

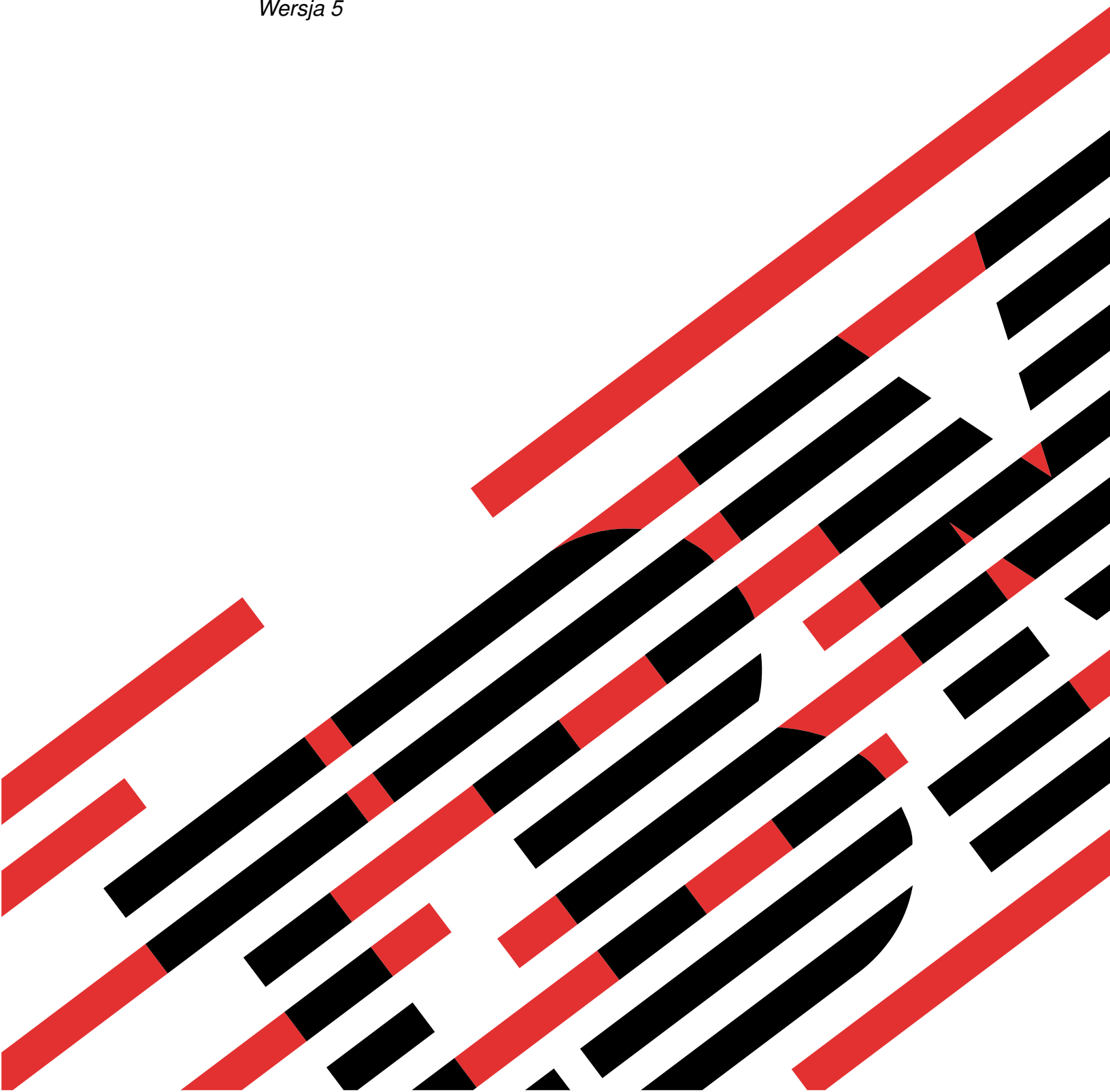
IBM

@server

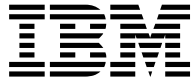
iSeries

Wprowadzenie do zintegrowanego  
systemu plików

*Wersja 5*







@server

iSeries

Wprowadzenie do zintegrowanego  
systemu plików

*Wersja 5*



---

# Spis treści

<b>Informacje o książce Wprowadzenie do zintegrowanego systemu plików</b> . . . . .	vii
Dla kogo przeznaczona jest książka Wprowadzenie do zintegrowanego systemu plików . . . . .	vii
Wyłączenie odpowiedzialności za kod . . . . .	vii
<b>Rozdział 1. Wprowadzenie do zintegrowanego systemu plików</b> . . . . .	1
Czym jest zintegrowany system plików . . . . .	1
Dlaczego warto korzystać ze zintegrowanego systemu plików . . . . .	1
<b>Rozdział 2. Zagadnienia związane ze zintegrowanym systemem plików</b> . . . . .	3
Plik strumieniowy . . . . .	3
Pliki strumieniowe *TYPE1 i *TYPE2 . . . . .	3
System plików w zintegrowanym systemie plików . . . . .	4
Katalog . . . . .	6
Katalog bieżący i katalog osobisty . . . . .	7
Katalogi *TYPE2 . . . . .	8
Korzystanie z katalogów *TYPE2 w systemie OS/400 V5R1 . . . . .	10
Konwersja do katalogu *TYPE2 . . . . .	10
Niedostępność bazowego systemu plików "Root", QOpenSys lub UDFS . . . . .	10
Wymagania pamięci dyskowej . . . . .	11
Uwagi na temat dowiązań symbolicznych . . . . .	11
Niezależne pule pamięci dyskowej (ASP) . . . . .	11
Uwagi na temat składowania/odzyskiwania . . . . .	12
Przygotowanie do konwersji do formatu *TYPE2 . . . . .	12
Przetwarzanie konwersji . . . . .	13
Przykład: konwersja wszystkich systemów plików (mała liczba obiektów) . . . . .	14
Przykład: konwersja wszystkich systemów plików (duża liczba obiektów) . . . . .	15
Przykład: konwersja wybranych pul ASP . . . . .	15
Nazwa ścieżki . . . . .	16
Dowiązanie . . . . .	17
Dowiązanie stałe . . . . .	18
Dowiązanie symboliczne . . . . .	19
Porównanie: dowiązania stałe i symboliczne . . . . .	20
Atrybuty rozszerzone . . . . .	21
Niezmiennosc nazw . . . . .	22
<b>Rozdział 3. Dostęp do zintegrowanego systemu plików poprzez tradycyjny interfejs systemu</b> . . . . .	23
Wykonywanie operacji za pomocą menu i ekranów serwera iSeries . . . . .	23
Wykonywanie operacji za pomocą komend CL . . . . .	24
Reguły dotyczące nazw ścieżek dla komend CL i ekranów . . . . .	27
Wykonywanie operacji za pomocą komputera PC . . . . .	29
Przesyłanie plików za pomocą protokołu FTP . . . . .	30
Praca z plikami za pomocą iSeries NetServer . . . . .	31
Przenoszenie obiektów do innego systemu plików . . . . .	32
Uwagi dotyczące przenoszenia obiektów do innego systemu plików . . . . .	32
Katalogi udostępnione w zintegrowanym systemie plików . . . . .	33
<b>Rozdział 4. Dostęp do zintegrowanego systemu plików za pomocą programu iSeries Navigator</b> . . . . .	35
Odblokowanie pliku . . . . .	35
Zablokowanie pliku . . . . .	36
Nadawanie uprawnień do pliku lub folderu . . . . .	36
Konfigurowanie konwersji plików tekstowych . . . . .	36
Wysyłanie pliku lub folderu do innego systemu . . . . .	37
Zmiana opcji dla definicji pakietu . . . . .	37

Planowanie daty i godziny wysyłki pliku lub folderu . . . . .	37
Tworzenie folderu . . . . .	38
Usuwanie folderu . . . . .	38
Tworzenie współużytkowanego zasobu plikowego. . . . .	38
Zmiana współużytkowanego zasobu plikowego. . . . .	38
Tworzenie nowego systemu plików użytkownika . . . . .	39
Podłączanie systemu plików użytkownika . . . . .	39
Odłączanie systemu plików użytkownika . . . . .	40
Uruchomienie kronikowania . . . . .	40
Zakończenie kronikowania . . . . .	40
<b>Rozdział 5. Używanie zintegrowanego systemu plików w programach . . . . .</b>	<b>41</b>
Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych . . . . .	41
Kopiowanie danych za pomocą komend CL . . . . .	42
Kopiowanie danych za pomocą funkcji API . . . . .	43
Kopiowanie danych za pomocą funkcji przesyłania . . . . .	43
Kopiowanie danych pomiędzy plikami strumieniowymi a zbiorami składowania . . . . .	45
Wykonywanie operacji za pomocą funkcji API . . . . .	45
Funkcje ILE C/400 . . . . .	51
Obsługa dużych plików przez funkcje API . . . . .	52
Reguły dotyczące ścieżek dla funkcji API . . . . .	52
Deskryptor pliku . . . . .	53
Ochrona . . . . .	54
Obsługa gniazd . . . . .	54
Nazewnictwo i obsługa różnych języków . . . . .	55
Konwersja danych . . . . .	55
<b>Rozdział 6. Systemy plików w zintegrowanym systemem plików . . . . .</b>	<b>57</b>
Porównanie systemów plików . . . . .	57
Bazowy system plików "root" (/) . . . . .	60
Korzystanie z bazowego systemu plików "root" (/) . . . . .	61
System plików systemów otwartych (QOpenSys) . . . . .	62
Używanie QOpenSys . . . . .	62
System plików definiowany przez użytkownika (UDFS) . . . . .	63
Koncepcja UDFS . . . . .	63
Korzystanie z UDFS poprzez interfejs zintegrowanego systemu plików . . . . .	64
System plików bibliotek (QSYS.LIB) . . . . .	67
Korzystanie z QSYS.LIB poprzez interfejs zintegrowanego systemu plików. . . . .	68
System plików QSYS.LIB niezależnej ASP . . . . .	70
Korzystanie z systemu plików QSYS.LIB niezależnej puli ASP poprzez interfejs zintegrowanego systemu plików . . . . .	70
System plików obsługujący obiekty biblioteki dokumentów (QDLS). . . . .	72
Korzystanie z QDLS poprzez interfejs zintegrowanego systemu plików . . . . .	73
System plików nośników optycznych (QOPT) . . . . .	74
Korzystanie z QOPT poprzez interfejs zintegrowanego systemu plików . . . . .	75
System plików NetWare (QNetWare) . . . . .	76
Podłączanie systemów plików NetWare . . . . .	77
Struktura katalogów QNetWare . . . . .	77
Korzystanie z QNetWare poprzez interfejs zintegrowanego systemu plików . . . . .	77
System plików serwera Windows NT (QNTC) . . . . .	79
Korzystanie z QNTC poprzez interfejs zintegrowanego systemu plików . . . . .	79
System plików serwera plików OS/400 (QFileSvr.400) . . . . .	81
Korzystanie z QFileSvr.400 poprzez interfejs zintegrowanego systemu plików. . . . .	82
Sieciowy system plików (Network File System - NFS) . . . . .	85
Korzystanie z systemów plików NFS poprzez interfejs zintegrowanego systemu plików . . . . .	85

<b>Rozdział 7. Obsługa kronikowania dla obiektów zintegrowanego systemu plików</b>	89
Zarządzanie kronikami.	89
Jakie obiekty należy uwzględnić w kronikowaniu	89
Kronikowane obiekty zintegrowanego systemu plików	90
Kronikowane operacje.	91
Specjalne zagadnienia dotyczące pozycji kronik	91
<b>Dodatek A. Wywołanie TI-RPC (Transport Independent Remote Procedure Call)</b>	93
Wybór sieci.	93
Konwersja nazwy na adres	93
Zewnętrzna reprezentacja danych (XDR)	94
Uwierzytelnienie	95
TI-RPC (Transport Independent RPC)	95
Uprozczone funkcje API aplikacji TI-RPC	96
Funkcje API poziomu najwyższego aplikacji TI-RPC	96
Funkcje API poziomu średniego aplikacji TI-RPC	96
Funkcje API poziomu eksperckiego aplikacji TI-RPC	96
Inne funkcje API aplikacji TI-RPC	97
<b>Dodatek B. Przykładowy program wykorzystujący funkcje C zintegrowanego systemu plików</b>	99
<b>Dodatek C. Przykład programu w języku RPG dla zintegrowanego systemu plików</b>	105
<b>Bibliografia</b>	107
<b>Indeks</b>	109





---

## **Informacje o książce Wprowadzenie do zintegrowanego systemu plików**

Książka ta zawiera przegląd informacji na temat zintegrowanego systemu plików, w tym:

- definicję zintegrowanego systemu plików,
- jego zastosowania,
- pojęcia i terminologię dotyczącą zintegrowanego systemu plików,
- interfejsy, których można używać do pracy ze zintegrowanym systemem plików,
- funkcje API i techniki, których można używać tworząc programy współpracujące ze zintegrowanym systemem plików,
- cechy poszczególnych systemów plików.

---

## **Dla kogo przeznaczona jest książka Wprowadzenie do zintegrowanego systemu plików**

Książka ta przeznaczona jest dla użytkowników, programistów i osób zarządzających serwerami iSeries, które chcą zrozumieć, jak działa zintegrowany system plików i jak można go wykorzystać.

---

## **Wyłączenie odpowiedzialności za kod**

Ten dokument zawiera przykłady programów.

Firma IBM udziela użytkownikowi licencji na wykorzystanie wszystkich przykładowych kodów programów, z których można uzyskać podobną funkcjonalność, dostosowując je do własnych potrzeb.

Wszystkie przykłady programów firma IBM umieściła tu tylko w celu ilustracji. Przykłady te nie zostały gruntownie przetestowane. Dlatego IBM nie gwarantuje niezawodności, funkcjonalności ani prawidłowego działania tych programów.

Wszystkie zawarte na tej stronie programy dostarczone są użytkownikowi w takim stanie, w jakim się znajdują ("AS IS"). Firma IBM nie udziela żadnych implikowanych gwarancji nienaruszalności, sprzedawalności i przydatności do określonych celów.



---

# Rozdział 1. Wprowadzenie do zintegrowanego systemu plików

W poniższych sekcjach opisano zintegrowany system plików na serwerze iSeries oraz przedstawiono informacje o sposobie korzystania z tego systemu.

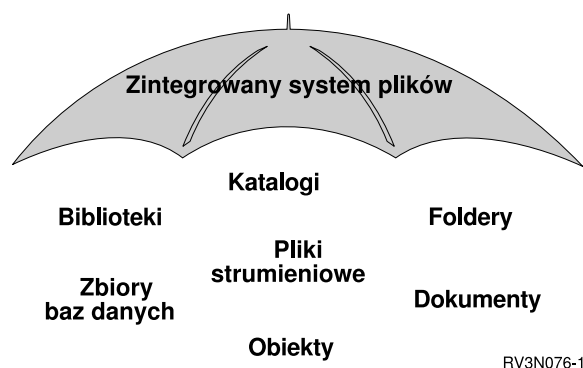
---

## Czym jest zintegrowany system plików

**Zintegrowany system plików** jest częścią systemu OS/400. Obsługuje ona strumieniowe wejście/wyjście i zarządzanie pamięcią, podobnie jak systemy operacyjne komputerów osobistych i system UNIX. Udostępnia również strukturę integrującą wszystkie informacje przechowywane na serwerze.

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, w której obiekty mogą być przedstawione jako owoce na gałęziach drzewa. Obiekt jest dostępny, kiedy zostanie podana ścieżka do niego zawierająca wszystkie katalogi.
- Wspólny interfejs umożliwiający użytkownikom i aplikacjom dostęp nie tylko do plików strumieniowych, ale także do zbiorów baz danych, dokumentów i innych obiektów przechowywanych na serwerze.
- Jednolity podgląd plików strumieniowych przechowywanych na serwerze lokalnym Integrated xSeries Server for iSeries lub na zdalnym serwerze Windows NT. Pliki strumieniowe mogą być również przechowywane zdalnie na serwerze sieci LAN, na serwerze Novell NetWare, na innym zdalnym serwerze iSeries lub na serwerze NFS (Network File System).



Rysunek 1. Struktura wszystkich informacji przechowywanych na serwerze iSeries

---

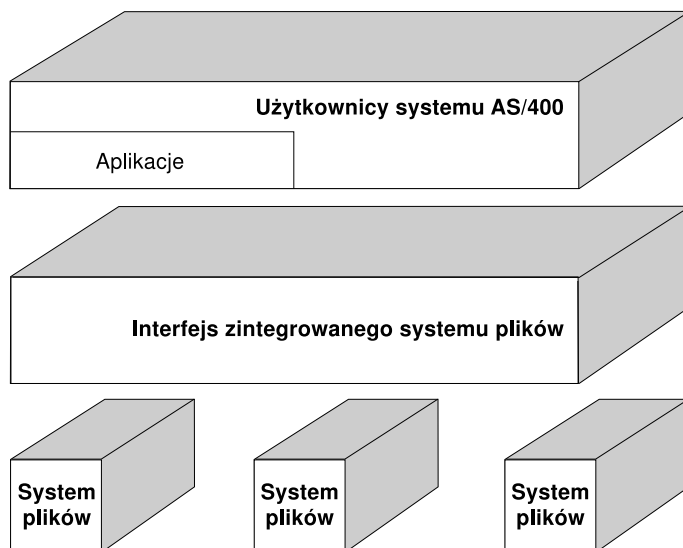
## Dlaczego warto korzystać ze zintegrowanego systemu plików

Zintegrowany system plików powiększa i tak szerokie możliwości zarządzania danymi przez system OS/400 o lepszą obsługę istniejących i przyszłych sposobów przetwarzania informacji, takich jak klient/serwer, systemy otwarte i multimedia.

Zintegrowany system plików pozwala na:

- Szybszy dostęp do danych w systemie OS/400, szczególnie dla takich aplikacji, jak Client Access, które wykorzystują serwer plików OS/400.
- 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 XPG. Ta struktura plików i katalogów stanowi także środowisko znane użytkownikom systemów operacyjnych PC, takich jak DOS, Windows czy Windows NT.

- 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.

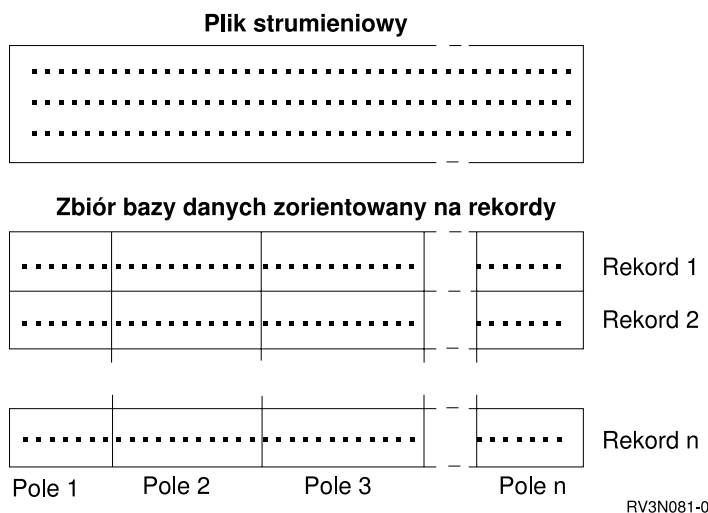
- Lepsze wykorzystanie graficznego interfejsu użytkownika przez użytkowników komputerów osobistych. Na przykład użytkownicy Windows mogą używać narzędzi graficznych Windows do pracy z plikami strumieniowymi i innymi obiektami serwera iSeries w taki sam sposób, w jaki pracują z plikami przechowywanymi w komputerach osobistych.
- 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.

## Rozdział 2. Zagadnienia związane ze zintegrowanym systemem plików

### 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 serwera są plikami strumieniowymi. Innymi przykładami plików strumieniowych są pliki komputera osobistego i systemu UNIX. Plik strumieniowy zintegrowanego systemu plików jest obiektem systemowym typu \*STMF.

Łatwiej zrozumieć ideę plików strumieniowych, jeśli porówna się je ze zbiorami baz danych serwera iSeries. Zbiór bazy danych zawiera wstępnie zdefiniowane części składowe złożone z jednego lub kilku pól o konkretnych cechach, takich jak długość i typ danych.



Rysunek 3. 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 bogatego oprogramowania serwera możliwy jest dostęp do poszczególnych pól zbioru zawierającego rekordy oraz manipulacja tymi polami. Z kolei plik strumieniowy bardziej nadaje się do przechowywania takich informacji, jak zdjęcie klienta składające 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.

Więcej informacji na temat plików strumieniowych można znaleźć w sekcjach:

- “Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych” na stronie 41.
- “Pliki strumieniowe \*TYPE1 i \*TYPE2”.

### Pliki strumieniowe \*TYPE1 i \*TYPE2

Plik ma ustawioną jedną z dwóch opcji formatu: plik strumieniowy \*TYPE1 lub plik strumieniowy \*TYPE2.

Plik strumieniowy \*TYPE1 ma ten sam format co pliki strumieniowe utworzone we wcześniejszych wydaniach systemu OS/400, do wersji 4, wydanie 4. Jest on zapisywany szybciej niż plik strumieniowy \*TYPE2 w wydaniach systemu OS/400, do wersji 4, wydanie 4. Jego minimalna wielkość to 4096 bajtów.

Plik strumieniowy \*TYPE2 charakteryzuje się wysoką wydajnością dostępu do pliku i jest nowością w wersji 4, wydaniu 4 systemu operacyjnego OS/400. Jest on zapisywany wolniej niż plik strumieniowy \*TYPE1 we wcześniejszych wydaniach systemu OS/400, do wersji 4, wydanie 4. Minimalna wielkość obiektu to 4096 bajtów. Wszystkie pliki utworzone w systemie V4R4 i w nowszych są plikami strumieniowymi \*TYPE2.

Ponieważ pliki strumieniowe \*TYPE2 działają jedynie w systemie V4R4 i w nowszych, można zapisać pliki strumieniowe \*TYPE2 w celu odtworzenia w systemach wcześniejszych niż V4R4. Proces ten może być jednak czasochłonny.

---

## System plików w zintegrowanym systemie plików

**System plików (file system)** umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, 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 OS/400 dostępu do zbiorów baz danych i obiektów innych typów poprzez biblioteki może być traktowane jako system plików. Podobnie obsługiwane przez OS/400 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.

Porównanie właściwości i ograniczeń każdego z systemów plików znajduje się w sekcji "Porównanie systemów plików" na stronie 57.

Zintegrowany system plików obsługuje następujące systemy plików:

### "root" (/)

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

### QOpenSys

System plików systemów otwartych. Jest zgodny ze standardami systemów otwartych bazujących na systemie UNIX, takich jak POSIX i XPG. Podobnie jak bazowy system plików, ten system plików również wykorzystuje obsługę plików strumieniowych i katalogów oferowaną przez zintegrowany system plików. Ponadto system ten rozróżnia wielkie i małe litery w nazwach obiektów.

**UDFS** System plików definiowany przez użytkownika. Znajduje się w puli pamięci dyskowej (ASP) lub w niezależnej puli pamięci dyskowej (IASP) wybranej przez użytkownika. System ten jest tworzony i zarządzany przez użytkownika.

### QSYS.LIB

System plików bibliotek. Obsługuje strukturę bibliotek serwera. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów serwera iSeries obsługiwanych przez biblioteki w systemie i podstawowe pule ASP użytkowników.

### QSYS.LIB niezależnej ASP

System plików QSYS.LIB niezależnej ASP. Obsługuje strukturę biblioteki serwera w każdej niezależnej ASP utworzonej i zdefiniowanej przez użytkownika. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów serwera iSeries obsługiwanych przez biblioteki.

