć te informacje.

### **Informacje pokrewne**

Komenda Konwersja katalogu (Convert Directory - CVTDIR)

### **Tworzenie profili użytkowników:**

Funkcja konwersji tworzy profil użytkownika, który używany jest w trakcie działania funkcji. Ten profil ma nazwę QP0FCWA. Jest wykorzystywany przez funkcję konwersji jako właściciel przetwarzanych katalogów w systemie plików, jeśli nie można im przypisać oryginalnego właściciela.

Profil ten jest usuwany po zakończeniu konwersji, o ile jest to możliwe. Komunikat CPIA08B jest wysyłany do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR, jeśli temu użytkownikowi zostaną przydzielone prawa własności katalogu.

## Odsyłacze pokrewne


“Zmiana właściciela katalogu” na stronie 88

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika.

## Zmiana nazw obiektów:

Katalogi \*TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16.

Reguła nazewnictwa dla katalogów \*TYPE2 różni się od reguły dla katalogów \*TYPE1, które mają nazwy UCS2 Poziom 1. Dlatego podczas konwersji katalogów mogą wystąpić nieprawidłowe lub zduplikowane nazwy. Gdy znaleziona zostanie niepoprawna lub powielona nazwa, zmieniana jest ona na unikalną, poprawną nazwę UTF-16, a do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08A podający oryginalną i nową nazwę. Znaki złożone lub nieprawidłowe pary znaków odpowiedników znajdujące się w nazwie mogą spowodować zmianę nazwy obiektu.

Więcej informacji o kodzie UTF-16 można znaleźć na stronie głównej Unicode (<http://www.unicode.org> ).

### *Znaki złożone:*

Niektóre znaki mogą składać się z więcej niż jednego znaku Unicode.

Znaki posiadające akcent (na przykład é lub à) lub umlaut (na przykład ä lub ö) muszą być zmienione lub *znormalizowane* do wspólnego formatu zanim zostaną umieszczone w katalogu, tak aby wszystkie obiekty miały unikalne nazwy. Normalizacja znaków złożonych jest procesem, w którym znak jest przekształcany do nowego i przewidywalnego formatu. Format wybrany dla katalogu \*TYPE2 to *złożona postać kanoniczna*. Jeśli w katalogu \*TYPE1 znajdują się dwa obiekty zawierające te same znaki złożone, są one normalizowane do tej samej nazwy. Powoduje to kolizję, nawet jeśli jeden obiekt zawiera połączone złożone znaki, a drugi zawiera rozłączone złożone znaki. Dlatego jeden z nich musi mieć zmienioną nazwę zanim zostanie dowiązany w katalogu \*TYPE2.

### *Znaki odpowiedników:*

Niektóre znaki nie mają prawidłowych odpowiedników w systemie Unicode.

Znaki te mają specjalne wartości, które składają się z dwóch znaków Unicode, w dwóch określonych przedziałach, z których pierwszy znak znajduje się w pierwszym zakresie (na przykład 0xD800-0xD8FF), a drugi w drugim zakresie (na przykład 0xDC00-0xDCFF). Nazywa się to parą odpowiedników.

Jeśli jeden ze znaków Unicode zostanie usunięty lub są one podane w złej kolejności (tylko jeden znak), nazwa jest nieprawidłowa. Nazwy tego typu są dozwolone w katalogach \*TYPE1, ale nie są dozwolone w katalogach \*TYPE2. Aby konwersja mogła być kontynuowana, jeśli nazwa zawiera jeden z nieprawidłowych znaków, jest zmieniana zanim obiekt zostanie dowiązany w katalogu \*TYPE2.

## Uwagi na temat profili użytkowników:

Podczas konwersji system próbuje zapewnić, aby ten sam profil użytkownika, który jest właścicielem dowolnego katalogu \*TYPE1 był również właścicielem odpowiedniego katalogu \*TYPE2.

Ponieważ katalogi \*TYPE1 i \*TYPE2 przez pewien czas istnieją jednocześnie, ma to wpływ na wielkość pamięci należącej do profilu użytkownika i liczbę pozycji w profilu użytkownika.

### *Zmiana maksymalnej ilości pamięci dla profilu użytkownika:*

Podczas konwersji katalogu właścicielem pewnej liczby katalogów istniejących w obu formatach jednocześnie jest ten sam profil użytkownika.

Jeśli w trakcie konwersji zostanie przekroczony maksymalny limit pamięci dla profilu użytkownika, limit ten jest zwiększany. Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08C.

#### *Zmiana właściciela katalogu:*

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika.

Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08B i konwersja jest kontynuowana.

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika. Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08B i konwersja jest kontynuowana.

#### **Odsyłacze pokrewne**

“Tworzenie profili użytkowników” na stronie 86

Funkcja konwersji tworzy profil użytkownika, który używany jest w trakcie działania funkcji. Ten profil ma nazwę QP0FCWA. Jest wykorzystywany przez funkcję konwersji jako właściciel przetwarzanych katalogów w systemie plików, jeśli nie można im przypisać oryginalnego właściciela.

#### **Wymagania pamięci dyskowej:**

Wymagania pamięci dyskowej należy rozważyć, gdy w systemie rozpocznie się konwersja katalogów w systemie plików do formatu \*TYPE2.

Poniżej znajduje się kilka uwag odnoszących się do wymaganej przestrzeni dyskowej:

- końcowa wielkość katalogów po konwersji do formatu \*TYPE2,
- dodatkowa przestrzeń dyskowa wymagana, gdy funkcja konwersji jest uruchomiona.

W wielu przypadkach, końcowa wielkość katalogu \*TYPE2 jest mniejsza niż katalogu \*TYPE1. Zazwyczaj katalogi \*TYPE2 zawierające mniej niż 350 obiektów wymagają mniej pamięci dyskowej niż katalogi \*TYPE1 zawierające tę samą liczbę obiektów. Katalogi \*TYPE2 zawierające więcej niż 350 obiektów są o 10 procent większe (średnio) niż katalogi \*TYPE1.

Gdy funkcja konwersji jest uruchomiona, wymagana jest dodatkowa przestrzeń dyskowa. Funkcja konwersji wymaga, aby obie wersje katalogów, w formacie \*TYPE1 i \*TYPE2 istniały jednocześnie.

**Uwaga:** Zanim rozpocznie się instalacja wersji i5/OS V5R3MO lub nowszej wersji systemu operacyjnego, należy rozważyć uruchomienie komendy systemu OS/400 V5R2 (CVTDIR) z opcją \*ESTIMATE, która dostarczy dokładnych danych szacunkowych ilości wolnej pamięci dyskowej wymaganej do przeprowadzenia operacji konwersji.

#### **Informacje pokrewne**

Komenda Konwersja katalogu (Convert Directory - CVTDIR)

#### **Wskazówki: dowiązanie symboliczne:**

Dowiązania symboliczne są obiektami w zintegrowanym systemie plików zawierającymi ścieżkę do innego obiektu.

W trakcie konwersji istnieje kilka sytuacji, w których nazwa obiektu może zostać zmieniona. Jeśli podczas konwersji zostanie zmieniona nazwa jednego z elementów ścieżki w dowiązaniu symbolicznym, to zawartość dowiązania symbolicznego nie będzie już wskazywać na obiekt.

#### **Pojęcia pokrewne**



“Dowiązanie” na stronie 11

*Dowiązanie (link)* jest nazwą połączenia między katalogiem a obiektem. Podając nazwę dowiązania do tego obiektu, użytkownik lub program może poinformować system, gdzie znajduje się obiekt. Dowiązania można użyć jako ścieżki lub jako części ścieżki.

#### **Odsyłacze pokrewne**

“Zmiana nazw obiektów” na stronie 87

Katalogi \*TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16.

#### **Informacje pokrewne**

symlink()--Tworzenie dowiązania symbolicznego

### **Wskazówki: niezależne ASP:**

Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2 lub nowszej.

### **Wskazówki: składowanie i odtwarzanie:**

Katalogi \*TYPE1 mogą być zapisane i odtworzone w systemie plików, który został przekonwertowany do \*TYPE2.

Podobnie, katalogi \*TYPE2 mogą być zapisane i odtworzone w systemie plików o formacie \*TYPE1, pod warunkiem, że nie zostało przekroczone żadne ograniczenie formatu \*TYPE1, gdy katalog istniał jako katalog \*TYPE2.

### **Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików:**

Gdy system przetwarza systemy plików "root" (/), QOpenSys i UDFS w puli ASP użytkownika do obsługi formatu katalogów \*TYPE2, nie można uruchomić komend Odzyskiwanie pamięci (RCLSTG) oraz Odzyskanie dowiązań obiektu (RCLLNK) w żadnym katalogu zintegrowanego systemu plików, również w katalogach niezależnej ASP.

Można jednak użyć parametru OMIT(\*DIR) w komendzie RCLSTG, aby pominąć katalogi zintegrowanego systemu plików i umożliwić odzyskanie obiektów niezwiązanych ze zintegrowanym systemem plików.

#### **Pojęcia pokrewne**

“Operacja odzyskiwania systemu plików "root" (/), systemu plików QOpenSys i systemu plików UDFS” na stronie 98

Odzyskanie systemów plików "root" (/), QOpenSys i UDFS jest możliwe przy użyciu komend Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

#### **Informacje pokrewne**

Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

### **Skanowanie zintegrowanego systemu plików:**

Obiekty w systemie plików "root" (/), QOpenSys i UDFS ASP użytkownika nie będą skanowane z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików do czasu zakończenia konwersji systemów do formatu katalogów \*TYPE2.

Atrybuty związane ze skanowaniem można przypisać do obiektów w katalogach \*TYPE1 i \*TYPE2 w celu określenia, czy obiekty mają być skanowane, czy nie, nawet jeśli system plików nie został w pełni przetworzony.

Gdy system przetwarza obiekty z formatu \*TYPE1 do \*TYPE2, uwzględnia wartość systemową sterowania skanowaniem ('Skanowanie przy następnym dostępie po odtworzeniu obiektu') tak, jakby obiekty były odtwarzane. Na przykład, jeśli wartość Skanowanie przy następnym dostępie po odtworzeniu obiektu została podana w trakcie konwersji, wówczas obiekt z katalogu \*TYPE1 mający atrybut obiekt nie będzie skanowany zostanie poddany skanowaniu przynajmniej raz po zakończeniu pełnej konwersji.

## Pojęcia pokrewne

“Obsługa skanowania” na stronie 18

System operacyjny i5/OS umożliwia skanowanie obiektów zintegrowanego systemu plików.

“Pokrewne wartości systemowe” na stronie 19

Wartości systemowe QSCANFS i QSCANFCTL służą do zdefiniowania środowiska skanowania w systemie.

## Konwersja nazw w celu obsługi dodatkowych znaków

Systemy plików przechowują nazwy w standardzie Unicode. Na systemy plików, które nie rozróżniają wielkości znaków, mają wpływ zmiany w znakach i regułach wielkości znaków związane z konkretnym standardem Unicode. Systemy plików, które nie uwzględniają wielkości znaków, takie jak "root" (/) i systemy UDFS utworzone przy użyciu funkcji CASE(\*MONO) od wydania i5/OS V6R1 obsługują standard Unicode 4.0.

Bezpośrednio po zainstalowaniu systemu i5/OS system rozpocznie automatyczną konwersję katalogów do formatu Unicode 4.0 dla wszystkich systemów plików, które jeszcze nie zostały przekształcone na ten standard Unicode. Konwersja ta nie powinna znacząco wpłynąć na działanie systemu, ponieważ będzie działać w zadaniu w tle o niskim priorytecie.

### Informacje pokrewne



<http://www.unicode.org>

## Przegląd automatycznej konwersji nazw

Niektóre systemy plików, które nie rozróżniają wielkości liter, na przykład system "root" (/) i UDFS utworzone przy użyciu funkcji CASE(\*MONO), obsługują nazwy zapisane w standardzie Unicode 4.0. System wykonuje automatyczną konwersję nazw, aby obsłużyć dodatkowe znaki w tych nazwach.

W zintegrowanym systemie plików wyszukiwanie nazw jest realizowane, jeśli nazwa jest podana w komendzie CL lub funkcji API zintegrowanego systemu plików, w celu zidentyfikowania obiektu, na którym wykonywane są działania. Jeśli zdefiniowane są dodatkowe znaki lub reguły wielkości liter są aktualizowane w celu obsługi nowych znaków, to nazwy zawierające znaki, na które ma wpływ dokonana zmiana, mogą nie zostać znalezione.

Zintegrowany system plików przechowuje wszystkie nazwy w standardzie Unicode. W wydaniach systemu i5/OS wcześniejszych niż V6R1 zintegrowany system plików obsługuje standard Unicode 2.0. Od wydania V6R1 zintegrowany system plików obsługuje standard Unicode 4.0. Program narzędziowy do konwersji nazw jest uruchamiany automatycznie w celu aktualizacji katalogów w systemach plików, które nie rozróżniają wielkości liter, w celu obsługi standardu Unicode 4.0. Systemy plików, które nie uwzględniają wielkości nazw i które obejmuje ta konwersja to: "root" (/) oraz systemy UDFS (w puli pamięci dyskowej) utworzone za pomocą funkcji CASE(\*MONO).

Konwersja nazw w systemie plików "root" (/) i w systemach UDFS w podstawowych pulach pamięci dyskowej użytkownika (1-32) jest rozpoczynana automatycznie, niedługo po zainstalowaniu systemu operacyjnego i5/OS. Działa ona w wątku o niskim priorytecie w zadaniu systemowym QFILESYS1. Jeśli funkcja konwersji nie została zakończona, a system wykonuje program IPL, funkcja konwersji jest wznawiana po zakończeniu IPL. Konwersja jest wznawiana po każdym IPL, do czasu gdy wszystkie odpowiednie systemy plików zostaną w pełni przekształcone.

Podczas operacji udostępniania niezależnej ASP, zadanie QFSYS##### dla grupy niezależnych pul dyskowych ##### jest uruchamiane automatycznie, a konwersja nazw w systemie UDFS w niezależnej ASP ##### działa w wątku o niskim priorytecie w tym zadaniu systemowym. Konwersja nazw dla systemów UDFS w niezależnych ASP nie rozpocznie przekształcania katalogów do czasu, aż nie zostanie zakończona konwersja nazw dla systemów "root" (/) i systemów UDFS w podstawowej ASP użytkownika. Jeśli funkcja konwersji nie została zakończona, a niezależna ASP jest odłączona lub system wykonuje program IPL, funkcja konwersji jest wznawiana po zakończeniu IPL i udostępnieniu niezależnej ASP.

**Uwaga:** Jeśli odłączona jest jednocześnie więcej niż jedna niezależna ASP, to w danym czasie może być aktywna tylko jedna procedura konwersji nazw niezależnej ASP. Kiedy jedna konwersja zostanie zakończona, następna może stać się aktywna.

Aby zaplanować tę konwersję nazw, można użyć komendy Analiza konwersji obiektów (Analyse Object Conversion - ANZOBJCVN) w poprzednim wydaniu. Komenda ANZOBJCVN zbiera lub raportuje informacje dotyczące konwersji obiektów w systemie. Obejmuje to informacje, jakie zintegrowane systemy plików zawierają nazwy, w których występuje co najmniej jeden znak, na który mogą mieć wpływ nowe reguły znaków Unicode i wielkości nazw w nowej wersji. Nazwy obiektów, na które ma to wpływ, mogą być potencjalnie zmienione z uwzględnieniem nowych znaków Unicode i reguł wielkości znaków przy wykonywaniu automatycznej konwersji po aktualizacji do nowego wydania. Ponieważ zmienione obiekty mogą wpływać na aplikacje, warto zmienić nazwy wszelkich obiektów, na które może mieć wpływ ta sytuacja, przed aktualizacją do nowego wydania.

#### **Pojęcia pokrewne**

“Niezmienność nazw” na stronie 16

Użytkownicy systemu plików “root” (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmiennosc znaków w nazwach obiektów.

#### **Informacje pokrewne**

 <http://www.unicode.org>

Sugerowane: Instalowanie poprawek PTF i analiza systemu pod kątem konwersji obiektów

Planowanie ochrony zintegrowanego systemu plików

### **Zagadnienia związane z konwersją nazw**

Poniżej przedstawiono kilka zagadnień, które należy wziąć pod uwagę podczas przeprowadzania konwersji nazw.

#### **Określenie statusu konwersji:**

Ogólne komunikaty informujące o tym, które systemy plików zostały przekształcone, oraz różne komunikaty o postępie prac są przesyłane do protokołu zadania, które wykonuje konwersję.

W przypadku systemów plików “root” (/) i UDFS w pulach ASP 1-32 komunikaty są wysyłane do protokołu zadania QFILESYS1. W przypadku systemów UDFS w niezależnych pulach ASP komunikaty są wysyłane do protokołu zadania QFSYS##### dla grupy niezależnych pul dysków #####. Komunikaty te obejmują takie informacje, jak dane o konwertowanych systemach plików oraz liczba dowiązań, które zostały przetworzone w systemie plików. Większość ogólnych komunikatów jest też wysyłana do kolejki komunikatów QSYSOPR. Komunikaty o błędach są wysyłane do protokołu historii QHST.

Dlatego też warto zachowywać protokoły QHST, które zawierają te komunikaty. Gdy systemy plików zostaną w pełni przekształcone, a zintegrowany system plików działa poprawnie, można usunąć te informacje.

#### **Zmiana nazw obiektów:**

Jeśli używany jest system plików nierozróżniający wielkości liter, podczas wyszukiwania nazw w tym systemie nie są rozróżniane małe i wielkie litery. Wszelka zmiana reguł rozróżniania wielkości liter wpłynie na przetwarzanie znaków, które przy braku rozróżnienia między małymi a wielkimi literami są traktowane jako identyczne.

Podczas automatycznej konwersji nazw może się zdarzyć, że nazwy, które były różne w kodowaniu Unicode Standard 2.0, zostaną w kodowaniu Unicode Standard 4.0 uznane za identyczne. W takiej sytuacji konieczna jest zmiana nazwy jednego z obiektów. Gdy dokonywana jest zmiana nazwy w celu zapewnienia jej unikalności, do kolejki QHST wysyłany jest komunikat CPDA0BC, zawierający pierwotną i nową nazwę.

Więcej informacji na temat różnych wersji standardu Unicode można znaleźć na stronie głównej standardu ([www.unicode.org](http://www.unicode.org)).

#### **Uwagi na temat profili użytkowników:**

Podczas działania procesu konwersji system wymaga dodatkowej pamięci, co ma wpływ na ilość pamięci będącej własnością profilu użytkownika.

| Jeśli podczas procesu konwersji nazw osiągnięty zostanie maksymalny limit pamięci dla profilu użytkownika, system zwiększa ten limit dla profilu użytkownika i wysyła komunikat CPIA08C do kolejki komunikatów QSYSOPR.

| **Wskazówki: dowiązanie symboliczne:**

| Dowiązania symboliczne są obiektami w zintegrowanym systemie plików, które zawierają nazwy ścieżek wskazujące na inne obiekty. Jeśli nazwa jednego z elementów nazwy ścieżki w dowiązaniu symbolicznym zostanie zmieniona podczas konwersji, zawartość dowiązania symbolicznego nie będzie już wskazywać na inny obiekt.

| **Wskazówki: niezależne ASP:**

| Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do standardu Unicode dla bieżącego wydania, zostanie uruchomione zadanie mające na celu wykonanie tej konwersji po udostępnieniu niezależnej ASP.

| System uruchamia zadanie QFSYS##### dla grupy niezależnych pul dyskowych #####. Automatyczna konwersja nazw działa w wątku o niskim priorytecie należącym do tego zadania, a dla każdej niezależnej puli ASP w grupie uruchamiany jest jeden wątek. Jeśli automatyczna konwersja nazw nie została zakończona, system rozpoczyna to zadanie za każdym razem, gdy grupa niezależnych pul dyskowych zostaje udostępniona.

| **Wskazówki: składowanie i odtwarzanie:**

| Katalogi wykorzystujące standard Unicode 2.0 mogą być składowane i odtwarzane do systemu plików, który wykorzystuje standard Unicode 4.0, i odwrotnie.

| Przy odtwarzaniu obiektów są one dowiązywane do katalogu. Standard Unicode katalogu, do którego obiekt jest dowiązywany, dotyczy nazwy odtwarzanego obiektu. Dlatego też obiekty nie mogą być odtwarzane, jeśli nazwa nie jest unikalna w standardzie Unicode wykorzystywanym przez katalog. W takich wypadkach system wysyła komunikat CPD37B9 z kodem powodu 1 do protokołu zadania.

| **Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików:**

| Jeśli podczas automatycznej konwersji nazw wystąpią jakiegokolwiek błędy, komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) i Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) pozwolą zakończyć konwersję katalogu i jego podkatalogów. Ma to miejsce tylko po zakończeniu przetwarzania automatycznej konwersji nazw w systemie plików.

| Dodatkowo, komenda RCLSTG konwertuje dowolne zagubione katalogi na standard Unicode 4.0.

| **Pojęcia pokrewne**

| “Operacja odzyskiwania systemu plików "root" (/), systemu plików QOpenSys i systemu plików UDFS” na stronie 98

| Odzyskanie systemów plików "root" (/), QOpenSys i UDFS jest możliwe przy użyciu komend Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

| **Informacje pokrewne**

| Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

| Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

---

## Kronikowanie obiektów

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Dodatkowo, główną funkcją kronikowania jest asysta w powielaniu zmian obiektów w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia.