**QDLS** System plików obsługujący biblioteki dokumentów. Umożliwia dostęp do dokumentów i folderów.

## QOPT, system plików

System plików na nośnikach optycznych. Umożliwia dostęp do danych strumieniowych przechowywanych na nośnikach optycznych.

## QNetWare

System plików QNetWare. Umożliwia dostęp do lokalnych oraz zdalnych danych i obiektów, które są przechowywane na serwerze z uruchomionym oprogramowaniem Novell NetWare 4.10 lub 4.11 lub na autonomicznych serwerach PC z uruchomionym oprogramowaniem Novell NetWare 3.12, 4.10 4.11 lub 5.0. Użytkownik może dynamicznie dołączyć system plików NetWare do istniejących lokalnie systemów plików.

**QNTC** System plików Windows NT Server. Zapewnia dostęp do danych i obiektów przechowywanych na serwerze, na którym został uruchomiony Windows NT w wersji 4.0 lub nowszej. Umożliwia to aplikacjom serwera iSeries używanie tych samych danych, z których korzystają klienci Windows NT. Zapewnia dostęp do danych na serwerze Windows NT, który został uruchomiony na Zintegrowanym serwerze PC. Więcej informacji można znaleźć w książce OS/400-AS/400 Integration with Windows NT Server, SC41-5439-01 (SC41-5439).

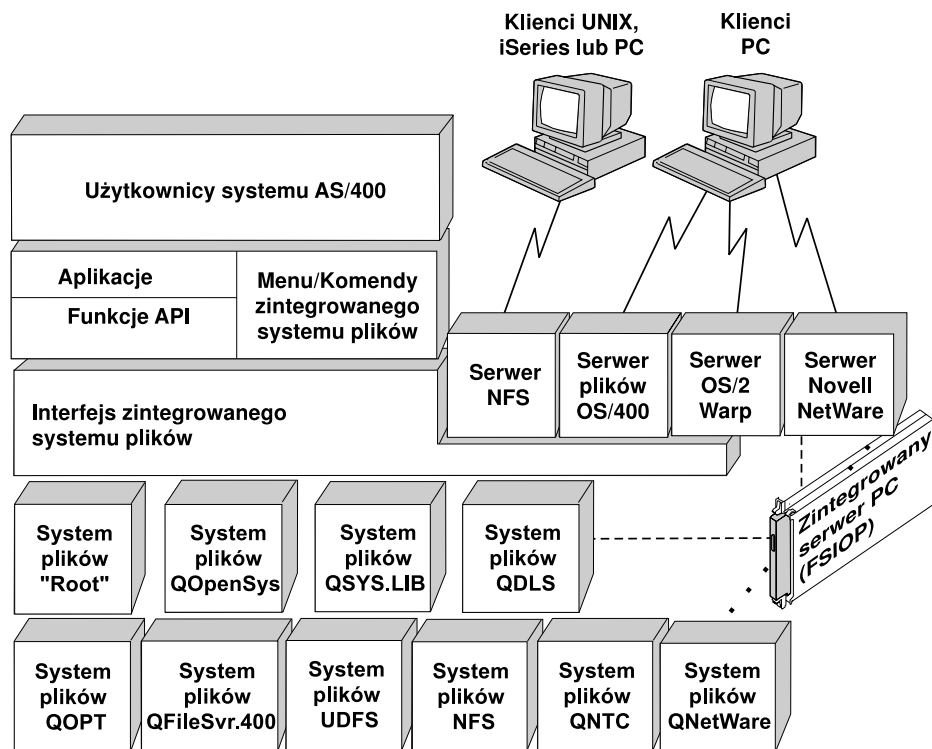
## QFileSvr.400

Umożliwia dostęp do innych systemów plików znajdujących się na zdalnych serwerach iSeries.

**NFS** Sieciowy system plików. Daje użytkownikom dostęp do danych i obiektów przechowywanych na zdalnym serwerze NFS. Sieciowy system plików może być eksportowany przez serwer NFS, a następnie dynamicznie dołączany przez klientów NFS.

Użytkownik może współdziałać z dowolnym systemem plików poprzez wspólny interfejs. Jest on zoptymalizowany dla strumieniowego wejścia/wyjścia, inaczej niż wejście/wyjście ukierunkowane na rekordy, które jest charakterystyczne dla interfejsów zarządzania danymi. Do współdziałania z systemami plików poprzez ten wspólny interfejs służą komendy, menu i ekrany oraz interfejsy API.

Więcej informacji na ten temat zawierają następujące sekcje tej książki oraz publikacje:



Rysunek 4. Systemy plików, serwery plików i interfejs zintegrowanego systemu plików

- Optical Support 

---

## Katalog

**Katalog (directory)** 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 na serwerze. 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 QSYS.LIB 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 (poddrzewo QDLS) foldery obsługiwane są podobnie jak podkatalogi, a dokumenty w folderach podobnie jak obiekty w podkatalogu.

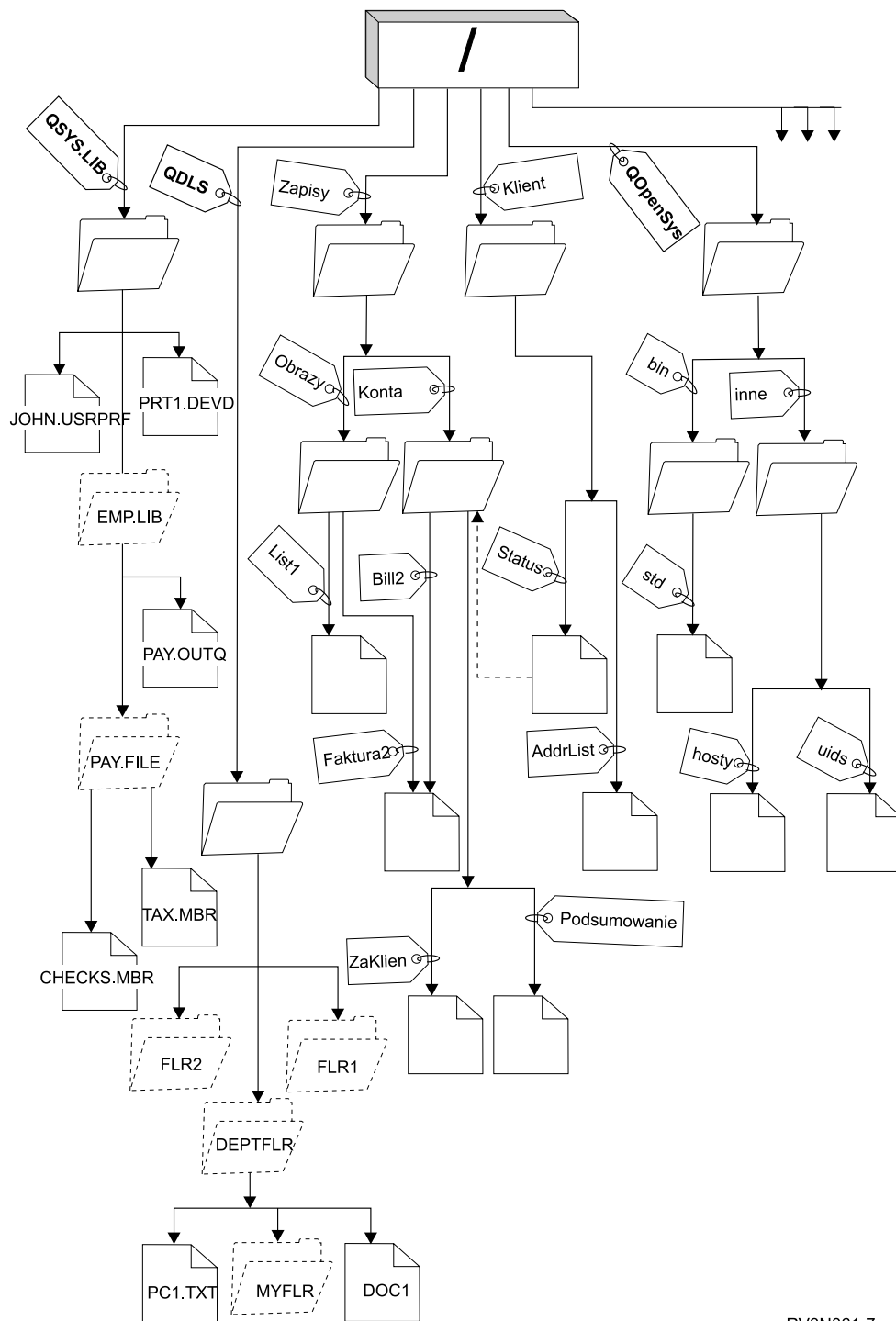
Ze względu na różnice w systemach plików pewne operacje można wykonywać tylko w niektórych poddrzewach struktury katalogów.

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.

Informacje o katalogach zintegrowanego systemu plików zawierają następujące sekcje zamieszczone w dalszej części tego dokumentu:

- "Katalog bieżący i katalog osobisty" na stronie 7
- "Katalogi udostępnione w zintegrowanym systemie plików" na stronie 33
- "Katalogi \*TYPE2" na stronie 8





RV3N061-7

Rysunek 5. Systemy plików i ich obiekty stanowią gałęzie drzewa katalogów zintegrowanego systemu plików.

## Katalog bieżący i katalog osobisty

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

**Katalog osobisty (home directory)** 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, powinien także utworzyć 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 `fchdir()`.

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. 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.

Szczegóły dotyczące funkcji API QWTCHGJB można znaleźć w sekcji Interfejsy API.

---

## Katalogi \*TYPE2

Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w zintegrowanym systemie plików 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.

Zalety katalogów \*TYPE2 są następujące:

- zwiększona wydajność,
- zwiększona niezawodność,
- zwiększona funkcjonalność,
- użyta mniejsza przestrzeń pamięci dyskowej (w większości przypadków).

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:

1. Katalogi \*TYPE2 obsługują zmianę wielkości liter w nazwie zapisanej w systemie plików stosującym wielkie litery (na przykład zmianę A na a).
2. Obiekt w katalogu \*TYPE2 może mieć do miliona dowiązań, natomiast w katalogach \*TYPE1 liczba ta wynosi 32767 dowiązań. Oznacza to, że można utworzyć do miliona stałych dowiązań do pliku strumieniowego, a katalog \*TYPE2 może zawierać do miliona podkatalogów.
3. Podczas korzystania z programu iSeries Navigator, podczas otwierania katalogu w formacie \*TYPE2 lista pozycji jest automatycznie sortowana w porządku binarnym.

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:

- System plików użytkownika (UDFS) w niezależnej puli pamięci dyskowej (ASP) jest przekształcany do formatu \*TYPE2 podczas pierwszego udostępniania puli ASP w systemie OS/400 V5R2.
- Wszystkie inne obsługiwane systemy plików, z wyjątkiem systemów UDFS w niezależnej puli ASP muszą być przekształcone do formatu \*TYPE2 przez użycie komendy Konwersja katalogu (Convert Directory - CVTDIR).
- Nowe serwery iSeries, na których jest wstępnie załadowany system OS/400 V5R2 mają katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.
- Po zainstalowaniu systemu OS/400 V5R2 na serwerze iSeries, w systemie są katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.

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

CVTDIR OPTION(\*CHECK).

**Uwaga:** Katalogi \*TYPE2 są obsługiwane w systemie OS/400 V5R1, ale obsługa różni się nieznacznie od obsługi normalnych katalogów \*TYPE2. Aby uzyskać więcej informacji, patrz sekcja Korzystanie z katalogów \*TYPE2 w systemie OS/400 V5R1.

Więcej informacji na temat katalogów \*TYPE2 zawierają następujące sekcje:

- Konwersja do katalogu \*TYPE2
- Niedostępność bazowego systemu plików "Root", QOpenSys lub UDFS
- Wymagania pamięci dyskowej
- Uwagi na temat dowiązań symbolicznych
- Niezależne pule pamięci dyskowej (ASP)
- Uwagi na temat składowania/odtworzenia
- Przygotowanie do konwersji do formatu \*TYPE2
- Przetwarzanie konwersji
- Przykład: konwersja wszystkich systemów plików (mała liczba obiektów)
- Przykład: konwersja wszystkich systemów plików (duża liczba obiektów)
- Przykład: konwersja określonych pul ASP

## Korzystanie z katalogów \*TYPE2 w systemie OS/400 V5R1


Bazowy system plików "root" (/), QOpenSys oraz zdefiniowane przez użytkownika systemy plików (UDFS) w zintegrowanym systemie plików obsługują format katalogu \*TYPE2. Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1. Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1.

W systemie w wersji V5R1 można przekształcić katalogi V5R1 do katalogów formatu \*TYPE2. Zaleca się przekształcenie katalogów formatu \*TYPE2 **przed** instalacją nowego wydania systemu OS/400. Jest to konieczne, ponieważ przekształcenie katalogów może zostać automatycznie wykonane podczas instalacji. Wadą automatycznej konwersji podczas instalacji jest to, że wymagany czas instalacji ulega **znacznemu** wydłużeniu.

**Uwaga:** W przypadku aktualizacji systemu OS/400 V5R1 do wersji V5R2, automatyczna konwersja katalogów do formatu \*TYPE2 **nie jest** wykonywana. Nie ma potrzeby dokonywania konwersji katalogów we wcześniejszych instalacjach.

Obsługa katalogów \*TYPE2 w wersji V5R1 jest dostępna po zainstalowaniu odpowiednich poprawek (PTF). Program narzędziowy do konwersji różni się nieznacznie od programu w wersji V5R2. Pełna dokumentacja dotycząca katalogów \*TYPE2 w wersji V5R1 znajduje się w raporcie informacyjnym APAR II13161. Aby uzyskać dostęp do raportu APAR, należy użyć jednego z poniższych sposobów:

1. Pobierz raport informacyjny APAR do serwera iSeries i przejrzyj go. Użyj następujących komend:  

```
SNDPTFORD PTFID((II13161))  
DSPPTFCVR LICPGM(INFOAS4) SELECT(II13161)
```
2. Przejdź do witryny pod adresem <http://www-912.ibm.com> , aby przejrzeć raport informacyjny APAR w trybie online. Wybierz pozycję **Authorized Program Analysis Reports (APARs) → V5R1 APARs → APAR number II13161**.

## Konwersja do katalogu \*TYPE2

Komenda CVTDIR dokonuje konwersji z katalogu \*TYPE1 do katalogu \*TYPE2. Ponadto, udostępnia informacje o tym, jak przekształcić systemy plików do katalogu formatu \*TYPE2. Komenda CVTDIR wykonuje następujące czynności:

- Wyświetla bieżący format katalogu dla bieżących systemów plików, które obsługują katalogi \*TYPE2.
- Określa czas, jaki zajmie proces konwersji.
- Określa wymagania przestrzeni dyskowej dla konwersji.
- Konwertuje systemy plików do formatu \*TYPE2. Wszystkie istniejące katalogi są konwertowane do formatu \*TYPE2, a wszystkie nowo utworzone foldery po konwersji są formatu \*TYPE2.

Istnieje kilka sposobów konwersji katalogów w jednym systemie plików:

- Ręcznie, za pomocą komendy CVTDIR.
- Automatycznie, gdy niezależna pula ASP jest pierwszy raz dołączana do systemu, na którym zainstalowany jest system OS/400 V5R2.
- Podczas pracy programu IPL, jeśli system określi, że nieprawidłowe zakończenie pracy systemu nastąpiło podczas konwersji systemu plików.
- Podczas wykonywania komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG SELECT(\*ALL)), jeśli wykryta zostanie utrata katalogów \*TYPE1, które są częścią systemu plików przekształconego do formatu \*TYPE2.

## Niedostępność bazowego systemu plików "Root", QOpenSys lub UDFS

Konwersja bazowego systemu plików "root" (/) lub systemu plików QOpenSys musi być wykonana, gdy system znajduje się w stanie zastrzeżonym. Podczas konwertowania UDFS, system nie musi znajdować się w stanie zastrzeżonym, jednak UDFS w tej puli ASP nie będzie dostępny podczas konwersji. Czas wymagany na dokonanie konwersji zależy od wielkości systemu plików. Aby więc wybrać najlepszy czas na

wykonanie konwersji, niezbędne jest planowanie. Opcja \*ESTIMATE w komendzie CVTDIR szacuje czas potrzebny na dokonanie konwersji określonego systemu plików. Oszacowany czas ma najdłuższą szacowaną wartość. Szacowany jest czas konwersji w przypadku uruchomienia konwersji jako pojedynczego wątku. Rzeczywista konwersja wykorzystuje wiele wątków i powinna zająć mniej czasu niż szacowana wartość. Typowo, system plików, który ma więcej niż 40000 dowiązań, może zostać przekonwertowany w czasie równym od 30 do 50 procent szacowanego czasu. Rzeczywisty czas zależy jednak od sprzętu i konfiguracji serwera.

Jeśli system nieprawidłowo zakończy pracę podczas wykonywania komendy CVTDIR, to podczas kolejnego IPL funkcja konwersji uruchamiana jest w zadaniu SCPF. Zadanie SCPF nie zezwala na uaktywnienie kilku wątków. Z tego względu, jeśli konwersja systemu plików musi być dokonana podczas IPL, jest uruchamiana w pojedynczym wątku. Funkcja konwersji jest uruchamiana, gdy podczas pracy programu IPL wyświetlony zostanie komunikat SRC C900 2A85 i wyświetlony zostanie komunikat statusu CPIA089 wskazujący postęp konwersji.

Funkcja konwersji jest uruchamiana podczas wykonywania komendy RCLSTG, jeśli istnieją utracone katalogi \*TYPE1 w systemie plików, który został już przekonwertowany do \*TYPE2. Funkcja konwersji jest uruchamiana w zadaniu, które wywołało komendę RCLSTG. Jeśli zostaną znalezione jakiegokolwiek utracone katalogi, które muszą być przekonwertowane, to z powodu ograniczeń systemu konwersja uruchamiana jest w pojedynczym wątku.

## Wymagania pamięci dyskowej

Przed dokonaniem konwersji katalogów w systemie plików do formatu \*TYPE2, należy rozważyć kwestię wymaganej pamięci dyskowej. Istnieje kilka problemów 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 określona liczba katalogów była jednocześnie zarówno w wersji \*TYPE1, jak i \*TYPE2. Ta liczba jest zależna od konfiguracji serwera iSeries i struktury katalogów konwertowanego systemu plików.

Opcja \*ESTIMATE w komendzie CVTDIR oferuje informacje, które wskazują na szacowaną wielkość przestrzeni dyskowej wymaganej do dokonania konwersji.

## Uwagi na temat dowiązań symbolicznych

Dowiązania symboliczne to obiekty w zintegrowanym systemie plików, które zawierają ścieżkę do innego obiektu. Podczas konwersji w niektórych przypadkach nazwa obiektu może ulec zmianie. Jeśli podczas konwersji zostanie zmieniona nazwa jednego z elementów ścieżki w dowiązaniu symbolicznym, to zawartość dowiązania symbolicznego nie będzie już wskazywać na obiekt. Szczegóły dotyczące zmiany nazw obiektów .ożna znaleźć w sekcji Obiekty o zmienionej nazwie.

## Niezależne pule pamięci dyskowej (ASP)

Gdy niezależna pula ASP jest pierwszy raz dołączana do systemu, na którym zainstalowany jest system OS/400 V5R2, katalogi są konwertowane do formatu \*TYPE2. Dla celów planowania, w systemie OS/400 V5R1 znajduje się funkcja szacująca, która podaje informację o czasie trwania konwersji. Przed dołączeniem niezależnej puli ASP do serwera V5R2, należy uruchomić następującą funkcję API w systemie V5R1, gdy niezależna pula ASP (o nazwie ASP\_NAME) jest dołączona i aktywna:

CALL QP0FCVT2 (\*ESTIMATE ASP\_NAME \*TYPE2)

**Uwaga:** Przed wywołaniem tej funkcji zaleca się uruchomienie komendy RCLSTG dla niezależnej puli ASP w systemie V5R1.

## Uwagi na temat składowania/odzyskiwania

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.

## Przygotowanie do konwersji do formatu \*TYPE2

Przed dokonaniem konwersji katalogów do formatu \*TYPE2 zaleca się użycie następujących komend i parametrów:


- Odzyskanie pamięci (Reclaim Storage - RCLSTG)

Użycie komendy RCLSTG SELECT(\*ALL) przed dokonaniem konwersji jakiegokolwiek systemu plików powoduje wyczyszczenie katalogów i zapewnia, że katalogi są poprawne. Nie eliminuje to wszystkich możliwych problemów, na które można się natknąć podczas konwersji katalogów, ale zapewnia, że możliwy będzie odczyt katalogu w systemie plików.

Komendę tę należy uruchomić tylko raz, przed wykonaniem jakiegokolwiek opcji komendy CVTDIR.

- Składowanie systemu (Save System - SAVSYS)

Pełne składowanie systemu serwera iSeries powinno zostać wykonane po wykonaniu komendy RCLSTG i przed użyciem opcji \*CONVERT komendy CVTDIR. Aby wykonać składowanie systemu, należy użyć menu Składowanie (Save) w serwerze iSeries. Aby przejść do menu Składowanie (Save), w dowolnym wierszu komend wpisz GO SAVE i wybierz opcję 21. Więcej informacji można znaleźć w podręczniku

Składowanie i odtwarzanie  .

Komendę tę należy uruchomić tylko raz, przed wykonaniem jakiegokolwiek opcji komendy CVTDIR.

- Opcja \*ESTIMATE komendy CVTDIR

Opcji \*ESTIMATE komendy CVTDIR należy użyć do określenia czasu wymaganego na konwersję katalogów.

Oprócz informowania o czasie i o szacowanej ilości pamięci dyskowej, opcja \*ESTIMATE oferuje jeszcze dodatkowe korzyści. Tworzy ona pewne dodatkowe obiekty powiązane z katalogami \*TYPE1, które umożliwiają szybsze wykonanie konwersji (ponieważ nie muszą być one tworzone w trakcie jej trwania). Te dodatkowe obiekty istnieją dopóki opcja \*CONVERT nie zostanie użyta do konwersji systemu plików do formatu \*TYPE2. Opcja \*ESTIMATE odczytuje także wszystkie katalogi w systemie plików, co jest równoznaczne ze sprawdzeniem katalogów. Opcja \*ESTIMATE nie gwarantuje odnalezienia wszystkich możliwych błędów, które mogą wystąpić podczas rzeczywistej konwersji, ale pomaga wykryć część z nich.

Po uruchomieniu opcji \*ESTIMATE należy sprawdzić w protokole błędów, czy wystąpiły błędy, i wykonać zalecane czynności przed dokonaniem konwersji systemu plików. Ponowne uruchomienie opcji po wykonaniu zalecanych czynności nie jest wymagane, ale może być pożądane w celu sprawdzenia, czy nie wystąpią inne problemy.

- Wymagania pamięci dyskowej

Należy sprawdzić dostępność pamięci dyskowej dla puli ASP, która zawiera konwertowany system plików. Opcja CVTDIR \*ESTIMATE wyświetla komunikat CPIA090, który wskazuje wielkość dostępnej pamięci dyskowej dla puli ASP. Ponadto, wyświetla wielkość szacowanej pamięci dyskowej, która będzie potrzebna w czasie konwersji. Wyświetlany jest także komunikat CPIA091, który określa, czy całkowita wielkość katalogów \*TYPE2 w systemie plików po konwersji będzie większa czy mniejsza niż istniejące katalogi \*TYPE1. Dostępna pamięć w puli ASP powinna być równa sumie niewykorzystanej pamięci dyskowej (wyświetlonej w komunikacie CPIA090) i różnicy pomiędzy wielkością katalogu \*TYPE1 i \*TYPE2 (wyświetloną w komunikacie CPIA091).

Innym sposobem uzyskania informacji o wielkości dostępnej pamięci dyskowej jest uruchomienie komendy STRSST i wybranie opcji Praca z jednostkami dysków (Work With Disk Units).

**Uwaga:** Jeśli w systemie zdefiniowana jest tylko jedna pula ASP, komenda Praca ze statusem systemu (Work with System Status - WRKSYSSTS) jest wystarczająca do uzyskania informacji o dostępnej pamięci dyskowej.

Dobrze jest wykonać ogólne czyszczenie systemu przed użyciem jakiegokolwiek opcji komendy CVTDIR. Jeśli w systemie znajdują się niepotrzebne katalogi lub pliki, należy je usunąć przed użyciem opcji komendy CVTDIR. Czynność ta spowoduje zwolnienie pamięci dyskowej, zapewnia bardziej dokładne szacowanie dostępnej pamięci dyskowej i umożliwia konwersję w krótszym czasie, ponieważ liczba obiektów do przetworzenia jest mniejsza.

- Należy rozważyć zmianę działania kolejki komunikatów na \*PRTWRAP dla zadania, które wywołuje komendę CVTDIR. Wykonanie tej czynności:
  1. Chroni przed nieprawidłowym zakończeniem zadania w sytuacji wypełnienia protokołu zadań;
  2. Jeśli protokół zadania zawinie się, drukuje nadpisywane komunikaty do pliku bufora, aby nie utracić żadnych ważnych komunikatów.
- W systemach z niezależnymi pulami ASP: przed włączeniem niezależnej puli ASP do systemu OS/400 V5R2, należy użyć funkcji V5R1 \*ESTIMATE na wszystkich niezależnych pulach ASP. Zapewnia to oszacowanie czasu pierwszego podłączenia niezależnej puli ASP po instalacji. Więcej informacji znajduje się w sekcji Niezależne pule pamięci dyskowej (ASP).

## Przetwarzanie konwersji

Komenda CVTDIR konwertuje katalogi \*TYPE1 do katalogów \*TYPE2. Podczas przeprowadzania konwersji, należy wziąć pod uwagę następujące zagadnienia:

- Konwersja do systemu plików "root" (/) lub QOpenSys
- Konwersja do systemów plików zdefiniowanych przez użytkownika
- Tworzenie profili użytkowników
- Zmiana nazw obiektów
- Uwagi na temat profilu użytkownika

### Konwersja do systemu plików "root" lub QOpenSys

Podczas konwersji do systemów plików "root" lub QOpenSys, system **musi** być w stanie zastrzeżonym. Nie można używać żadnego z systemów. Wszystkie systemy plików UDFS i NFS są odłączane przez komendę CVTDIR i nie są ponownie podłączane po zakończeniu konwersji. Komenda Podłączanie (MOUNT) może być użyta do ponownego podłączenia systemu plików UDFS lub NFS.

### Konwersja do systemów plików zdefiniowanych przez użytkownika

Konwertowane systemy plików UDFS w pulach ASP 1-32 nie są dostępne dla użytkowników. Dla każdej puli ASP, w katalogu /dev jest tworzony katalog QASPxx. Komenda CVTDIR usuwa oba katalogi QASPxx z przestrzeni nazw, aby uniemożliwić użytkownikom dostęp do systemów plików UDFS w pulach ASP. Gdy komenda CVTDIR zakończy przetwarzanie wszystkich obiektów, w tym katalogów QASPxx, obiekty są ponownie umieszczane w przestrzeni nazw i są dostępne dla użytkowników w systemie. Systemy plików UDFS dla pul ASP są odłączane przez komendę CVTDIR i nie są ponownie podłączane po zakończeniu konwersji. Aby ponownie podłączyć system plików UDFS, można użyć komendy Podłączanie (MOUNT).

**Uwaga:** Katalog QASP01 znajduje się w każdym systemie.


### Tworzenie profili użytkowników

Ta funkcja konwersji tworzy profile użytkowników, które są wykorzystywane podczas konwersji. Te profile użytkowników mają nazwy w postaci QP0FCVxxxx, gdzie xxxx to liczba, na przykład 0001. Są one wykorzystywane przez funkcję konwersji w taki sposób, że stają się właścicielami katalogów w konwertowanym systemie plików, na wypadek, gdyby pierwotny właściciel nie mógł być właścicielem swoich katalogów.

Te profile użytkowników są usuwane po zakończeniu konwersji, jeśli jest to możliwe. Jeśli jeden z tych profili jest właścicielem jakiegoś katalogu, jest wysyłany komunikat CPIA08B.

### **Zmiana nazw obiektów**

Katalogi \*TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16. Katalogi \*TYPE1 mają nazwy UCS2 Poziom 1. Dlatego podczas konwersji katalogów mogą pojawiać się nieprawidłowe lub zduplikowane nazwy. Jeśli system wykryje taką nazwę, zmienia ją na unikalną, a do protokołu zadania wysyła komunikat CPIA08A zawierający oryginalną nazwę i nową nazwę. Znaki złożone lub nieprawidłowe pary znaków odpowiedników znajdujące się w nazwie mogą spowodować zmianę nazwy obiektu.

Więcej informacji na temat 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. Na przykład są to znaki akcentowane lub umlaut. Te znaki muszą być zmienione lub znormalizowane do wspólnego formatu zanim zostaną umieszczone w katalogu, 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 odpowiedniki:** 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 profilu użytkownika**

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 należąca do profilu użytkownika i liczbę pozycji w profilu użytkownika.

**Zmiana maksymalnej ilości pamięci dla profilu użytkownika:** Podczas konwersji katalogu, pewna liczba katalogów istnieje w obu formatach jednocześnie i ich właścicielem jest ten sam profil użytkownika. Jeśli katalog \*TYPE2 nie może być utworzony z powodu osiągnięcia przez profil użytkownika maksymalnego limitu pamięci, limit dla profilu użytkownika jest zwiększany. Do protokołu zadania jest wysyłany komunikat CPIA08C i konwersja jest kontynuowana.

**Zmiana właściciela katalogu:** Jeśli profil użytkownika jest właścicielem katalogu \*TYPE1 nie może być właścicielem tworzonego katalogu \*TYPE2, jako właściciel katalogu \*TYPE2 jest ustawiany jeden z dodatkowych profili użytkowników opisanych w sekcji Tworzenie profili użytkowników. Do protokołu zadania wysyłany jest komunikat CPIA08B i konwersja jest kontynuowana.

### **Przykład: konwersja wszystkich systemów plików (mała liczba obiektów)**

W systemie A jest skonfigurowanych pięć pul ASP: 1 (systemowa pula ASP), 3, 5, 11 i 25. Żaden z tych systemów plików nie został przekonwertowany z typu \*TYPE1 do \*TYPE2 w systemie. Użytkownik może przekonwertować wszystkie te systemy plików. W systemie plików nie ma dużej liczby obiektów, więc wszystkie kroki można wykonać jednego dnia.

Aby przekonwertować katalogi zawierające małą liczbę obiektów we wszystkich systemach plików:



1. Wprowadź system w stan zastrzeżony.
2. W wierszu komend wpisz RCLSTG SELECT(\*ALL).
3. Zachowaj system za pomocą menu Składowanie (Save). W wierszu komend wpisz GO SAVE i wybierz opcję 21.
4. W wierszu komend wpisz CVTDIR OPTION(\*ESTIMATE) FILESYS(\*ALL) FORMAT(\*TYPE2).
5. Za pomocą funkcji \*ESTIMATE sprawdź, czy nie ma komunikatów o błędach.
6. Sprawdź, czy wszystkie pule ASP mają wystarczająco dużo pamięci dyskowej.
7. W wierszu komend wpisz CVTDIR OPTION(\*CONVERT) FILESYS(\*ALL) FORMAT(\*TYPE2).

**Uwaga:** Podczas konwertowania wszystkich systemów plików (\*ALL), wyświetlany jest komunikat CPAA084 i użytkownik musi potwierdzić, czy wymienione systemy plików mają być konwertowane.

8. Za pomocą funkcji \*CONVERT sprawdź, czy nie ma komunikatów o błędach.
9. Wyprowadź system ze stanu zastrzeżonego.

## Przykład: konwersja wszystkich systemów plików (duża liczba obiektów)

W systemie B jest skonfigurowanych pięć pul pamięci dyskowych (ASP): 1 (systemowa pula ASP), 3, 5, 11 i 25. Żaden z tych systemów plików nie został przekonwertowany z typu \*TYPE1 do \*TYPE2 w systemie. Użytkownik może przekonwertować wszystkie te systemy plików. Systemy plików zawierają dużą liczbę obiektów, więc operację tę można podzielić na dwa kroki, wykonywane podczas dwóch weekendów.

### Weekend pierwszy:

1. Wprowadź system w stan zastrzeżony.
2. W wierszu komend wpisz RCLSTG SELECT(\*ALL).
3. Zachowaj system za pomocą menu Składowanie (Save). W wierszu komend wpisz GO SAVE i wybierz opcję 21.
4. Wyprowadź system ze stanu zastrzeżonego.

### W tygodniu:

5. W wierszu komend wpisz CVTDIR OPTION(\*ESTIMATE) FILESYS(\*ALL) FORMAT(\*TYPE2).
6. Za pomocą funkcji \*ESTIMATE sprawdź, czy nie ma komunikatów o błędach.
7. Sprawdź, czy wszystkie pule ASP mają wystarczająco dużo pamięci dyskowej.

### Weekend drugi:

8. Wprowadź system w stan zastrzeżony.
9. W wierszu komend wpisz CVTDIR OPTION(\*CONVERT) FILESYS(\*ALL) FORMAT(\*TYPE2).

**Uwaga:** Podczas konwertowania wszystkich systemów plików (\*ALL), wyświetlany jest komunikat CPAA084 i użytkownik musi potwierdzić, czy wymienione systemy plików mają być konwertowane.

10. Za pomocą funkcji \*CONVERT sprawdź, czy nie ma komunikatów o błędach.
11. Wyprowadź system ze stanu zastrzeżonego.

## Przykład: konwersja wybranych pul ASP

W systemie A jest skonfigurowanych sześć pul pamięci dyskowych (ASP): 1 (systemowa pula ASP), 2, 4, 8, 10 i 30. Żaden z tych systemów plików nie został przekonwertowany w systemie. Należy przekonwertować jedynie systemy plików UDFS w pulach ASP 4, 10 i 30.

Aby przekonwertować katalogi w systemie plików UDFS w określonych pulach ASP:

1. Sprawdź format katalogu systemów plików. W wierszu komend wpisz CVTDIR OPTION(\*CHECK).
2. Wprowadź system w stan zastrzeżony.
3. W wierszu komend wpisz RCLSTG SELECT(\*ALL).
4. Zachowaj system za pomocą menu Składowanie (Save). W wierszu komend wpisz GO SAVE i wybierz opcję 21.
5. Wyprowadź system ze stanu zastrzeżonego.
6. W wierszu komend wpisz CVTDIR OPTION(\*ESTIMATE) FILESYS(\*UDFS) ASP(4) FORMAT(\*TYPE2).
7. Za pomocą funkcji \*ESTIMATE sprawdź, czy nie ma komunikatów o błędach.
8. W wierszu komend wpisz CVTDIR OPTION(\*ESTIMATE) FILESYS(\*UDFS) ASP(10) FORMAT(\*TYPE2).
9. Za pomocą funkcji \*ESTIMATE sprawdź, czy nie ma komunikatów o błędach.
10. W wierszu komend wpisz CVTDIR OPTION(\*ESTIMATE) FILESYS(\*UDFS) ASP(30) FORMAT(\*TYPE2).
11. Za pomocą funkcji \*ESTIMATE sprawdź, czy nie ma komunikatów o błędach.
12. Sprawdź, czy wszystkie pule ASP mają wystarczająco dużo pamięci dyskowej.
13. W wierszu komend wpisz CVTDIR OPTION(\*CONVERT) FILESYS(\*UDFS) ASP(4) FORMAT(\*TYPE2).
14. Za pomocą funkcji \*CONVERT sprawdź, czy nie ma komunikatów o błędach.
15. W wierszu komend wpisz CVTDIR OPTION(\*CONVERT) FILESYS(\*UDFS) ASP(10) FORMAT(\*TYPE2).
16. Za pomocą funkcji \*CONVERT sprawdź, czy nie ma komunikatów o błędach.
17. W wierszu komend wpisz CVTDIR OPTION(\*CONVERT) FILESYS(\*UDFS) ASP(30) FORMAT(\*TYPE2).
18. Za pomocą funkcji \*CONVERT sprawdź, czy nie ma komunikatów o błędach.

---

## Nazwa ścieżki

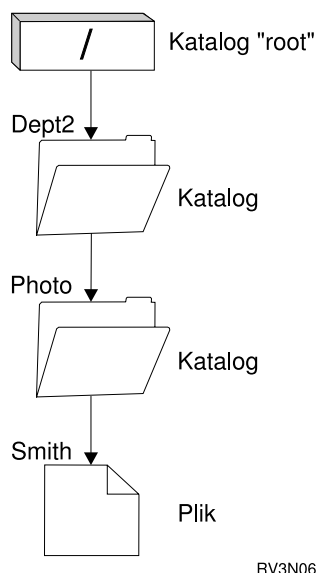
**Nazwa ścieżki (path name)** informuje serwer, jak znaleźć dany obiekt. Jest to sekwencja nazw katalogów zakończona nazwą obiektu. Poszczególne nazwy katalogów i obiektów rozdziela znak ukośnika (/); na przykład:

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.

## Dowiązanie

**Dowiązanie (link)** jest nazwą połączenia pomiędzy 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 serwer. W rzeczywistości obiekt identyfikuje się przy pomocy ścieżki katalogów. Czasami można mieć dostęp do obiektu poprzez podanie tylko jego "nazwy". Jest to możliwe tylko 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 wspólnie używać pliku, gdy każdy z nich ma dowiązanie do tego pliku ze swojego katalogu osobistego (patrz sekcja "Katalog bieżący i katalog osobisty" na stronie 7). Pewne typy dowiązań mogą przekraczać granice systemów plików i mogą istnieć bez obiektów.

Istnieją dwa typy dowiązań: **stałe** i **symboliczne**.

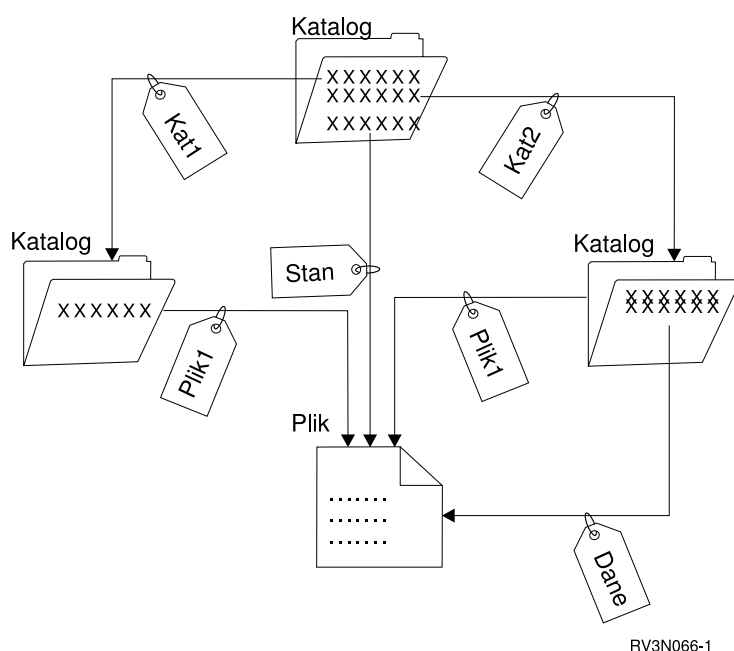
Informacje o dowiązaniach zawierają następujące sekcje zamieszczone w dalszej części tego dokumentu:

- Dowiązanie stałe
- Dowiązanie symboliczne
- Porównanie: Dowiązania stałe i symboliczne

## 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 pomiędzy 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 7. 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 serwera, 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 zostanie usunięty z serwera. Obiektu nie można otworzyć po usunięciu ostatniego dowiązania stałego.

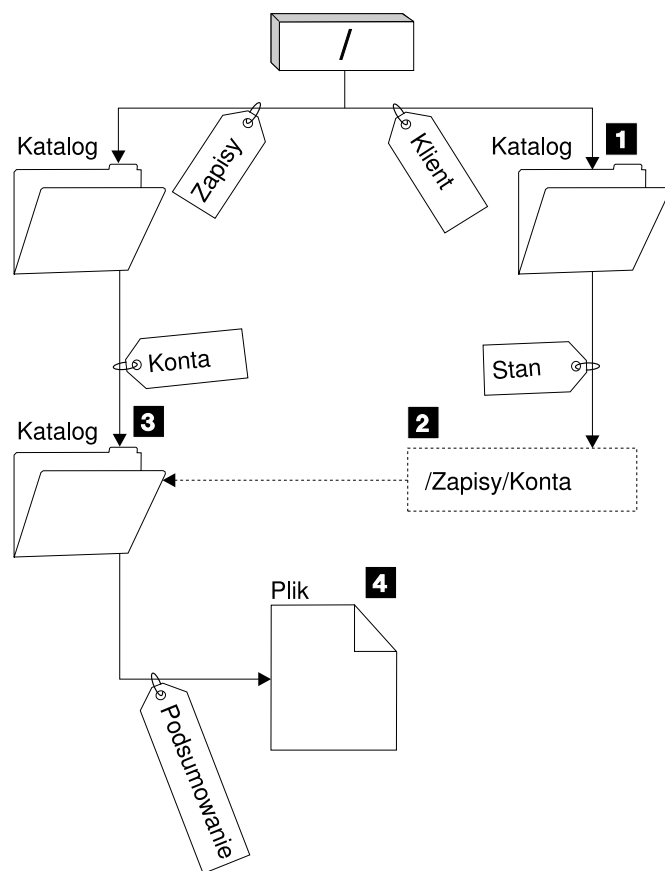
Pojęcie dowiązania stałego, z pewnymi ograniczeniami, może się także odnosić do systemu plików QSYS.LIB lub QSYS.LIB niezależnej puli ASP i do systemu plików obsługującego obiekty biblioteki dokumentów (QDLS). Tak więc biblioteka ma jedno dowiązanie stałe do każdego zawartego w niej obiektu. Podobnie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

Dowiązania stałego nie można zakładać pomiędzy systemami plików. Na przykład katalog w systemie plików QOpenSys nie może mieć dowiązania stałego do obiektu w systemie plików QSYS.LIB lub w niezależnej puli ASP QSYS.LIB 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 8. Przykład zastosowanie 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. Przykładowo można użyć dowiązania symbolicznego, aby dostarczyć ścieżkę do pliku, który ma zostać dodany lub zastąpiony później.

Ponadto, inaczej niż dowiązanie stałe, dowiązanie symboliczne może przekraczać granice systemów plików. Na przykład użytkownik pracujący w jednym systemie plików może używać dowiązania symbolicznego, aby uzyskać dostęp do innego systemu plików. Chociaż systemy plików QSYS.LIB, QSYS.LIB niezależnej puli ASP i QDLS nie dopuszczają tworzenia i przechowywania dowiązań symbolicznych, w systemie plików "root" (/) lub QOpenSys można tworzyć dowiązania symboliczne, które umożliwią:

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

Patrz także sekcja "Porównanie: dowiązania stałe i symboliczne".

## Porównanie: dowiązania stałe i symboliczne

Korzystając z nazw ścieżek w programach, można używać dowiązań stałych lub dowiązań symbolicznych (patrz "Dowiązanie" na stronie 17). 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

Aspekt	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 dynamiczne (atributy zmieniające się)	Wolniejsze. W każdym dowiązaniu stałym jest przechowywanych wiele atrybutów obiektu. Zmiany w obiektach dynamicznych są więc wolniejsze w miarę wzrostu liczby stałych dowiązań do obiektu.	Szybsze. Dowiązania symboliczne nie wpływają na zmiany w obiektach dynamicznych.
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.
Zasięg	Ograniczony. Dowiązania stałe nie mogą przekraczać granic systemów plików.	Nieograniczony. Dowiązania symboliczne mogą przekraczać granice systemów plików.

---

## Atrybuty rozszerzone

Atrybut rozszerzony (extended attribute) 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" (/), systemach otwartych (QOpenSys) i UDFS katalogi, pliki strumieniowe i dowiązania symboliczne mogą mieć atrybuty rozszerzone dowolnego typu. Inne obiekty natomiast mogą wcale nie posiadać atrybutów rozszerzonych.

Do wyświetlenia atrybutu rozszerzonego .SUBJECT dla danego obiektu można użyć komendy Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK). 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). Więcej informacji na temat tych systemów plików znajduje się w sekcjach "System plików obsługujący obiekty biblioteki dokumentów (QDLS)" na stronie 72 i "System plików nośników optycznych (QOPT)" na stronie 74.

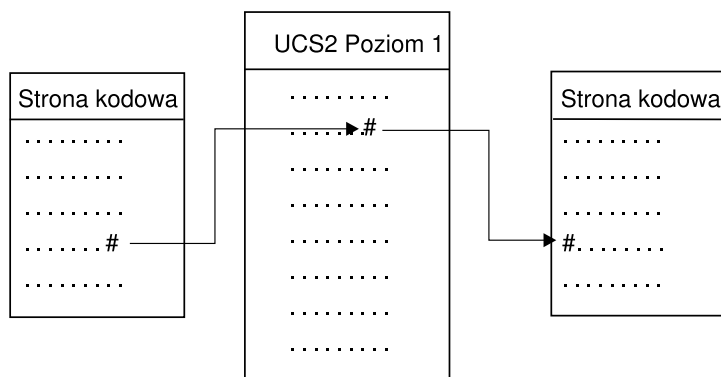
Jeśli klient PC jest połączony z serwerem iSeries poprzez OS/2 lub Windows, wówczas do sprawdzenia i ustawienia atrybutów rozszerzonych dowolnego obiektu typu plik można użyć interfejsów programistycznych odpowiedniego systemu operacyjnego (takich jak DosQueryFileInfo i DosSetFileInfo). Użytkownicy systemu OS/2 mogą ponadto zmieniać atrybuty rozszerzone obiektu na pulpicie, używając notesu ustawień, to znaczy wybierając pozycję Ustawienia w menu wywoływanym tego obiektu.

Przy definiowaniu atrybutów rozszerzonych należy uwzględnić 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:

## Niezmienność nazw

Użytkownicy "bazowego" (/) systemu plików, systemu plików QOpenSys i systemu plików użytkownika korzystają z funkcji systemowej zapewniającej niezmienność znaków w nazwach obiektów. W szczególności dotyczy to użycia tych systemów plików na serwerach iSeries i podłączonych urządzeniach o różnych schematach kodowania znaków (stronach kodowych). Serwer 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. Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi \*TYPE2. UCS2 Poziom 1 i UTF-16 są podzbiorami standardu ISO 10646. Gdy jest używana nazwa, system przekształca przechowywany format znaków do odpowiedniej reprezentacji znaków w aktualnie używanej stronie kodowej. Nazwy rozszerzonych atrybutów związanych z każdym obiektem traktowane są w taki sam sposób.



RV3N141-0

Rysunek 9. Utrzymywanie niezmienności znaków w różnych schematach kodowania

Obsługa ta ułatwia współdzielenie serwera z urządzeniami używającymi innych stron kodowych. Na przykład użytkownicy komputerów osobistych mogą mieć dostęp do zbioru serwera iSeries, korzystając z tej samej nazwy, nawet jeśli ich komputery osobiste nie używają tej samej strony kodowej, co serwer iSeries. Konwersja stron kodowych jest wykonywana automatycznie przez serwer. Oczywiście, urządzenie musi używać strony kodowej zawierającej znaki wykorzystywane w danej nazwie.