W poniższym temacie przedstawiono krótki przegląd zarządzania kronikami, a także uwagi dotyczące kronikowania obiektów zintegrowanego systemu plików i opis obsługi kronikowania dla obiektów zintegrowanego systemu plików.

## Informacje pokrewne

Zarządzanie kronikami

## Przegląd kronikowania

Poniższe tematy opisują obsługę kronikowania obiektów zintegrowanego systemu plików.

### Informacje pokrewne

Zarządzanie kronikami

## Zarządzanie kronikami

Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu.

Zarządzanie kronikami może również służyć do:

- zapisów kontrolnych dla działań dotyczących obiektów w systemie;
- rejestrowania innych niż kronikowane działań dotyczących obiektów;
- szybszego odtwarzania z nośników, na których przeprowadza się składowanie podczas użycia;
- asysty w powielaniu zmian w obiektach w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia;
- asysty przy testowaniu programów użytkowych.

Za pomocą kroniki można zdefiniować, które obiekty zostaną zabezpieczone poprzez kronikowanie. W zintegrowanym systemie plików można kronikować pliki strumieniowe, katalogi i dowiązania symboliczne. Obsługiwane są tylko obiekty następujących systemów plików: bazowego (root, "/"), QOpenSys i UDFS.

### Pojęcia pokrewne

“Jakie obiekty należy uwzględnić w kronikowaniu”

Przed podjęciem decyzji o kronikowaniu obiektu zintegrowanego systemu plików należy rozważyć pewne kwestie.

## Jakie obiekty należy uwzględnić w kronikowaniu

Przed podjęciem decyzji o kronikowaniu obiektu zintegrowanego systemu plików należy rozważyć pewne kwestie.

Aby określić, które obiekty powinny być kronikowane, należy wziąć po uwagę następujące czynniki:

- Jak bardzo obiekt się zmienia? Dobrymi kandydatami do kronikowania są obiekty, które ulegają znacznym zmianom między kolejnymi składowaniami.
- Jak trudno jest odtworzyć zmiany wprowadzane w obiekcie? Jak wiele zmian obiektu nie ma formy pisemnej? Trudniej na przykład odtworzyć obiekt wykorzystywany do rejestrowania zamówień zgłaszanych przez telefon, niż obiekt używany do rejestrowania zamówień składanych na formularzach przesyłanych pocztą tradycyjną.
- Jakie znaczenie mają informacje zapisane w obiekcie? Gdyby zaszła konieczność przywrócenia obiektu do postaci zapisanej podczas ostatniej operacji składowania, jaki wpływ na działanie firmy miałyby opóźnienia spowodowane koniecznością odtworzenia zmian?
- W jaki sposób obiekt jest powiązany z innymi obiektami w systemie? Może się okazać, że choć dane określonego obiektu nie ulegają częstym zmianom, to jednak mają one kluczowe znaczenie do innych, bardziej dynamicznych obiektów w systemie. Przykładowo, wiele obiektów może być zależnych od głównego zbioru danych klientów. Jeśli dokonywana jest rekonstrukcja zamówień, główny zbiór danych klientów musi uwzględniać nowych klientów lub zmiany limitów kredytowych, jakie zostały wprowadzone od czasu ostatniej operacji składowania.

## Kronikowane obiekty zintegrowanego systemu plików

Obsługa kronikowania w systemie i5/OS umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików.

Obsługiwanymi typami obiektów są pliki strumieniowe, katalogi i dowiązania symboliczne. Jedynymi systemami plików obsługującymi kronikowanie obiektów tych typów są bazowy system plików (/), QOpenSys i UDFS. Obiekty zintegrowanego systemu plików mogą być kronikowane zarówno poprzez tradycyjny interfejs systemowy (komendy

CL lub funkcje API), jak i z wykorzystaniem programu System i Navigator. Operacje Uruchomienie kronikowania i Zakończenie kronikowania można wykonać w aplikacji System i Navigator, podobnie jak można wyświetlić informacje o statusie kronikowania.

**Uwaga:** Pliki strumieniowe mapowania pamięci, pliki woluminów wirtualnych i pliki strumieniowe używane jako przestrzeń pamięci programu Integrated xSeries Server (IXS) nie mogą być kronikowane. Katalogi, które mogą zawierać blokowe obiekty plików specjalnych (\*BLKSF), nie mogą być kronikowane. Przykładami są: /dev/QASP01, /dev/QASP22 i /dev/IASPNAM.

Poniższa lista zawiera podsumowanie obsługi kronikowania w zintegrowanym systemie plików:

- Do wykonywania operacji kronikowania na obsługiwanych typach obiektów można używać komend ogólnych i funkcji API. Interfejsy te akceptują identyfikację obiektów poprzez nazwę ścieżki, identyfikator zbioru lub obie metody jednocześnie.
- Niektóre komendy operacji kronikowania, takie jak Uruchomienie kronikowania (Start Journal), Zakończenie kronikowania (End Journal), Zmiana obiektu kronikowanego (Change Journaled Object) i Zastosowanie kronikowanych zmian (Apply Journaled Changes) można wykonać dla całych poddrzew obiektów zintegrowanego systemu plików. Opcjonalnie możliwe jest zastosowanie list włączających i wyłączających z wzorcami nazw obiektów wykorzystującymi znaki zastępcze. Można na przykład użyć komendy Uruchomienie kronikowania (Start Journal) dla wszystkich obiektów drzewa "/MojaFirma", które pasują do wzorca "\*.data", z wyłączeniem obiektów odpowiadających wzorcom "A\*.data" i "B\*.data".
- Obsługa kronikowania katalogów obejmuje takie działania na katalogach, jak dodawanie i usuwanie dowiązań oraz tworzenie, zmiana nazwy i przemieszczanie obiektów w obrębie katalogu.

Kronikowane katalogi mają atrybut, który można ustawić tak, aby nowe obiekty w poddrzewie dziedziczyły bieżące ustawienia kronikowania dla katalogu. Włączenie tego atrybutu dla kronikowanego katalogu spowoduje, że wszystkie pliki strumieniowe, katalogi i dowiązania symboliczne utworzone w tym katalogu lub dowiązane do niego (przez dodanie dowiązania stałego lub przez zmianę nazwy bądź przeniesienie obiektu) będą automatycznie kronikowane przez system.

**Uwaga:** Zagadnienia związane z atrybutem dziedziczenia kronikowania:

- Jeśli nazwa obiektu zostanie zmieniona w tym samym katalogu, w którym obecnie obiekt się znajduje, kronikowanie nie zostanie uruchomione dla obiektu, nawet jeśli dla katalogu włączono atrybut dziedziczenia stanu kronikowania.
- Gdy katalog zostanie przeniesiony do katalogu z włączonym atrybutem dziedziczenia kronikowania, kronikowanie zostanie uruchomione tylko dla przeniesionego katalogu, jeśli jest to możliwe. Obiekty w przeniesionym katalogu nie będą objęte kronikowaniem.
- Jeśli obiekt zostanie odtworzony w katalogu z włączonym atrybutem dziedziczenia kronikowania, kronikowanie nie zostanie uruchomione dla tego obiektu, jeśli obiekt był już kronikowany.
- Użycie komendy Zastosowanie kronikowanych zmian (Apply Journaled Changes - APYJRNCHG) powoduje, że bieżąca wartość atrybutu dziedziczenia kronikowania nie zostanie użyta dla żadnego katalogu. Zamiast niej, dla obiektów utworzonych jako część działania komendy kronikowanie zostanie uruchomione w zależności od przebiegu działania uruchomionej komendy.
- Nazwy obiektów i pełne nazwy ścieżek są zapisane w kilku pozycjach kronik obiektów zintegrowanego systemu plików. Dla nazwy obiektów i nazw ścieżek włączona jest obsługa języków narodowych (NLS).
- W przypadku nieprawidłowego zakończenia pracy systemu kronikowane obiekty zintegrowanego systemu plików objęte zostaną odtwarzaniem przy IPL.
- Maksymalna wielkość pozycji kroniki obsługiwana przez różne interfejsy zapisu zawiera się w przedziale 2 GB - 1; jeśli podano wartość RCVSIZOPT (\*MAXOPT2 lub \*MAXOPT3), wartość ta wynosi 4 000 000 000 bajtów. W przeciwnym wypadku, maksymalna wielkość pozycji kroniki wynosi 15 761 440 bajtów. Jeśli kronikowany jest plik strumieniowy, a programy piszące przekraczają wielkość 15 761 440 bajtów, należy użyć obsługi \*MAXOPT2 lub \*MAXOPT3, aby uniknąć wystąpienia błędów.

Szczegóły dotyczące zawartości i formatów danych właściwych dla pozycji kronik , a także informacje o różnych układach pozycji kronik zintegrowanego systemu plików zawiera plik w języku C o nazwie qp0ljrn1.h, dostarczany w podzbiorze QSYSINC/H (QP0LJRN1).

### **Pojęcia pokrewne**

“Plik strumieniowy” na stronie 15

*Plik strumieniowy* jest dostępną losowo sekwencją bajtów, pozbawioną struktury narzucanej przez system.

“Katalog” na stronie 4

*Katalog* jest specjalnym obiektem używanym do wyszukiwania obiektów według nazw podanych przez użytkownika. Każdy katalog zawiera listę dołączonych do niego obiektów. Lista ta może zawierać inne katalogi.

“Dowiązanie symboliczne” na stronie 13

*Dowiązanie symboliczne* (symbolic link), zwane także dowiązaniem miękkim (soft link), jest ścieżką zawartą w pliku.

### **Zadania pokrewne**

“Uruchamianie kronikowania” na stronie 97

Aby uruchomić kronikowanie, wykonaj w programie System i Navigator następujące operacje na obiekcie.

“Przerywanie kronikowania” na stronie 98

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

“Zmiana kronikowania” na stronie 97

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ).

### **Informacje pokrewne**

Zarządzanie kronikami

Narzędzie wyszukiwania informacji o pozycjach kroniki

## **Kronikowane operacje**

Wymienione niżej operacje są kronikowane tylko wtedy, kiedy dotyczą obiektów lub dowiązań obsługiwanych przy kronikowaniu.

- utworzenie obiektu,
- dodanie dowiązania do istniejącego obiektu,
- usunięcie dowiązania,
- zmiana nazwy dowiązania,
- zmiana identyfikatora pliku,
- przeniesienie dowiązania do katalogu lub z katalogu.

W przypadku plików strumieniowych kronikowane są następujące działania:

- zapis danych lub usunięcie zawartości,
- obcięcie/rozszerzenie pliku,
- wymuszenie pliku danych,
- składowanie z opróżnieniem pamięci.

Dla wszystkich kronikowanych obiektów kronikowane są następujące operacje:

- Zmiany atrybutów (w tym zmiany ochrony, np. zmiany uprawnień i praw własności)
- Otwarcie
- Zamknięcie
- Uruchomienie kronikowania
- Komenda Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ)
- Zakończenie kronikowania

- Komenda Rozpoczęcie zastosowania kronikowanych zmian (Start the Apply Journalized Changes - APYJRNCHG)
- Komenda Zakończenie zastosowania kronikowanych zmian (End the Apply Journalized Changes - APYJRNCHG)
- Składowanie
- Odtwarzanie

### **Informacje pokrewne**

Zarządzanie kronikami

Narzędzie wyszukiwania informacji o pozycjach kroniki

## **Uwagi specjalne dotyczące pozycji kronik**

Wiele kronikowanych operacji w zintegrowanym systemie plików korzysta z kontroli transakcji w celu uformowania pojedynczej transakcji z wielu funkcji wykonywanych podczas działania.

Tych kronikowanych operacji nie można uznać za kompletne, dopóki w cyklu kontroli transakcji nie pojawi się pozycja kroniki Zatwierdzenie (Commit) (Kod kroniki C, Typ CM). Operacje kronikowane zawierające pozycję kroniki Wycofanie (Rollback) (Kod kroniki C, Typ RB) w cyklu kontroli transakcji to operacje, które się nie powiodły, a pozycje kronik dla tych operacji nie powinny być powtarzane ani replikowane.

Kronikowane pozycje zintegrowanego systemu plików (Kod kroniki B) używające w ten sposób kontroli transakcji obejmują:

- AA - Zmianę wartości kontroli (Change Audit Value),
- B0 - Rozpoczęcie tworzenia (Begin Create),
- B1 - Utworzenie podsumowania (Create Summary),
- B2 - Dodanie dowiązania (Add Link),
- B3 — Zmianę nazwy/Przeniesienie (Rename/Move)
- B4 - Usunięcie dowiązania (katalog nadrzędny) (Unlink - Parent Directory),
- B5 Usunięcie dowiązania (dowiązanie) (Unlink - link),
- B7 — Informacje o uprawnieniach utworzonego obiektu (Created Object Authority Information),
- FA - Zmianę atrybutu (Attribute Change),
- JT - Uruchomienie kronikowania (Start Journal) (tylko gdy kronikowanie się rozpoczyna, ponieważ operacja została wykonana w katalogu z włączonym atrybutem kronikowania),
- OA - Zmianę uprawnień (Authority Change),
- OG - Zmianę grupy głównej obiektu (Object Primary Group Change),
- OO - Zmianę właściciela obiektu (Object Owner Change).

Kilka pozycji kronik dla zintegrowanego systemu plików ma specyficzne pola wskazujące, czy dana pozycja jest pozycją podsumowania. Operacje wysyłające pozycje kronik zawierające podsumowania, przesyłają dwie pozycje do kroniki. Pierwsza z nich zawiera podzbiór danych właściwych dla danej pozycji. Druga pozycja zawiera pełny zbiór danych właściwych dla pozycji i jest oznaczona jako pozycja podsumowania. Programy, które powielają obiekt lub ponownie wykonują operację, będą głównie zainteresowane pozycjami podsumowań.

Dla operacji tworzenia w katalogu kronikowanym pozycją zawierającą podsumowanie jest pozycja kroniki B1 (Utworzenie podsumowania - Create Summary).

Niektóre operacje kronikowane muszą wysyłać pozycje kronik, które są przeciwne do tych operacji. Na przykład cykl kontroli transakcji zawierający pozycje kroniki B4 (Usunięcie dowiązania - Unlink) może również zawierać pozycję B2 (Dodanie dowiązania - Add Link). Jednak taki przypadek wystąpi tylko podczas operacji, w wyniku których pojawi się pozycja kroniki Wycofanie (Rollback) (C - RB).

Może się tak zdarzyć z dwóch powodów:

1. Wystąpiły problemy podczas operacji i pozycja ta była potrzebna wewnętrznie w procedurze czyszczącej błąd ścieżki.



2. Operacja została przerwana przez wyłączenie systemu i podczas procedury IPL wykonano odtwarzanie konieczne do wysłania pozycji w celu wycofania przerwanej operacji.

#### **Informacje pokrewne**

Narzędzie wyszukiwania informacji o pozycjach kroniki

## **Zagadnienia dotyczące używania wielu napędów dysków i kronikowania**

Jeśli istnieje kilka dowiązań stałych do kronikowanego obiektu zintegrowanego systemu plików, należy wszystkie te dowiązania zeszkładować i odtworzyć razem, aby zachowana została definicja dowiązania razem z odpowiednimi informacjami kroniki.

Podanie - w niektórych komendach związanych z kroniką - nazw, które są wieloma dowiązaniem stałymi spowoduje, że obiekt będzie przetwarzany jednorazowo dla wszystkich nazw. Pozostałe dowiązania stałe zostaną praktycznie zignorowane.

Ponieważ wiele dowiązań stałych wskazuje ten sam obiekt, a pozycja kroniki zawiera tylko identyfikator pliku (File ID), który jest ten sam dla obiektu, wszystkie interfejsy kroniki wyświetlą nazwę ścieżki. Na przykład komenda Wyświetlenie kroniki (Display Journal - DSPJRN) spowoduje wyświetlenie nazwy tylko jednego dowiązania dla obiektu. Nie powinno to jednak powodować błędów, ponieważ obiekt można obsługiwać z użyciem dowolnej nazwy i uzyskać ten sam rezultat.

#### **Pojęcia pokrewne**

“Dowiązanie stałe” na stronie 12

*Dowiązanie stałe*, zwane po prostu dowiązaniem, nie może istnieć bez przyłączenia do rzeczywistego obiektu.

## **Uruchamianie kronikowania**

Aby uruchomić kronikowanie, wykonaj w programie System i Navigator następujące operacje na obiekcie.

1. W oknie **System i Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Kliknij prawym przyciskiem myszy obiekt, który ma być kronikowany, i wybierz opcję **Kronikowanie** (Journaling).
4. Po wybraniu odpowiednich opcji kronikowania, kliknij przycisk **Uruchom**.

Aby uruchomić kronikowanie obiektu przy użyciu interfejsu znakowego, można użyć komendy Uruchomienie kroniki (Start Journal - STRJRN) lub funkcji API QjoStartJournal.

#### **Informacje pokrewne**

Komenda Uruchomienie kroniki (Start Journal - STRJRN)

Funkcja API Uruchomienie kroniki (Start Journal - QjoStartJournal)

Zarządzanie kronikami

## **Zmiana kronikowania**

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ).

#### **Zadania pokrewne**

“Uruchamianie kronikowania”

Aby uruchomić kronikowanie, wykonaj w programie System i Navigator następujące operacje na obiekcie.

“Przerywanie kronikowania” na stronie 98

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

#### **Informacje pokrewne**

Komenda Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ)

## Przerywanie kronikowania

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

Aby zatrzymać kronikowanie obiektu przy użyciu programu System i Navigator, wykonaj następujące czynności:

1. W oknie **System i Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Kliknij prawym przyciskiem myszy obiekt, którego kronikowanie ma być zakończone i wybierz opcję **Kronikowanie** (Journalling).
4. Kliknij **Zakończ**.

Aby zatrzymać kronikowanie obiektu przy użyciu interfejsu znakowego, można użyć komendy Zatrzymanie kroniki (End Journal - ENDJRN) lub funkcji API QjoEndJournal.

### Zadania pokrewne

“Uruchamianie kronikowania” na stronie 97

Aby uruchomić kronikowanie, wykonaj w programie System i Navigator następujące operacje na obiekcie.

### Informacje pokrewne

Komenda Zakończenie kronikowania (End Journal - ENDJRN)

Funkcja API Zakończenie kronikowania (End Journal - QjoEndJournal)

Zarządzanie kronikami

---

## Operacja odzyskiwania systemu plików "root" (/), systemu plików QOpenSys i systemu plików UDFS

Odzyskanie systemów plików "root" (/), QOpenSys i UDFS jest możliwe przy użyciu komend Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

Komendy RCLLNK i RCLSTG umożliwiają wykonanie następujących zadań:

- Usunięcie problemów z profilami użytkownika
- Usunięcie problemów z systemami plików użytkownika
- Usunięcie problemów z obiektami wewnętrznymi
- Usunięcie niepoprawnych dowiązań obiektu
- Obsługę uszkodzonych obiektów
- Utworzenie brakujących obiektów systemowych
- Usunięcie wewnętrznych problemów systemów plików (tylko RCLSTG)
- Wyszukiwanie utraconych obiektów (tylko RCLSTG)
- Zakończenie konwersji katalogów na standard Unicode 4.0, jeśli podczas automatycznej konwersji nazw wystąpiły błędy.

## Porównanie komend Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komendy Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) umożliwiają rozwiązanie problemów z systemami plików "root" (/), QOpenSys i UDFS.

Komenda RCLLNK identyfikuje i, jeśli jest to możliwe, usuwa problemy z używanymi podłączonymi systemami plików. Komenda RCLSTG nie ma takiej funkcjonalności. Komenda RCLSTG może jednak usunąć problemy, których komenda RCLLNK nie jest w stanie zidentyfikować i usunąć. Poniższa tabela zawiera bardziej szczegółowe porównanie obu komend.