---

## Rozdział 3. Dostęp do zintegrowanego systemu plików poprzez tradycyjny interfejs systemu

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 serwera 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 jest również dostępny poprzez iSeries Navigator, graficzny interfejs użytkownika stosowany do zarządzania i administrowania serwerem z poziomu pulpitu Windows.

Istnieje kilka sposobów komunikowania się z zintegrowanym systemem plików:

### **Korzystanie z interfejsów API**

Interfejsy API wykonujące działania na katalogach i plikach strumieniowych zintegrowanego systemu plików mają formę funkcji w języku C.

### **Używanie komend CL**

Przy użyciu komend CL można wykonywać działania na zbiorach, plikach i innych obiektach w dowolnym systemie plików, który jest dostępny poprzez interfejs zintegrowanego systemu plików.

### **Używanie menu i ekranów iSeries**

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

### **Używanie programu iSeries Navigator**

iSeries Navigator to graficzny interfejs użytkownika służący do zarządzania i administrowania serwerami z poziomu pulpitu Windows.

### **Używanie komputera PC**

Jeśli klient PC jest podłączony do serwera iSeries, może pracować z katalogami i obiektami zintegrowanego systemu plików tak, jakby były one przechowywane na tym komputerze PC.

---

## Wykonywanie operacji za pomocą menu i ekranów serwera iSeries

W zintegrowanym systemie plików można wykonywać operacje na zbiorach i innych obiektach, używając zestawu menu i ekranów dostępnych na serwerze. Aby wyświetlić menu zintegrowanego systemu plików:

1. Wpisz się do serwera.
2. Naciśnij **Enter**, aby kontynuować.
3. Wybierz opcję **Zbiory, biblioteki i foldery (Files, Libraries, and Folders)** z menu głównego iSeries.
4. Wybierz opcję **Zintegrowany system plików (Integrated File System)** z menu Zbiory, biblioteki i foldery (Files, Libraries, and Folders).

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 serwera 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 (Integrated File System).

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, na 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;
- kopiować dane pomiędzy 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. Ograniczenia dotyczące poszczególnych systemów plików opisano w sekcji "System plików w zintegrowanym systemie plików" na stronie 4.

Więcej informacji o menu i ekranach zintegrowanego systemu plików można znaleźć w następującej sekcji:

- Reguły dotyczące nazw ścieżek dla komend CL i ekranów

---

## Wykonywanie operacji za pomocą komend CL

Wszystkie operacje, które można wykonywać poprzez menu zintegrowanego systemu plików (patrz sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23), można zrealizować używając komend 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 na temat komend języka CL, szczególnie tych, które dotyczą systemów plików definiowanych przez użytkownika (UDFS), sieciowego systemu plików (NFS) i, ogólnie, dołączanych (mounted) systemów plików, zawierają sekcje "System plików definiowany przez użytkownika (UDFS)" na stronie 63, "Sieciowy system plików (Network File System - NFS)" na stronie 85. Gdy komenda wykonuje te same operacje co komenda OS/2 lub DOS, dla wygody użytkowników systemów OS/2 i DOS w tabeli podano alternatywną nazwę komendy (alias).

*Tabela 2. Komendy zintegrowanego systemu plików*

Komenda	Opis	Alias
ADDLNK	Dodanie dowiązania (Add Link). Dodaje dowiązanie pomiędzy katalogiem i obiektem.	

Tabela 2. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
ADDMFS	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.	MOUNT
APYJRNCHG <sup>2</sup>	Zastosowanie kronikowanych zmian (Apply Journaled Changes). Korzysta z pozycji kronik do wprowadzenia zmian, które wystąpiły od ostatniego składowania obiektu kronikowanego lub do wprowadzenia zmian do określonego punktu.	
CHGATR	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	Zmiana wartości kontroli (Change Auditing Value). Włącza lub wyłącza kontrolę dla obiektu.	
CHGAUT	Zmiana uprawnień (Change Authority). Nadaje konkretne uprawnienia do obiektu użytkownikowi lub grupie użytkowników.	
CHGCURDIR	Zmiana bieżącego katalogu (Change Current Directory). Zmienia katalog, który ma być używany jako bieżący.	CD, CHDIR
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	Zmiana właściciela (Change Owner). Zmienia właściciela obiektu.	
CHGPGP	Zmiana grupy podstawowej (Change Primary Group). Zmienia grupę podstawową z jednego użytkownika na innego.	
CHKIN	Odblokowanie (Check In). Odblokowuje obiekt, który był uprzednio zablokowany.	
CHKOUT	Zablokowanie (Check Out). Blokuje obiekt, co uniemożliwia innym użytkownikom dokonywanie w nim zmian.	
CPY	Kopiowanie (Copy). Kopiuje pojedynczy obiekt lub grupę obiektów.	COPY
CPYFRMSTMF	Kopiowanie z pliku strumieniowego (Copy from Stream File). Kopiuje dane z pliku strumieniowego do podzbioru bazy danych.	
CPYTOSTMF	Kopiowanie do pliku strumieniowego (Copy to Stream File). Kopiuje dane z podzbioru bazy danych do pliku strumieniowego.	
CRTDIR	Tworzenie katalogu (Create Directory). Dodaje nowy katalog do systemu.	MD, MKDIR
CRTUDFS	Tworzenie systemu plików UDFS (Create UDFS). Tworzy system plików definiowany przez użytkownika (UDFS).	
CVTDIR	Konwersja katalogu (Convert directory). Podaje informacje o konwertowaniu katalogów zintegrowanego systemu plików z formatu *TYPE1 do formatu *TYPE2 lub wykonuje konwersji.	
CVTRPCSRC	Konwersja kodu źródłowego RPC (Convert RPC Source). Generuje kod w języku C na podstawie pliku wejściowego napisanego w języku RPC.	RPCGEN
DLTUDFS	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.	

Tabela 2. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
DSPCURDIR	Wyświetlenie bieżącego katalogu (Display Current Directory). Wyświetla nazwę bieżącego katalogu.	
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</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	Przeniesienie (Move). Przenosi obiekt do innego katalogu.	MOVE
RLSIFSLCK	Zwolnienie blokad zintegrowanego systemu plików (Release in Integrated File System Locks). Zwalnia wszystkie blokady, na poziomie bajtów, sieciowego systemu plików nałożone przez klienta lub na obiekt.	
RMVDIR	Usuwanie katalogu (Remove Directory). Usuwa katalog z systemu.	RD, RMDIR
RMVLNK	Usuwanie dowiązania (Remove Link). Usuwa dowiązanie do obiektu.	DEL, ERASE
RMVMFS	Usuwanie podłączonego FSS (Remove Mounted File System). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnych katalogów klienta.	UNMOUNT
RNM	Zmiana nazwy (Rename). Zmienia nazwę obiektu w katalogu.	REN
RPCBIND	Uruchomienie demona konsolidacji RPC (Start RPC Binder Daemon). Uruchamia demon RPCBind.	
RST	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).	
SAV	Składowanie (Save). Kopiuje obiekt lub grupę obiektów z systemu do urządzenia składowania.	
SNDJRNE <sup>2</sup>	Wysłanie pozycji do kroniki (Send Journal Entry). Dodaje do dziennika pozycje kroniki użytkownika opcjonalnie powiązane z kronikowanym obiektem.	
STRJRN <sup>2</sup>	Uruchomienie kronikowania (Start Journal). Uruchamia kronikowanie zmian (dotyczących obiektu lub listy obiektów w określonej kronice).	

Tabela 2. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
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 uprawnień oraz udostępnia opcje umożliwiające dodawanie użytkowników, zmianę uprawnień użytkowników lub usuwanie użytkowników.	
WRKLNK	Praca z dowiązaniem obiektów (Work with Object Links). Wyświetla listę obiektów w katalogu i udostępnia opcje umożliwiające wykonywanie działań na tych obiektach.	
WRKOBJOWN <sup>1</sup>	Praca z obiektami według właścicieli (Work with Objects by Owner). Wyświetla listę obiektów, których właścicielem jest dany profil użytkownika i udostępnia opcje umożliwiające wykonywanie działań na tych obiektach.	
WRKOBJPGP <sup>1</sup>	Praca z obiektami według grupy głównej (Work with Objects by Primary Group). Wyświetla listę obiektów kontrolowanych przez grupę główną, zawiera opcje umożliwiające wykonywanie działań na tych obiektach.	

**Uwaga:**

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 można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries.

Informacje na temat komend CD zintegrowanego systemu plików i restrykcji dotyczących używania tych komend w określonych systemach plików można znaleźć w następujących sekcjach:

- Systemy plików w zintegrowanym systemie plików
- Reguły dotyczące nazw ścieżek dla komend CL i ekranów
- Sekcja CL w Centrum informacyjnym iSeries

## Reguły dotyczące nazw ścieżek dla komend CL i ekranów

Używając do pracy z obiektem komend lub ekranów zintegrowanego systemu plików, użytkownik identyfikuje obiekt podając ścieżkę do niego. Poniżej przedstawiono zasady, o których należy pamiętać w trakcie podawania ścieżek. Pojęcie **obiekt** oznacza w nich katalog, plik, dowiązanie lub inny obiekt.

- Nazwy obiektów wewnątrz każdego katalogu muszą być unikalne.
- Nazwa ścieżki przekazana do komendy CL zintegrowanego systemu plików musi być przedstawiona za pomocą identyfikatora 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.
- W wierszu komend ścieżki muszą się znajdować w apostrofach ('). 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 ścieżek są rozdzielane ukośnikiem (/) lub ukośnikiem odwrotnym (\), na przykład:

'Kat1/Kat2/Kat3/ZbiórUż'

lub

```
'Kat1\Kat2\Kat3\ZbiórUż'
```

- Znaków / lub \ i znaków pustych nie można używać w poszczególnych składnikach ścieżki (ponieważ znaki / i \ są używane jako separatory). 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.

Sekcja Systemy plików w zintegrowanym systemie plików zawiera limity ścieżek w każdym systemie plików.

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

```
'/Kat1/Kat2/Kat3/ZbiórUż'
```

- Jeśli ścieżka nie rozpoczyna się znakiem / lub \, 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 (~) z ukośnikiem (lub ukośnikiem odwrotnym) na początku ścieżki oznacza, że ścieżka rozpoczyna się w katalogu osobistym użytkownika, który wprowadził komendę, na przykład:

```
'~/KatUżytk/ObiektUżytk'
```

- Znak tyldy (~), po którym następuje nazwa użytkownika a następnie ukośnik (lub ukośnik odwrotny), 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 w

```
'/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 serwera iSeries, 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'
```

Sekcja "System plików bibliotek (QSYS.LIB)" na stronie 67 zawiera więcej szczegółów na ten temat.

- 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
```

Więcej szczegółów zawiera sekcja "System plików QSYS.LIB niezależnej ASP" na stronie 70.

- Ścieżka musi znajdować się w dodatkowych apostrofach (') lub w cudzysłowie ("), jeśli użyto jednego z poniższych znaków:
  - gwiazdka (\*),
  - znak zapytania (?),
  - apostrof ('),
  - 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'''
```

Działanie takie nie jest zalecane, ponieważ można pomylić znaczenie znaku w ciągu komend i wpisać ciąg komend błędnie.

- 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:

```
crtmdir 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 (zobacz "Wykonywanie operacji za pomocą funkcji API" na stronie 45).

Informacje o ograniczeniach dotyczących używania określonych komend można znaleźć w pomocy do komendy lub w sekcji Język CL w Centrum informacyjnym iSeries.

---

## Wykonywanie operacji za pomocą komputera PC

Jeśli klient PC jest podłączony do serwera iSeries, może pracować z katalogami i obiektami zintegrowanego systemu plików tak, jakby były one przechowywane na tym komputerze PC. Korzystając z metody "przeciągnij i upuść" programu Eksploracja w systemie Windows można kopiować obiekty między katalogami. Jeśli jest to konieczne, można nawet skopiować obiekt z serwera do komputera PC, zaznaczając obiekt na dysku serwera i przeciągając go na dysk komputera PC.

Każdy obiekt serwera iSeries kopiowany do klienta PC może zostać automatycznie poddany konwersji z kodu EBCDIC na kod ASCII. Kod EBCDIC (Extended Binary-Coded Decimal Interchange Code) jest stosowany przez IBM, natomiast kod ASCII (American National Standard Code for Information Interchange) przez większość innych firm. Program iSeries Access może zostać tak skonfigurowany, aby wykonywał konwersję automatycznie, lub też aby wykonywał konwersję na plikach z konkretnymi rozszerzeniami. Począwszy od wersji OS/400 V4R4, iSeries NetServer można również tak skonfigurować, aby automatycznie wykonywał konwersję plików.

W zależności od typu obiektu, użytkownik może używać do pracy z nim interfejsów i aplikacji komputera PC. Na przykład do edycji pliku strumieniowego zawierającego tekst można używać edytora komputera PC.

Po podłączeniu klienta PC do serwera iSeries, zintegrowany system plików udostępnia katalogi i obiekty serwera na kliencie PC. Użytkownicy PC mogą pracować z plikami w zintegrowanym systemie plików za pomocą klientów umożliwiających współużytkowanie plików wbudowanych w system operacyjny Windows, klienta FTP lub programu iSeries Navigator (części programu iSeries Access). Komputer PC wykorzystuje klientów umożliwiających współużytkowanie plików do uzyskania dostępu do serwera iSeries NetServer, który uruchomiony jest na serwerze iSeries.

#### **Przesyłanie plików za pomocą protokołu FTP**

Klient FTP umożliwia przesyłanie plików znalezionych na serwerze iSeries również w systemach plików "root" (/), QSYS.LIB, QSYS.LIB niezależnej puli ASP, QOpenSys, QOPT oraz QFileSvr.400. Pozwala także na przesyłanie folderów i dokumentów w systemie plików usługi biblioteki dokumentów (QDLS).

#### **Praca z plikami za pomocą programu iSeries Navigator**

iSeries Access zawiera iSeries Navigator, który łączy się z serwerem iSeries i sprawia, że zintegrowany system plików jest dostępny dla komputera PC. iSeries Navigator to graficzny interfejs użytkownika służący do zarządzania i administrowania serwerem iSeries z poziomu pulpitu Windows.

#### **Praca z plikami za pomocą iSeries NetServer**

iSeries NetServer jest to część systemu OS/400, która umożliwia współpracę serwera z wbudowanymi w Windows funkcjami udostępniania plików i drukarek.

**Uwaga:** W nowej wersji iSeries Access, dostęp do zintegrowanego systemu plików opiera się wyłącznie na programie NetServer. Obsługa NetServer jest możliwa tylko dla połączeń TCP/IP z serwerem iSeries na którym działa system OS/400 w wersji V4R2 i nowszej.

## **Przesyłanie plików za pomocą protokołu FTP**

Klient FTP umożliwia przesyłanie plików znalezionych w serwerze iSeries również w systemach plików "root" (/), QOpenSys, QSYS.LIB, QSYS.LIB niezależnej puli ASP, QOPT oraz QFileSvr.400. Pozwala także na przesyłanie folderów i dokumentów w systemie plików usługi biblioteki dokumentów (QDLS). Klienta FTP można uruchomić interaktywnie w trybie nienadzorowanego zadania wsadowego, w którym komendy są czytane z pliku, a odpowiedzi na nie zapisywane do pliku. Zapewnia on także inne funkcje umożliwiające przetwarzanie plików na serwerze, na którym działa.

Obsługę protokołu FTP można używać 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 obsługujący obiekty biblioteki dokumentów (QDLS),
- system plików nośników optycznych (QOPT),
- sieciowy system plików (Network File System - NFS),
- system plików NetWare (QNetWare),
- system plików serwera Windows NT (QNTC).

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 na temat protokołu FTP można znaleźć w kategorii **Sieć** w Centrum informacyjnym iSeries:

- FTP
- Przesyłanie plików za pomocą FTP

## Praca z plikami za pomocą iSeries NetServer

iSeries Support for Windows Network Neighborhood (iSeries NetServer) jest funkcją systemu operacyjnego IBM Operating System/400 wersja 5 (OS/400), która umożliwia klientom Windows dostęp do współużytkowanych katalogów i kolejek wyjściowych systemu OS/400. iSeries NetServer umożliwia jednolity dostęp do danych i drukarek systemu iSeries komputerom PC z oprogramowaniem Windows. Sieciowe programy klientów PC używają wbudowanych w ich systemy operacyjne funkcji współużytkowania zasobów plikowych i drukarkowych. Oznacza to, że aby skorzystać z iSeries NetServer, nie trzeba na komputerze PC instalować żadnego dodatkowego oprogramowania.

Klienci LINUX z zainstalowanym oprogramowaniem klienckim Samba również mogą uzyskać dostęp do danych i drukarek za pomocą iSeries NetServer. Systemy plików Samba (smbfs) mogą być podłączone z iSeries NetServer w podobny sposób, w jaki podłączane są systemy plików NFS z iSeries. Więcej informacji można znaleźć w sekcji iSeries NetServer w Centrum informacyjnym iSeries.

Współużytkowany zasób plikowy iSeries NetServer jest ścieżką do katalogu, którą iSeries NetServer udostępnia klientom PC w sieci iSeries. Współużytkowany zasób plikowy może składać się z dowolnych katalogów zintegrowanego systemu plików w iSeries. Zanim będzie można pracować z współużytkowaniem plików za pomocą iSeries NetServer, należy utworzyć współużytkowany zasób iSeries i, jeśli jest to konieczne, zmienić współużytkowany zasób iSeries NetServer za pomocą iSeries Navigator.

Aby uzyskać dostęp do współużytkowanych zasobów plikowych zintegrowanego systemu plików za pomocą iSeries NetServer:

1. Kliknij prawym przyciskiem myszy **Start** i wybierz **Eksploruj**, aby otworzyć Eksplorator Windows w komputerze PC.
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 iSeries NetServer. Na przykład można użyć następującej składni: **\\QSYSTEM1\Sharename**

**Uwaga:** QSYSTEM1 jest nazwą serwera iSeries NetServer, a Sharename to nazwa współużytkowanego zasobu plikowego, z którego chcesz skorzystać.

5. Kliknij OK.

**Uwaga:** Jeśli korzysta się z iSeries NetServer, nazwa serwera może różnić się od nazwy używanej przez program iSeries Access. Na przykład nazwą iSeries NetServer może być QAS400X, a ścieżka umożliwiająca pracę z plikami może mieć postać:

\\QAS400X\QDLS\MÓJFOLDER.FLR\MÓJPLIK.DOC. Jednak nazwą iSeries Access może być AS400X, a ścieżka umożliwiająca pracę z plikami może mieć postać  
\\AS400X\QDLS\MÓJFOLDER.FLR\MÓJPLIK.DOC.

Za pomocą iSeries NetServer można wybrać, które katalogi mają być współużytkowane w sieci. Katalogi te są wyświetlane na pierwszym poziomie poniżej nazwy systemu. Na przykład, jeśli administrator wybierze do współużytkowania katalog /home/fred z nazwą fredmdir, użytkownik będzie miał dostęp z komputera PC do tego katalogu poprzez nazwę \\QAS400X\FREDSDIR lub z klienta systemu Linux poprzez nazwę //qas400x/fredmdir.

Znacznie większą wydajność udostępniania plików dla klienta PC w porównaniu z innymi systemami plików iSeries zapewnia bazowy system plików "root" (/). Użytkownik może przenosić pliki z bazowego systemu plików "root" (/). Więcej informacji zawiera sekcja Uwagi dotyczące przenoszenia obiektów do innego systemu plików.

Więcej informacji na temat iSeries NetServer i współużytkowanych zasobów plikowych można znaleźć w następujących sekcjach w kategorii **Sieć** w Centrum informacyjnym iSeries:

- iSeries NetServer
- Współużytkowane zasoby plikowe iSeries NetServer
- Dostęp do współużytkowanych zasobów plikowych iSeries NetServer z klienta PC Windows

---

## Przenoszenie obiektów do innego systemu plików

Przed użyciem zintegrowanego systemu plików do przenoszenia obiektów między systemami plików należy przeczytać sekcję "Uwagi dotyczące przenoszenia obiektów do innego systemu plików".

Aby przenieść obiekty do innego systemu plików, 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 (Move - 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 (Copy - 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. Najbezpieczniejszym sposobem przywrócenia obiektu jest odtworzenie jego zeszkładowanej wersji.

## Uwagi dotyczące przenoszenia obiektów 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 zintegrowanego systemu plików i ich charakterystykami. Więcej informacji zawiera sekcja "System plików w zintegrowanym systemie plików" na stronie 4.

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). Obsługują również funkcje API i komendy, które służą do pracy z dokumentami i folderami. Nie można używać tych interfejsów w przypadku obiektów znajdujących się w innych systemach plików.

- Jak 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. System QDLS, w przeciwieństwie do innych systemów, obsługuje wiele atrybutów aplikacji Office. Dlatego dokumenty Office lepiej przechowywać w systemie QDLS.

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, QNetWare i QNTC. Ponieważ systemy te mają wiele cech systemu plików OS/2, zapewniają one szybszy dostęp do plików.

---

## Katalogi udostępnione w zintegrowanym systemie plików

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, następujące katalogi:

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

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

Katalogiem /tmp i zawartymi w nim plikami 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 plikami funkcji API, które obsługują zintegrowany system plików (patrz sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45). Na przykład program użytkowy może usunąć plik z katalogu /tmp używając funkcji API unlink().

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

**/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. Więcej informacji znajduje się w sekcji "Katalog bieżący i katalog osobisty" na stronie 7.

**/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 wykorzystywanym do przechowywania danych dotyczących 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 dla systemu plików QOpenSys i jest wykorzystywany do przechowywania danych dotyczących 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.

---

## Rozdział 4. Dostęp do zintegrowanego systemu plików za pomocą programu iSeries Navigator

iSeries Navigator to graficzny interfejs użytkownika służący do zarządzania i administrowania systemami z poziomu pulpitu Windows. iSeries Navigator ułatwia eksploatację systemu i zarządzanie nim oraz przyczynia się do zwiększenia produktywności. Można na przykład skopiować profil użytkownika z jednego serwera iSeries do innego, przeciągając go myszą. Konfigurowanie ochrony i usług TCP/IP w aplikacjach odbywa się pod nadzorem kreatorów.

Korzystając z programu iSeries Navigator można wykonać wiele zadań. Poniżej wymieniono sekcje opisujące niektóre najczęściej wykonywane zadania systemu plików:

### Praca z plikami i folderami

- “Tworzenie folderu” na stronie 38
- “Usuwanie folderu” na stronie 38
- “Odblokowanie pliku”
- “Zablokowanie pliku” na stronie 36
- “Nadawanie uprawnień do pliku lub folderu” na stronie 36
- “Konfigurowanie konwersji plików tekstowych” na stronie 36
- “Wysyłanie pliku lub folderu do innego systemu” na stronie 37
- “Zmiana opcji dla definicji pakietu” na stronie 37
- “Planowanie daty i godziny wysyłki pliku lub folderu” na stronie 37

### Praca z współużytkowanymi zasobami plikowymi

- “Tworzenie współużytkowanego zasobu plikowego” na stronie 38
- “Zmiana współużytkowanego zasobu plikowego” na stronie 38

### Praca z systemami plików użytkownika

- “Tworzenie nowego systemu plików użytkownika” na stronie 39
- “Podłączanie systemu plików użytkownika” na stronie 39
- “Odłączanie systemu plików użytkownika” na stronie 40

### Obiekty kroniki

- “Uruchomienie kronikowania” na stronie 40
- “Zakończenie kronikowania” na stronie 40

---

## Odblokowanie pliku

Aby odblokować plik, wykonaj następujące czynności:

1. W programie **iSeries Navigator** kliknij prawym przyciskiem myszy plik, który chcesz zwrócić.
2. Wybierz **Właściwości**.
3. Wybierz **Właściwości pliku** → zakładka **Użyj**.
4. Kliknij **Sprawdź**.

---

## Zablokowanie pliku

Aby zablokować plik, wykonaj następujące czynności:

1. W programie **iSeries Navigator** kliknij prawym przyciskiem myszy plik, który chcesz pobrać.
2. Wybierz **Właściwości**.
3. Wybierz **Właściwości pliku** → zakładka **Użyj**.
4. Kliknij **Zwróć**.

---

## Nadawanie uprawnień do pliku lub folderu

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 nadać uprawnienia do pliku lub folderu, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system, którego chcesz użyć.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony obiekt, do którego chcesz nadać uprawnienia.
4. Kliknij obiekt prawym klawiszem myszy i wybierz **Uprawnienia**.
5. W oknie dialogowym **Uprawnienia** kliknij **Dodaj**.
6. W oknie dialogowym **Dodaj** wybierz przynajmniej jednego użytkownika lub grupę albo wpisz odpowiednią nazwę w polu nazwy użytkownika lub grupy.
7. Kliknij przycisk **OK**. Spowoduje to dodanie użytkowników lub grup i umieszczenie ich na początku listy.
8. Aby wprowadzić szczegółowe uprawnienia, kliknij przycisk **Dostosuj...**
9. Zastosuj odpowiednie uprawnienia dla użytkownika, zaznaczając właściwe pola wyboru.
10. Kliknij przycisk **OK**.

---

## Konfigurowanie konwersji plików tekstowych

W programie iSeries Navigator można ustawić automatyczną konwersję pliku tekstowego. Automatyczna konwersja pliku tekstowego umożliwia użycie rozszerzeń plików w przypadku konwersji plików z danymi. Zintegrowany system plików może przekonwertować plik danych, gdy jest on przenoszony pomiędzy serwerem iSeries a komputerem PC. W przypadku dostępu do pliku z komputera PC, jest on obsługiwany jak plik ASCII.

Aby skonfigurować konwersję plików tekstowych, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system, którego chcesz użyć.
2. Rozwiń gałąź **Systemy plików**.
3. Kliknij prawym klawiszem myszy **Zintegrowany system plików** i wybierz **Właściwości**.
4. W pole tekstowe **Rozszerzenia plików dla automatycznej konwersji pliku tekstowego** wpisz rozszerzenie nazw plików, które mają być automatycznie poddane konwersji i kliknij **Dodaj**.
5. Powtórz czynność 4 dla wszystkich rozszerzeń nazw plików, które chcesz automatycznie poddać konwersji.
6. Kliknij przycisk **OK**.

---

## Wysyłanie pliku lub folderu do innego systemu

Aby wysłać plik lub folder do innego systemu, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system, którego chcesz użyć.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony plik lub folder, który chcesz wysłać.
4. Kliknij plik lub folder prawym klawiszem myszy i wybierz **Wyślij**. Plik lub folder zostanie wyświetlony na liście **Wybrane pliki i foldery**, w oknie dialogowym **Wysyłanie plików z**.
5. Rozwiń listę dostępnych systemów i grup.
6. 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.
7. Aby wysłać plik lub folder z bieżącymi definicjami pakietu i informacjami harmonogramu, kliknij **OK**.

Informacje te można jednak zmienić, co zostało opisane w sekcjach “Zmiana opcji dla definicji pakietu” i “Planowanie daty i godziny wysyłki pliku lub folderu”.

Po utworzeniu definicji pakietu jest ona zapisywana i można ją ponownie wykorzystać do wysłania zdefiniowanego zestawu plików i folderów do kilku systemów końcowych lub grup systemów. Jeśli utworzony zostanie obraz plików, można zachować kilka wersji kopii tego samego zestawu plików. Wysyłanie obrazu zapewnia, że podczas dystrybucji pliki nie zostaną zaktualizowane, a więc ostatni system docelowy odbierze te same obiekty co pierwszy.

---

## Zmiana opcji dla definicji pakietu

Definiowanie pakietów umożliwia grupowanie zestawów obiektów OS/400 lub plików zintegrowanego systemu plików. Umożliwia też przeglądanie tej samej grupy plików traktowanej jako zestaw logiczny lub zestaw fizyczny. Pobierany jest obraz plików, aby uniemożliwić ich dalszą dystrybucję.

Aby zmienić opcje dla definicji pakietu, wykonaj następujące czynności:

1. Zakończ czynności opisane w sekcji “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 **OK**, aby wysłać plik lub kliknij **Harmonogram**, aby ustawić godzinę wysyłki pliku.

Tematy pokrewne:

- “Planowanie daty i godziny wysyłki pliku lub folderu”.

---

## Planowanie daty i godziny wysyłki pliku lub folderu

Wykorzystanie funkcji programu do planowania zadań zapewnia elastyczność i wykonywanie zadań w czasie najbardziej dogodnym dla użytkownika. Aby zaplanować datę i godzinę wysyłki pliku lub folderu:

1. Zakończ czynności opisane w sekcji “Wysyłanie pliku lub folderu do innego systemu”.
2. Kliknij **Harmonogram**.
3. Określ porę wysłania pliku lub folderu.

---

## Tworzenie folderu

Aby utworzyć folder, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system, którego chcesz użyć.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Kliknij prawym klawiszem myszy system plików, do którego chcesz dodać folder, i wybierz **Nowy folder**.
5. W oknie dialogowym **Nowy folder** wpisz nazwę nowego obiektu.
6. Kliknij przycisk **OK**.

Podczas tworzenia folderu na serwerze iSeries należy rozważyć, czy nowy folder (lub obiekt) ma być chroniony za pomocą zarządzania kronikami. Więcej informacji można znaleźć w sekcji Zarządzanie kronikami.

Tematy pokrewne:

- Uruchomienie kronikowania
- Zakończenie kronikowania

---

## Usuwanie folderu

Aby usunąć folder, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system, którego chcesz użyć.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony plik lub folder, który chcesz usunąć.
4. Kliknij plik lub folder prawym klawiszem myszy i wybierz **Usuń**.

---

## Tworzenie współużytkowanego zasobu plikowego

Współużytkowany zasób plikowy jest ścieżką do katalogu, którą iSeries NetServer udostępnia klientom PC w sieci iSeries. Współużytkowany zasób plikowy może składać się z dowolnych katalogów zintegrowanego systemu plików w iSeries.

Aby utworzyć współużytkowany zasób plikowy, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń system plików zawierający folder, dla którego chcesz utworzyć współużytkowany zasób plikowy.
5. Prawym klawiszem myszy kliknij folder, dla którego chcesz utworzyć współużytkowany zasób plikowy, i wybierz **Współużytkowanie**.
6. Wybierz **Nowy zasób współużytkowany**.

---

## Zmiana współużytkowanego zasobu plikowego

Współużytkowany zasób plikowy jest ścieżką do katalogu, którą iSeries NetServer udostępnia klientom PC w sieci iSeries. Współużytkowany zasób plikowy może składać się z dowolnych katalogów zintegrowanego systemu plików w iSeries.

Aby zmienić współużytkowany zasób plikowy, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.



3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń folder ze zdefiniowanym zasobem współużytkowanym, który chcesz zmienić.
5. Prawym klawiszem myszy kliknij ten folder i wybierz **Współużytkowanie**.
6. Wybierz **Nowy zasób współużytkowany**.

---

## Tworzenie nowego systemu plików użytkownika

System plików użytkownika (UDFS) to system plików, który tworzy i którego atrybuty definiuje użytkownik. Systemy plików UDFS znajdują się w pulach pamięci dyskowej (APS) w systemie.

Aby utworzyć nowy system plików użytkownika (UDFS), wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń gałąź **Root**.
5. Rozwiń gałąź **Dev**.
6. Kliknij pulę pamięci dyskowej (ASP), która ma zawierać nowy system UDFS.
7. Z menu **Plik** wybierz **Nowy UDFS**.
8. W oknie dialogowym **Nowy system plików użytkownika** podaj nazwę systemu UDFS, opis (opcjonalnie), wartości kontroli, domyślny format plików oraz określ, czy w nazwach plików w nowym systemie UDFS ma być rozróżniana wielkość liter.

---

## Podłączanie systemu plików użytkownika

System plików użytkownika (UDFS) to system plików, który tworzy i którego atrybuty definiuje użytkownik. Systemy plików UDFS znajdują się w pulach pamięci dyskowej (APS) w systemie. Aby uzyskać dostęp lub przejrzeć dane przechowywane w UDFS, należy podłączyć UDFS.

Po podłączeniu, system plików UDFS, przysłania wszystkie systemy plików, katalogi i obiekty, które istnieją poniżej punktu podłączenia w hierarchii folderów. To sprawia, że te systemy plików, katalogi i obiekty są niedostępne do chwili odłączenia systemu plików UDFS. Aby zapewnić, że zachowany będzie dostęp do wszystkich danych w zintegrowanym systemem plików, należy podłączać system UDFS do pustego folderu. Gdy system plików UDFS jest podłączony, znajdujące się w nim pliki będą dostępne w tym folderze. Wszelkie zmiany dokonane w tym folderze będą wprowadzone w systemie plików UDFS, a nie w zakrytym folderze.

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

Aby podłączyć system plików użytkownika (UDFS), wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń gałąź **Root**.
5. Rozwiń gałąź **Dev**.
6. Kliknij pulę pamięci dyskowej (ASP) zawierającą system UDFS, który chcesz podłączyć.
7. W prawym panelu Operations Navigator, w kolumnie **Nazwa UDFS** kliknij prawym klawiszem myszy system UDFS, który chcesz podłączyć.
8. Wybierz **Podłącz**.

System UDFS można również podłączyć, przeciągając go do folderu wewnątrz zintegrowanego systemu plików na tym samym serwerze. Nie można jednak przeciągnąć systemu UDFS do folderów /dev, /dev/QASPxx, /dev/asp\_name, do innego systemu ani na pulpit.

---

## Odlączenie systemu plików użytkownika

Po podłączeniu, system plików UDFS przesłania wszystkie systemy plików, katalogi i obiekty, które istnieją poniżej punktu podłączenia w hierarchii folderów. To sprawia, że te systemy plików, katalogi i obiekty są niedostępne do chwili odlączenia systemu plików UDFS.

Aby odłączyć system plików użytkownika (UDFS), wykonaj następujące czynności:

1. W oknie **Operations Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń gałąź **Root**.
5. Rozwiń gałąź **Dev**.
6. Kliknij pulę pamięci dyskowej (ASP) zawierającą system UDFS, który chcesz odłączyć.
7. W prawym panelu iSeries Navigator, w kolumnie **Nazwa UDFS** kliknij prawym klawiszem myszy system UDFS, który chcesz odłączyć.
8. Wybierz **Odłącz**.

---

## Uruchomienie kronikowania

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu.

Aby uruchomić kronikowanie obiektu:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Kliknij prawym przyciskiem myszy obiekt, który ma być kronikowany, i wybierz opcję **Kronikowanie...**
4. Po wybraniu odpowiednich opcji kronikowania, kliknij przycisk **Uruchom**.

Szczegółowe informacje na temat kronikowania obiektów zintegrowanego systemu plików można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries.

---

## Zakończenie kronikowania

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o tym, jak można rozpocząć kronikowanie obiektu, znajduje się w sekcji Uruchomienie kronikowania. Gdy kronikowanie obiektu zostanie uruchomione, istnieje możliwość przerwania kronikowania tego obiektu.

Aby zakończyć kronikowanie obiektu:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Kliknij prawym przyciskiem myszy obiekt, którego kronikowanie ma być zakończone, i wybierz opcję **Kronikowanie...**
4. Kliknij **Zakończ**.

Szczegółowe informacje na temat kronikowania obiektów zintegrowanego systemu plików można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries.

---

## Rozdział 5. Używanie zintegrowanego systemu plików w programach

Dodanie zintegrowanego systemu plików do serwera iSeries nie ma wpływu na istniejące aplikacje serwera iSeries. Języki programowania i narzędzia, takie jak język opisu struktur danych DDS, działają tak samo, jak przed wprowadzeniem zintegrowanego systemu plików.

Aby korzystać z plików strumieniowych, katalogów i innych elementów zintegrowanego systemu plików, trzeba używać zestawu funkcji API w języku C, które umożliwiają dostęp do funkcji dostarczanych przez zintegrowany system plików.

Ponadto, dodanie zintegrowanego systemu plików umożliwia kopiowanie danych pomiędzy fizycznymi zbiorami baz danych a plikami strumieniowymi. Kopiowanie to można wykonać za pomocą komend CD, funkcji przesyłania danych programu iSeries Access lub funkcji API.

Następujące sekcje wyjaśniają sposób użycia funkcji kopiowania plików strumieniowych zintegrowanego systemu plików i stanowią wprowadzenie do zastosowania funkcji API w celu uzyskania dostępu do funkcji zintegrowanego systemu plików:

- Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych
- Kopiowanie danych pomiędzy plikami strumieniowymi a plikami składowania
- Wykonywanie operacji za pomocą funkcji API
- Obsługa gniazd
- Nazywanie i obsługa międzynarodowa
- Konwersja danych

---

### Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych

Użytkownicy znający zasady pracy ze zbiorami baz danych za pomocą narzędzi do przetwarzania rekordów, takich jak język opisu struktur danych (DDS), mogą stwierdzić istnienie zasadniczych różnic 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.

W zintegrowanym systemie plików istnieją trzy metody kopiowania danych między plikami strumieniowymi a zbiorami baz danych:

- Kopiowanie danych za pomocą komend CL
- Kopiowanie danych za pomocą funkcji API
- Kopiowanie danych za pomocą funkcji przesyłania

## Kopiowanie danych za pomocą komend CL

Istnieją dwa zestawy komend CL, które umożliwiają kopiowanie danych między plikami strumieniowymi a podzbiorami zbiorów baz danych:

- CPYTOSTMF i CPYFRMSTMF,
- CPYTOIMPF i CPYFRMIMPF.

### Komendy CPYTOSTMF i CPYFRMSTMF

Do kopiowania danych pomiędzy 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 CPYTOSTMF. 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, zewnętrznie 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 pomiędzy 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 zewnętrznie opisanego do źródłowego zbioru fizycznego.
2. Użyj komend CPYTOSTMF i CPYFRMSTMF (które umożliwiają pełną konwersję danych niezależnie od tego, czy zbiór docelowy istnieje) do skopiowania danych ze źródłowego zbioru fizycznego do pliku strumieniowego.

A oto przykład:

```
CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML(X'07') STRDLML('"') 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 na serwerze. Jednak ich interfejsy są złożone.

Więcej informacji można znaleźć w pomocy dotyczącej komend lub w sekcji Język CL w Centrum informacyjnym iSeries.

## Kopiowanie danych za pomocą funkcji API

Do kopiowania podzbiorów bazy danych do pliku strumieniowego można użyć funkcji zintegrowanego systemu plików open(), read() i write(), które odpowiednio otwierają podzbiór oraz odczytują z niego dane i zapisują je w nim. W sekcji Funkcje API zintegrowanego systemu plików w Centrum informacyjnym iSeries można znaleźć więcej informacji.

## Kopiowanie danych za pomocą funkcji przesyłania

Zaletami aplikacji przesyłania danych iSeries Access są: łatwy w użyciu interfejs graficzny, automatyczna numeracja i konwersja danych znakowych. Funkcja przesyłania danych wymaga jednak zainstalowania produktu iSeries Access i używania zasobów komputera PC i serwera iSeries, a także korzystania z komunikacji pomiędzy nimi.

Jeśli na komputerze PC i na serwerze jest zainstalowany produkt iSeries Access, można go wykorzystać do przesyłania danych między plikami strumieniowymi i zbiorami baz danych. Można również przysyłać 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.

Następujące zadania pomagają kopiować i przenosić dane za pomocą aplikacji do przesyłania danych:

- Przesyłanie danych ze zbioru bazy danych do pliku strumieniowego
- Przesyłanie danych z pliku strumieniowego do zbioru bazy danych
- Przesyłanie danych do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru
- Tworzenie pliku opisu formatu

## Przesyłanie danych ze zbioru bazy danych do pliku strumieniowego

Aby przesłać dane ze zbioru bazy danych do pliku strumieniowego na serwerze, wykonaj następujące czynności:

1. Połącz się z serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. W oknie iSeries Access for Windows wybierz ikonę **Transfer danych z serwera iSeries**.
4. Wybierz serwer, z którego chcesz przysyłać dane.
5. Używając baz danych biblioteki iSeries i nazwy pliku, wybierz nazwę pliku, z którego będziesz kopiował, a także napęd sieciowy dla wynikowego pliku strumieniowego. Możesz też wybrać przycisk **Szczegóły** w sekcji **PC** w celu wybrania formatu pliku PC dla pliku strumieniowego. Przesyłanie 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 iSeries**, 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 Transfer danych do serwera iSeries tworzy plik .DTT lub .TFR. Natomiast aplikacja Transfer danych z serwera iSeries tworzy plik .DTF lub .TTO. W katalogu iSeries 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. Aby podać program, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), oraz termin jego uruchomienia, możesz użyć programu System Agent Tool (część pakietu Microsoft Plus Pack).

### Przesyłanie danych z pliku strumieniowego do zbioru bazy danych

Aby przesłać dane z pliku strumieniowego do zbioru bazy danych na serwerze, wykonaj następujące czynności:

1. Połącz się z serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. W oknie iSeries Access for Windows wybierz ikonę **Transfer danych do serwera iSeries**.
4. Na komputerze PC wybierz nazwę pliku, który chcesz przesłać. Dla nazwy pliku PC możesz użyć przycisku **Przełączaj** (dla przypisanego wcześniej napędu sieciowego) i wybrać plik strumieniowy. Możesz też użyć pliku strumieniowego znajdującego się na komputerze PC.
5. Wybierz serwer, na którym chcesz umieścić zewnętrznie opisany zbiór bazy danych.
6. Kliknij przycisk **Przesyłanie danych do iSeries**, aby uruchomić przesyłanie plików.

**Uwaga:** Jeśli dane są przesyłane do istniejącej definicji zbioru bazy danych, aplikacja Transfer danych do serwera iSeries wymaga użycia powiązanego pliku opisu formatu (FDF). Plik FDF opisuje format pliku strumieniowego. Jest on tworzony przez aplikację Transfer danych z serwera iSeries po przesłaniu danych ze zbioru bazy danych do pliku strumieniowego. Aby zakończyć przesyłanie danych z pliku strumieniowego do zbioru bazy danych, kliknij przycisk **Przesyłanie danych do iSeries**. Jeśli istniejący plik .FDF nie jest dostępny, można go szybko utworzyć.

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 Transfer danych do serwera iSeries tworzy plik .DTT lub .TFR. Natomiast aplikacja Transfer danych z serwera iSeries tworzy plik .DTF lub .TTO. W katalogu iSeries 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. Aby podać program, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), oraz termin jego uruchomienia, możesz użyć programu System Agent Tool (część pakietu Microsoft Plus Pack).

### 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 serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. W oknie iSeries Access for Windows wybierz ikonę **Transfer danych do serwera iSeries**.
4. Otwórz menu **Narzędzia** aplikacji Transfer danych do serwera iSeries.
5. Wybierz **Tworzenie zbioru bazy danych iSeries....**

Zostanie uruchomiony kreator, który pomoże utworzyć nowy zbiór bazy danych iSeries z istniejącego pliku PC. Konieczne będzie podanie nazwy pliku PC, z którego zostanie utworzony zbiór iSeries, nazwy tworzonego zbioru iSeries 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 na serwerze.

## Tworzenie pliku opisu formatu

Jeśli dane są przesyłane do istniejącej definicji zbioru bazy danych, aplikacja Transfer danych do iSeries wymaga użycia powiązanego pliku opisu formatu (FDF). Plik FDF opisuje format pliku strumieniowego. Jest on tworzony przez aplikację Transfer danych z serwera iSeries 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. Użyj funkcji Transfer danych z serwera iSeries, aby utworzyć plik strumieniowy i powiązany plik .FDF z tego zbioru bazy danych.
4. Teraz możesz użyć funkcji Transfer danych do serwera iSeries. Określ plik .FDF oraz źródłowy plik strumieniowy, który ma być przesłany.

---

## Kopiowanie danych pomiędzy 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).

Korzystając z klienta PC można również uzyskać dostęp do zbioru składowania i skopiować dane do pamięci PC lub do sieci LAN. Należy jednak pamiętać, że system plików NFS nie umożliwia dostępu do danych w zbiorze składowania.

---

## Wykonywanie operacji za pomocą funkcji API

Interfejsy API wykonujące działania na katalogach i plikach strumieniowych zintegrowanego systemu plików mają formę funkcji w języku C. Można wybrać jeden z dwóch zestawów funkcji; każdego z nich można używać w programach tworzonych za pomocą zintegrowanego środowiska językowego (ILE) C/400. Są to:

- funkcje zintegrowanego systemu plików w języku C zawarte w systemie OS/400,
- funkcje w języku C będące częścią programu licencjonowanego ILE C/400.

Funkcje zintegrowanego systemu plików działają tylko poprzez obsługę strumieniowego wejścia/wyjścia zintegrowanego systemu plików. Następujące funkcje API są obsługiwane:

*Tabela 3. Zintegrowany system plików API*

Funkcja	Opis
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

Tabela 3. Zintegrowany system plików API (kontynuacja)

<b>Funkcja</b>	<b>Opis</b>
closedir()	Zamknięcie katalogu
creat()	Utworzenie nowego lub wyczyszczenie istniejącego pliku
creat64()	Utworzenie nowego lub wyczyszczenie istniejącego pliku (również dla dużych plików)
DosSetFileLocks()	Blokowanie i odblokowanie pliku na poziomie bajtów
DosSetFileLocks64()	Blokowanie i odblokowanie pliku na poziomie bajtów (również dla dużych 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
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
getwpanam()	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



Tabela 3. Zintegrowany system plików API (kontynuacja)

Funkcja	Opis
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)
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
mkdir()	Utworzenie katalogu
mkfifo()	Utworzenie specjalnego pliku FIFO
open()	Otwarcie pliku
open64()	Otwarcie pliku (również dla dużych plików)
opendir()	Otwarcie katalogu
pathconf()	Pobranie konfigurowalnych zmiennych z nazwami ścieżek
pipe()	Utworzenie międzyprocesowego kanału z gniazdami
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
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)
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)

Tabela 3. Zintegrowany system plików API (kontynuacja)

<b>Funkcja</b>	<b>Opis</b>
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)
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)

Tabela 3. Zintegrowany system plików API (kontynuacja)

Funkcja	Opis
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
Qp0ICvtPathToSYSObjName()	Tłumaczenie nazwy ścieżki zintegrowanego systemu plików na nazwę obiektu QSYS
Qp0IFLOP()	Wykonanie różnych operacji na obiektach

Tabela 3. Zintegrowany system plików API (kontynuacja)

Funkcja	Opis
Qp0IGetAttr()	Pobranie atrybutów systemowych dla obiektu
Qp0IGetPathFromFileID()	Pobranie ścieżki obiektu z jego ID pliku
Qp0IOpen()	Otwarcie pliku z nazwą ścieżki dostępną dla NLS
Qp0IProcessSubtree()	Przetworzenie katalogów lub obiektów wewnątrz drzewa katalogów
Qp0IRenameKeep()	Zmiana nazwy pliku lub katalogu z zachowaniem <i>nowego</i> , jeśli istnieje
Qp0IRenameUnlink()	Zmiana nazwy pliku lub katalogu z usunięciem dowiązania do <i>nowego</i> , jeśli istnieje
QP0LROR()	Wczytanie referencji do obiektu
Qp0ISaveStgFree()	Zeskładowanie danych obiektów i zwolnienie ich pamięci
Qp0ISetAttr()	Ustawienie atrybutów systemowych dla obiektu
Qp0IUnlink()	Odłączenie pliku z nazwą ścieżki dostępną dla NLS
qsyssetegid()	Ustawienie obowiązującego ID grupy
qsysseteuid()	Ustawienie obowiązującego ID użytkownika
qsyssetgid()	Ustawienie ID grupy
qsyssetregid()	Ustawienie rzeczywistego i obowiązującego ID grupy
qsyssetreuid()	Ustawienie rzeczywistego i obowiązującego ID użytkownika
qsyssetuid()	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 Qp0IRenameKeep() lub Qp0IRenameUnlink()
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
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

Tabela 3. Zintegrowany system plików API (kontynuacja)

Funkcja	Opis
write()	Zapis do pliku
writenv()	Zapis do pliku (wektor)

**Uwaga:** Niektóre z tych funkcji są używane również dla gniazd OS/400. Ograniczenia dotyczące korzystania z tych funkcji w poszczególnych systemach plików opisuje sekcja “System plików w zintegrowanym systemie plików” na stronie 4. Dodatek B, “Przykładowy program wykorzystujący funkcje C zintegrowanego systemu plików” na stronie 99 zawiera przykład programu w języku C, używającego funkcji zintegrowanego systemu plików.