Tabela 11. Porównanie komend RCLLNK i RCLSTG

	RCLLNK OBJ(' /MyDir/MyObj ')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
<b>Czy system musi być w stanie zastrzeżonym?</b>	Nie	Tak	Nie
<b>Czy podczas odzyskiwania możliwe jest użycie wszystkich systemów plików?</b>	Tak	Nie	Użycie systemów w niezależnej puli ASP jest niemożliwe.
<b>Obiekty z których puli ASP można odzyskać?</b>	Odzyskuje obiekty w systemowych i niezależnych pulach ASP oraz w pulach ASP użytkownika.	Odzyskuje obiekty w systemowych pulach ASP oraz w pulach ASP użytkownika.	Odzyskuje obiekty w niezależnych ASP.
<b>Jak odzyskiwane są obiekty?</b>	Obiekty są odzyskiwane pojedynczo lub poddrzewami, zgodnie z parametrami komendy.	Odzyskiwane są obiekty w całym systemie.	Odzyskiwane są obiekty w niezależnych pulach ASP.
<b>Które ze znanych problemów systemu plików są identyfikowane i usuwane, jeśli możliwe?</b>	Większość (więcej informacji na ten temat znajduje się w sekcji "Operacja odzyskiwania systemu plików "root" (/), systemu plików QOpenSys i systemu plików UDFS" na stronie 98).	Wszystkie	Wszystkie
<b>Czy utracone obiekty są odnajdywane?</b>	Nie	Tak	Tak
<b>Czy odzyskiwane są obiekty w odłączonych systemach plików?</b>	Nie	Tak	Tak
<b>Czy komenda jest wątkowo bezpieczna?</b>	Tak	Nie	Nie
<b>Ile instancji komendy może zostać wywołanych jednocześnie?</b>	Wiele	Jedna	Jedna
<b>Które z obiektów zintegrowanego systemu plików są ponownie tworzone w razie potrzeby?</b>	Wszystkie	Większość (więcej informacji na ten temat znajduje się w sekcji "Ponowne tworzenie obiektów zintegrowanego systemu plików" na stronie 100).	Brak
<b>Czy możliwe jest zidentyfikowanie uszkodzonych obiektów bez ich odtwarzania?</b>	Tak	Nie	Nie

### Pojęcia pokrewne

"Przykłady: komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)" na stronie 101  
 Poniższe przykłady opisują sytuacje, w których komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) może zostać użyta do odzyskania obiektów w systemach plików "root" (/), QOpenSys i podłączonych systemach plików użytkownika.

### Odsyłacze pokrewne

"Ponowne tworzenie obiektów zintegrowanego systemu plików" na stronie 100  
 Poniższa tabela zawiera obiekty zintegrowanego systemu plików, które są ponownie tworzone przez komendę Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK), jeśli nie istnieją. Są one zazwyczaj tworzone

podczas ładowania programu początkowego (IPL). Istnieje również możliwość ponownego utworzenia w razie potrzeby niektórych z tych obiektów przy użyciu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

### Informacje pokrewne

Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

## Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) umożliwia zidentyfikowanie i naprawienie uszkodzonych obiektów w systemach plików "root" (/), QOpenSts i i podłączonych systemach plików użytkownika; system nie musi znajdować się w stanie zastrzeżonym. Umożliwia to usunięcie problemów z systemami plików bez obniżania efektywności.

Komenda RCLLNK może być w wielu sytuacjach używana jako alternatywa do komendy Odzyskiwanie pamięci (RCLSTG). Użycie komendy RCLLNK jest na przykład dobrym sposobem identyfikowania i usuwania problemów w następujących sytuacjach:

- Problemy są związane z jednym obiektem.
- Problemy są związane z grupą obiektów.
- Zachodzi potrzeba zidentyfikowania lub usunięcia uszkodzonych obiektów.
- System nie może znajdować się w stanie zastrzeżonym podczas operacji odzyskiwania.
- Niezależne pule ASP muszą być dostępne podczas operacji odzyskiwania.

## Ponowne tworzenie obiektów zintegrowanego systemu plików

Poniższa tabela zawiera obiekty zintegrowanego systemu plików, które są ponownie tworzone przez komendę Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK), jeśli nie istnieją. Są one zazwyczaj tworzone podczas ładowania programu początkowego (IPL). Istnieje również możliwość ponownego utworzenia w razie potrzeby niektórych z tych obiektów przy użyciu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

Tabela 12. Obiekty zintegrowanego systemu plików ponownie tworzone przez komendy RCLLNK i RCLSTG

Nazwa ścieżki	Typ	Ponownie tworzony przez komendę RCLLNK	Ponownie tworzony przez komendę RCLSTG ASPDEV(*SYSBASE)
/dev/zero	*CHRFS	Tak	Tak
/dev/null	*CHRFS	Tak	Tak
/dev/xti/tcp	*CHRFS	Tak	Nie
/dev/xti/udp	*CHRFS	Tak	Nie
/etc/vfs	*STMF	Tak	Nie

Aby możliwe było ponowne utworzenie obiektów zintegrowanego systemu plików przez komendę RCLLNK, należy ją wywołać, podając katalog nadrzędny, przy parametrze SUBTREE mającym wartość \*DIR lub \*ALL. Możliwe musi być pomyślne odzyskanie przez komendę katalogu nadrzędnego obiektu systemowego. Na przykład komenda RCLLNK OBJ('/dev') SUBTREE(\*DIR)

ponownie tworzy obiekty /dev/zero i /dev/null \*CHRFS, jeśli nie istnieją.

Aby możliwe było ponowne utworzenie obiektów zintegrowanego systemu plików przez komendę RCLSTG, należy ją wywołać przy parametrze ASPDEV mającym wartość \*SYSBASE; nie można pominąć fragmentu odzyskiwania odpowiedzialnego za odzyskiwanie katalogów.

### Pojęcia pokrewne

“Dostarczane katalogi” na stronie 7

Po restarcie systemu zintegrowany system plików tworzy niżej wymienione katalogi, jeśli jeszcze nie istnieją. Po utworzeniu katalogów przez system nie należy ich przenosić ani zmieniać ich nazw.

### Informacje pokrewne

Komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

## Przykłady: komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK)

Poniższe przykłady opisują sytuacje, w których komenda Odzyskanie dowiązań obiektu (Reclaim Object Links - RCLLNK) może zostać użyta do odzyskania obiektów w systemach plików "root" (/), QOpenSys i podłączonych systemach plików użytkownika.

### Przykład: usuwanie problemów z obiektem

W tej sytuacji problemy są znane i dotyczą jednego obiektu. Obiekt jest uszkodzony, użycie go jest niemożliwe oraz niemożliwe jest odtworzenie wersji zapasowej z nośnika. Zachodzi potrzeba szybkiego usunięcia problemu bez zakłócania normalnych działań w systemie plików.

Aby odtworzyć obiekt, należy wywołać następującą komendę:

```
RCLLNK OBJ('/Kat/UszkObiekt') SUBTREE(*NONE)
```

, gdzie wartość /Kat/UszkObiekt odpowiada nazwie uszkodzonego obiektu.

### Przykład: usuwanie problemów występujących w strukturze katalogów

W tej sytuacji problemy są znane i dotyczą grupy obiektów w poddrzewie katalogów. Z ich powodu występuje błąd aplikacji. Zachodzi potrzeba szybkiego usunięcia problemu bez zakłócania normalnych działań w systemie plików.

Aby odzyskać obiekty w poddrzewie katalogów, należy wywołać następującą komendę:

```
RCLLNK OBJ('/KatInstalacyjnyAplikacji') SUBTREE(*ALL)
```

, gdzie KatInstalacyjnyAplikacji jest nazwą katalogu zawierającego obiekty, z którymi występują problemy.

### Przykład: usunięcie wszystkich uszkodzonych obiektów w systemach plików "root" (/), QOpenSys i UDFS

W tej sytuacji awaria dysku powoduje uszkodzenie pewnej liczby obiektów. Zachodzi potrzeba zidentyfikowania uszkodzonych obiektów przed określeniem właściwego sposobu ich odzyskania.

Konieczne jest rozwiązanie umożliwiające wyszukanie uszkodzonych obiektów bez usuwania problemów. Operacja nie może mieć wpływu na normalne działania na systemie plików.

Aby zidentyfikować uszkodzone obiekty, należy wywołać następującą komendę:

```
RCLLNK OBJ('/') SUBTREE(*ALL) DMGOBJOPT(*KEEP *KEEP)
```

Dodatkowo komenda usunie podczas wyszukiwania problemy nie dotyczące uszkodzonych obiektów.

### Przykład: usunięcie wszystkich uszkodzonych obiektów w systemach plików "root" (/), QOpenSys i UDFS

W tej sytuacji awaria dysku powoduje uszkodzenie pewnej liczby obiektów. Uszkodzone obiekty należy usunąć, aby możliwe było odtworzenie kopii zapasowej z nośnika.

Aby usunąć uszkodzone obiekty, należy wywołać następującą komendę:

```
RCLLNK OBJ('/') SUBTREE(*ALL) DMGOBJOPT(*DELETE *DELETE)
```

Uszkodzone obiekty są usuwane; operacja ta nie ma wpływu na normalne działania na systemie plików. Dodatkowo podczas usuwania uszkodzonych obiektów usuwane są inne problemy.

## Przykład: uruchomienie wielu instancji komendy RCLLNK w celu szybkiego odzyskania wszystkich obiektów w systemach plików "root" (/), QOpenSys i UDFS

W tej sytuacji wszystkie obiekty w systemach plików "root" (/), QOpenSys i podłączonych systemach plików użytkownika są odzyskiwane w trakcie rutynowej obsługi systemu. Użytkownik może pragnąć zakończyć operację odzyskiwania tak szybko, jak to możliwe, aby zrealizować dodatkową obsługę.

Podzielenie operacji odzyskiwania na grupy umożliwia jednoczesne wykonanie wielu instancji komendy RCLLNK, co skraca czas odzyskiwania.

Aby uruchomić wiele operacji odzyskiwania katalogów systemowych i katalogów najwyższego poziomu, należy wywołać poniższe komendy (każdą w osobnym wątku lub zadaniu).

```
RCLLNK OBJ('/') SUBTREE(*DIR)
RCLLNK OBJ('/tmp') SUBTREE(*ALL)
RCLLNK OBJ('/home') SUBTREE(*ALL)
RCLLNK OBJ('/etc') SUBTREE(*ALL)
RCLLNK OBJ('/usr') SUBTREE(*ALL)
RCLLNK OBJ('/QIBM') SUBTREE(*ALL)
RCLLNK OBJ('/QOpenSys') SUBTREE(*ALL)
RCLLNK OBJ('/IaspName') SUBTREE(*ALL)
RCLLNK OBJ('/dev') SUBTREE(*ALL)
RCLLNK OBJ('/InneKatalogiNajwyzszegoPoziomu') SUBTREE(*ALL)
```

gdzie wartość InneKatalogiNajwyzszegoPoziomu oznacza pozostałe katalogi, które mają zostać odzyskane.

---

## Obsługa programistyczna

Aby korzystać z plików strumieniowych, katalogów i innych elementów zintegrowanego systemu plików, należy używać zestawu funkcji API, które umożliwiają dostęp do funkcji dostarczanych przez zintegrowany system plików.

Ponadto dodanie obsługi zintegrowanego systemu plików umożliwia kopiowanie danych między fizycznymi zbiorami baz danych a plikami strumieniowymi. Kopiowanie to można wykonać za pomocą komend CD, funkcji przesyłania danych programu rodzina produktów System i Access lub funkcji API.

## Kopiowanie danych między plikami strumieniowymi a zbiorami baz danych

Użytkownicy znający zasady pracy ze zbiorami baz danych za pomocą narzędzi do przetwarzania rekordów, takich jak język opisu struktur danych (DDS), zauważą zapewne zasadnicze różnice w pracy z plikami strumieniowymi.

Wynikają one z innej struktury (lub też braku struktury) plików strumieniowych w porównaniu ze zbiorami baz danych. Dostęp do pliku strumieniowego uzyskuje się podając przesunięcie w bajtach (byte offset) i długość. Zwykle dostęp do bazy danych uzyskuje się określając, które pola mają zostać użyte i ile rekordów ma zostać przetworzonych.

Ponieważ format i charakterystyka pliku opartego na rekordach definiowane są z góry, system operacyjny zna układ pliku i jest w stanie pomóc uniknąć działań niewłaściwych dla danego formatu i charakterystyki pliku. W przypadku plików strumieniowych system operacyjny ma niewielką lub wręcz żadną wiedzę na temat formatu pliku. Program użytkowy musi wiedzieć, jak wygląda plik i jak poprawnie wykonywać na nim działania. Pliki strumieniowe oferują niezwykle elastyczne środowisko programistyczne, ale dzieje się to kosztem rezygnacji z pomocy systemu operacyjnego. Pliki strumieniowe bardziej nadają się do użycia w niektórych przypadkach programowania; w innych przypadkach lepiej jest użyć plików opartych na rekordach.

### Pojęcia pokrewne

“Plik strumieniowy” na stronie 15

*Plik strumieniowy* jest dostępną losowo sekwencją bajtów, pozbawioną struktury narzucanej przez system.

## Kopiowanie danych za pomocą komend CL

Istnieją dwa zestawy komend CL, które umożliwiają kopiowanie danych między plikami strumieniowymi a podzbiorami zbiorów baz danych.

## Komendy CPYTOSTMF i CPYFRMSTMF

Do kopiowania danych między plikami strumieniowymi a podzbiorami baz danych można używać komendy Kopiowanie z pliku strumieniowego (Copy from Stream File - CPYFRMSTMF) i Kopiowanie do pliku strumieniowego (Copy to Stream File - CPYTOSTMF). Plik strumieniowy można utworzyć z podzbioru bazy danych za pomocą komendy CCPYTOSTMF. Można też utworzyć podzbiór bazy danych z pliku strumieniowego, używając komendy CPYFRMSTMF. Jeśli zbiór lub podzbiór, do którego są kopiowane dane, nie istnieje, to zostanie on utworzony.

Pojawiają się jednak pewne ograniczenia. Zbiór bazy danych musi być albo zbiorem fizycznym opisywanym przez program i zawierającym tylko jedno pole, albo źródłowym zbiorem fizycznym zawierającym tylko jedno pole tekstowe. Omawiane komendy obejmują wiele opcji konwersji i zmiany formatu kopiowanych danych.

Komendy CPYTOSTMF i CPYFRMSTMF mogą również zostać użyte do kopiowania danych między plikiem strumieniowym a zbiorem składowania.

## Komendy CPYTOIMPF i CPYFRMIMPF

Do kopiowania danych między plikami strumieniowymi a podzbiorami baz danych można również użyć komend Kopiowanie do zbioru importu (Copy to Import File - CPYTOIMPF) i Kopiowanie ze zbioru importu (Copy from Import File - CPYFRMIMPF). Komendy CPYTOSTMF i CPYFRMSTMF uniemożliwiają przenoszenie danych ze złożonych, zewnątrznie opisanych (za pomocą DDS) zbiorów baz danych. Określenie *plik importu (import file)* odnosi się do pliku typu strumieniowego; termin ten zazwyczaj oznacza plik utworzony w celu kopiowania danych między heterogenicznymi bazami danych.

Podczas kopiowania z pliku strumieniowego (lub pliku importu) komenda CPYFRMIMPF pozwala użytkownikowi określić plik definicji pola (field definition file - FDF), który opisuje dane zawarte w tym pliku strumieniowym. Użytkownik może też określić, czy plik strumieniowy jest ograniczony i jakie znaki zostaną użyte do oznaczenia granic łańcuchów, pól i rekordów. Dostępne są też opcje przekształcania specjalnych typów danych, takich jak data i godzina.

Konwersja danych jest w tych komendach dostępna, jeśli docelowy plik strumieniowy lub podzbiór bazy danych już istnieje. Jeśli ten plik nie istnieje, można wykorzystać przedstawioną poniżej dwustopniową metodę uzyskiwania przekształconych danych.

1. Użyj komend CPYTOIMPF i CPYFRMIMPF do skopiowania danych z pliku zewnątrznie opisanego do źródłowego zbioru fizycznego.
2. Komend CPYTOSTMF i CPYFRMSTMF (które przeprowadzają pełną konwersję danych bez względu na to, gdzie znajduje się plik docelowy) można użyć do skopiowania danych między fizycznym plikiem źródłowym, a plikiem strumieniowym.

A oto przykład:

```
CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML('X'07') STRDLML(*DBLQUOTE) DATFMT(*USA) TIMFMT(*USA)
```

Parametr DTAFMT określa, czy strumień wejściowy (plik importu) jest ograniczony; można też wybrać parametr DTAFMT(\*FIXED), który wymaga podania pliku definicji pola. Parametry FLDDLML, RCDDLML i STRDLML określają znaki, które działają jako ograniczniki lub separatory pól, rekordów i łańcuchów.

Parametry DATFMT i TIMFMT określają format wszelkich informacji dotyczących daty i godziny, które są kopiowane do pliku importu.

Komendy te są użyteczne, ponieważ mogą zostać umieszczone w programie i działają wyłącznie w systemie. Jednak ich interfejsy są złożone.

### Informacje pokrewne

Komenda Kopiowanie do pliku strumieniowego (Copy to Stream File - CPYTOSTMF)

Komenda Kopiowanie z pliku strumieniowego (Copy from Stream File - CPYFRMSTMF)

Komenda Kopiowanie do zbioru importu (Copy to Import File - CPYTOIMPF)

Kopiowanie ze zbioru importu (Copy from Import File - CPYFRMIMPF)

Język CL

## Kopiowanie danych za pomocą funkcji API

Aby skopiować podzbiory zbioru bazy danych do pliku strumieniowego w aplikacji, należy użyć funkcji `open()`, `read()` i `write()` w celu otwarcia podzbioru, odczytania danych i zapisania ich w nim lub w innym pliku.

### Informacje pokrewne

`open()`--funkcja API Open File

`read()`--funkcja API Read from Descriptor

`write()`--funkcja API Write to Descriptor

Funkcje API zintegrowanego systemu plików

## Kopiowanie danych za pomocą funkcji przesyłania danych

Zaletami aplikacji przesyłania danych rodziny produktów System i Access są: łatwy w użyciu interfejs graficzny, automatyczna numeracja i konwersja danych znakowych.

Funkcja przesyłania danych wymaga jednak zainstalowania rodziny produktów System i Access i używania zarówno zasobów komputera PC, jak i zasobów systemu i5/OS, a także korzystania z komunikacji między nimi.

Jeśli na komputerze PC i w systemie jest zainstalowana rodzina produktów System i Access, można ją wykorzystać do przesyłania danych między plikami strumieniowymi i zbiorami baz danych. Można również przysłać dane do nowego zbioru bazy danych na podstawie istniejącego zbioru bazy danych, do zewnętrznie opisanego zbioru bazy danych lub do nowej definicji zbioru bazy danych i do nowego zbioru.

### Przesyłanie danych ze zbioru bazy danych do pliku strumieniowego:

Aby przesłać dane ze zbioru bazy danych do pliku strumieniowego w systemie, wykonaj następujące czynności:

1. Połącz się z systemem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików i5/OS.
3. W oknie programu System i Access for Windows kliknij opcję **Przesyłanie danych z systemu System i** (Data Transfer From System i).
4. Wybierz system, z którego chcesz przysłać dane.
5. Używając baz danych biblioteki i5/OS i nazwy pliku, wybierz nazwę pliku, z którego będą kopiowane dane, a także napęd sieciowy dla wynikowego pliku strumieniowego. Można też kliknąć przycisk **Szczegóły pliku PC** (PC File Details) w celu wybrania formatu pliku PC dla pliku strumieniowego. Transfer danych obsługuje najczęściej spotykane typy plików PC, takie jak: tekstowy ASCII, BIFF3, CSV, DIF, tekstowy ograniczony znakami tabulacji oraz WK4.
6. Kliknij przycisk **Przesyłanie danych z systemu System i**, aby uruchomić przesyłanie plików.

Tego typu przesyłanie danych można też przeprowadzić uruchamiając aplikację do przesyłania danych w zadaniu wsadowym. W tym celu należy wykonać czynności opisane powyżej, ale z wybraniem opcji menu **Plik**, aby zapisać żądanie przesyłania. Aplikacja Data Transfer To System i tworzy plik `.DTT` lub `.TFR`. Natomiast aplikacja Data Transfer From System i tworzy plik `.DTF` lub `.TTO`. W katalogu rodzina produktów System i Access z wiersza komend można uruchomić wsadowo dwa programy:

- RTOPCB, którego parametrem jest plik `.DTF` lub plik `.TTO`
- RFROMPCB, którego parametrem jest plik `.DTT` lub plik `.TFR`

Używając programu planującego można w określonym terminie uruchomić dowolną z tych komend. Na przykład, można użyć narzędzia System Agent Tool (część pakietu Plus Pack firmy Microsoft) do wskazania programu, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), i godziny jego uruchomienia.

### Przesyłanie danych z pliku strumieniowego do zbioru bazy danych:



Aby przesłać dane z pliku strumieniowego do pliku bazy danych w systemie, wykonaj następujące czynności:

1. Połącz się z systemem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików i5/OS.
3. W oknie programu System i Access for Windows kliknij opcję **Przesyłanie danych do systemu System i** (Data Transfer To System i).
4. Na komputerze PC wybierz nazwę pliku, który ma być przesłany. Dla nazwy pliku PC możesz użyć przycisku **Przeglądaj** (dla przypisanego wcześniej napędu sieciowego) i wybrać plik strumieniowy. Możesz też użyć pliku strumieniowego znajdującego się na komputerze PC.
5. Wybierz system, na którym chcesz umieścić zewnętrznie opisany zbiór bazy danych.
6. Kliknij przycisk **Przesyłanie danych do systemu System i** (Transfer data to System i), aby uruchomić przesłanie plików.

**Uwaga:** Jeśli dane są przesyłane do istniejącej definicji zbioru bazy danych, aplikacja Data Transfer to System i wymaga użycia powiązanego pliku opisu formatu (FDF). Plik FDF opisuje format pliku strumieniowego. Jest on tworzony przez aplikację Data Transfer from System i po przesłaniu danych ze zbioru bazy danych do pliku strumieniowego. Aby zakończyć przesłanie danych z pliku strumieniowego do pliku bazy danych, kliknij przycisk **Przesyłanie danych do systemu System i**. Jeśli istniejący plik .FDF nie jest dostępny, można go szybko utworzyć.

Tego typu przesłanie danych można też przeprowadzić uruchamiając aplikację do przesyłania danych w zadaniu wsadowym. W tym celu należy wykonać czynności opisane powyżej, ale z wybraniem opcji menu **Plik**, aby zapisać żądanie przesyłania. Aplikacja Data Transfer To System i tworzy plik .DTT lub .TFR. Natomiast aplikacja Data Transfer From System i tworzy plik .DTF lub .TTO. W katalogu rodzina produktów System i Access z wiersza komend można uruchomić wsadowo dwa programy:

- RTOPCB, którego parametrem jest plik .DTF lub plik .TTO
- RFROMPCB, którego parametrem jest plik .DTT lub plik .TFR

Używając programu planującego można w określonym terminie uruchomić dowolną z tych komend. Na przykład, można użyć narzędzia System Agent Tool (część pakietu Plus Pack firmy Microsoft) do wskazania programu, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), i godziny jego uruchomienia.

#### **Odsyłacze pokrewne**

“Tworzenie pliku opisu formatu”

Jeśli dane są przesyłane do istniejącego zbioru bazy danych w systemie, aplikacja Data Transfer to System i wymaga użycia powiązanego pliku opisu formatu (FDF).

#### **Przesyłanie danych do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru:**

Aby przesłać dane do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru, wykonaj następujące czynności:

1. Połącz się z systemem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików i5/OS.
3. W oknie System i Access for Windows kliknij opcję **Przesyłanie danych do systemu System i**.
4. Otwórz menu Narzędzia aplikacji Data Transfer to System i.
5. Kliknij polecenie **Tworzenie zbioru bazy danych System i**.

Zostanie uruchomiony kreator, który pomoże utworzyć nowy zbiór bazy danych System i z istniejącego pliku PC. Konieczne będzie podanie nazwy pliku PC, z którego zostanie utworzony zbiór System i, nazwy tworzonego zbioru System i oraz kilku innych potrzebnych szczegółów. To narzędzie analizuje składnię podanego pliku strumieniowego, w celu określenia liczby, typu i wielkości pól wymaganych w wynikowym zbiorze bazy danych. Narzędzia tego można następnie użyć do utworzenia definicji zbioru bazy danych w systemie.

#### **Tworzenie pliku opisu formatu:**

Jeśli dane są przesyłane do istniejącego zbioru bazy danych w systemie, aplikacja Data Transfer to System i wymaga użycia powiązanego pliku opisu formatu (FDF).

Plik FDF opisuje format pliku strumieniowego. Jest on tworzony przez aplikację Data Transfer from System i po przesłaniu danych ze zbioru bazy danych do pliku strumieniowego.

Aby utworzyć plik .FDF, wykonaj następujące czynności:

1. Utwórz opisany zewnętrznie zbiór bazy danych o formacie odpowiadającym źródłowemu plikowi strumieniowemu (liczba pól, typy danych).
2. Utwórz jeden tymczasowy rekord w zbiorze bazy danych.
3. Funkcja Data Transfer from System i służy do tworzenia pliku strumieniowego i przypisanego do niego pliku .FDF ze zbioru bazy danych.

Teraz można użyć funkcji Data Transfer to System i. Określ plik .FDF oraz źródłowy plik strumieniowy, który ma być przesłany.

#### **Odsyłacze pokrewne**

“Przesyłanie danych ze zbioru bazy danych do pliku strumieniowego” na stronie 104

Aby przesłać dane ze zbioru bazy danych do pliku strumieniowego w systemie, wykonaj następujące czynności:

“Przesyłanie danych z pliku strumieniowego do zbioru bazy danych” na stronie 104

Aby przesłać dane z pliku strumieniowego do pliku bazy danych w systemie, wykonaj następujące czynności:

## **Kopiowanie danych między plikami strumieniowymi a zbiorami składowania**

Zbiór składowania jest używany, wraz z komendami składowania i odtwarzania, do zachowywania danych, które w przeciwnym wypadku byłyby zapisywane na taśmie lub na dyskietce.

Zbioru tego można również używać jak zbioru bazy danych do odczytywania i zapisywania rekordów zawierających informacje o składowaniu/odtworzeniu. Zbioru składowania można ponadto używać do wysyłania obiektów do innych użytkowników w sieci SNADS.

Do kopiowania zbioru składowania do pliku strumieniowego i odwrotnie służy komenda CPY. Jednak przy kopiowaniu pliku strumieniowego z powrotem do zbioru składowania, dane muszą mieć poprawny format danych składowania (muszą to być dane pochodzące ze zbioru składowania i skopiowane do pliku strumieniowego).

Używając klienta PC można uzyskać dostęp do zbioru składowania i skopiować dane do pamięci komputera PC lub sieci LAN. Należy jednak pamiętać, że system plików NFS nie umożliwia dostępu do danych w zbiorze składowania.

#### **Informacje pokrewne**

Komenda Kopiowanie obiektu (Copy Object - CPY)

## **Wykonywanie operacji za pomocą funkcji API**

Wiele interfejsów API, które wykonują operacje na obiektach zintegrowanego systemu plików, ma postać funkcji napisanych w języku C.

Dostępne są dwa zestawy funkcji, z których obu można używać w programach tworzonych przy pomocy środowiska Integrated Language Environment (ILE) języka C:

- funkcje zintegrowanego systemu plików w języku C zawarte w systemie operacyjnym i5/OS,
- funkcje języka C udostępniane przez program licencjonowany ILE.

Informacje o programach obsługi wyjścia obsługiwanych przez zintegrowany system plików można uzyskać po wybraniu odsyłacza Tabela 14 na stronie 112.

Funkcje zintegrowanego systemu plików działają jedynie poprzez opcję obsługi wejścia/wyjścia strumieni zintegrowanego systemu plików. Następujące funkcje API są obsługiwane:

Tabela 13. Funkcje API zintegrowanego systemu plików

<b>Funkcja</b>	<b>Opis</b>
access()	Określenie dostępności pliku
accessx()	Określenie dostępności pliku dla klasy użytkowników
chdir()	Zmiana bieżącego katalogu
chmod()	Zmiana uprawnień do pliku
chown()	Zmiana właściciela i grupy dla pliku
close()	Zamknięcie deskryptora pliku
closedir()	Zamknięcie katalogu
creat()	Utworzenie nowego lub wyczyszczenie istniejącego pliku
creat64()	Utworzenie nowego lub wyczyszczenie istniejącego pliku (również dla dużych plików)
DosSetFileLocks()	Blokowanie i anulowanie blokady zakresu bajtów w pliku
DosSetFileLocks64()	Blokowanie i anulowanie blokady zakresu bajtów w pliku (obsługa wielkich plików)
DosSetRelMaxFH()	Zmiana maksymalnej liczby deskryptorów plików
dup()	Powielenie deskryptora otwartego pliku
dup2()	Powielenie deskryptora otwartego pliku do nowego deskryptora
faccessx()	Określenie dostępności pliku dla klasy użytkowników przez deskryptor
fchdir()	Zmiana bieżącego katalogu przez deskryptor
fchmod()	Zmiana uprawnień do pliku przez deskryptor
fchown()	Zmiana właściciela i grupy dla pliku przez deskryptor
fclear()	Usunięcie zawartości pliku
fclear64()	Usunięcie zawartości pliku (obsługa wielkich plików)
fcntl()	Wykonanie kontroli pliku
fpathconf()	Pobranie konfigurowalnych zmiennych z nazwami ścieżek przez deskryptor
fstat()	Pobranie informacji o pliku przez deskryptor
fstat64()	Pobranie informacji o pliku przez deskryptor (również dla dużych plików)
fstatvfs()	Pobranie informacji przez deskryptor
fstatvfs64()	Pobranie informacji przez deskryptor (uaktywniony tryb 64-bitowy)
fsync()	Synchronizacja zmian w pliku
ftruncate()	Obcięcie pliku
ftruncate64()	Obcięcie pliku (również dla dużych plików)
getcwd()	Pobranie ścieżki bieżącego katalogu
getegid()	Pobranie obowiązującego ID grupy
geteuid()	Pobranie obowiązującego ID użytkownika
getgid()	Pobranie rzeczywistego ID grupy
getgrgid()	Pobranie informacji o grupie za pomocą ID grupy
getgrnam()	Pobranie informacji o grupie za pomocą nazwy grupy
getgroups()	Pobranie ID grupy

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
getpwnam()	Pobranie informacji o użytkowniku dla danej nazwy użytkownika
getpwuid()	Pobranie informacji o użytkowniku dla danego ID użytkownika
getuid()	Pobranie rzeczywistego ID użytkownika
givedescriptor()	Przekazanie dostępu do pliku do innego zadania
ioctl()	Wykonanie kontroli we/wy pliku
link()	Utworzenie dowiązania do pliku
lseek()	Ustawienie przesunięcia (offset) odczytu/zapisu dla pliku
lseek64()	Ustawienie przesunięcia (offset) odczytu/zapisu dla pliku (również dla dużych plików)
lstat()	Pobranie informacji o pliku lub dowiązaniu
lstat64()	Pobranie informacji o pliku lub dowiązaniu (również dla dużych plików)
mkdir()	Utworzenie katalogu
mkfifo()	Utworzenie specjalnego pliku FIFO
mmap()	Utworzenie mapy pamięci
mmap64()	Utworzenie mapy pamięci (również dla dużych plików)
mprotect()	Zmiana ochrony mapy pamięci
msync()	Synchronizacja mapy pamięci
munmap()	Usunięcie mapy pamięci
open()	Otwarcie pliku
open64()	Otwarcie pliku (również dla dużych plików)
opendir()	Otwarcie katalogu
pathconf()	Pobranie konfigurowalnych zmiennych z nazwami ścieżek
pread()	Wczytanie z deskryptora z przesunięciem
pread64()	Wczytanie z deskryptora z przesunięciem (również dla dużych plików)
pwrite()	Zapis do deskryptora z przesunięciem
pwrite64()	Zapis do deskryptora z przesunięciem (również dla dużych plików)
QjoEndJournal()	Zakończenie kronikowania
QjoRetrieveJournal Information()	Odtworzenie informacji kroniki
QjoRetrieveJournalEntries()	Odtworzenie pozycji kroniki
QJORJIDI()	Odtworzenie informacji o identyfikatorze kroniki
QJOSJRNE()	Wysłanie pozycji do kroniki
QjoStartJournal()	Uruchomienie kronikowania
QlgAccess()	Określenie dostępności pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgAccessx()	kreślenie dostępności pliku dla klasy użytkowników (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgChdir()	Zmiana bieżącego katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

<b>Funkcja</b>	<b>Opis</b>
QlgChmod()	Zmiana autoryzacji pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgChown()	Zmiana właściciela i grupy dla pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgCreat()	Utworzenie nowego lub przepisanie istniejącego pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgCreat64()	Utworzenie nowego lub przepisanie istniejącego pliku (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgCvtPathToQSYSObjName()	Tłumaczenie nazwy ścieżki zintegrowanego systemu plików na nazwę obiektu QSYS (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetAttr()	Pobranie atrybutów systemowych dla obiektu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetcwd()	Pobranie nazwy ścieżki dla bieżącego katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetPathFromFileID()	Pobranie nazwy ścieżki obiektu z jego identyfikatora (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetpwnam()	Pobranie informacji o użytkowniku dla danej nazwy użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetpwnam_r()	Pobranie informacji o użytkowniku dla danej nazwy użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetpwuid()	Pobranie informacji o użytkowniku dla danego identyfikatora użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgGetpwuid_r()	Pobranie informacji o użytkowniku dla danego identyfikatora użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgLchown()	Zmiana właściciela i grupy dla dowiązania symbolicznego (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgLink()	Utworzenie dowiązania do pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgLstat()	Pobranie informacji o pliku lub dowiązaniu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgLstat64()	Pobranie informacji o pliku lub dowiązaniu (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgMkdir()	Utworzenie katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgMkfifo()	Utworzenie specjalnego pliku FIFO (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgOpen()	Otwarcie pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
QlgOpen64()	Otwarcie pliku (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgOpendir()	Otwarcie katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgPathconf()	Pobranie zmiennych konfiguracyjnych nazwy ścieżki (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgProcessSubtree()	Przetworzenie katalogów lub obiektów wewnątrz drzewa katalogów (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgReaddir()	Odczytanie pozycji katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgReaddir_r()	Odczytanie pozycji katalogu (z ochroną wątków i wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgReadlink()	Odczytanie wartości dowiązania symbolicznego (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgRenameKeep()	Zmiana nazwy pliku lub katalogu z zachowaniem atrybutu <i>nowy</i> , jeśli już istnieje (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgRenameUnlink()	Zmiana nazwy pliku lub katalogu z usunięciem dowiązania do <i>nowego</i> , jeśli już istnieje (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgRmdir()	Usunięcie katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgSaveStgFree()	Zeskładowanie danych obiektów i zwolnienie ich pamięci (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgSetAttr()	Ustawienie atrybutów systemowych dla obiektu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStat()	Pobranie informacji o pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStat64()	Pobranie informacji o pliku (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStatvfs()	Pobranie informacji o systemie plików (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStatvfs64()	Pobranie informacji o systemie plików (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgSymlink()	Utworzenie dowiązania symbolicznego (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgUnlink()	Usunięcie dowiązania do pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgUtime()	Ustawienie godziny dostępu do pliku i jego modyfikacji (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QP0FPTOS()	Wykonanie różnych funkcji systemowych
QP0LCHSG()	Zmiana sygnatury skanowania

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
Qp0lCvtPathToSYSObjName()	Tłumaczenie nazwy ścieżki zintegrowanego systemu plików na nazwę obiektu QSYS
QP0LFLOP()	Wykonanie różnych operacji na obiektach
Qp0lGetAttr()	Pobranie atrybutów systemowych dla obiektu
Qp0lGetPathFromFileID()	Pobranie ścieżki obiektu z jego ID pliku
Qp0lOpen()	Otwarcie pliku z nazwą ścieżki dostępną dla NLS
Qp0lProcessSubtree()	Przetworzenie katalogów lub obiektów wewnątrz drzewa katalogów
Qp0lRenameKeep()	Zmiana nazwy pliku lub katalogu z zachowaniem <i>nowego</i> , jeśli istnieje
Qp0lRenameUnlink()	Zmiana nazwy pliku lub katalogu z usunięciem dowiązania do <i>nowego</i> , jeśli istnieje
QP0LROR()	Wczytanie referencji do obiektu
QP0LRRO()	Pobranie obiektów odniesienia
QP0LRTSG()	Pobranie sygnatury skanowania
Qp0lSaveStgFree()	Zeskładowanie danych obiektów i zwolnienie ich pamięci
Qp0lSetAttr()	Ustawienie atrybutów systemowych dla obiektu
Qp0lUnlink()	Odłączenie pliku z nazwą ścieżki dostępną dla NLS
Qp0zPipe()	Utworzenie międzyprocesowego kanału z gniazdami
qsysetegid()	Ustawienie obowiązującego ID grupy
qsyseteuid()	Ustawienie obowiązującego ID użytkownika
qsysetid()	Ustawienie ID grupy
qsysetregid()	Ustawienie rzeczywistego i obowiązującego ID grupy
qsysetreuid()	Ustawienie rzeczywistego i obowiązującego ID użytkownika
qsysetuid()	Ustawienie ID użytkownika
QZNFRTVE()	Odzyskanie informacji dotyczących eksportu NFS
read()	Wczytanie z pliku
readdir()	Wczytanie pozycji katalogu
readdir_r()	Wczytanie pozycji katalogu (bezpieczne przy wątkach)
readlink()	Wczytanie wartości dowiązania symbolicznego
readv()	Wczytanie z pliku (wektor)
rename()	Zmiana nazwy pliku lub katalogu. Można ją zdefiniować ze składnią Qp0lRenameKeep() lub Qp0lRenameUnlink()
rewinddir()	Ustawienie początkowe (reset) strumienia katalogu
rmdir()	Usunięcie katalogu
select()	Sprawdzenie statusu we/wy wielu deskryptorów plików
stat()	Pobranie informacji o pliku
stat64()	Pobranie informacji o pliku (również dla dużych plików)
statvfs()	Pobranie informacji o systemie plików
statvfs64()	Pobranie informacji o systemie plików (również dla dużych plików)
symlink()	Utworzenie dowiązania symbolicznego
sysconf()	Pobranie zmiennych konfiguracyjnych systemu

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
takedescriptor()	Przejęcie dostępu do pliku od innego zadania
umask()	Ustawienie maski uprawnień dla zadania
unlink()	Usunięcie dowiązania do pliku
utime()	Ustawienie czasów dostępu i modyfikacji pliku
write()	Zapis do pliku
writev()	Zapis do pliku (wektor)

**Uwaga:** Niektóre z tych funkcji używane są także dla gniazd systemu i5/OS.

Tabela 14. Programy obsługi wyjścia zintegrowanego systemu plików

Funkcja	Opis
Program obsługi wyjścia zintegrowanego systemu plików Skanowanie przy zamykaniu (Scan on Close)	Wywoływana w trakcie przetwarzania po zamknięciu, na przykład funkcją API close(). Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Program obsługi wyjścia zintegrowanego systemu plików Skanowanie przy otwieraniu (Scan on Open)	Wywoływana w trakcie przetwarzania po otwarciu, na przykład funkcją API open(). Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Przetworzenie nazwy ścieżki	Wywołane przez funkcję API Qp0IProcessSubtree() dla każdego obiektu wyszukanego przez funkcję API, który spełnia kryteria wyboru. Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Zwolnienie pamięci składowania	Wywoływane przez funkcję API Qp0ISaveStgFree() w celu zeskładowania typu obiektu *STMF. Ten program obsługi wyjścia musi być utworzony przez użytkownika.

### Pojęcia pokrewne

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

### Odsyłacze pokrewne

“Przykład: funkcje C zintegrowanego systemu plików” na stronie 118

Kolejne strony zawierają prosty program napisany w języku C, przedstawiający użycie pewnych funkcji zintegrowanego systemu plików.

“Kopiowanie danych za pomocą funkcji API” na stronie 104

Aby skopiować podzbiory zbioru bazy danych do pliku strumieniowego w aplikacji, należy użyć funkcji open(), read() i write() w celu otwarcia podzbioru, odczytania danych i zapisania ich w nim lub w innym pliku.

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Funkcje ILE C

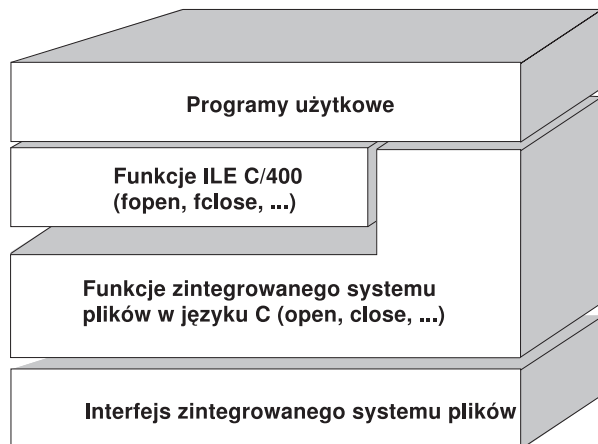
Funkcje ILE C zawierają standardowe funkcje C zdefiniowane przez instytut American National Standards Institute (ANSI).