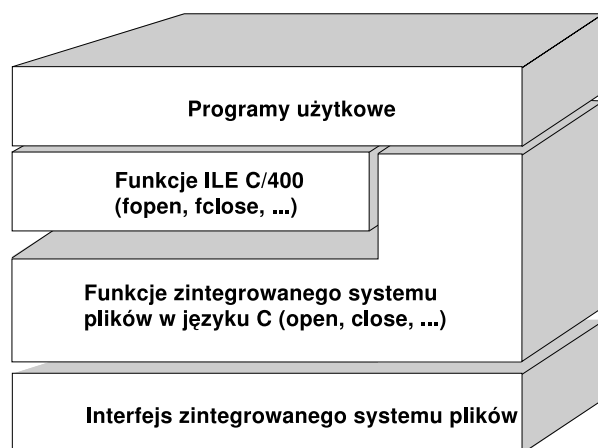
Więcej informacji o funkcjach API zintegrowanego systemu plików można znaleźć w następujących sekcjach:

- Funkcje ILE C/400
- Obsługa dużych plików przez funkcje API
- Reguły dotyczące ścieżek dla funkcji API
- Deskryptor pliku
- Ochrona
- Sekcja Aplikacyjne interfejsy programowa (API) w Centrum informacyjnym iSeries

## Funkcje ILE C/400

ILE C/400 obejmuje standardowe funkcje w języku C zdefiniowane przez 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 kompilator używał strumieniowego wejścia/wyjścia zintegrowanego systemu plików, należy wpisać wartość \*IFSIO jako parametr Opcja interfejsu systemowego (System interface option - SYSIFCOPT) w komendzie Utworzenie modułu ILE C/400 (Create ILE C/400 Module - CRTCMOD) lub Utworzenie programu łączonego 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 efekcie funkcje ILE C/400 używają podczas wykonywania operacji wejścia/wyjścia funkcji zintegrowanego systemu plików.




RV3N070-3

Rysunek 10. Funkcje ILE C/400 używają funkcji strumieniowego wejścia/wyjścia zintegrowanego systemu plików.

Więcej informacji o korzystaniu z funkcji ILE C/400 podczas obsługi strumieniowego wejścia/wyjścia zintegrowanego systemu plików zawiera publikacja WebSphere Development Studio: ILE C/C++

Programmers Guide  . Szczegółowy opis każdej funkcji ILE C/400 C można znaleźć w publikacji

WebSphere Development Studio: C/C++ Language Reference  .

## Obsługa dużych plików przez funkcje API

Funkcje API zintegrowanego systemu plików zostały tak rozbudowane, aby umożliwić aplikacjom przechowywanie w tym systemie bardzo dużych plików. Zintegrowany system plików akceptuje plik strumieniowy o wielkości do 256 gigabajtów, znajdujący się w bazowym systemie plików "root" (/), systemie plików Open Systems (QOpenSys) i systemach plików zdefiniowanych przez 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. Informacje na temat każdej 64-bitowej funkcji API można znaleźć w sekcji Funkcje API zintegrowanego systemu plików w Centrum informacyjnym iSeries.

Poniższe etykiety umożliwiają aplikacjom użycie obsługi dużych plików:

1. 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` .
2. 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.

## Reguły dotyczące ścieżek dla funkcji API

Używając funkcji API zintegrowanego systemu plików lub ILE C/400 do wykonywania działań na obiekcie, obiekt określa się podając jego ścieżkę. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API. Pojęcie **obiekt** w tych regułach oznacza katalog, plik, dowiązanie lub inny obiekt.

- Ścieżki podawane są w porządku hierarchicznym, poczynając od najwyższego poziomu w hierarchii katalogów. Nazwy składników ścieżki oddzielane są ukośnikiem (/), na przykład:

```
Kat1/Kat2/Kat3/PlikUz
```

Odwrotny ukośnik (\) 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. Sekcja "Porównanie systemów plików" na stronie 57 opisuje ograniczenia występujące w każdym systemie 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 serwera iSeries, 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'

Należy wiedzieć, że 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

Sekcja “System plików bibliotek (QSYS.LIB)” na stronie 67 zawiera więcej szczegółów na ten temat.

- 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

Więcej szczegółów zawiera sekcja “System plików QSYS.LIB niezależnej ASP” na stronie 70.

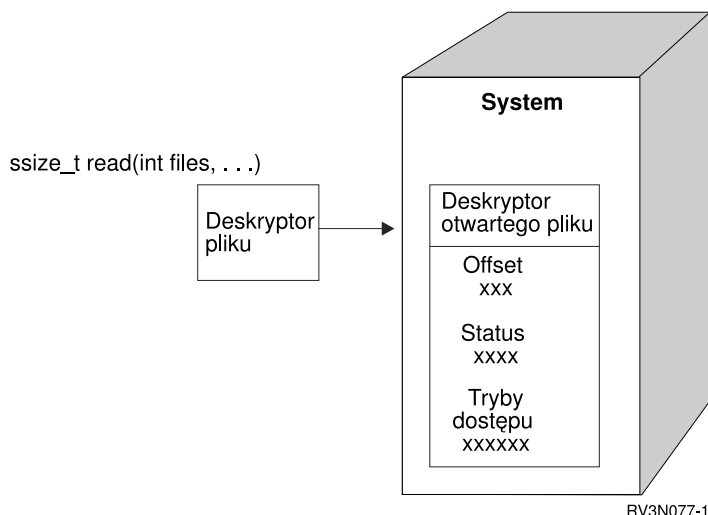
- W nazwach ścieżek nie wolno używać dwukropka (:). Ma on specjalne znaczenie w systemie.
- W odróżnieniu od ścieżek w komendach zintegrowanego systemu plików (patrz “Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 27) gwiazdka (\*), znak zapytania (?), apostrof (') oraz cudzysłów (") i tylda (~) nie mają specjalnego znaczenia. Traktowane są one jak zwykłe znaki w nazwie. Jedyne funkcje API, które stanowią wyjątek od tej reguły to QjoEndJournal() i QjoStartJournal().

## Deskryptor pliku

Używając do wykonywania operacji na plikach funkcji strumieniowego wejścia/wyjścia ILE C/400 zgodnych z definicją ANSI (American National Standards Institute), plik identyfikuje się za pomocą wskaźników.

Używając funkcji C zintegrowanego systemu plików, plik określa się podając **deskryptor pliku**. Jest on 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 *filides*, 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.

Jeśli funkcja strumieniowego wejścia/wyjścia ILE C/400 używana jest ze zintegrowanym systemem plików, obsługa modułu wykonawczego (runtime) ILE C/400 przekształca 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()` i `takedescriptor()` przekazujące deskryptory pomiędzy zdaniami. Opis tych funkcji można znaleźć w publikacji Programowanie gniazd lub w sekcji Gniazda funkcji API w Centrum informacyjnym iSeries.

## Ochrona

Używając funkcji API zintegrowanego systemu plików można ograniczyć dostęp do obiektów, podobnie jak przy korzystaniu z interfejsów zarządzania danymi. 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. Zadania serwera NFS są jedynym odstępstwem od tej reguły. Żądania skierowane do serwera NFS są obsługiwane za pomocą profilu użytkownika, którego numer identyfikacyjny (UID) serwer NFS otrzymał w momencie zgłoszenia żądania.

Uprawnienia (authorities) na serwerze są odpowiednikiem **praw dostępu (permissions)** 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 praw dostępu można zmieniać używając funkcji “zmiany trybu” `chmod()` lub `fchmod()`. Do sterowania ustawieniami bitów praw dostępu do plików tworzonych przez dane zadanie można również wykorzystywać funkcję `umask()`.

Szczegółowe informacje o ochronie i uprawnieniach można znaleźć w publikacji Security — Reference




---

## 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 zdaniami 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()`. Funkcje te oraz ogólne koncepcje gniazd opisano w sekcji Programowanie z użyciem gniazd w Centrum informacyjnym iSeries.

Po nawiązaniu połączenia oba zadania mogą przekazywać sobie dane, korzystając z 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 pomiędzy 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.

---

## Nazwenictwo i obsługa różnych języków

Obsługa systemów plików "root" (/) i QOpenSys zapewnia niezmiennosc znaków w nazwach obiektów przy przechodzeniu pomiędzy 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 (patrz "Niezmienność nazw" na stronie 22). W chwili użycia nazwy następuje jej konwersja do postaci kodowanej właściwej dla używanej strony kodowej.

Jeśli strona kodowa, do której następuje konwersja nazwy, nie zawiera znaku użytego w nazwie, nazwa jest odrzucana jako niepoprawna.

Ponieważ znaki pozostają niezmienione we wszystkich stronach kodowych, wykonując działania nie powinno się zakładać, że dany znak zostanie zamieniony na inny określony znak, gdy użyta zostanie konkretna strona kodowa. Na przykład nie powinno się przyjmować, że symbol liczby 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.

Więcej informacji o identyfikatorach CCSID zawiera sekcja Globalizacja w Centrum informacyjnym iSeries.

---

## 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 żądanego 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.

W przypadku funkcji `open()` zintegrowanego systemu plików, domyślnie używana jest opcja *Binarny, wymuszony (Binary, Forced)*.

## Rozdział 6. Systemy plików w zintegrowanym systemem plików

Zintegrowany system plików obsługuje następujące systemy plików:

- “root”
- QOpenSys
- UDFS
- QSYS.LIB
- QSYS.LIB niezależnej puli ASP
- QDLS
- QOPT
- QNetWare
- QNTC
- QFileSvr.400
- NFS

Przegląd różnych systemów plików zawiera sekcja Porównanie systemów plików.

### Porównanie systemów plików

Tabela 4 i Tabela 5 na stronie 59 zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

Tabela 4. Podsumowanie systemów plików (część 1 z 2)

Cecha	“root” (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
Standardowa część OS/400	Tak	Tak	Tak	Tak	Tak
Typ zbioru	Strumieniowy	Strumieniowy	Rekord <sup>12</sup>	Strumieniowy	Strumieniowy
Zintegrowany z OfficeVision (na przykład można wysyłać pliki)	Nie	Nie	Nie	Tak	Nie
Dostęp poprzez serwer plików OS/400	Tak	Tak	Tak	Tak	Tak
Bezpośredni dostęp poprzez procesor wejścia/wyjścia 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 zn	255 zn	10.6 zn <sup>6</sup>	8.3 zn <sup>7</sup>	255 zn
Maksymalna długość ścieżki <sup>8</sup>	16 MB	16 MB	55 – 66 zn <sup>4</sup>	82 zn	255 zn
Maksymalna długość atrybutów rozszerzonych obiektu	2 GB	2 GB	Zmienia się <sup>9</sup>	32 kB	64 kB
Maksymalna liczba poziomów hierarchii katalogów w systemie plików	Nieograni-czona <sup>10</sup>	Nieograni-czona <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/zbiór może mieć właściciela	Tak	Tak	Tak	Tak	Nie

Tabela 4. Podsumowanie systemów plików (część 1 z 2) (kontynuacja)

Cecha	“root” (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
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	Tak <sup>14</sup>	Nie	Nie

**Uwagi:**

1. LAN Server używa procesora we/wy serwera plików.
2. Przy dostępie przez serwer plików OS/400.
3. Przy dostępie do klienta PC przez LAN Server. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny.
4. Maksymalna długość ścieżki w systemie plików QSYS.LIB jest równa 55 znaków. Szczegóły zawiera sekcja “System plików bibliotek (QSYS.LIB)” na stronie 67. Maksymalna długość ścieżki w systemie plików QSYS.LIB niezależnej puli ASP jest równa 66 znaków. Szczegóły zawiera sekcja “System plików QSYS.LIB niezależnej ASP” na stronie 70.
5. Szczegóły zawiera sekcja “System plików obsługujący obiekty biblioteki dokumentów (QDLS)” na stronie 72.
6. Do 10 znaków nazwy obiektu i do 6 znaków typu obiektu. Sekcja “System plików bibliotek (QSYS.LIB)” na stronie 67 zawiera więcej szczegółów na ten temat.
7. Do 8 znaków nazwy i od 1 do 3 znaków rozszerzenia określającego typ pliku (jeśli ma występować). Szczegóły zawiera sekcja “System plików obsługujący obiekty biblioteki dokumentów (QDLS)” na stronie 72.
8. Przy 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, .CODEPAGE i .TYPE. 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. Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu.
12. Obszary użytkowników w systemie plików SYS.LIB i QSYS.LIB 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. QSYS.LIB i niezależna pula ASP QSYS.LIB może obsługiwać kronikowanie obiektów innego typu niż systemy plików bazowy (root), UDFS i QOpenSys. Więcej informacji o kronikowaniu obiektów znajdujących się w systemie plików QSYS.LIB lub w niezależnej puli ASP QSYS.LIB zawiera dokument Zarządzanie kronikowaniem w Centrum informacyjnym iSeries.
15. Katalogi \*TYPE2 posiadają limit miliona dowiązań na obiekt. Katalogi \*TYPE1 posiadają limit 32767 dowiązań na obiekt. Więcej informacji zawiera sekcja Katalogi \*TYPE2.
16. Dane w tej kolumnie odnoszą się zarówno do systemu plików QSYS.LIB jak i do systemu plików QSYS.LIB niezależnej puli ASP.

**Skróty**

zn = znaki

B = bajty    kB = kilobajty    MB = megabajty    GB = gigabajty

Tabela 5. Podsumowanie systemów plików (część 2 z 2)

Cecha	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Standardowa część OS/400	Tak	Tak	Tak	Tak	Nie
Typ zbioru	Strumieniowy	Strumieniowy	Strumieniowy	Strumieniowy	Strumieniowy
Zintegrowany z OfficeVision (na przykład można wysyłać pliki)	Nie	Nie	Nie	Nie	Nie
Dostęp poprzez serwer plików OS/400	Tak	Tak	Tak	Tak	Tak
Bezpośredni dostęp poprzez Zintegrowany serwer PC (FSIOP) <sup>1</sup>	Nie	Nie	Nie	Nie	Tak
Względna szybkość operacji otwarcia/zamknięcia	Niska	Niska <sup>2</sup>	Średnia <sup>2</sup>	Średnia <sup>2</sup>	Wysoka <sup>11</sup>
Wyszukiwanie nazw z rozróżnianiem małych i wielkich liter	Nie	Nie <sup>2</sup>	Tak <sup>12</sup>	Zmienia się <sup>2</sup>	Nie
Maksymalna długość każdego składnika ścieżki	Zmienia się <sup>4</sup>	Zmienia się <sup>2</sup>	255 zn	Zmienia się <sup>2</sup>	255 zn <sup>13</sup>
Maksymalna długość ścieżki	294 zn	Nieograniczona <sup>2</sup>	16 MB	Nieograniczona <sup>2</sup>	255 zn
Maksymalna długość atrybutów rozszerzonych obiektu	8 MB	0 <sup>6</sup>	2 GB <sup>10</sup>	0 <sup>6</sup>	64 kB
Maksymalna liczba poziomów hierarchii katalogów w systemie plików	Nieograniczona <sup>7</sup>	Nieograniczona <sup>2</sup>	Nieograniczona <sup>7</sup>	Nieograniczona <sup>2</sup>	100
Maksymalna liczba dowiązań do jednego obiektu <sup>7</sup>	1	1	Zmienia się <sup>15</sup>	Zmienia się <sup>2</sup>	1
Obsługa dowiązań symbolicznych	Nie	Nie	Tak	Tak <sup>2</sup>	Nie
Obiekt/zbiór może mieć właściciela	Nie	Nie <sup>9</sup>	Tak	Tak <sup>2</sup>	Tak
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)	Tak	Nie	Nie	Nie <sup>2</sup>	Nie
Ochrona wątków <sup>14</sup>	Tak	Nie	Tak	Nie	Nie
Obsługa kronikowania obiektów	Nie	Nie	Tak	Nie	Nie

Tabela 5. Podsumowanie systemów plików (część 2 z 2) (kontynuacja)

Cecha	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
<b>Uwagi:</b>					
<ol style="list-style-type: none"> <li>LAN Server używa procesora we/wy serwera plików.</li> <li>Zależy od tego, który zdalny system plików jest dostępny.</li> <li>Przy dostępie przez serwer plików OS/400.</li> <li>Szczegóły zawiera sekcja "System plików nośników optycznych (QOPT)" na stronie 74.</li> <li>Przy założeniu ścieżki bezwzględnej rozpoczynającej się od znaku / po którym następuje nazwa systemu plików.</li> <li>System plików QFileSvr.400 nie zwraca atrybutów rozszerzonych, nawet jeśli używany właśnie system plików je obsługuje.</li> <li>W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu.</li> <li>Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu.</li> <li>Dostępny system plików może obsługiwać prawa własności do obiektów.</li> <li>Maksymalna długość atrybutów rozszerzonych całego UDFS nie może przekroczyć 40 B.</li> <li>Przy dostępie poprzez klienta Novell NetWare. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny.</li> <li>Rozróżnianie wielkości liter jest definiowane w momencie tworzenia UDFS. Jeśli w trakcie tworzenia UDFS zostanie użyty parametr *MIXED, spowoduje to rozróżnianie wielkości liter.</li> <li>Obiekty NDS (NetWare Directory Services) mogą mieć maksymalnie 255 znaków. Nazwy plików i katalogów są ograniczone wg konwencji DOS: 8.3.</li> <li>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.</li> <li>Katalogi *TYPE2 posiadają limit miliona dowiązań na obiekt. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt. Więcej informacji zawiera sekcja Katalogi *TYPE2.</li> </ol>					
<b>Skróty</b>					
zn = znaki					
B = bajty   kB = kilobajty   MB = megabajty   GB = gigabajty					

## Bazowy system plików "root" (/)

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

Dodatkowo, system ten:

- jest zoptymalizowany w kierunku wejścia/wyjścia plików strumieniowych,
- obsługuje wiele dowiązań stałych i symbolicznych,
- obsługuje gniazda lokalne,
- obsługuje obiekty \*OOPOOL,
- obsługuje funkcje API z ochroną wątków,
- obsługuje obiekty \*FIFO,
- obsługuje obiekty /dev/null i /dev/zero \*CHRSF a także obiekty \*CHRSF,
- obsługuje kronikowanie zmian obiektów.

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

zwyczajne 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.

Więcej informacji na temat bazowego systemu plików "root" (/) można znaleźć w sekcji Korzystanie z bazowego systemu plików "root" (/).

## Korzystanie z bazowego systemu plików "root" (/)

Dostęp do bazowego systemu plików można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji API w języku C, których dostarcza serwer plików OS/400 lub zintegrowany system plików.

### Rozróżnianie wielkości znaków w bazowym systemie plików

Ten system plików zachowuje małe i wielkie litery we wprowadzanej nazwie obiektu, lecz nie rozróżnia wielkości liter podczas wyszukiwania nazwy.

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

- Ścieżki dostępu mają następującą postać:

Katalog/Katalog . . . /Obiekt

- Elementy nazw ścieżek występujące po nazwie współużytkowanej są tworzone zgodnie z regułami dotyczącymi nazw ścieżek w Windows NT. Złożone nazwy katalogów i nazwy obiektów mogą mieć do 255 znaków.
- 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) (patrz "Niezmienność nazw" na stronie 22). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi \*TYPE2.

### Dowiązania w bazowym systemie plików "root" (/)

W bazowym systemie plików dozwolonych jest wiele dowiązań stałych do jednego obiektu. W pełni obsługiwane są dowiązania symboliczne. Można używać dowiązań symbolicznych z bazowego systemu plików do obiektów w innych systemach, takich jak QSYS.LIB, QSYS.LIB niezależnej puli ASP lub QDLS.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

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

W bazowym systemie plików działają wszystkie komendy, opisane w sekcji "Wykonywanie operacji za pomocą komend CL" na stronie 24, i ekrany, opisane w sekcji "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

### Korzystanie z funkcji API zintegrowanego systemu plików w bazowym systemie plików

Wszystkie funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają na systemie plików "root" (/) w sposób bezpieczny dla wielowątkowości.

### Kronikowanie zmian w obiektach w bazowym systemie plików "root" (/)

Obiekty w bazowym systemie plików "root" (/) mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji na temat kronikowania zmian w obiektach w bazowym systemie plików "root" (/) zawiera Rozdział 7, "Obsługa kronikowania dla obiektów zintegrowanego systemu plików" na stronie 89.

---

## System plików systemów otwartych (QOpenSys)

System plików QOpenSys jest zgodny ze standardami systemów otwartych bazujących na systemie UNIX, takich jak POSIX i XPG. Podobnie jak bazowy system plików "root" (/), ten system plików wykorzystuje możliwości plików strumieniowych i katalogów dostarczanych przez zintegrowany system plików.

Dodatkowo, system ten:

- jest dostępny poprzez hierarchiczną strukturę katalogów podobną do istniejącej w systemach UNIX,
- jest zoptymalizowany w kierunku wejścia/wyjścia plików strumieniowych,
- obsługuje wiele dowiązań stałych i symbolicznych,
- obsługuje nazwy z rozróżnieniem małych i wielkich liter,
- obsługuje gniazda lokalne,
- obsługuje obiekty \*OOPOOL,
- obsługuje funkcje API z ochroną wątków,
- obsługuje obiekty \*FIFO,
- obsługuje kronikowanie zmian obiektów.

System plików QOpenSys ma te same cechy co bazowy system plików, poza tym, że rozróżnia on małe i wielkie litery w celu umożliwienia obsługi standardów systemów otwartych, opartych na systemie UNIX.

Więcej informacji zawiera sekcja Katalogi \*QOpenSys.

Więcej informacji o konwersji katalogów \*TYPE 1 lub \*TYPE2 i ograniczeń systemu plików QOpenSys można znaleźć w sekcji Katalogi \*TYPE2.

## Używanie QOpenSys

Dostęp do QOpenSys można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji API w języku C, dostarczanych przez serwer plików OS/400 lub zintegrowany system plików.

### Rozróżnianie wielkości znaków w systemie plików QOpenSys

W odróżnieniu od bazowego 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 przynajmniej jeden znak napisany jest raz małą, a raz wielką literą. W tym samym katalogu QOpenSys można przechowywać obiekt o nazwie Payroll, obiekt o nazwie PayRoll i obiekt o nazwie PAYROLL.