Funkcje te mogą obsługiwać zarządzanie danymi wejścia/wyjścia lub strumieniowe wejście/wyjście zintegrowanego systemu plików, zależnie od deklaracji przy tworzeniu programu w języku C. Domyślnie kompilator używa zarządzania danymi wejścia/wyjścia.

Aby wskazać programowi kompilującemu użycie strumieniowych danych we/wy zintegrowanego systemu plików, należy podać wartość \*IFSIO parametru Interfejs systemowy (SYSIFCOPT) w komendzie Utworzenie Modułu ILE



C(Create ILE C Module - CRTCMOD) lub Utworzenie przypisanego programu C (Create Bound C Program - CRTBNDC). Po wpisaniu wartości \*IFSIO, zamiast funkcji wejścia/wyjścia zarządzania danymi, dołączane są funkcje wejścia/wyjścia zintegrowanego systemu plików. W rezultacie, funkcje C ILE C używają funkcji zintegrowanego systemu plików do przeprowadzania operacji we/wy.



RV3N070-3

Rysunek 10. Funkcje ILE C używają funkcji strumieniowego we/wy zintegrowanego systemu plików

Więcej informacji na temat używania funkcji ILE C z danymi strumieniowymi we/wy zintegrowanego systemu plików

zawiera publikacja WebSphere Development Studio: ILE C/C++ Programmer's Guide . Szczegółowe informacje o każdej funkcji C ILE C można znaleźć w publikacji WebSphere Development Studio: C/C++ Language Reference



## Obsługa dużych plików

Funkcje API zintegrowanego systemu plików zostały tak rozbudowane, aby umożliwić aplikacjom przechowywanie w tym systemie bardzo dużych plików. Maksymalna wielkość obiektu wynosi dla nich około 1 TB (1 TB jest równy około 1 099 511 627 776 bajtom) w systemach plików "root" (/), QOpenSys i użytkownika.

Zintegrowany system plików wyposażony jest w zestaw 64-bitowych funkcji API typu UNIX i umożliwia łatwe odwzorowanie istniejących 32-bitowych funkcji API na funkcje API 64-bitowe, które mogą uzyskać dostęp do dużych plików za pomocą 8-bajtowych argumentów całkowitych.

Poniższe sytuacje umożliwiają aplikacjom użycie obsługi dużych plików:

- Jeśli w czasie kompilacji zostanie zdefiniowana etykieta makro `_LARGE_FILE_API`, aplikacje mają dostęp do 64-bitowych funkcji API i struktur danych. Na przykład aplikacja, która zamierza używać funkcji API `stat64()` i struktury `stat64`, będzie wymagała zdefiniowania podczas kompilacji etykiety `_LARGE_FILE_API`.
- Jeśli etykieta makro `_LARGE_FILES` zostanie zdefiniowana przez aplikację podczas kompilacji, istniejące funkcje API i struktury danych są odwzorowane na swoje wersje 64-bitowe. Na przykład, jeśli aplikacja definiuje podczas kompilacji etykietę `_LARGE_FILES`, to wywołanie funkcji API `stat()` zostanie odwzorowane na funkcję API `stat64()`, a struktura `stat()` zostanie odwzorowana na strukturę `stat64()`.

Aplikacje, które będą obsługiwały duże pliki, mogą albo zdefiniować podczas kompilacji etykietę `_LARGE_FILE_API` i bezpośrednio używać 64-bitowych funkcji API, albo zdefiniować podczas kompilacji etykietę `_LARGE_FILES`. Wszystkie odpowiednie funkcje API i struktury danych zostaną automatycznie odwzorowane na wersje 64-bitowe.

Aplikacje, które nie będą obsługiwały dużych plików, mogą używać funkcji API zintegrowanego systemu plików w dotychczasowej postaci.

## Informacje pokrewne

Funkcje API zintegrowanego systemu plików

stat64()--funkcja API Get File Information (Large File Enabled)

stat()--funkcja API Get File Information

## Reguły dotyczące ścieżek dla funkcji API

Gdy do wykonywania działań na obiekcie używane są zintegrowany system plików lub funkcja API ILE C, obiekt identyfikowany jest przez podanie ścieżki do katalogu. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API.

Określenie *obiekt* w tych zasadach odnosi się do dowolnego katalogu, pliku, dowiązania lub innego obiektu.

- Ścieżki podawane są w porządku hierarchicznym, poczynając od najwyższego poziomu w hierarchii katalogów. Nazwy składników ścieżki oddzielane są ukośnikami (/), na przykład:

Kat1/Kat2/Kat3/PlikUz

Ukośnik odwrotny (\) nie jest rozpoznawany jako separator. Jest on traktowany jak zwykły znak w nazwie.

- Nazwy obiektów w ramach katalogu nie mogą się powtarzać.
- Maksymalna długość każdego składnika ścieżki i maksymalna długość łańcucha określającego ścieżkę mogą się zmieniać w zależności od systemu plików.
- Znak / na początku ścieżki oznacza, że zaczyna się ona w katalogu głównym ("root"/), na przykład:

/Kat1/Kat2/Kat3/PlikUz

- Jeśli ścieżka nie zaczyna się od znaku /, przyjmuje się, że zaczyna się ona w bieżącym katalogu, na przykład:

MójKat/MójPlik

gdzie *MójKat* jest podkatalogiem bieżącego katalogu.

- Aby uniknąć pomyłek związanych z wartościami specjalnymi systemu i5/OS, nazwy ścieżek nie mogą rozpoczynać się od pojedynczego znaku gwiazdki (\*). Aby podać nazwę ścieżki, która rozpoczyna się od dowolnych cyfr lub znaków, należy użyć dwóch gwiazdek (\*), na przykład:

'\*\*.plik'

Dotyczy to tylko względnych nazw ścieżek, w których przed gwiazdką (\*) nie występują inne znaki.

- Podczas działań na obiektach w systemie plików QSYS.LIB nazwy składników muszą przybrać postać *nazwa.typ-obiektu*, na przykład:

/QSYS.LIB/PAYROLL.LIB/PAY.FILE

- Podczas działań na obiektach w systemie plików QSYS.LIB niezależnej puli ASP nazwy składników muszą przybrać postać *nazwa.typ-obiektu*, na przykład:

/nazwa\_asp/QSYS.LIB/PAYDAVE.LIB/PAY.FILE

- W nazwach ścieżek nie wolno używać dwukropka (:). Ma on specjalne znaczenie w systemie.
- W przeciwieństwie do nazw ścieżek w komendach zintegrowanego systemu plików, gwiazdka(\*), znak zapytania (?), cudzysłów pojedynczy ('), cudzysłów (") oraz tylda (~) nie mają specjalnego znaczenia. Traktowane są one jak zwykłe znaki w nazwie. Aby uniknąć pomyłek związanych z wartościami specjalnymi systemu i5/OS, nazwy ścieżek nie mogą rozpoczynać się od pojedynczego znaku gwiazdki (\*). Jedynymi funkcjami API, których zasada ta nie dotyczy, są funkcje QjoEndJournal i QjoStartJournal.
- Gdy używane są interfejsy API Qlg (z użyciem nazw ścieżek obsługiwanych w systemie plików NLS), znaki null w nazwach ścieżek są niedozwolone, chyba że znak ten podany zostanie jako ogranicznik nazwy ścieżki.

## Odsyłacze pokrewne

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 66

Gdy do wykonywania działań na obiekcie używane są komendy lub ekrany zintegrowanego systemu plików, obiekt identyfikowany jest przez podanie jego właściwej ścieżki.

## Informacje pokrewne

Funkcja API Zakończenie kronikowania (End Journal - QjoEndJournal)

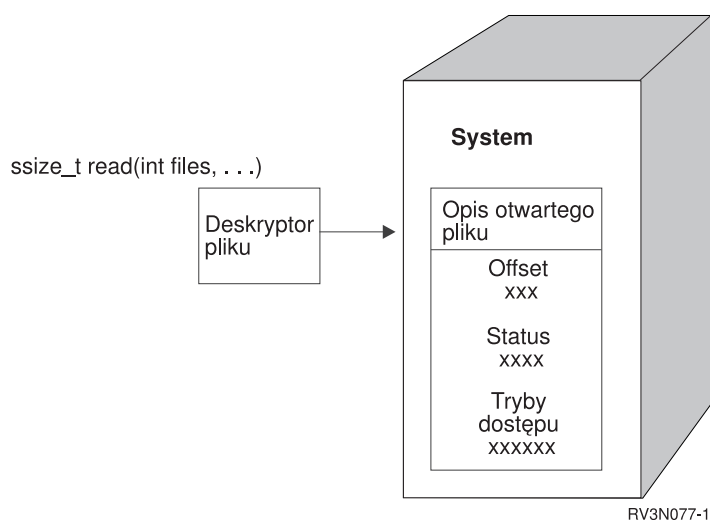
Funkcja API Uruchomienie kroniki (Start Journal - QjoStartJournal)

## Deskryptor pliku

Gdy funkcje ILE C obsługi strumieniowych danych we/wy zdefiniowane przez instytut American National Standards Institute (ANSI) używane są do operacji wykonywanych na pliku, plik identyfikowany jest przy użyciu wskaźników. Używając funkcji C zintegrowanego systemu plików, plik określa się podając deskryptor pliku. *Deskryptor pliku* jest dodatnią liczbą całkowitą, która musi być unikalna dla każdego zadania.

Zadanie używa deskryptora pliku do identyfikacji otwartego pliku podczas przeprowadzania na nim operacji. Deskryptor pliku reprezentowany jest w funkcjach w języku C działających na zintegrowanym systemie plików przez zmienną *files*, a w funkcjach w języku C działających na gniazdach przez zmienną *descriptor*.

Każdy deskryptor pliku powiązany jest z *opisem otwartego pliku*, który zawiera informacje, takie jak przesunięcie (offset), status pliku i tryby dostępu do niego. Z tym samym opisem otwartego pliku może być powiązanych wiele deskryptorów, ale deskryptor pliku może wskazywać tylko na jeden opis otwartego pliku.



Rysunek 11. Deskryptor pliku i opis otwartego pliku.

Gdy funkcje ILE C obsługi strumieniowych danych we/wy używane są ze zintegrowanym systemem plików, obsługa czasu wykonywania funkcji ILE C zmienia wskaźnik pliku na deskryptor pliku.

- | Używając systemów plików “root” (/), QOpenSys lub UDFS, można przekazywać dostęp do opisu otwartego pliku z
- | jednego zadania do innego, umożliwiając mu w ten sposób dostęp do tego pliku. Służą do tego funkcje
- | `givedescriptor()`, `takedescriptor()`, `sendmsg()`, i `recvmsg()` przekazujące deskryptory między zadaniami.

### Informacje pokrewne

- `givedescriptor()`--funkcja API Pass Descriptor Access to Another Job
- `takedescriptor()`--funkcja API Receive Socket Access from Another Job
- `sendmsg()`--funkcja API Send a Message Over a Socket
- `recvmsg()`--funkcja API Receive a Message Over a Socket
- Programowanie z użyciem gniazd
- Interfejsy API gniazd

## Bezpieczeństwo

Korzystając z funkcji API zintegrowanego systemu plików, podobnie jak w przypadku interfejsów zarządzania danymi, istnieje możliwość ograniczenia dostępu do obiektów. Trzeba jednak mieć świadomość, że przejmowanie uprawnień nie jest obsługiwane. Funkcja API zintegrowanego systemu plików korzysta z uprawnień pobieranych z profilu użytkownika, który jest właścicielem zadania.

- | Każdy system plików może mieć własne wymagania dotyczące uprawnień. Serwer NFS oraz zadania serwera plików
- | wymagają specjalnego potraktowania. Zadania te zazwyczaj wykonują funkcje na rzecz użytkowników, którzy mogą
- | nie być właścicielami profilu użytkownika dla zadania. Żądania serwera NFS są uruchamiane w profilu użytkownika,
- | którego identyfikator użytkownika (UID) został otrzymany przez serwer NFS w czasie żądania. Inne zadania serwera
- | plików realizują żądania dla użytkowników, którzy nie są podłączeni do serwera.

Uprawnienia w systemie są odpowiednikiem *praw* w systemach UNIX. Prawa dostępu mogą dotyczyć odczytu i zapisu (dla pliku lub katalogu), wykonywania (dla pliku) i przeszukiwania (dla katalogu). Prawa dostępu określa zestaw bitów praw dostępu składających się na tryb dostępu do pliku lub katalogu. Bity uprawnień można zmienić używając funkcji zmiany trybu `chmod()` lub `fchmod()`. Do sterowania ustawieniami bitów praw dostępu do plików tworzonych przez dane zadanie można również wykorzystywać funkcję `umask()`.

#### **Informacje pokrewne**

`chmod()`--funkcja API Change File Authorizations

`fchmod()`--funkcja API Change File Authorizations by Descriptor

`umask()`--funkcja API Set Authorization Mask for Job

Funkcje API zintegrowanego systemu plików

Informacje o bezpieczeństwie

## **Obsługa gniazd**

Jeśli aplikacja korzysta z systemu plików “root” (/), QOpenSys lub UDFS, można wykorzystać obsługę gniazd lokalnych *local sockets* zintegrowanego systemu plików. Obiekt typu gniazdo lokalne (typ obiektu \*SOCKET) umożliwia nawiązanie komunikacji między dwoma zadaniami działającymi w tym samym systemie.

Jedno z zadań ustanawia punkt połączeniowy tworząc gniazdo lokalne za pomocą funkcji `bind()` w języku C. Drugie zadanie podaje nazwę gniazda lokalnego w funkcji `connect()`, `sendto()` lub `sendmsg()`.

Po nawiązaniu połączenia, zadania mogą wzajemnie wysłać i odebrać dane używając funkcji zintegrowanego systemu plików, takich jak `write()` i `read()`. Żadne z przesyłanych danych nie przechodzą przez gniazdo lokalne. Gniazdo jest jedynie miejscem spotkania, w którym jedno zadanie może znaleźć drugie.

Po zakończeniu komunikacji między zadaniami każde z nich używa funkcji `close()` w celu zamknięcia połączenia przez gniazdo. Lokalny obiekt gniazda pozostaje w systemie dopóki nie zostanie usunięty za pomocą funkcji `unlink()` lub komendy Usuwanie dowiązania (Remove Link - RMVLNK).

Gniazda lokalnego nie można zeszkładować.

#### **Informacje pokrewne**

Programowanie z użyciem gniazd

`write()`--funkcja API Write to Descriptor

`read()`--funkcja API Read from Descriptor

`close()`--funkcja API Close File or Socket Descriptor

`unlink()`--funkcja API Remove Link to File

Komenda Usuwanie dowiązania (Remove link - RMVLNK)

## **Nazewnictwo i obsługa różnych języków**

Obsługa systemów plików “root” (/) i QOpenSys zapewnia niezmiennosc znaków w nazwach obiektów przy przechodzeniu między schematami kodowania używanymi w różnych językach narodowych i w różnych urządzeniach.

- | Podczas przekazywania nazwy obiektu do systemu każdy jej znak jest poddawany konwersji do postaci 16-bitowej, w
- | której wszystkie znaki mają standardowo kodowaną reprezentację. W chwili podania nazwy jako danych wejściowych
- | interfejsu zintegrowanego system plików jest ona konwertowana ze strony kodowej używanej przez wywołującego na

l odpowiednie kodowanie docelowe. Jeśli docelowa strona kodowa, na którą konwertowana jest nazwa, nie zawiera  
l jednego ze znaków użytych w nazwie wejściowej, może zostać zgłoszony błąd lub w zwracanych danych mogą się  
l znaleźć znaki niemożliwe do wyświetlenia.

Ponieważ znaki pozostają niezmiennione w różnych stronach kodowych, wykonując operacje nie powinno się zakładać, że dany znak zostanie zamieniony na inny określony znak w wyniku zastosowania konkretnej strony kodowej. Na przykład nie powinno się przyjmować, że znak krzyżyka zostanie zamieniony na znak funta szterlinga, nawet jeśli mają one tę samą reprezentację kodową w różnych stronach kodowych.

Należy pamiętać, że atrybuty rozszerzone obiektu poddawane są konwersji w ten sam sposób, co nazwa obiektu; stosują się więc do nich te same uwagi.

#### **Pojęcia pokrewne**

“Niezmienność nazw” na stronie 16

Użytkownicy systemu plików “root” (/), systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmiennosc znaków w nazwach obiektów.

“Przegląd automatycznej konwersji nazw” na stronie 90

Niektóre systemy plików, które nie rozróżniają wielkości liter, na przykład system “root” (/) i UDFS utworzone przy użyciu funkcji CASE(\*MONO), obsługują nazwy zapisane w standardzie Unicode 4.0. System wykonuje automatyczną konwersję nazw, aby obsłużyć dodatkowe znaki w tych nazwach.

## **Konwersja danych**

W trakcie dostępu do plików poprzez zintegrowany system plików zawarte w tych plikach dane mogą zostać przekształcone lub nie, w zależności od trybu otwarcia żadanego w momencie otwierania tego pliku.

Otwarty plik może znajdować się w jednym z dwóch trybów otwarcia:

#### **Binarnym (Binary)**

Dane są odczytywane z pliku i zapisywane do niego bez konwersji. Za operowanie danymi odpowiada aplikacja.

#### **Tekstowym (Text)**

Dane są odczytywane z pliku i zapisywane do niego przy założeniu, że mają one postać tekstową. Podczas odczytu danych z pliku dokonywana jest ich konwersja z identyfikatora kodowanego zestawu znaków (CCSID) pliku na identyfikator CCSID aplikacji, zadania lub systemu odbierającego dane. Przy zapisie danych do pliku są one poddawane konwersji z identyfikatora CCSID aplikacji, zadania lub systemu na identyfikator CCSID pliku. Dla rzeczywistych plików strumieniowych wszystkie znaki formatowania wiersza (takie jak znak powrotu karetki, znak tabulacji i znak końca pliku) są po prostu poddawane konwersji z jednego identyfikatora CCSID na inny.

Podczas odczytu z plików o strukturze opartej na rekordach, które są traktowane jako pliki strumieniowe, na końcu danych z każdego rekordu dołączane są znaki końca wiersza (znak końca wiersza i znak nowego wiersza). Przy zapisie do pliku o strukturze opartej na rekordach:

- znaki końca wiersza są usuwane,
- znaki tabulacji zastępowane są odpowiednią liczbą spacji do następnej pozycji tabulacji,
- wiersze uzupełniane są spacjami (w przypadku podzbiorów fizycznych zbiorów źródłowych) lub znakami pustymi (w przypadku podzbiorów fizycznych zbiorów danych) do końca rekordu.

W żądaniu otwarcia można wpisać jedną z opcji:

#### **Binarny, wymuszony (Binary, Forced)**

Dane są przetwarzane jako binarne, niezależnie od faktycznej zawartości pliku. Za poprawne operowanie danymi odpowiada aplikacja.

#### **Tekstowy, wymuszony (Text, Forced)**

Zakłada się, że dane są tekstem. Dane są poddawane konwersji z identyfikatora CCSID pliku na identyfikator CCSID aplikacji.

Domyślna opcja *Binarne*, *wymuszone* użyta jest dla funkcji `open()`.

### Informacje pokrewne

`open()`--funkcja API Open File

## Przykład: funkcje C zintegrowanego systemu plików

Kolejne strony zawierają prosty program napisany w języku C, przedstawiający użycie pewnych funkcji zintegrowanego systemu plików.

Program wykonuje następujące operacje:

- 1 Używa funkcji `getuid()` do określenia rzeczywistego identyfikatora użytkownika (`uid`).
- 2 Używa funkcji `getcwd()` do określenia bieżącego katalogu.
- 3 Używa funkcji `open()` do utworzenia pliku. Nadaje ona właścicielowi (osobie, która utworzyła zbiór) uprawnienia do zapisu, odczytu i wykonywania pliku.
- 4 Używa funkcji `write()` do zapisu ciągu bajtów do pliku. Plik identyfikuje deskryptor pliku, który został dostarczony podczas operacji otwarcia (3).
- 5 Używa funkcji `close()` do zamknięcia pliku.
- 6 Używa funkcji `mkdir()` do utworzenia nowego podkatalogu w katalogu bieżącym. Właściciel ma prawo do czytania, zapisu i wykonywania tego podkatalogu.
- 7 Używa funkcji `chdir()` do zmiany podkatalogu bieżącego na nowy podkatalog.
- 8 Używa funkcji `link()`, aby utworzyć dowiązanie do poprzednio utworzonego pliku (3).
- 9 Używa funkcji `open()` do otwarcia pliku tylko do odczytu. Dowiązanie, które zostało utworzone w (8), umożliwia dostęp do danego pliku.
- 10 Używa funkcji `read()` do odczytu ciągu bajtów z pliku. Plik identyfikuje deskryptor pliku, który został dostarczony podczas operacji otwarcia (9).
- 11 Używa funkcji `close()` do zamknięcia pliku.
- 12 Używa funkcji `unlink()`, aby usunąć dowiązanie do pliku.
- 13 Używa funkcji `chdir()`, aby zmienić bieżący katalog z powrotem na katalog nadrzędny, w którym został utworzony nowy podkatalog.
- 14 Używa funkcji `rmdir()`, aby usunąć poprzednio utworzony podkatalog (6).
- 15 Używa funkcji `unlink()`, aby usunąć poprzednio utworzony plik (3).

**Uwaga:** Ten przykładowy program będzie funkcjonował poprawnie w systemach, w których identyfikator CCSID zadania, w którym został uruchomiony, wynosi 37. Funkcje API zintegrowanego systemu plików muszą mieć nazwy obiektów i ścieżek zakodowane w identyfikatorze CCSID zadania; jednak kompilator języka C przechowuje stałe znakowe w identyfikatorze CCSID 37. Dla pełnej zgodności, przed przekazaniem funkcji API do identyfikatora CCSID zadania, należy przetłumaczyć stałe znakowe, takie jak nazwy obiektów i ścieżek.

**Uwaga:** Korzystając z przykładów kodu, użytkownik wyraża zgodę na warunki zapisane w sekcji “Licencja na kod oraz Informacje dotyczące kodu” na stronie 136.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
```

```
#define BUFFER_SIZE      2048
```

```

#define NEW_DIRECTORY    "testdir"
#define TEST_FILE       "test.file"
#define TEST_DATA       "Hello World!"
#define TEST_DATA       "Hello World!"
#define USER_ID         "user_id_"
#define PARENT_DIRECTORY ".."

char InitialFile[BUFFER_SIZE];
char LinkName[BUFFER_SIZE];
char InitialDirectory[BUFFER_SIZE] = ".";
char Buffer[32];
int FilDes = -1;
int BytesRead;
int BytesWritten;
uid_t UserID;

void CleanUpOnError(int level)
{
    printf("Napotkano błąd, trwa sprzątanie.\n");
    switch ( level )
    {
        case 1:
            printf("Nie można pobrać bieżącego katalogu roboczego.\n");
            break;
        case 2:
            printf("Nie można utworzyć pliku %s.\n",TEST_FILE);
            break;
        case 3:
            printf("Nie można pisać do pliku %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 4:
            printf("Nie można zamknąć pliku %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 5:
            printf("Nie można utworzyć katalogu %s.\n",NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 6:
            printf("Nie można zmienić na katalog %s.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 7:
            printf("Nie można utworzyć dowiązania %s to %s.\n",LinkName,InitialFile);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 8:
            printf("Nie można utworzyć dowiązania %s.\n",LinkName);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 9:
            printf("Nie można odczytać dowiązania %s.\n",LinkName);
            close(FilDes);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
    }
}

```

```

        unlink(TEST_FILE);
        break;
    case 10:
        printf("Nie można zamknąć dowiązania %s.\n",LinkName);
        close(FilDes);
        unlink(TEST_FILE);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 11:
        printf("Nie można zamknąć dowiązania %s.\n",LinkName);
        unlink(TEST_FILE);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 12:
        printf("Nie można zmienić katalogu %s.\n",PARENT_DIRECTORY);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 13:
        printf("Nie można usunąć katalogu %s.\n",NEW_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 14:
        printf("Nie można usunąć dowiązania pliku %s.\n",TEST_FILE);
        unlink(TEST_FILE);
        break;
    default:
        break;
}
printf("Program zakończony błędem.\n\
      "Nie wszystkie testowe pliki i katalogi mogły zostać usunięte.\n");
}

int main ()
{
    1
    /* Pobranie i wydruk rzeczywistego id użytkownika przy wykorzystaniu funkcji getuid(). */
    UserID = getuid();
    printf("Rzeczywistym identyfikatorem użytkownika jest %u. \n",UserID);

    2
    /* Pobranie bieżącego katalogu roboczego i zachowanie go w InitialDirectory. */
    if ( NULL == getcwd(InitialDirectory,BUFFER_SIZE) )
    {
        perror("Błąd funkcji getcwd");
        CleanupOnError(1);
        return 0;
    }
    printf("Bieżącym katalogiem roboczym jest %s. \n",InitialDirectory);

    3
    /* Utworzenie pliku TEST_FILE do zapisu, jeśli taki nie istnieje.
       Nadanie właścicielowi uprawnień do odczytu, zapisu i wykonywania. */
    FilDes = open(TEST_FILE, O_WRONLY | O_CREAT | O_EXCL, S_IRWXU);
    if ( -1 == FilDes )
    {
        perror("Błąd funkcji open");
        CleanupOnError(2);
        return 0;
    }
    printf("Utworzono %s w katalogu %s.\n",TEST_FILE,InitialDirectory);
}

```



```

4
/* Wpisz TEST_DATA do TEST_FILE przez FilDes */
BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
if ( -1 == BytesWritten )
{
    perror("Błąd funkcji write");
    CleanupOnError(3);
    return 0;
}
printf("Wpisano %s do pliku %s.\n",TEST_DATA,TEST_FILE);

5
/* Zamknięcie TEST_FILE przez FilDes */
if ( -1 == close(FilDes) )
{
    perror("Błąd funkcji close");
    CleanupOnError(4);
    return 0;
}
FilDes = -1;
printf("Zamknięcie pliku %s.\n",TEST_FILE);

6
/* Utworzenie nowego katalogu w bieżącym katalogu roboczym
i nadanie właścicielowi uprawnień do odczytu, zapisu i wykonania */
if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )
{
    perror("Błąd funkcji mkdir");
    CleanupOnError(5);
    return 0;
}
printf("Utworzono katalog %s w katalogu %s.\n",NEW_DIRECTORY,InitialDirectory);

7
/* Zmiana bieżącego katalogu roboczego na
właśnie utworzony katalog NEW_DIRECTORY. */
if ( -1 == chdir(NEW_DIRECTORY) )
{
    perror("Błąd funkcji chdir");
    CleanupOnError(6);
    return 0;
}
printf("Zmiana na katalog %s/%s.\n",InitialDirectory,NEW_DIRECTORY);

/* Skopiowanie PARENT_DIRECTORY do InitialFile i
dodanie "/" i TEST_FILE do InitialFile. */
strcpy(InitialFile,PARENT_DIRECTORY);
strcat(InitialFile,"/");
strcat(InitialFile,TEST_FILE);

/* Skopiowanie Copy USER_ID do LinkName, a następnie dołączenie
UserID jako łańcucha do LinkName. */
strcpy(LinkName, USER_ID);
sprintf(Buffer, "%d\0", (int)UserID);
strcat(LinkName, Buffer);

8
/* Utworzenie dowiązania do nazwy InitialFile z LinkName. */
if ( -1 == link(InitialFile,LinkName) )
{
    perror("Błąd funkcji link");
    CleanupOnError(7);
    return 0;
}
printf("Utworzono dowiązanie %s do %s.\n",LinkName,InitialFile);

```

```

9
/* Otwarcie pliku LinkName tylko do odczytu. */
if ( -1 == (FilDes = open(LinkName, O_RDONLY)) )
{
    perror("Błąd funkcji open");
    CleanupOnError(8);
    return 0;
}
printf("Otwarto %s do odczytu.\n", LinkName);

10
/* Czytanie z pliku LinkName poprzez FilDes do Buffer. */
BytesRead = read(FilDes, Buffer, sizeof(Buffer));
if ( -1 == BytesRead )
{
    perror("Błąd funkcji read");
    CleanupOnError(9);
    return 0;
}
printf("Odczyt %s z %s.\n", Buffer, LinkName);
if ( BytesRead != BytesWritten )
{
    printf("OSTRZEŻENIE: liczba odczytanych bajtów "\
        "jest różna od liczby wpisanych bajtów.\n");
}

11
/* Zamknięcie pliku LinkName poprzez FilDes. */
if ( -1 == close(FilDes) )
{
    perror("Błąd funkcji close");
    CleanupOnError(10);
    return 0;
}
FilDes = -1;
printf("Zamknięto %s.\n", LinkName);

12
/* Odłączenie dowiązania LinkName do InitialFile. */
if ( -1 == unlink(LinkName) )
{
    perror("Błąd funkcji unlink");
    CleanupOnError(11);
    return 0;
}
printf("Dowiązanie %s zostało usunięte.\n", LinkName);

13
/* Zmiana bieżącego katalogu roboczego
z powrotem na katalog początkowy. */
if ( -1 == chdir(PARENT_DIRECTORY) )
{
    perror("Błąd funkcji chdir");
    CleanupOnError(12);
    return 0;
}
printf("zmiana katalogu na %s.\n", InitialDirectory);

14
/* Usunięcie katalogu NEW_DIRECTORY */
if ( -1 == rmdir(NEW_DIRECTORY) )
{
    perror("Błąd funkcji rmdir");
    CleanupOnError(13);
    return 0;
}
printf("Usunięto katalog %s.\n", NEW_DIRECTORY);

```

```

15
/* Odłączenie pliku TEST_FILE */
if ( -1 == unlink(TEST_FILE) )
{
    perror ("Błąd funkcji unlink");
    CleanupOnError(14);
    return 0;
}
printf("Odłączono plik %s.\n",TEST_FILE);

printf("Program został zakończony poprawnie.\n");
return 0;
}

```

---

## Praca z plikami i folderami w programie System i Navigator

W sekcji opisano dostępne operacje na plikach i folderach.

### Tworzenie folderu

Aby utworzyć folder, wykonaj następujące czynności.

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym klawiszem myszy system plików, do którego chcesz dodać folder, i wybierz **Nowy folder**.
3. W oknie dialogowym **Nowy folder** wpisz nazwę nowego folderu i podaj jego atrybuty.
  - W przypadku folderów utworzonych w systemach plików "root" (/), QOpenSys lub UDFS można podać wartości następujących pól: **Ograniczenie zmiany nazw i usuwania dowiązań** (Restrict rename and unlink), **Kontrola obiektów utworzonych w folderze** (Audit objects created in folder) i **Skanowanie obiektów utworzonych w folderze** (Scan objects created in folder).
  - W przypadku folderów utworzonych w innych systemach plików można podać wartość pola **Kontrola obiektów utworzonych w folderze** (Audit objects created in folder).
4. Kliknij przycisk **OK**.

Gdy na platformie System i tworzony jest folder, należy zdecydować, czy nowy folder (lub obiekt) ma być chroniony przez kronikowanie. Należy także rozważyć, czy obiekty utworzone w tym folderze mają być skanowane, czy nie.

#### Zadania pokrewne

"Określanie, czy obiekty mają być skanowane" na stronie 129

Określanie obiektów, które mają być skanowane, jest możliwe w systemach plików "root" (/), QOpenSys i użytkownika. Wykonaj następujące czynności, aby ustawić opcje skanowania.

#### Informacje pokrewne

Zarządzanie kronikami

### Usuwanie pliku lub folderu

Aby usunąć plik lub folder, wykonaj następujące czynności.

1. W oknie programu System i Navigator rozwiń **Moje połączenia** (**My Connections**) → *system* → **Systemy plików** (**File Systems**) → **Zintegrowany system plików** (**Integrated File System**). Rozwijaj kolejne gałęzie do momentu wyświetlenia obiektu, który chcesz usunąć.
  2. Kliknij prawym klawiszem myszy plik lub folder, a następnie wybierz opcję **Usuń**. Zostanie wyświetlone okno potwierdzenia, przedstawiające wykaz wszystkich elementów wybranych do usunięcia. Jeśli nie chcesz usuwać określonego obiektu, anuluj jego zaznaczenie. W panelu potwierdzenia kliknij przycisk **Usuń** (Delete).
- Uwaga:** Usunięcie folderu powoduje również usunięcie wszystkich obiektów w nim zawartych.

## Przenoszenie plików lub folderów do innego systemu plików

Każdy system plików ma swoją własną charakterystykę. Jednak przeniesienie obiektów do innego systemu plików może oznaczać utratę korzyści płynących z systemu plików, w którym obiekt jest aktualnie przechowywany. Można jednak tak przenosić obiekty z jednego systemu plików do innego, aby wykorzystywać zalety jego charakterystyki.

Przed przeniesieniem obiektów do innego systemu plików należy zatem zapoznać się z systemami plików w zintegrowanym systemie plików i ich charakterystykami.

Należy także rozważyć następujące kwestie:

- Czy używa się aplikacji wykorzystujących zalety systemu plików, w którym obecnie znajdują się obiekty?

Niektóre systemy plików obsługują interfejsy, których nie obsługuje zintegrowany system plików. Aplikacje używające tych interfejsów mogą nie mieć dostępu do obiektów przeniesionych do innego systemu. Na przykład systemy plików QDLS i QOPT obsługują hierarchiczny system plików (HFS). Nie można używać tych interfejsów w przypadku obiektów znajdujących się w innych systemach plików.

- Jakie cechy obiektów są ważne?

Nie wszystkie cechy są obsługiwane przez każdy system plików. Na przykład system plików QSYS.LIB lub QSYS.LIB niezależnej puli ASP obsługuje zapisywanie i odczytywanie tylko kilku atrybutów rozszerzonych, a systemy plików "root" (/) i QOpenSys obsługują zapisywanie i odczytywanie wszystkich atrybutów rozszerzonych. Dlatego systemów plików QSYS.LIB i QSYS.LIB niezależnej puli ASP nie należy używać do przechowywania obiektów posiadających atrybuty rozszerzone.

Dobrym rozwiązaniem jest przeniesienie plików PC znajdujących się w systemie QDLS. Większość aplikacji komputera PC powinna bez zakłóceń kontynuować pracę z plikami komputera PC przeniesionymi z systemu QDLS do innych systemów plików. Do przechowywania tych plików PC dobrze nadają się systemy plików: bazowy ("root", /), QOpenSys i QNTC. Ponieważ te systemy plików obsługują wiele charakterystyk systemu plików OS/2, mogą zapewnić szybszy dostęp do plików.

- | W programie System i Navigator można przenieść plik lub folder do innego systemu plików, przeciągając obiekt i upuszczając w nowej lokalizacji. Można też tego dokonać przy użyciu funkcji Kopiuj i wklej lub Wytnij i wklej.

Aby przenieść obiekty do innego systemu plików za pomocą komend języka CL, wykonaj następujące czynności:

1. Utwórz kopie wszystkich obiektów przeznaczonych do przeniesienia.

Posiadanie kopii zapasowej umożliwia odtworzenie obiektów w pierwotnym systemie plików, jeśli aplikacje nie będą miały dostępu do obiektów w systemie plików, do którego przeniesiono obiekty.

**Uwaga:** Nie można zeszkładować obiektów w jednym systemie plików i odtworzyć ich w innym.

2. Użyj komendy Utworzenie katalogu (Create Directory - CRTDIR), aby utworzyć katalogi w systemie plików, do którego przenosisz obiekty.

Należy uważnie przeanalizować atrybuty katalogu, w którym obecnie znajdują się obiekty, aby określić, czy mają zostać powielone w tworzonych katalogach. Na przykład, czy właścicielem obiektu zostaje użytkownik, który go tworzy, a nie użytkownik, który jest właścicielem starego katalogu. Jeśli system plików obsługuje określanie własności katalogu, po utworzeniu katalogu można zmienić jego właściciela.

3. Użyj komendy Przeniesienie obiektu (Move Object - MOV), aby przenieść pliki do wybranego systemu plików.

Komenda MOV jest zalecana, ponieważ zachowuje właścicieli obiektów, jeśli system plików obsługuje określanie własności obiektów. Zachowanie własności obiektów jest możliwe również za pomocą komendy Kopiowanie obiektu (Copy Object - CPY) z parametrem OWNER(\*KEEP). Metoda ta działa jednak tylko w systemach plików, które umożliwiają przydzielenie praw własności do obiektów. Należy pamiętać, że użycie komendy MOV lub CPY może spowodować:

- niezgodność atrybutów i ich odrzucenie,
- odrzucenie atrybutów rozszerzonych,
- odrzucenie uprawnień, które nie mają odpowiedników.

Oznacza to, że jeśli chce się ponownie przenieść obiekty do systemu plików, z którego zostały skopiowane, to nie można ich po prostu przenieść lub skopiować z powrotem, ponieważ część atrybutów mogła zostać utracona. Najbezpieczniej sposobem przywrócenia obiektu jest odtworzenie jego zeszkładowanej wersji.

### Pojęcia pokrewne

“Systemy plików” na stronie 23

*System plików* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są w systemie pliki, katalogi, biblioteki i obiekty.

### Odsyłacze pokrewne

“Porównanie systemów plików” na stronie 24

Poniższe tabele zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

### Informacje pokrewne

Komenda Utworzenie katalogu (Create Directory - CRTDIR)

Komenda Przeniesienie obiektu (Move Object - MOV)

Komenda Kopiowanie obiektu (Copy Object - COPY)

## Ustawianie uprawnień

Nadawanie uprawnień do obiektów pozwala na kontrolę manipulacji tymi obiektami przez innych użytkowników. Dzięki otrzymanym uprawnieniom niektórzy użytkownicy będą mogli jedynie wyświetlać obiekty, podczas gdy inni będą mogli poddawać je edycji.

Aby ustawić uprawnienia dla pliku lub folderu, wykonaj następujące czynności:

1. W oknie programu System i Navigator rozwiń **Moje połączenia (My Connections)** → *system* → **Systemy plików (File Systems)** → **Zintegrowany system plików (Integrated File System)**. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony obiekt, do którego chcesz nadać uprawnienia.
2. Kliknij obiekt prawym klawiszem myszy i wybierz **Uprawnienia**.
3. Kliknij przycisk **Dodaj** w oknie dialogowym **Uprawnienia**.
4. W oknie dialogowym **Dodaj** wybierz przynajmniej jednego użytkownika lub grupę albo wpisz odpowiednią nazwę w polu nazwy użytkownika lub grupy.
5. Kliknij przycisk **OK**. Spowoduje to dodanie użytkowników lub grup i umieszczenie ich na początku listy.
6. Kliknij przycisk **Szczegóły** (Details) aby wprowadzić szczegółowe uprawnienia.
7. Zaznaczenie odpowiedniego pola wyboru powoduje nadanie wymaganych uprawnień użytkownikowi.
8. Kliknij przycisk **OK**.

## Konfigurowanie konwersji plików tekstowych

W programie System i Navigator można ustawić automatyczną konwersję pliku tekstowego. Automatyczna konwersja pliku tekstowego umożliwia użycie rozszerzeń plików w przypadku konwersji plików z danymi.

Zintegrowany system plików umożliwia automatyczną konwersję danych w pliku podczas przesyłania plików między platformą System i a komputerem PC. W przypadku dostępu do pliku z komputera PC, jest on obsługiwany jak plik ASCII.

Aby skonfigurować konwersję tekstu pliku, wykonaj następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia**(My Connections) *system* **Systemy plików** (File Systems).
2. Kliknij prawym klawiszem myszy **Zintegrowany system plików** i wybierz **Właściwości**.
3. W pole tekstowe **Rozszerzenia plików dla automatycznej konwersji pliku tekstowego** wpisz rozszerzenie nazw plików, które mają być automatycznie poddane konwersji i kliknij **Dodaj**.
4. Powtórz czynność 3 dla wszystkich rozszerzeń nazw plików, które chcesz automatycznie poddać konwersji.
5. Kliknij przycisk **OK**.

## Wysyłanie pliku lub folderu do innego systemu

Aby wysłać plik lub folder do innego systemu, wykonaj następujące czynności.

1. W oknie programu System i Navigator rozwiń **Moje połączenia (My Connections)** → *system* → **Systemy plików (File Systems)** → **Zintegrowany system plików (Integrated File System)**. Rozwijaj kolejne gałęzie do momentu wyświetlenia obiektu, który chcesz wysłać.
2. Kliknij plik lub folder prawym klawiszem myszy i wybierz polecenie **Wyślij**. Plik lub folder zostanie wyświetlony na liście Wybrane pliki i foldery w oknie dialogowym Wysyłanie plików z.
3. Rozwiń listę dostępnych systemów i grup.
4. Wybierz system i kliknij **Dodaj**, aby dodać go do listy **Systemy i grupy docelowe**. Powtórz tę czynność dla wszystkich systemów, do których chcesz wysłać wybrany plik lub folder.
5. Kliknij przycisk **OK**, by wysłać plik lub folder.

### Zadania pokrewne

“Zmiana opcji dla wysyłania pliku lub folderu”

Przy wysyłaniu pliku lub folderu do innego systemu można określić, czy podfoldery mają również być wysyłane i czy zastępować istniejący plik. Można też określić porę wysłania pliku lub folderu przez system. Aby zmienić opcje dla wysyłania pliku, wykonaj następujące czynności:

## Zmiana opcji dla wysyłania pliku lub folderu

Przy wysyłaniu pliku lub folderu do innego systemu można określić, czy podfoldery mają również być wysyłane i czy zastępować istniejący plik. Można też określić porę wysłania pliku lub folderu przez system. Aby zmienić opcje dla wysyłania pliku, wykonaj następujące czynności:

1. Wykonaj kroki opisane w temacie “Wysyłanie pliku lub folderu do innego systemu”.
2. Kliknij zakładkę **Opcje**. Domyślnie włączone są opcje dołączania podfolderów podczas pakowania i wysyłania plików oraz zastępowania istniejących plików plikami wysyłanymi.
3. Zmień te opcje zgodnie z wymaganiami.
4. Aby ustawić opcje składowania i odtwarzania, kliknij przycisk **Zaawansowane**.
5. Kliknij **OK**, aby zapisać opcje zaawansowane.
6. Kliknij przycisk **OK**, aby wysłać plik lub folder, lub kliknij przycisk **Harmonogram (Schedule)**, aby ustalić czas wysyłania pliku lub folderu.
7. Określ porę wysłania pliku lub folderu. Funkcja harmonogramu daje możliwość wykonania pracy w najbardziej dogodnym dla użytkownika czasie.

## Tworzenie współużytkowanego zasobu plikowego

*Współużytkowany zasób plikowy* jest ścieżką do katalogu, który i5/OS NetServer udostępnia klientom PC w sieci System i. Zasób taki może wskazywać dowolny katalog zintegrowanego systemu plików na platformie System i.

Aby utworzyć współużytkowany zasób plikowy, wykonaj następujące czynności.

1. W programie System i Navigator rozwiń **Moje połączenia** → **(My Connections) system** → **Systemy plików (File Systems)** **Zintegrowany system plików (Integrated File System)**.
2. Rozwiń system plików zawierający folder, dla którego chcesz utworzyć współużytkowany zasób plikowy.
3. Prawym klawiszem myszy kliknij folder, dla którego chcesz utworzyć współużytkowany zasób plikowy, i wybierz **Współużytkowanie**.
4. Wybierz **Nowy zasób współużytkowany**.
5. W oknie dialogowym Współużytkowany zasób plikowy (File Share) podaj atrybuty nowego współużytkowanego zasobu plikowego i kliknij przycisk **OK**.

## Zmiana współużytkowanego zasobu plikowego

*Współużytkowany zasób plikowy* jest ścieżką do katalogu, który i5/OS NetServer udostępnia klientom PC w sieci System i. Zasób taki może wskazywać dowolny katalog zintegrowanego systemu plików na platformie System i.

Aby zmienić współużytkowany zasób plikowy, wykonaj następujące czynności.

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems)**Zintegrowany system plików** (Integrated File System).
2. Rozwiń folder ze zdefiniowanym zasobem współużytkowanym, który chcesz zmienić.
3. Kliknij prawym przyciskiem myszy folder, który zawiera współużytkowany zasób plikowy do zmiany, a następnie wybierz opcję **Współużytkowanie** (Sharing).
4. Wybierz nazwę współużytkowanego zasobu plikowego, który ma zostać zmieniony.
5. W oknie dialogowym *Współużytkowany zasób plikowy* (File Share) zmień atrybuty współużytkowanego zasobu plikowego i kliknij przycisk **OK**, aby potwierdzić zmiany.

## Usuwanie współużytkowanego zasobu plikowego

*Współużytkowany zasób plikowy* jest ścieżką do katalogu, którą i5/OS NetServer udostępnia klientom PC w sieci System i. Zasób taki może wskazywać dowolny katalog zintegrowanego systemu plików na platformie System i. Można zatrzymać udostępnianie istniejącego współużytkowanego zasobu plikowego przy użyciu programu System i Navigator.

Aby usunąć współużytkowany zasób plikowy, wykonaj następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems)**Zintegrowany system plików** (Integrated File System).
2. Rozwiń system plików zawierający współużytkowany zasób plikowy, który ma nie być już udostępniany.
3. Kliknij prawym przyciskiem myszy współużytkowany katalog, który ma nie być już udostępniany, a następnie wybierz kolejno opcje **Współużytkowanie**(Sharing) **Zakończ współużytkowanie** (Stop Sharing).
4. W wyświetlonym oknie **Zakończenie współużytkowania** (Stop sharing) kliknij przycisk **OK**.

## Tworzenie nowego systemu plików użytkownika

System plików użytkownika (UDFS) jest systemem, dla którego należy utworzyć oraz zdefiniować atrybuty. Systemy UDFS znajdują się w pulach ASP i niezależnych pulach ASP systemu.

Aby utworzyć nowy system plików użytkownika (UDFS), wykonaj następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia**(My Connections) *system* **Systemy plików** (File Systems) **Zintegrowany system plików** (Integrated File System) **Rootdev**.
2. Kliknij ikonę puli ASP, w której chcesz umieścić nowy system plików użytkownika (UDFS).
3. Z menu folderu wybierz opcję **Nowy system plików użytkownika (UDFS)**.
4. W oknie dialogowym **Nowy system plików UDFS** (New User-Defined File System) określ nazwę systemu UDFS, jego opis (opcjonalnie), wartości kontroli, domyślny format plików, domyślny atrybut przeglądania, domyślny sposób przydzielania przestrzeni dyskowej, domyślny sposób przydzielania pamięci oraz podaj, czy w nowym systemie plików UDFS ma być rozróżniana wielkość liter w nazwach plików.

**Uwaga:** Opcje **Domyślny sposób przydzielania przestrzeni dyskowej** i **Domyślny sposób przydzielania pamięci** są dostępne tylko w wersji V6R1 lub nowszej systemu.

## Podłączanie systemu plików utworzonego przez użytkownika

Dostęp do danych przechowywanych w systemie plików UDFS jest możliwy po podłączeniu go po ładowaniu IPL.

Po podłączeniu systemu plików UDFS następuje przesłonięcie systemów plików, katalogów i obiektów znajdującym się pod punktem podłączenia w hierarchii folderów. Wszystkie te systemy plików, katalogi lub obiekty będą

niewidoczne aż do odłączenia UDFS. Aby zapewnić dostęp do wszystkich danych zintegrowanego systemu plików, należy podłączyć UDFS do pustego folderu. Po podłączeniu UDFS wszystkie pliki zawarte w nim będą dostępne w tym folderze. Wszelki zmiany dokonane w folderze spowodują zmianę w systemie UDFS, a nie w przysłoniętym folderze.

**Uwaga:** System UDFS znajdujący się w niezależnej puli ASP nie może być podłączony.

Aby podłączyć system plików zdefiniowany przez użytkownika (UDFS), należy wykonać następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia**(My Connections) **system Systemy plików** (File Systems) **Zintegrowany system plików** (Integrated File System) **Rootdev**.
2. Kliknij pulę pamięci dyskowej (ASP) zawierającą system plików UDFS, który ma zostać podłączony.
3. W kolumnie **Nazwa UDFS** (UDFS Name) prawego panelu programu System i Navigator kliknij prawym przyciskiem myszy system UDFS, który ma zostać podłączony.
4. Wybierz polecenie **Podłącz**.
5. W wyświetlonym oknie dialogowym Podłączanie systemu UDFS (Mount UDFS) podaj ścieżkę katalogu podłączenia, typ dostępu (tylko do odczytu, odczyt i zapis) oraz określ, czy zezwolić na ustawienia identyfikatorów użytkownika i grupy. Następnie kliknij **OK**.

Jeśli używana jest technika przeciągnij i upuść, można podłączyć UDFS, przeciągając go do folderu zintegrowanego systemu plików w tym samym systemie. Nie można jednak przeciągnąć systemu UDFS do folderów /dev, /dev/QASPxx, /dev/asp\_name, do innego systemu ani na pulpit.

## Odłączanie systemu plików użytkownika

Po podłączeniu systemu plików UDFS następuje przesłonięcie systemów plików, katalogów i obiektów znajdującym się pod punktem podłączenia w hierarchii folderów. Wszystkie te systemy plików, katalogi lub obiekty będą niewidoczne aż do odłączenia UDFS.

Aby odłączyć system plików zdefiniowany przez użytkownika (UDFS), należy wykonać następujące czynności:

1. W oknie programu System i Navigator rozwiń **Moje połączenia (My Connections)** → **system** → **Systemy plików (File Systems)** → **Zintegrowany system plików (Integrated File System)** → **Root** → **dev**.
2. Kliknij pulę pamięci dyskowej (ASP) zawierającą system plików UDFS, który ma zostać odłączony.
3. W kolumnie **Nazwa** (Name) w prawym panelu programu System i Navigator kliknij prawym przyciskiem myszy system plików UDFS, który chcesz odłączyć.
4. Wybierz polecenie **Odłącz**. Zostanie wyświetlone okno potwierdzenia odłączenia, pokazujące wybrane do odłączenia systemy plików UDFS.
5. Anuluj zaznaczenia wszelkich systemów plików UDFS, których nie chcesz odłączać. Następnie kliknij przycisk **Odłącz** (Unmount).

## Praca z dynamicznie podłączanymi systemami plików

Można użyć funkcji Informacje o podłączeniu dynamicznym (Dynamic Mount Information), aby sprawdzić, jakie dynamicznie podłączane systemy plików są aktualnie podłączone, wyświetlić ich właściwości lub odłączyć dowolne z nich.

Aby użyć tej funkcji, wykonaj następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia** → **(My Connections) system** → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym przyciskiem myszy opcję **Zintegrowany system plików** (Integrated File System).
3. W menu wywoływanym wybierz opcję **Informacje o podłączeniach dynamicznych** (Dynamic Mount Information).
4. Wyświetlone zostanie okno Informacje o podłączeniach dynamicznych (Dynamic Mount Information), w którym znajdują się wszystkie aktualnie podłączone systemy plików. W oknie wyświetlane są nazwy podłączonych



systemów plików, zdalne systemy, do których podłączone są systemy plików, oraz typ podłączenia. Obsługiwane typy połączeń to System plików użytkownika (UDFS), Sieciowy system plików w wersji 2 (NFSv2), Sieciowy system plików w wersji 3 (NFSv3) i Sieciowy system plików w wersji 4 (NFSv4). Można odłączyć dowolne wyświetlone systemy plików lub wyświetlić właściwości konkretnego systemu plików.

**Uwaga:** System NFSv4 jest dostępny dopiero od wersji V6R1.

- Aby odłączyć system plików, wybierz go z listy, a następnie kliknij przycisk **Odłącz** (Unmount). Wyświetlone zostanie okno z prośbą o potwierdzenie odłączenia (Confirm Unmount). Sprawdź, czy wyświetlony system jest tym, który ma być odłączony. Można usunąć elementy, które nie mają być odłączone. Następnie kliknij przycisk **Odłącz** (Unmount), aby potwierdzić czynność.
- Aby wyświetlić właściwości systemu plików, wybierz go z listy i kliknij przycisk **Właściwości** (Properties). Wyświetlone zostanie okno Właściwości podłączenia (Mount Properties).
  - W przypadku systemu UDFS okno Właściwości podłączenia (Mount Properties) zawiera kartę **Ogólne** (General). Na tej karcie wyświetlane są następujące właściwości: nazwa, ścieżka do której podłączony jest UDFS, typ podłączenia, czas podłączenia, informacja, czy system plików jest systemem tylko do odczytu, a także informacja, czy system plików zezwala na ustawienia użytkowników i grup.
  - W przypadku systemu plików NFS okno Właściwości podłączenia (Mount Properties) zawiera kartę **Ogólne** (General) i kartę **Zaawansowane** (Advanced). Karta **Ogólne** (General) zawiera następujące właściwości: nazwę, nazwa zdalnego serwera, ścieżka, do której podłączony jest NFS, typ podłączenia, czas podłączenia, informacja, czy system plików jest systemem tylko do odczytu oraz informacja, czy system plików zezwala na ustawienia użytkowników i grup. Karta **Zaawansowane** (Advanced) zawiera następujące właściwości: typ podłączenia, wartość limitu czasu, wielkość bufora odczytu, wielkość bufora zapisu, liczba ponownych prób, liczba prób retransmisji, minimalny i maksymalny czas regularnego atrybutu obiektu, minimalny i maksymalny czas atrybutu folderu, informacja, czy wymuszać odświeżenie atrybutów przy otwieraniu oraz czy zezwalać na przechowywanie w pamięci podręcznej atrybutów i nazw.

## Określanie, czy obiekty mają być skanowane

Określanie obiektów, które mają być skanowane, jest możliwe w systemach plików "root" (/), QOpenSys i użytkownika. Wykonaj następujące czynności, aby ustawić opcje skanowania.

1. W oknie programu System i Navigator rozwiń **Moje połączenia (My Connections)** → *system* → **Systemy plików (File Systems)** → **Zintegrowany system plików (Integrated File System)**. Rozwijaj kolejne gałęzie do momentu wyświetlenia odpowiedniego obiektu.
2. Kliknij prawym klawiszem myszy folder lub plik, a następnie wybierz opcję **Właściwości**
3. Kliknij zakładkę **Ochrona**.
4. Wybierz opcję w polu **Skanowanie obiektów**.

Więcej informacji o opcjach znajduje się w poniższych tematach. Opisy tych opcji dotyczą plików. Skanować można tylko pliki. Dla folderów i systemów plików użytkownika można zdefiniować przydzielanie atrybutu skanowania dla plików tworzonych w wybranym folderze lub systemie plików użytkownika.

- Tak  
Obiekt będzie podlegał skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.
- Nie  
Obiekt nie będzie podlegał skanowaniu przez programy wyjściowe dotyczące skanowania.

**Uwaga:** Jeśli opcja Skanuj przy kolejnym dostępie po przywróceniu obiektu jest wybrana w wartościach systemowych, obiekt będzie podlegał skanowaniu co najmniej raz po przywróceniu.

- Tylko po zmianie obiektu

Obiekt będzie podlegał skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania tylko w przypadku modyfikacji od momentu ostatniego skanowania obiektu. Nie będzie podlegał skanowaniu w przypadku aktualizacji oprogramowania do skanowania.

Jeśli wartość systemowa Użyj atrybutu "tylko w przypadku zmiany obiektów" do kontroli skanowania nie jest określona, atrybut tylko w przypadku zmiany obiektów nie będzie używany, a obiekt będzie podlegał skanowaniu po modyfikacji i gdy oprogramowanie do skanowania wskaże aktualizację.

#### Uwagi:

1. W zakładce tej dla plików można określić status skanowania obiektu.
2. Jeśli opcja Skanuj przy kolejnym dostępie po przywróceniu obiektu jest wybrana w wartościach systemowych, obiekt będzie podlegał skanowaniu co najmniej raz po przywróceniu.

## Zwracanie obiektów

Można zwrócić plik lub wszystkie zakwalifikowane obiekty w folderze, używając opcji **Zwróć** (Check In) w menu wywoływanym lub na stronie **Właściwości** (Properties).

Można zwracać obiekty, które spełniają następujące wymagania:

- Typ obiektu jest obsługiwany przez komendę Zwrócenie obiektu (Check In Object - CHKIN).
- Obiekt jest aktualnie pobrany.

Aby zwrócić obiekty przy użyciu menu wywoływanego, wykonaj następujące czynności:

**Uwaga:** Ta metoda jest dostępna tylko w programie System i Navigator w wersji V6R1 lub nowszej. W przypadku starszych wersji należy użyć metody **Właściwości** (Properties).

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym przyciskiem myszy plik, który ma zostać zwrócony lub folder, którego cała zawartość ma być zwrócona.
3. W menu wywoływanym wybierz opcję **Zwróć** (Check In).

Aby zwrócić obiekty za pomocą strony **Właściwości** (Properties), wykonaj następujące czynności:

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym przyciskiem myszy plik, który ma zostać zwrócony lub folder, którego cała zawartość ma być zwrócona.

**Uwaga:** Funkcja zwracania zawartości całego folderu jest dostępna tylko w systemie w wersji V6R1 lub nowszej.

3. W menu wywoływanym wybierz opcję **Właściwości** (Properties).
4. W oknie **Właściwości** (Properties) kliknij zakładkę **Użyj** (Use).
5. Zwróć plik lub wszystkie obiekty w folderze.
  - Jeśli chcesz zwrócić plik, kliknij opcję **Zwróć** (Check In).
  - Jeśli chcesz zwrócić wszystkie obiekty w folderze, kliknij opcję **Zwróć** (Check In). Wyświetlone zostanie okno z komunikatem potwierdzającym. Kliknij przycisk **Kontynuuj** (Continue), aby kontynuować operację zwracania. Operacja ta może długo potrwać, w zależności od liczby zwracanych obiektów.

## Pobieranie obiektów

Można pobrać plik lub wszystkie zakwalifikowane obiekty w folderze, używając opcji **Pobierz** (Check Out) w menu wywoływanym lub na stronie **Właściwości** (Properties).

Można pobierać obiekty, które spełniają następujące wymagania:

- Typ obiektu jest obsługiwany przez komendę Pobranie obiektu (Check Out Object - CHKOUT).

• Obiekt jest aktualnie zwrócony.

Aby pobrać obiekty przy użyciu menu wywoływanego, wykonaj następujące czynności:

**Uwaga:** Ta metoda jest dostępna tylko w programie System i Navigator w wersji V6R1 lub nowszej. W przypadku starszych wersji należy użyć metody **Właściwości** (Properties).

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym przyciskiem myszy plik, który ma zostać pobrany lub folder, którego cała zawartość ma być pobrana.
3. W menu wywoływanym wybierz opcję **Pobierz** (Check Out).

Aby pobrać obiekty za pomocą strony **Właściwości** (Properties), wykonaj następujące czynności.

1. W programie System i Navigator rozwiń **Moje połączenia** → (**My Connections**) *system* → **Systemy plików**(File Systems) **Zintegrowany system plików** (Integrated File System).
2. Kliknij prawym przyciskiem myszy plik, który ma zostać pobrany lub folder, którego cała zawartość ma być pobrana.

**Uwaga:** Funkcja pobierania zawartości całego folderu jest dostępna tylko w systemie w wersji V6R1 lub nowszej.

3. W menu wywoływanym wybierz opcję **Właściwości** (Properties).
4. W oknie **Właściwości** (Properties) kliknij zakładkę **Użyj** (Use).
5. Pobierz plik lub wszystkie obiekty w folderze.
  - Jeśli chcesz pobrać plik, kliknij opcję **Pobierz** (Check Out).
  - Jeśli chcesz pobrać wszystkie obiekty w folderze, kliknij opcję **Pobierz** (Check Out). Wyświetlone zostanie okno z komunikatem potwierdzającym. Kliknij przycisk **Kontynuuj** (Continue), aby kontynuować operację pobierania. Operacja ta może długo potrwać, w zależności od liczby pobieranych obiektów.

---

## Zdalne wywołanie procedury niezależne od transportów

Opracowane przez firmę Sun Microsystems wywołanie RPC (remote procedure call - zdalne wywołanie procedury) w prosty sposób oddziela aplikacje klienta od mechanizmu serwera.

Obejmuje ono standardową reprezentację danych o nazwie XDR (eXternal Data Representation), która umożliwia wielu typom maszyn dostęp do przesyłanych danych. Najnowszą wersją RPC jest TI-RPC (Transport-independent RPC). Jest to metoda odizolowania od protokołu, który jest używany w warstwie sieci, co umożliwia płynne przejścia między protokołami. Jedynymi protokołami aktualnie dostępnymi na platformie System i są TCP i UDP.

Tworzenie rozproszonych aplikacji w sieci odbywa się za pomocą wywołania RPC. Chodzi tu przede wszystkim o aplikacje ukierunkowane na rozproszone interfejsy użytkownika lub rozproszone pobieranie danych.

## Funkcje API wyboru sieci

Poniższe funkcje API zapewniają środki do wyboru transportu aplikacji.

Te funkcje API wymagają, aby w systemie istniał plik \*STMF /etc/netconfig. Jeśli w katalogu /etc nie ma pliku netconfig, użytkownik musi skopiować go z katalogu /QIBM/ProdData/OS400/RPC. Plik netconfig zawsze jest w katalogu /QIBM/ProdData/OS400/RPC.

funkcja API	Opis
endnetconfig()	Zwalnia wskaźnik do rekordów przechowywanych w pliku netconfig.
freenetconfig()	Zwalnia strukturę netconfig, która została zwrócona na skutek wywołania do funkcji getnetconfig().

funkcja API	Opis
getnetconfig()	Zwraca wskaźnik do bieżącego rekordu w pliku netconfig i ustawia wskaźnik na następnym rekordzie.
getnetconfigent()	Zwraca wskaźnik do struktury netconfig, która odpowiada wejściowemu identyfikatorowi sieci (netid).
setnetconfig()	Ustawia wskaźnik rekordu na pierwszą pozycję pliku netconfig. Funkcji setnetconfig() należy używać przed pierwszym użyciem funkcji getnetconfig(). Funkcja setnetconfig() zwraca unikalny uchwyt (wskaźnik do rekordów przechowywanych w pliku netconfig), który ma zostać użyty przez funkcję getnetconfig().

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API konwersji nazwy na adres

Poniższe funkcje API umożliwiają aplikacji otrzymywanie adresu usługi lub podanego hosta w sposób niezależny od transportu.

funkcja API	Opis
netdir_free()	Zwalnia struktury, które zostały przydzielone przez funkcje API konwersji nazwy na adres.
netdir_getbyaddr()	Odwzorowuje adresy na nazwy hostów i nazwy usług.
netdir_getbyname()	Odwzorowuje podane w parametrze usługi, nazwę hosta oraz nazwę usługi na zbiór adresów, które są zgodne z transportem zidentyfikowanym w strukturze netconfig.
netdir_options()	Stanowi interfejs dla funkcji transportu, takich jak adres rozgłośni i zarezerwowany port TCP i UDP.
netdir_sperror()	Zwraca komunikaty opisowe, określające dlaczego jedna z funkcji API konwersji nazwy na adres nie powiodła się.
taddr2uaddr()	Dokonuje konwersji adresu specyficznego dla danego transportu (lokalnego) na adres niezależny od danego transportu (uniwersalny).
uaddr2taddr()	Dokonuje konwersji adresu niezależnego od danego transportu (uniwersalnego) na adres specyficzny dla danego transportu (lokalny) (struktura netbuf).

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Zewnętrzna reprezentacja danych (XDR)

Poniższe funkcje API umożliwiają aplikacjom RPC obsługę dowolnych struktur danych, bez względu na różnice między hostami dotyczące uporządkowania bajtów lub sposobu rozmieszczenia składowych struktury.

funkcja API	Opis
xdr_array()	Filtr, który dokonuje konwersji między tablicami o zmiennej długości i odpowiadającymi im zewnętrznymi reprezentacjami. Funkcja ta jest wywoływana do kodowania i dekodowania każdego elementu tablicy.
xdr_bool()	Filtr, który dokonuje konwersji między zmiennymi typu Boolean (zmiennie całkowite w języku C) i ich zewnętrznymi reprezentacjami. Podczas kodowania danych filtr ten tworzy wartości równe 1 lub 0.

<b>funkcja API</b>	<b>Opis</b>
xdr_bytes()	Filtr, który dokonuje konwersji między tablicami bajtów i ich zewnętrznymi reprezentacjami. Funkcja ta operuje na tablicach, których elementy mają rozmiar 1 bajtu z wbudowanym zewnętrznym opisem każdego elementu. Długość ciągu bajtów jest jawnie podana jako liczba całkowita bez znaku (unsigned integer). Ciąg bajtów nie jest zakończony znakiem null. Zewnętrzna i wewnętrzna reprezentacja bajtów jest taka sama.
xdr_char()	Filtr, który dokonuje konwersji między znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_double()	Filtr, który dokonuje konwersji między liczbami o podwójnej precyzji języka C i ich zewnętrznymi reprezentacjami.
xdr_double_char()	Filtr, który dokonuje konwersji między dwubajtowymi znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_enum()	Filtr, który dokonuje konwersji między liczbami typu wyliczeniowego (enum) języka C i ich zewnętrznymi reprezentacjami.
xdr_free()	Rekurencyjnie zwalnia obiekt wskazany przez przekazany wskaźnik.
xdr_float()	Filtr, który dokonuje konwersji między liczbami zmiennoprzecinkowymi (znormalizowane liczby zmiennoprzecinkowe pojedynczej precyzji) języka C i ich zewnętrznymi reprezentacjami.
xdr_int()	Filtr, który dokonuje konwersji między liczbami całkowitymi języka C i ich zewnętrznymi reprezentacjami.
xdr_long()	Filtr, który dokonuje konwersji między długimi liczbami całkowitymi (long integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_netobj()	Filtr, który dokonuje konwersji między nieprzezroczystymi (opaque) danymi o zmiennej długości i ich zewnętrznymi reprezentacjami.
xdr_opaque()	Filtr podstawowy, który dokonuje konwersji między nieprzezroczystymi danymi o stałej długości i ich zewnętrznymi reprezentacjami.
xdr_pointer()	Zapewnia uchwycenie (chasing) wskaźnika wewnątrz struktur i szereguje wskaźniki null (dokonuje ich konwersji do postaci XDR). Może reprezentować rekurencyjne struktury danych, takie jak drzewa binarne lub listy.
xdr_reference()	Filtr, który zapewnia uchwycenie wskaźnika wewnątrz struktur. Umożliwia on szeregowanie, deszeregowanie i zwalnianie wskaźników wewnątrz jednej struktury, do której istnieją odniesienia z innej struktury. Funkcja xdr_reference() nie nadaje specjalnego znaczenia wskaźnikowi null podczas szeregowania i przesłanie adresu wskaźnika null może spowodować błąd pamięci. Dlatego programista musi opisywać dane za pomocą unii o dwóch składowych. Jedna składowa jest używana, gdy wskaźnik jest poprawny, druga, gdy wskazuje na wartość null.
xdr_short()	Filtr, który dokonuje konwersji między krótkimi liczbami całkowitymi (short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_string()	Filtr, który dokonuje konwersji między ciągami znaków języka C i ich zewnętrznymi reprezentacjami.
xdr_u_char()	Filtr, który dokonuje konwersji między znakami języka C (unsigned char) i ich zewnętrznymi reprezentacjami.
xdr_u_int()	Filtr, który dokonuje konwersji między liczbami całkowitymi bez znaku (unsigned integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_long()	Filtr, który dokonuje konwersji między długimi liczbami całkowitymi bez znaku (unsigned long integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_short()	Filtr, który dokonuje konwersji między krótkimi liczbami całkowitymi bez znaku (unsigned short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_union()	Filtr, który dokonuje konwersji między uniami języka C i ich zewnętrznymi reprezentacjami.

funkcja API	Opis
xdr_vector()	Filtr, który dokonuje konwersji między tablicami o stałej długości i odpowiadającymi im zewnętrznymi reprezentacjami.
xdr_void()	Bez parametrów. Jest przesyłana do innych funkcji RPC, które wymagają parametrów, ale nie przesyłają danych.
xdr_wrapstring()	Funkcja, która wywołuje funkcję API xdr_string(xdr, sp, maxuint), gdzie maxuint jest maksymalną wartością liczby całkowitej bez znaku. Funkcja xdr_wrapstring() jest użyteczna, ponieważ pakiet RPC przesyła maksymalnie dwie funkcje XDR jako parametry, a funkcja xdr_string() wymaga trzech.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Uwierzytelnianie funkcji API

Poniższe funkcje API uwierzytelniają aplikacje TI-RPC.

funkcja API	Opis
auth_destroy()	Niszczy strukturę informacji dotyczącą uwierzytelnienia, które zostało wskazane przez parametr auth.
authnone_create()	Tworzy i zwraca domyślny uchwyt zalegalizowania RPC, który za każdym zdalnym wywołaniem procedury przesyła puste (null) informacje dotyczące uwierzytelnienia.
authsys_create()	Tworzy i zwraca uchwyt uwierzytelnienia RPC, który zawiera informacje dotyczące uwierzytelnienia.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API wywołań RPC niezależnych od transportu (TI-RPC)

Poniższe funkcje API udostępniają środowisko projektowania aplikacji rozproszonych poprzez odizolowanie aplikacji od specyfiki funkcji transportu. Ułatwia to korzystanie z transportu.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Uproszczone funkcje API wywołań TI-RPC

Poniższe uproszczone funkcje API określają typ transportu, którego należy użyć. Aplikacje używające tego poziomu nie muszą jawnie tworzyć uchwytów.

funkcja API	Opis
rpc_call()	Wywołuje zdalną procedurę w podanym systemie.
rpc_reg()	Rejestruje procedurę z pakietem usług RPC.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API najwyższego poziomu TI-RPC

Poniższe funkcje API umożliwiają aplikacji podanie typu transportu.

funkcja API	Opis
clnt_call()	Wywołuje zdalną procedurę związaną z klientem.
clnt_control()	Zmienia informacje na temat obiektu klienta.

<b>funkcja API</b>	<b>Opis</b>
clnt_create()	Tworzy ogólny uchwyt klienta
clnt_destroy()	Niszczy uchwyt RPC klienta.
svc_create()	Tworzy uchwyt serwera.
svc_destroy()	Niszczy uchwyt transportu usługi RPC.

#### **Informacje pokrewne**

Narzędzie wyszukiwania funkcji API

### **Funkcje API poziomu pośredniego aplikacji TI-RPC**

Poniższe funkcje API są podobne do funkcji API najwyższego poziomu, ale aplikacje użytkownika wybierają informacje specyficzne dla transportu za pomocą funkcji API służących do wyboru sieci.

<b>funkcja API</b>	<b>Opis</b>
clnt_tp_create()	Tworzy uchwyt klienta.
svc_tp_create()	Tworzy uchwyt serwera.

#### **Informacje pokrewne**

Narzędzie wyszukiwania funkcji API

### **Funkcje API poziomu najwyższego aplikacji TI-RPC**

Poniższe funkcje API umożliwiają aplikacji wybór typu transportu. Oferują także zwiększony zakres sterowania szczegółami uchwytów CLIENT i SVCXPRT. Te funkcje API są podobne do funkcji API pośredniego poziomu z dodatkowym sterowaniem zapewnionym poprzez używanie funkcji API służących do konwersji nazwy na adres.

Używanie funkcji API służących do konwersji nazwy na adres zapewnia dodatkowe sterowanie.

<b>funkcja API</b>	<b>Opis</b>
clnt_tli_create()	Tworzy uchwyt klienta.
rpcb_getaddr()	Znajduje uniwersalny adres usługi.
rpcb_set()	Rejestruje adres serwera za pomocą RPCbind.
rpcb_unset()	Używana przez serwery do wyrejestrowania adresów.
svc_reg()	Wiąże program i jego wersję z programem szeregującym.
svc_tli_create()	Tworzy uchwyt serwera.
svc_unreg()	Usuwa dowiązanie ustawione za pomocą funkcji svc_reg().

#### **Informacje pokrewne**

Narzędzie wyszukiwania funkcji API

### **Inne funkcje API wywołań TI-RPC**

Te funkcje API w połączeniu z innymi funkcjami API (uproszczonymi, najwyższego, pośredniego i eksperckiego poziomu) umożliwiają różnym aplikacjom pracę.

<b>funkcja API</b>	<b>Opis</b>
clnt_freeres()	Zwalnia dane przydzielone przez system RPC lub XDR.
clnt_geterr()	Pobiera strukturę błędu z uchwytu klienta.
svc_freeargs()	Zwalnia dane przydzielone przez system RPC lub XDR.
svc_getargs()	Dekoduje argumenty żądania RPC.
svc_getrpcaller()	Pobiera adres sieciowy wywołującego.

funkcja API	Opis
svc_run()	Czeka na żądania RPC.
svc_sendreply()	Wysyła wyniki wywołania procedury do zdalnego klienta.
svcerr_decode()	Wysyła informacje do klienta związane z błędem dekodowania.
svcerr_noproc()	Wysyła do klienta informacje związane z błędem numerycznym procedury.
svcerr_systemerr()	Wysyła do klienta informacje związane z błędem systemowym.

### Informacje pokrewne





Narzędzie wyszukiwania funkcji API

---

## Informacje pokrewne dotyczące zintegrowanego systemu plików

Podręczniki produktów i inne kolekcje tematów w Centrum informacyjnym zawierają informacje, które wiążą się z kolekcją tematów Zintegrowany system plików. Wszystkie pliki PDF można wyświetlić lub wydrukować.

### Podręczniki

-  i5/OS Network File System Support . (2105 KB) W podręczniku tym opisano system plików NFS (Network File System), wykorzystując w tym celu szereg przykładów zaczerpniętych z rzeczywistych zastosowań. Zawarto w nim informacje na temat eksportu, dołączania, blokowania plików i na temat ochrony. Z podręcznika można się dowiedzieć, jak używać systemu NFS do zbudowania i rozwijania bezpiecznej sieciowej przestrzeni nazw.
-  WebSphere Development Studio: ILE C/C++ Language Reference . (4490 KB) Podręcznik ten zawiera informacje na temat projektowania, tworzenie, kompilowania, uruchamiania i debugowania programów ILE C na platformie System i.
-  APPC Programming . (1497 KB) Podręcznik ten zawiera informacje o zaawansowanej obsłudze komunikacji program-program (APPC) na platformach System i. Jest to przewodnik po programach do rozwoju aplikacji korzystających z APPC podczas definiowania środowiska komunikacyjnego dla APPC.
-  Odtwarzanie systemu . (8404 KB) Podręcznik ten zawiera ogólne informacje o opcjach odzyskiwania i dostępności dla platform System i.

### Inne informacje

#### • Raporty

Raporty te są pisane przez programistów firmy IBM, dokumentując ich bezpośrednie doświadczenia w realnym wprowadzaniu scenariuszy i rozwiązań. Pozwalają one korzystać z doświadczeń programistów IBM z konkretną implementacją rozwiązania System i; zawierają instrukcje krok po kroku oraz wskazówki. Raport Składowanie zintegrowanego systemu plików dotyczy plików i systemów plików.

- Język CL
- Globalizacja w systemie i5/OS
- Aplikacyjne interfejsy programistyczne (API)
- Zarządzanie kronikami
- Kontrola transakcji
- Podręcznik bezpieczeństwa

---

## Licencja na kod oraz Informacje dotyczące kodu

IBM udziela niewyłącznej licencji na prawa autorskie, stosowanej przy używaniu wszelkich przykładowych kodów programów, na podstawie których można wygenerować podobne funkcje dostosowane do indywidualnych wymagań.



Z ZASTRZEŻENIEM GWARANCJI WYNIKAJĄCYCH Z BEZWZGLĘDNE OBOWIĄZUJĄCYCH PRZEPISÓW PRAWA, IBM, PROGRAMIŚCI ANI DOSTAWCY IBM NIE UDZIELAJĄ NA NINIEJSZY PROGRAM ANI W ZAKRESIE EWENTUALNEGO WSPARCIA TECHNICZNEGO ŻADNYCH GWARANCJI, W TYM TAKŻE RĘKOJMI, NIE USTALAJĄ ŻADNYCH WARUNKÓW, WYRAŹNYCH CZY DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI CZY WARUNKÓW PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CZY NIENARUSZANIA PRAW STRON TRZECICH.

W ŻADNYCH OKOLICZNOŚCIACH IBM, ANI TEŻ PROGRAMIŚCI CZY DOSTAWCY PROGRAMÓW IBM, NIE PONOSZĄ ODPOWIEDZIALNOŚCI ZA PONIŻSZE SZKODY, NAWET JEŚLI ZOSTALI POINFORMOWANI O MOŻLIWOŚCI ICH WYSTĄPIENIA:

1. UTRATA LUB USZKODZENIE DANYCH;
2. SZKODY BEZPOŚREDNIE, SZCZEGÓLNE, UBOCZNE, POŚREDNIE ORAZ SZKODY, KTÓRYCH NIE MOŻNA BYŁO PRZEWIDZIEĆ PRZY ZAWIERANIU UMOWY, ANI TEŻ
3. UTRATA ZYSKÓW, KONTAKTÓW HANDLOWYCH, PRZYCHODÓW, REPUTACJI (GOODWILL) LUB PRZEWIDYWANYCH OSZCZĘDNOŚCI.

USTAWODAWSTWA NIEKTÓRYCH KRAJÓW NIE DOPUSZCZAJĄ WYŁĄCZENIA CZY OGRANICZENIA ODPOWIEDZIALNOŚCI ZA SZKODY BEZPOŚREDNIE, UBOCZNE LUB SZKODY, KTÓRYCH NIE MOŻNA BYŁO PRZEWIDZIEĆ PRZY ZAWIERANIU UMOWY, W ZWIĄZKU Z CZYM W ODNIESIENIU DO NIEKTÓRYCH KLIENTÓW POWYŻSZE WYŁĄCZENIE LUB OGRANICZENIE (TAK W CAŁOŚCI JAK I W CZĘŚCI) MOŻE NIE MIEĆ ZASTOSOWANIA.



---

## Dodatek. Uwagi

Niniejsza publikacja została przygotowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji, omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Pisemne zapytania w sprawie licencji można przysyłać na adres:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
USA

Zapytania w sprawie licencji na informacje dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub zgłaszać na piśmie pod adresem:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokio 106-0032, Japonia

**Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego:** INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE "AS IS" BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA NIE NARUSZA PRAW STRON TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych i domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy drukarskie. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych firm zostały wprowadzone wyłącznie dla wygody użytkownika i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do korzystania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjodawcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
USA

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, uiszczenie odpowiedniej opłaty.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program, Umowie Licencyjnej IBM na Kod Maszynowy lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary te wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów firm innych niż IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów firm innych niż IBM należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennych operacjach działalności gospodarczej. W celu kompleksowego ich zilustrowania, podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

#### LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programowym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów.

Każda kopia programu przykładowego lub jakiegokolwiek jego fragment, jak też jakiegokolwiek prace pochodne muszą zawierać następujące uwagi dotyczące praw autorskich:

© (nazwa przedsiębiorstwa użytkownika, rok). Fragmenty tego kodu pochodzą z © Copyright IBM Corp. (wpisać rok lub lata). Wszelkie prawa zastrzeżone.

W przypadku przeglądania niniejszych informacji w formie elektronicznej, zdjęcia i kolorowe ilustracje mogą nie być wyświetlane.

---

## Informacje dotyczące interfejsu programistycznego

Niniejsza publikacja opisuje planowane interfejsy programistyczne, pozwalające na pisanie programów umożliwiających korzystanie z usług systemu operacyjnego IBM i5/OS.

---

## Znaki towarowe

Następujące nazwy są znakami towarowymi International Business Machines Corporation w Stanach Zjednoczonych i/lub w innych krajach:

DB2  
i5/OS  
IBM  
IBM (logo)  
Integrated Language Environment  
NetServer  
OS/2  
OS/400  
System i  
System x  
WebSphere  
xSeries

Adobe, logo Adobe, PostScript oraz logo PostScript są zastrzeżonymi znakami towarowymi lub znakami towarowymi firmy Adobe Systems Incorporated w Stanach Zjednoczonych i/lub w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Microsoft, Windows, Windows NT i logo Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym Open Group w Stanach Zjednoczonych i w innych krajach.

Nazwy innych przedsiębiorstw, produktów i usług mogą być znakami towarowymi lub znakami usług innych podmiotów.

---

## Warunki

Zezwolenie na korzystanie z tych publikacji jest przyznawane na poniższych warunkach.

**Użytek osobisty:** Użytkownik ma prawo kopiować te publikacje do własnego, niekomercyjnego użytku pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa dystrybuować ani wyświetlać tych publikacji czy ich części, ani też wykonywać na ich podstawie prac pochodnych bez wyraźnej zgody IBM.

**Użytek służbowy:** Użytkownik ma prawo kopiować te publikacje, dystrybuować je i wyświetlać wyłącznie w ramach przedsiębiorstwa Użytkownika pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa wykonywać na podstawie tych publikacji ani ich fragmentów prac pochodnych, kopiować ich, dystrybuować ani wyświetlać poza przedsiębiorstwem Użytkownika bez wyraźnej zgody IBM.

Z wyjątkiem zezwoleń wyraźnie udzielonych w niniejszym dokumencie, nie udziela się jakichkolwiek innych zezwoleń, licencji ani praw, wyraźnych czy domniemanych, odnoszących się do tych publikacji czy jakichkolwiek informacji, danych, oprogramowania lub innej własności intelektualnej, o których mowa w niniejszym dokumencie.

IBM zastrzega sobie prawo do anulowania zezwolenia przyznanego w niniejszym dokumencie w każdej sytuacji, gdy, według uznania IBM, korzystanie z tych publikacji jest szkodliwe dla IBM lub jeśli IBM uzna, że warunki niniejszego dokumentu nie są przestrzegane.

Użytkownik ma prawo pobierać, eksportować lub reeksportować niniejsze informacje pod warunkiem zachowania bezwzględnej i pełnej zgodności z obowiązującym prawem i przepisami, w tym ze wszelkimi prawami i przepisami eksportowymi Stanów Zjednoczonych.

IBM NIE UDZIELA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, DOTYCZĄCYCH TREŚCI TYCH PUBLIKACJI. PUBLIKACJE TE SĄ DOSTARCZANE W STANIE, W JAKIM SIĘ ZNAJDUJĄ ("AS IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, WYRAŻNYCH CZY DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ NIENARUSZANIA PRAW STRON TRZECICH.





Drukowane w USA