### Nazwy ścieżek w systemie plików QOpenSys

- Ścieżki dostępu mają następującą postać:  
Katalog/Katalog/ . . . /Obiekt
- Każdy składnik ścieżki może mieć do 255 znaków. Pełna ścieżka może mieć do 16 megabajtów 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) (patrz "Niezmiennosc nazw" na stronie 22). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi \*TYPE2.

### 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.



Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 17.

## **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOpenSys**

W systemie plików QOpenSys działają wszystkie komendy, które zawiera sekcja “Wykonywanie operacji za pomocą komend CL” na stronie 24, i ekrany, które opisuje sekcja “Wykonywanie operacji za pomocą menu i ekranów serwera iSeries” na stronie 23. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

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

Wszystkie funkcje w języku C, które opisuje sekcja “Wykonywanie operacji za pomocą funkcji API” na stronie 45, działają na systemie plików QOpenSys w sposób bezpieczny dla wątków.

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

Obiekty w systemie plików QOpenSys mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji na temat kronikowania zmian obiektów w systemie plików QOpenSys zawiera Rozdział 7, “Obsługa kronikowania dla obiektów zintegrowanego systemu plików” na stronie 89.

---

## **System plików definiowany przez użytkownika (UDFS)**

System plików UDFS znajduje się w puli pamięci dyskowej (ASP) lub w puli niezależnej (IASP) wybranej przez użytkownika. Użytkownik tworzy te systemy i zarządza nimi.

Dodatkowo system ten:

- udostępnia hierarchiczną strukturę katalogów, podobną do występującej w systemach operacyjnych komputerów PC;
- jest zoptymalizowany w kierunku wejścia/wyjścia plików strumieniowych;
- obsługuje wiele dowiązań stałych i symbolicznych;
- obsługuje gniazda lokalne;
- obsługuje funkcje API z ochroną wątków,
- obsługuje obiekty \*FIFO,
- obsługuje kronikowanie zmian obiektów.

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.

Więcej informacji na temat systemów plików zdefiniowanych przez użytkownika można znaleźć w sekcjach:

- Koncepcja UDFS
- Korzystanie z UDFS poprzez interfejs zintegrowanego systemu plików

## **Koncepcja UDFS**


W systemie plików UDFS, tak jak w systemach “root” (/) i QOpenSys, użytkownik może tworzyć katalogi, pliki strumieniowe, dowiązania symboliczne, gniazda lokalne i obiekty SOM.

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.

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.

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

Aby dowiedzieć się więcej o podłączaniu systemów plików, należy przeczytać publikację OS/400 Network File System Support .

## Korzystanie z UDFS poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików UDFS można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą serwera plików serwera OS/400 lub komend, terminali użytkowników i funkcji API. Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

### Rozróżnianie wielkości znaków w zintegrowanym systemie plików UDFS

Podczas tworzenia systemu UDFS można określić, czy ma on rozróżniać wielkość liter w nazwach obiektów.

Jeśli użytkownik sobie tego zażyczy, podczas wyszukiwania nazw obiektów system będzie rozróżniał małe i wielkie litery. Na przykład nazwa podana wielkimi literami nie zostanie uznana za identyczną z tą samą nazwą, w której jakkolwiek litera jest mała. Dlatego /home/MURPH/ i /home/murph/ rozpoznane zostaną jako inne katalogi. Aby system UDFS rozróżniał wielkość liter, należy podczas użycia komendy CRTUDFS podać wartość \*MIXED dla parametru CASE.

Można również tak ustawić parametry, aby serwer nie rozróżniał wielkości liter podczas wyszukiwania nazw. Wówczas serwer potraktuje nazwy /home/CAYCE i /HOME/cayce jako nazwy tego samego katalogu, a nie dwu różnych. Aby UDFS nie rozróżniał wielkości liter, należy podczas użycia komendy CRTUDFS podać wartość \*MONO dla parametru CASE.

W obu przypadkach system plików zachowuje nazwy obiektów w takiej postaci, w jakiej zostały one podane przez użytkownika. Opcja rozróżniania wielkości liter odnosi się tylko do wyszukiwania przez użytkownika nazw na serwerze.

### Nazwy ścieżek w zintegrowanym systemie plików UDFS

W momencie, gdy występuje konieczność manipulacji całym systemem plików UDFS wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania UDFS służy specjalny plik blokowy (\*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 megabajtów 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) (patrz “Niezmienność nazw” na stronie 22). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi \*TYPE2.

## Dowiązania w zintegrowanym systemie plików UDFS

Może istnieć wiele dowiązań stałych do obiektów wewnątrz UDFS. System UDFS w pełni obsługuje również dowiązania symboliczne. Można utworzyć dowiązanie symboliczne z UDFS do obiektu w innym systemie plików.

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 17.

## Korzystanie z komend zintegrowanego systemu plików w systemie UDFS

Wszystkie komendy przedstawione w sekcji “Wykonywanie operacji za pomocą komend CL” na stronie 24 i ekrany opisane w sekcji “Wykonywanie operacji za pomocą menu i ekranów serwera iSeries” na stronie 23 działają również w systemie plików UDFS. Istnieją jednak komendy CL przeznaczone specjalnie dla UDFS i innych dołączanych systemów plików. Są one opisane w poniższej tabeli.

Tabela 6. Komendy CL dotyczące systemu plików definiowanego przez użytkownika

Komenda	Opis
ADDMFS	Dodanie podłączonego FS (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.
CRTUDFS	Utworzenie systemu plików UDFS (Create UDFS). Tworzy system plików definiowany przez użytkownika (UDFS).
DLTUDFS	Usunięcie systemu plików UDFS (Delete UDFS). Usuwa system plików definiowany przez użytkownika.
DSPMFSINF	Wyświetlenie danych podłączonego FS (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.
DSPUDFS	Wyświetlenie UDFS (Display UDFS). Wyświetla informacje na temat systemu plików definiowanego przez użytkownika.
MOUNT	Podłączenie systemu plików (Mount a File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. Jest to alternatywna postać komendy ADDMFS.

Tabela 6. Komendy CL dotyczące systemu plików definiowanego przez użytkownika (kontynuacja)

Komenda	Opis
RMVMFS	Usunięcie dołączonego systemu plików (Remove Mounted File System). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnej przestrzeni nazw klienta.
UNMOUNT	Odlączenie systemu plików (Unmount a file system). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnej przestrzeni nazw klienta. Jest to alternatywna postać komendy RMVMFS.

**Uwaga:** Aby komendy zintegrowanego systemu plików mogły wykonywać działania na obiektach systemu plików UDFS, musi on zostać podłączony.

### Korzystanie z funkcji API zintegrowanego systemu plików w systemie UDFS

Wszystkie funkcje API w języku C, które opisuje sekcja “Wykonywanie operacji za pomocą funkcji API” na stronie 45, działają w systemie plików UDFS.

**Uwaga:** Aby komendy zintegrowanego systemu plików mogły wykonywać działania na obiektach systemu plików UDFS, musi on zostać podłączony.

### Graficzny interfejs użytkownika dla systemu plików UDFS

iSeries Navigator, graficzny interfejs użytkownika w komputerze PC, zapewnia łatwy i dogodny dostęp do systemu plików UDFS. Umożliwia on użytkownikowi tworzenie, usuwanie, wyświetlanie, dołączanie i odłączanie UDFS z klienta Windows.

Działania w systemie UDFS można wykonywać poprzez iSeries Navigator. Podstawowe zadania opisuje sekcja:

- “Tworzenie nowego systemu plików użytkownika” na stronie 39.
- “Podłączanie systemu plików użytkownika” na stronie 39.
- “Odlączenie systemu plików użytkownika” na stronie 40.

### Utworzenie systemu plików UDFS w zintegrowanym systemie plików

Komenda Utworzenie systemu plików UDFS (Create User-Defined File System - CRTUDFS) tworzy system plików widoczny w przestrzeni nazw zintegrowanego systemu plików oraz dostępny przez funkcje API i komendy CL. Komendy ADDMFS lub MOUNT umieszczają UDFS “ponad” istniejącym lokalnie katalogiem. Użytkownik może tworzyć UDFS w dowolnej puli ASP lub niezależnej puli ASP. Może również określić rozróżnianie przez system wielkości liter.

### Usuwanie systemu plików UDFS w zintegrowanym systemie plików

Komenda Usunięcie systemu plików UDFS (Delete User-Defined File System - DLTUDFS) usuwa istniejący, niepodłączony system UDFS i wszystkie znajdujące się w nim obiekty. 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.

### Wyświetlanie systemu plików UDFS w zintegrowanym systemie plików

Komenda Wyświetlenie systemu plików UDFS (Display User-Defined File System - DSPUDFS) podaje atrybuty 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.

### Podłączanie systemu plików UDFS w zintegrowanym systemie plików

Komenda Dodanie podłączonego FS (Add Mounted File System - ADDMFS) oraz komenda MOUNT udostępniają obiekty systemu plików w przestrzeni nazw zintegrowanego systemu plików. Aby dołączyć UDFS, należy w komendzie ADDMFS dla parametru TYPE podać wartość \*UDFS.

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

### **Odlączenie systemu plików UDFS w zintegrowanym systemie plików**

Komenda odlączająca UDFS powoduje, że zawartość UDFS staje się niedostępna dla interfejsu zintegrowanego systemu plików. Obiekty znajdujące się w UDFS stają się niedostępne w momencie odlączenia UDFS. Komenda Usuwanie podłączonego FS (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, przestaniając którąkolwiek część starego UDFS, to starego UDFS nie można będzie odlączyć, dopóki nie zostanie on odslonięty.

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 odlączyć jenn.udfs, dopóki nie zostanie odlączony drugi system plików z /home/judy.

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

### **Składowanie i odtwarzanie systemu plików UDFS w zintegrowanym systemie plików**

Użytkownik ma możliwość składowania i odtwarzania wszystkich obiektów UDFS wraz ze związanymi z nimi uprawnieniami. Komenda Składowanie (Save - SAV) pozwala składać obiekty UDFS, natomiast komenda Odtworzenie (Restore - RST) pozwala użytkownikowi odtworzyć obiekty UDFS. Obydwie komendy funkcjonują bez względu na to, czy UDFS jest dołączony czy odlączony. Jednak aby poprawnie składać atrybuty UDFS, a nie tylko obiekty znajdujące się w UDFS, system plików powinien zostać odlączony.

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

Obiekty w zdefiniowanych przez użytkowników systemach plików mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji na temat kronikowania zmian obiektów w systemie plików UDFS zawiera Rozdział 7, "Obsługa kronikowania dla obiektów zintegrowanego systemu plików" na stronie 89.

---

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

System plików QSYS.LIB obsługuje strukturę bibliotek serwera iSeries. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów serwera iSeries obsługiwanych przez biblioteki, które znajdują się w systemowej puli ASP i w podstawowej puli ASP użytkowników.

Dodatkowo, system ten:

- obsługuje wszystkie interfejsy użytkownika i interfejsy programistyczne wykonujące działania na bibliotekach serwera iSeries 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 serwera iSeries,
- obsługuje operacje strumienia wejścia/wyjścia na podzbiórach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

W wersjach wcześniejszych niż wersja 3 systemu OS/400 system plików QSYS.LIB był nazywany po prostu *systemem plików serwera iSeries*. 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 QSYS.LIB poprzez interfejs zintegrowanego systemu plików.

Więcej informacji na temat QSYS.LIB można znaleźć w sekcji Korzystanie z QSYS.LIB poprzez interfejs zintegrowanego systemu plików.

## **Korzystanie z QSYS.LIB poprzez interfejs zintegrowanego systemu plików**

Dostęp do systemu plików QSYS.LIB można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji w języku C dostarczanych przez serwer plików OS/400 lub zintegrowany system plików. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

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

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 ze zdalnych klientów QSYS.LIB.

### **Obsługa ograniczeń 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 blokady na poziomie bajtów można znaleźć w opisie funkcji fcntl() w Centrum informacyjnym iSeries.)
- 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 użytkownika w systemie plików QSYS.LIB**

System plików QSYS.LIB obsługuje operacje strumieniowego wejścia/wyjścia do obiektów w obszarze 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 również obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez serwer systemu Network File System. Są one jednak dostępne z klientów PC i 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 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 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 o identyfikatorach CCSID zawiera sekcja Globalizacja w Centrum informacyjnym iSeries.

### **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 pomiędzy 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.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

### **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB**

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QSYS.LIB z następującymi wyjątkami:

- komenda 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 STRJRN i ENDJRN nie mogą być użyte na fizycznych zbiorach bazy danych.

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23.

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

Funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają w systemie plików QSYS.LIB z następującymi wyjątkami:

- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- funkcja symlink() może być używana jedynie do tworzenia dowiązań do obiektu w QSYS.LIB z innego systemu plików obsługującego dowiązania symboliczne,
- funkcje API QjoStartJournal() i QjoEndJournal() nie mogą być użyte na fizycznych zbiorach bazy danych.

---

## System plików QSYS.LIB niezależnej ASP

System plików QSYS.LIB niezależnej ASP obsługuje strukturę biblioteki serwera iSeries w niezależnej puli pamięci dyskowej ASP utworzonej i zdefiniowanej przez użytkownika. Umożliwia on dostęp do zbiorów baz danych i wszystkich innych typów obiektów serwera iSeries 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 serwera iSeries 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 serwera iSeries,
- obsługuje operacje strumienia wejścia/wyjścia na podzbiorach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

Więcej informacji na temat systemu plików QSYS.LIB niezależnej puli ASP można znaleźć w sekcji Korzystanie z niezależnej puli ASP QSYS.LIB poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z systemu plików QSYS.LIB niezależnej puli ASP poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QSYS.LIB niezależnej puli ASP można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji w języku C dostarczanych przez serwer plików OS/400 lub zintegrowany system plików. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

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

QPWFSEVER 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 niezależnej puli ASP QSYS.LIB ze zdalnych klientów.

## Obsługa ograniczeń 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 obsługiwanych w trybie tekstowym, opisane zewnętrznie zbiory fizyczne.



- Blokada na poziomie bajtów nie jest obsługiwana. (Więcej informacji na temat blokady na poziomie bajtów można znaleźć w opisie funkcji fcntl() w Centrum informacyjnym iSeries.)
- 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 użytkownika w systemie plików QSYS.LIB niezależnej puli ASP**

System plików QSYS.LIB niezależnej puli ASP obsługuje operacje strumieniowego wejścia/wyjścia do obiektów w obszarze 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 ASP 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 również obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez serwer systemu Network File System. Są one jednak dostępne z klientów PC i 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 QSYS.LIB, hierarchia katalogów w systemie plików QSYS.LIB niezależnej puli ASP może mieć sześć 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 o identyfikatorach CCSID zawiera sekcja Globalizacja w Centrum informacyjnym iSeries.

### **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 pomiędzy 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.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

### **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP**

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QSYS.LIB niezależnej puli ASP z następującymi wyjątkami:

- komenda 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 STRJRN i ENDJRN nie mogą być użyte na fizycznych zbiorach bazy danych.
- Nie można przenosić bibliotek w systemie plików QSYS.LIB niezależnej puli ASP do podstawowych pul pamięci dyskowej (ASP) za pomocą komendy MOV. 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 SAV lub 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/objekt.typ nie jest obsługiwana w SAV i RST.

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23.

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

Funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają w systemie plików QSYS.LIB niezależnej puli ASP z następującymi wyjątkami:

- operacje na zbiorach można wykonywać jedynie w odniesieniu do źródłowych zbiorów fizycznych i zbiorów fizycznych opisywanych przez program,
- funkcja symlink() może być używana jedynie do tworzenia dowiązań *do* obiektu w systemie plików QSYS.LIB niezależnej puli ASP z innego systemu plików obsługującego dowiązania symboliczne,
- funkcje API QjoStartJournal() i QjoEndJournal() nie mogą być użyte na fizycznych zbiorach bazy danych.

---

## **System plików obsługujący obiekty biblioteki dokumentów (QDLS)**

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

Dodatkowo, system ten:

- obsługuje foldery serwera iSeries i obiekty biblioteki dokumentów (DLO),
- obsługuje dane przechowywane w plikach strumieniowych.

Więcej informacji opisujących QDLS można znaleźć w sekcji Korzystanie z QDLS poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z QDLS poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QDLS można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji w języku C dostarczanych przez serwer plików OS/400 lub zintegrowany system plików. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

### Zintegrowany system plików i hierarchiczny system plików (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). Zintegrowany system plików jest oparty na modelu programowym ILE (integrated language environment), natomiast podstawą HFS jest oryginalny model programowy serwera iSeries.

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.

Więcej informacji na temat HFS można znaleźć w sekcji Funkcje API hierarchicznego systemu plików w Centrum informacyjnym iSeries.

### Wpisanie użytkownika na listę w systemie plików QDLS

Użytkownik pracujący z obiektami w 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**) użyte w nazwach obiektów na wielkie. Przy wyszukiwaniu nazw obiektów zawierających jedynie takie znaki nie są rozróżniane małe i wielkie litery.

W odniesieniu do wszystkich innych nazw małe litery traktowane są w systemie QDLS jako różne od wielkich.

Więcej szczegółów można znaleźć w sekcji Nazwy folderów i dokumentów w Centrum informacyjnym iSeries.

### Nazwy ścieżek w systemie plików QDLS

- Każdy składnik ścieżki może być po prostu nazwą, np:

```
/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 to 82 znaki.
- Hierarchia katalogów w ramach systemu QDLS może mieć do 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 zawiera sekcja Globalizacja w Centrum informacyjnym iSeries.

## 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 pomiędzy folderem a każdym spośród zawartych w nim obiektów. Jest więc możliwe tworzenie dowiązań z systemu plików obsługującego dowiązania symboliczne do systemu plików QDLS.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

## Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QDLS

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QDLS z następującymi wyjątkami:

- komendy ADDLNK można używać jedynie do tworzenia dowiązań *do* obiektu w QDLS z innego systemu plików, który obsługuje dowiązania symboliczne,
- komendy CHKIN i CHKOUT są obsługiwane jedynie dla plików, a nie dla katalogów,
- komendy APYJRNCHG, ENDJRN, SNDJRNE i STRJRN nie są obsługiwane.

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23.

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

Funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają w systemie plików QDLS z następującymi wyjątkami:

- funkcja symlink() może być używana jedynie do tworzenia dowiązań do obiektu w QDLS z innego systemu plików obsługującego dowiązania symboliczne,
- poniższe funkcje nie są obsługiwane:

- givedescriptor()
- ioctl()
- link()
- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()
- Qp0lGetPathFromFileID()
- readlink()
- takedescriptor()

---

## System plików nośników optycznych (QOPT)

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


Dodatkowo, system ten:

- udostępnia hierarchiczną strukturę katalogów, podobną do występującej w systemach operacyjnych komputerów PC,
- jest zoptymalizowany w kierunku wejścia/wyjścia plików strumieniowych,
- obsługuje dane przechowywane w plikach strumieniowych.

Więcej informacji na temat QOPT można znaleźć w sekcji Korzystanie z QOPT poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z QOPT poprzez interfejs zintegrowanego systemu plików


Dostęp do systemu plików QOPT można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji API dostarczanych przez serwer plików OS/400 lub zintegrowany system plików. Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

Więcej szczegółów można znaleźć w publikacji Optical Support .

## Zintegrowany system plików i hierarchiczny system plików (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). Zintegrowany system plików jest oparty na modelu programowym ILE (integrated language environment), natomiast podstawą HFS jest oryginalny model programowy serwera iSeries.

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 można znaleźć w sekcji Funkcje API hierarchicznego systemu plików w Centrum informacyjnym iSeries lub w publikacji Optical Support .

## 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 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 zbudowana 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.
- W nazwie ścieżki można po prostu wpisać /QOPT lub dołączyć przynajmniej jedną nazwę katalogu lub podkatalogu. Nazwy katalogów i plików mogą zawierać dowolne znaki oprócz znaków o kodach od X'00' do X'3F' i znaku o kodzie 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 szczegółowych informacji o regułach obowiązujących dla nazw ścieżek w systemie plików QOPT zawiera sekcja "Path Name Rules" w publikacji Optical Support .

## Dowiązania w systemie plików QOPT

System plików QOPT dopuszcza tylko jedno dowiązanie do obiektu. W QOPT nie można tworzyć ani przechowywać dowiązań symbolicznych. Można jednak uzyskać dostęp do plików w QOPT, używając dowiązania symbolicznego z systemu plików "root" (/) lub QOpenSys.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

## **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOPT**

Większość komend wymienionych w sekcji "Wykonywanie operacji za pomocą komend CL" na stronie 24 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, które opisuje sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23.

Następujące komendy zintegrowanego systemu plików nie są obsługiwane przez system plików QOPT:

- ADDLNK
- APYJRNCHG
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

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

Wszystkie funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają na systemie plików "root" (/) w sposób bezpieczny dla wielowątkowości, z wyjątkiem następujących:

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

---

## **System plików NetWare (QNetWare)**

System plików QNetWare umożliwia dostęp do danych przechowywanych na lokalnym lub zdalnym serwerze Integrated xSeries Server for iSeries, pracującym z Novell NetWare 4.10 lub 4.11, albo na autonomicznych serwerach PC, na których został uruchomiony Novell NetWare 3.12, 4.10, 4.11 lub 5.0.

Dodatkowo, system ten:

- zapewnia dostęp do obiektów NDS,
- obsługuje dane przechowywane w plikach strumieniowych,
- zapewnia dynamiczne dołączenie systemów plików Netware do lokalnych przestrzeni nazw.

**Uwaga:** System plików QNetWare jest dostępny tylko wtedy, gdy w systemie zainstalowano produkt NetWare Enhanced Integration for iSeries 400, opcja BOSS 25. Po jego instalacji i wykonaniu IPL katalog i podkatalogi systemu plików /QNetWare pojawiają się jako część struktury katalogów zintegrowanego systemu plików

Więcej informacji na temat systemu plików QNetWare można znaleźć w następujących sekcjach:

- Podłączanie systemów plików NetWare
- Struktura katalogów QNetWare

- Korzystanie z QNetWare poprzez interfejs zintegrowanego systemu plików

## Podłączanie systemów plików NetWare

Systemy plików NetWare umieszczone na serwerach Novell NetWare mogą zostać podłączone do bazowego systemu plików ("root", /), systemu plików QOpenSys i innych. Daje to łatwiejszy dostęp do systemu plików NetWare i zapewnia lepszą wydajność, niż poprzez katalog /QNetWare. Podłączenie systemów plików NetWare może być również użyte w celu wykorzystania różnych opcji komendy Dodanie podłączonego FS (Add Mounted File System - ADDMFS), na przykład podłączenia systemu plików z możliwym zapisem (read-write) w trybie tylko do odczytu (read-only).

Systemy plików NetWare można dołączyć podając ścieżkę NDS lub ścieżkę NetWare w postaci SERVER/VOLUME:katalog/katalog. Na przykład, aby dołączyć katalog doorway, który jest umieszczony w woluminie Nest na serwerze Dreyfuss, użytkownik może użyć następującej składni:

```
DREYFUSS/NEST: doorway
```

Składnia tej ścieżki jest bardzo podobna do składni komendy NetWare MAP. Ścieżki NDS mogą zostać użyte do podania ścieżki do woluminu NetWare, ale same nie mogą zostać dołączone.

## Struktura katalogów QNetWare

Struktura katalogów /QNetWare reprezentuje różne złożone systemy plików.

- Struktura ta reprezentuje znajdujące się w sieci serwery i woluminy Novell NetWare w następującej formie:

```
/QNetWare/SERVER.SVR/WOLUMIN
```

Rozszerzenie .SVR oznacza serwer Novell NetWare.

- Przy odwołaniu do woluminu serwera przez ekrany lub komendy zintegrowanego systemu plików lub przez funkcje API, katalog główny woluminu NetWare jest automatycznie dołączany do katalogu VOLUME znajdującego się w /QNetWare.
- QNetWare reprezentuje drzewa NDS znajdujące się w sieci w następujący sposób:

```
/QNetWare/CORP_TREE.TRE/USA.C/ORG.0/ORG_UNIT.OU/SVR1_VOL.CN
```

Rozszerzenia .TRE, .C, .O, .OU i .CN symbolizują odpowiednio: drzewo NDS, państwa, organizacje, jednostki organizacyjne, nazwy pospolite. Jeśli dostęp do woluminu Novell NetWare odbywa się poprzez ścieżkę NDS, poprzez obiekt woluminu lub alias obiektu woluminu, to katalog główny tego woluminu jest również automatycznie dołączany do obiektu NDS.

## Korzystanie z QNetWare poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QNetWare można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą serwera plików OS/400 lub komend, terminali użytkowników i funkcji API zintegrowanego systemu plików. Należy wziąć pod uwagę poniższe okoliczności, ograniczenia i zależności:

### Uprawnienia i prawa własności w systemie plików QNetWare

W systemie plików QNetWare pliki i katalogi są przechowywane i zarządzane przez serwery Novell NetWare. W trakcie używania funkcji API i komend do odszukania lub ustalenia uprawnień zarówno użytkowników, jak i wJeśli liczba znaków w nazwie użytkownika NetWare przekracza 10 lub jeśli nie istnieje odpowiedni użytkownik na serwerze iSeries, to uprawnienia nie zostaną odwzorowane. Użytkownicy, którzy nie mogą zostać odwzorowani, są automatycznie odwzorowywani na profil użytkownika QDFTOWN. Używając komend WRKAUT i CHGAUT można wyświetlać i zmieniać uprawnienia użytkowników. Podczas przekazywania uprawnień do i z serwera są one odwzorowywane na uprawnienia na serwerze iSeries.

## **Kontrola w systemie plików QNetWare**

Mimo że Novell NetWare obsługuje kontrolę plików i katalogów, to system plików QNetWare nie może zmienić wartości kontrolnych tych obiektów. Plików i katalogów, to system plików QNetWare nie może zmienić wartości kontrolnych tych obiektów. Dlatego komenda CHGAUD nie jest obsługiwana.

## **Pliki i katalogi w systemie plików QNetWare**

System plików QNetWare nie zachowuje wielkości liter w nazwach plików i katalogów wpisywanych w komendach lub funkcjach API. Podczas transmisji do serwera wszystkie litery w nazwach są zamieniane na wielkie. Novell NetWare obsługuje również przestrzenie nazw wielu różnych platform, takich jak DOS, OS/2, Apple Macintosh i NFS. Natomiast system QNetWare obsługuje tylko przestrzeń nazw DOS. Ponieważ we wszystkich woluminach Novell NetWare wymagana jest przestrzeń nazw DOS, wszystkie pliki i katalogi pojawiają się w systemie plików Q

## **Obiekty NDS w systemie plików QNetWare**

System plików QNetWare obsługuje wyświetlanie nazw NDS używając małych i wielkich liter.

## **Dowiązania w systemie plików QNetWare**

System plików QNetWare dopuszcza tylko jedno dowiązanie do obiektu. W systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych. W systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych. Mogą jednak zostać utworzone dowiązania symboliczne między katalogami b

## **Korzystanie z komend i terminali zintegrowanego systemu p**

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QNetWare; z następującymi wyjątkami: w systemie plików QNetWare; z następującymi wyjątkami:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRN
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Ponadto następujące komendy nie mogą być wykorzystywane do działań na obiektach NDS, serwerach i woluminach:

- CHGOWN
- CPYFRMSTMF
- CPYTOSTMF
- CRTDIR

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

Funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają w systemie plików QNetWare. Wyjątkami są:

- givedescriptor()
- link()
- QjoEndJournal()



QjoRetrieveJournalInformation()  
QJORJIDI()  
QJOSJRNE()  
QjoStartJournal()  
readlink()  
symlink()  
takedescriptor()

Ponadto następujące funkcje API nie mogą być wykorzystywane do działań na obiektach NDS, serwerach lub woluminach:

chmod()  
chown()  
create()  
fchmod()  
fchown()  
fcntl()  
ftruncate()  
lseek()  
mkdir()  
read()  
readv()  
unmask()  
write()  
writev()

---

## System plików serwera Windows NT (QNTC)

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych na lokalnym lub zdalnym serwerze Integrated xSeries Server for iSeries, na którym został uruchomiony Windows NT w wersji 4.0 lub nowszej bądź serwer autonomiczny. Umożliwia to aplikacjom serwera iSeries używanie tych samych danych, z których korzystają klienci Windows NT. System ten przechowuje dane w plikach strumieniowych.

System plików QNTC jest częścią podstawowego systemu operacyjnego OS/400. Aby używać systemu plików QNTC, należy zainstalować TCP/IP Connectivity Utilities for iSeries (numer części: 5769-TC1). Dostęp do /QNTC nie wymaga instalowania opcji 29 systemu operacyjnego produktu iSeries 400 Integration with Windows NT Server.

Więcej informacji na temat QNTC można znaleźć w sekcji Korzystanie z QNTC poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z QNTC poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QNTC można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji API dostarczanych przez serwer plików OS/400 lub zintegrowany system plików. Należy mieć na uwadze poniższe okoliczności i ograniczenia.

## 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.

Uprawnienia do plików i katalogów systemu NT są administrowane z 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 wprowadzanej nazwie obiektu, lecz nie rozróżnia wielkich i małych liter w nazwach. 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 musi zaczynać się od ukośnika (/) i może zawierać co najwyżej 255 znaków.
- W nazwach ścieżek rozróżniane są wielkie i małe litery.
- Ścieżka dostępu zbudowana jest z nazwy systemu plików, nazwy serwera Windows NT, nazwy współużytkowanej, nazwy katalogu i podkatalogu oraz nazwy obiektu. Ścieżki dostępu mają następującą postać:

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

- Nazwa serwera może mieć do 15 znaków. Musi być ona częścią ścieżki.
- Nazwa współużytkowana może mieć do 12 znaków.
- Każdy składnik ścieżki (występujący po nazwie współużytkowanej) 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.
- Każdy aktywny serwer Windows NT w podsieci lokalnej będzie się automatycznie pojawiał jako katalog pod /QNTC. Aby dodać serwery Windows NT spoza lokalnej podsieci, należy użyć komendy Utworzenie katalogu (Make Directory - MKDIR) (patrz Tabela 2 na stronie 24) lub funkcji API mkdir() (patrz "Wykonywanie operacji za pomocą funkcji API" na stronie 45).

### **Dowiązania w systemie plików QNTC**

System plików QNTC dopuszcza tylko jedno dowiązanie do obiektu. W QNTC nie można utworzyć ani przechowywać dowiązań symbolicznych. Można uzyskać dostęp do plików w QNTC, używając dowiązania symbolicznego z systemu plików "root" (/) lub QOpenSys.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

### **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QNTC**

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QNTC z następującymi wyjątkami:

```
ADDLNK  
APYJRNCHG  
CHGOWN  
CHGAUT  
CHGPGP  
CHKIN  
CHKOUT  
DSPAUT  
ENDJRN  
RST  
SAV  
SNDJRNE  
STRJRN
```

WRKAUT  
WRKOBJOWN  
WRKOBJPGP

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Wykonywanie operacji za pomocą menu i ekranów serwera iSeries” na stronie 23.

### **Korzystanie z komendy MKDIR w systemie plików QNTC**

Aby dodać katalog serwera do katalogu /QNTC, należy użyć komendy Utworzenie katalogu (Make Directory - MKDIR). Wszystkie aktywne serwery Windows NT w sieci lokalnej są tworzone automatycznie. Serwery Windows NT spoza lokalnej podsieci muszą zostać dodane za pomocą komendy MKDIR lub funkcji API mkdir(). Na przykład:

```
MKDIR '/QNTC/NTSRV1'
```

doda 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. Na przykład:

```
MKDIR '/QNTC/9.130.67.24'
```

doda serwer do struktury katalogów systemu plików QNTC.

**Uwaga:** Katalogi dodane do struktury katalogów za pomocą funkcji API mkdir() lub komendy CL MKDIR nie będą widoczne po IPL. Komenda MKDIR lub funkcja API mkdir() musi być ponownie użyta po każdym IPL systemu.

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

Funkcje API w języku C, które opisuje sekcja “Wykonywanie operacji za pomocą funkcji API” na stronie 45, działają w systemie plików QNTC z następującymi wyjątkami:

- użycie funkcji chmod(), fchmod(), utime() lub umask() do obiektów w QNTC nie będzie miało żadnego skutku, ale próba ich użycia nie spowoduje błędów;
- system plików QNTC nie obsługuje poniższych funkcji:

- chown()
- fchown()
- givedescriptor()
- link()
- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()
- QpOIGetPathFromFileID()
- readlink()
- symlink()
- takedescriptor()

---

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

System plików serwera plików OS/400 umożliwia przezroczysty dostęp do innych systemów plików znajdujących się na zdalnych serwerach iSeries. Jest to dostęp poprzez hierarchiczną strukturę katalogów.

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

Więcej informacji na temat QFileSvr.400 można znaleźć w sekcji Korzystanie z QFileSvr.400 poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z QFileSvr.400 poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QFileSvr.400 można uzyskać poprzez interfejs zintegrowanego systemu plików za pomocą komend, ekranów użytkownika i funkcji API dostarczanych przez serwer plików OS/400 lub przez zintegrowany system plików. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

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

### Rozróżnianie wielkości znaków w systemie plików serwera plików OS/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 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 OS/400.

### Nazwy ścieżek w systemie plików serwera plików OS/400

- Ścieżki dostępu mają następującą postać:

```
/QFileSvr.400/JednostkaZdalna/Katalog/Katalog . . . /Obiekt
```

Pierwszy poziom katalogów (w powyższym przykładzie JednostkaZdalna) może spełniać dwojaką rolę:

- Nazwy serwera docelowego, który zostanie użyty do nawiązania połączenia. Nazwa serwera docelowego może być:
  - nazwą hosta TCP/IP (na przykład beowulf.newyork.corp.com),
  - nazwą SNA LU 6.2 (na przykład appn.newyork).
- Katalogu "root" (/) serwera docelowego.

Dlatego też atrybuty podane podczas tworzenia katalogu pierwszego poziomu za pomocą interfejsu zintegrowanego systemu plików są ignorowane.

**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.
- Gdy nazwa jest zapisywana, składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (patrz "Niezmiennosc nazw" na stronie 22).

## Komunikacja w systemie plików serwera plików OS/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 QPWFSESV 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 zawiera publikacja 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 OS/400.
- Jeśli nie zostało ustanowione połączenie z serwerem docelowym, system plików QFileSvr.400 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.
  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 dwójaka, 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 OS/400.
    - 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 OS/400 na serwerze docelowym. Po ustanowieniu połączenia system QFileSvr.400 wymienia żądania i odpowiedzi z serwerem plików. W obrębie podsystemu QSERVER żądania prestartu QPWFSESVSO 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 pomiędzy APPC i serwerem plików OS/400.
- 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 zapętlenia. 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, to 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.

## Ochrona i uprawnienia do obiektu w systemie plików serwera plików OS/400

Jeśli w obu systemach skonfigurowany jest protokół Kerberos i użytkownik jest uwierzytelniony w protokole Kerberos, to protokół Kerberos może być użyty do uwierzytelniania systemu plików znajdującego się na docelowym serwerze iSeries. Jeśli uwierzytelnianie Kerberos nie powiedzie się, to do potwierdzenia dostępu może być użyty identyfikator użytkownika i hasło.

**Uwaga:** Jeśli ważność biletu uprawniającego do biletu lub biletu serwera wygaśnie po tym jak serwer docelowy potwierdzi dostęp, wygaśnięcie nie nastąpi, dopóki połączenie z serwerem docelowym nie zostanie rozłączone. Więcej informacji na temat protokołu Kerberos można znaleźć w sekcji Usługa uwierzytelniania sieci w Centrum informacyjnym iSeries.

- Jeśli do uwierzytelniania nie jest wykorzystywany protokół Kerberos, to aby użytkownik mógł uzyskać dostęp do systemu plików znajdującego się na docelowym serwerze iSeries, jego ID i hasło na tym serwerze muszą być zgodne z ID i hasłem znajdującym się na serwerze lokalnym.

**Uwaga:** Jeśli hasło użytkownika na serwerze lokalnym lub docelowym zostało zmienione po tym, jak serwer docelowy zweryfikował możliwość dostępu przez użytkownika, to zmiana ta nie jest uwzględniana, dopóki nie zostanie zakończone połączenie z serwerem docelowym. Zwłoka jednak nie wystąpi, jeśli na serwerze lokalnym zostanie usunięty profil użytkownika i utworzony zostanie inny profil z tym samym ID użytkownika. W tym przypadku system plików QFileSvr.400 potwierdza dostęp użytkownika do serwera docelowego.

- Uprawnienia do obiektu określone są przez profil użytkownika znajdujący się na serwerze docelowym. Oznacza to, że użytkownik uzyska dostęp do obiektu znajdującego się w systemie plików serwera docelowego tylko wtedy, gdy profil użytkownika na serwerze docelowym ma odpowiednie uprawnienia do tego obiektu.

### **Dowiązania w systemie plików serwera plików OS/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 w systemie QFileSvr.400, używając dowiązania symbolicznego z systemu plików "root" (/), QOpenSys lub UDFS.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

### **Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików serwera plików OS/400**

Komendy, które zawiera sekcja "Wykonywanie operacji za pomocą komend CL" na stronie 24, działają w systemie plików QFileSvr.400. Wyjątkami są:

ADDLNK  
APYJRNCHG  
CHGAUT  
CHGOWN  
DSPAUT  
ENDJRN  
RST  
SAV  
SNDJRNE  
STRJRN  
WRKOBJOWN  
WRKOBJPGP

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23.

### **Używanie funkcji API zintegrowanego systemu plików w systemie plików serwer plików OS/400**

Funkcje API w języku C, które opisuje sekcja "Wykonywanie operacji za pomocą funkcji API" na stronie 45, działają w systemie plików QFileSvr.400 z następującymi wyjątkami:

chown()  
fchown()

givedescriptor()  
link()  
QjoEndJournal()  
QjoRetrieveJournalInformation()  
QJORJIDI()  
QJOSJRNE  
QjoStartJournal  
Qp0lGetPathFromFileID()  
symlink()  
takedescriptor()

---

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

System plików NFS zapewnia użytkownikom dostęp do danych i obiektów przechowywanych na zdalnym serwerze 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.

Więcej informacji na temat NFS można znaleźć w sekcji Korzystanie z systemów plików NFS poprzez interfejs zintegrowanego systemu plików.

## Korzystanie z systemów plików NFS poprzez interfejs zintegrowanego systemu plików

NFS jest dostępny poprzez interfejs zintegrowanego systemu plików, należy pamiętać jednak o wymienionych poniżej uwagach i ograniczeniach.

### Charakterystyka sieciowego systemu plików

Charakterystyka każdego systemu plików dołączonego przez NFS zależy od tego, jaki system plików 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, działa 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 NFS funkcjonuje inaczej i inna jest jego charakterystyka:

1. użytkownik podłącza system plików z serwera iSeries do klienta,
2. użytkownik podłącza system plików z serwera UNIX do klienta,
3. użytkownik podłącza system plików z innego serwera do klienta.

W pierwszym przypadku podłączony system plików zachowuje się na kliencie podobnie, jak zachowywałby się na serwerze iSeries. 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 miał charakterystyki i ograniczenia systemu plików QDLS. Na przykład, w systemie plików QDLS składniki nazwy ścieżki są ograniczone do 8 znaków i 3 znaków rozszerzenia. Jednak podłączony system plików będzie miał także ograniczenia i charakterystykę systemu plików NFS. Na przykład, nie będzie można użyć komendy CHGAUD do zmiany wartości kontroli obiektu NFS.

W drugim przypadku warto zwrócić uwagę, że każdy system plików przyłączony z serwera UNIX zachowuje się podobnie do systemu plików QOpenSys serwera iSeries. Więcej informacji na temat systemu plików QOpenSys znajduje się w sekcji "System plików systemów otwartych (QOpenSys)" na stronie 62.

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

### Dowiązania w sieciowym systemie plików

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.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 17.

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

Wszystkie komendy przedstawione w sekcji "Wykonywanie operacji za pomocą komend CL" na stronie 24 i ekrany opisane w sekcji "Wykonywanie operacji za pomocą menu i ekranów serwera iSeries" na stronie 23 działają również w sieciowym systemie plików (NFS), z wyjątkiem następujących:

- APYJRNCHG
- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN

Istnieją jednak komendy odnoszące się tylko do pewnych podłączonych systemów plików, w tym do NFS. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego. Poniższa tabela opisuje te komendy. Pełny opis komend i ekranów odnoszących się do systemu Network File System

zawiera publikacja OS/400 Network File System Support  .

Tabela 7. Komendy CL dotyczące systemu plików NFS

Komenda	Opis
ADDMFS	Dodanie podłączonego FS (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.
CHGNFSEXP	Zmiana eksportu NFS (Change Network File System Export). Dodaje drzewa katalogów do lub usuwa je z tabeli eksportu systemów plików, które są eksportowane do klientów NFS.



Tabela 7. Komendy CL dotyczące systemu plików NFS (kontynuacja)


Komenda	Opis
DSPMFSINF	Wyświetlenie danych podłączonego FS (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	Eksport systemu plików (Export a File System). Dodaje drzewa katalogów do lub usuwa je z tabeli eksportu systemów plików, które są eksportowane do klientów NFS.
MOUNT	Podłączenie systemu plików (Mount a File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. Jest to alternatywna postać komendy ADDMFS.
RLSIFSLCK	Zwolnienie blokad zintegrowanego systemu plików (Release Integrated File System Locks). Zwalnia wszystkie blokady na poziomie bajtów sieciowego systemu plików posiadane przez klienta lub dotyczące obiektu.
RMVMFS	Usunięcie dołączonego systemu plików (Remove Mounted File System). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnej przestrzeni nazw klienta.
STRNFSSVR	Uruchomienie serwera NFS (Start NFS Server). Uruchamia jeden lub wszystkie demony NFS w serwerze.
UNMOUNT	Odlączenie systemu plików (Unmount a file system). Usuwa wyeksportowane systemy plików zdalnego serwera z lokalnej przestrzeni nazw klienta. Jest to alternatywna postać komendy RMVMFS.

**Uwaga:** Aby komendy zintegrowanego systemu plików mogły wykonywać działania na systemie plików NFS, musi on być podłączony.

### Korzystanie z funkcji API zintegrowanego systemu plików w sieciowym systemie plików

Wszystkie funkcje w języku C, które wymienione są w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 45 mogą pracować w sieciowym systemie plików, z wyjątkiem następujących:

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

Pełen opis funkcji języka C, które dotyczą systemu plików NFS można znaleźć w książce OS/400 Network File System Support .

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



---

## Rozdział 7. Obsługa kronikowania dla obiektów zintegrowanego systemu plików

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu.

W niniejszej sekcji przedstawiono krótki przegląd zarządzania kronikami, a także uwagi dotyczące kronikowania obiektów zintegrowanego systemu plików i opis obsługi kronikowania dla obiektów zintegrowanego systemu plików.

Obsługę kronikowania dla obiektów zintegrowanego systemu plików opisano w następujących sekcjach:

- “Zarządzanie kronikami”
- “Jakie obiekty należy uwzględnić w kronikowaniu”
- “Kronikowane obiekty zintegrowanego systemu plików” na stronie 90
- “Kronikowane operacje” na stronie 91
- “Specjalne zagadnienia dotyczące pozycji kronik” na stronie 91

Szczegółowe informacje na temat kronikowania obiektów zintegrowanego systemu plików można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries.

---

### 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 przy testowaniu programów użytkowych.

Za pomocą kroniki można zdefiniować, które obiekty zostaną zabezpieczone poprzez kronikowanie. Więcej uwag dotyczących kronikowania obiektów zawiera sekcja “Jakie obiekty należy uwzględnić w kronikowaniu”. W zintegrowanym systemie plików można kronikować pliki strumieniowe, katalogi i dowiązania symboliczne. Obsługiwane są tylko obiekty następujących systemów plików: bazowego (root, “/”), QOpenSys i UDFS.

---

### Jakie obiekty należy uwzględnić w kronikowaniu

Przed podjęciem decyzji o kronikowaniu obiektu zintegrowanego systemu plików należy rozważyć następujące zagadnienia:

- Jak bardzo obiekt się zmienia? Dobrymi kandydatami do kronikowania są obiekty, które ulegają znacznym zmianom pomiędzy kolejnymi składowaniami.
- Jak trudno byłoby odtworzyć zmiany obiektu? Jak wiele zmian obiektu nie ma formy pisemnej? Trudniej na przykład odtworzyć obiekt wykorzystywany dla rejestracji zamówień zgłaszanych przez telefon niż obiekt używany dla rejestracji zamówień składanych pisemnie lub w innej formie.
- Jakie znaczenie mają informacje zapisane w obiekcie? Jeśli obiekt musiałby zostać odtworzony w postaci zapisanej podczas ostatniej operacji składowania, jaki wpływ na działanie firmy miałyby opóźnienie spowodowane rekonstrukcją zmian?
- W jaki sposób obiekt jest powiązany z innymi obiektami na serwerze? Chociaż dane w konkretnym obiekcie mogą nie zmieniać się zbyt często, to jednak mogą one mieć krytyczne znaczenie dla innych, bardziej dynamicznych obiektów na serwerze. Na przykład wiele obiektów zależy od zbioru głównego klientów. Podczas rekonstrukcji zamówienia, główny zbiór klientów musi uwzględniać nowych klientów i zmiany limitów kredytowych wprowadzone od czasu ostatniej operacji składowania.

---

## Kronikowane obiekty zintegrowanego systemu plików

Obsługa kronikowania w systemie OS/400 umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików. Obsługiwane typy obiektów to 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 iSeries Navigator. iSeries Navigator pozwala rozpocząć i zakończyć kronikowanie oraz wyświetlić informacje o kronikowaniu.

**Uwaga:** Odzworowane w pamięci pliki strumieniowe oraz pliki strumieniowe wykorzystywane przez Integrated xSeries Server for iSeries (IXS) jako przestrzeń dyskowa napędu wirtualnego nie mogą być kronikowane.

Poniższa lista przedstawia podsumowanie dotyczące 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 te metody jednocześnie.
- Niektóre komendy kronikowania, w tym Uruchomienie kronikowania (Start Journaling), Zakończenie kronikowania (End Journaling) i Zastosowanie kronikowanych zmian (Apply Journaled Changes), mogą dotyczyć 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 Journaling) 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:** Jeśli kronikowanie obiektu zostanie zakończone, a następnie zmieniona zostanie nazwa obiektu w katalogu, w którym się on znajduje, kronikowanie obiektu nie będzie rozpoczęte, nawet jeśli katalog ma ustawiony dziedziczony bieżący atrybut stanu kronikowania.

- 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.
- Maksymalny limit zapisu obsługiwany przez funkcje API write() i writev() wynosi 2 gigabajty-1. Maksymalna wielkość pozycji kroniki, o ile wartość RCVSIZOPT (\*MAXOPT2) jest określona, wynosi 4 000 000 000. W przeciwnym razie maksymalna wielkość pozycji kroniki wynosi 15 761 440 bajtów. W przypadku zapisów do kroniki pliku strumieniowego przekraczających 15 761 440 bajtów, aby uniknąć błędów, konieczna jest obsługa \*MAXOPT2.

Szczegółowe informacje na temat kronikowania obiektów zintegrowanego systemu plików można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries.

Szczegóły dotyczące zawartości i formatów danych właściwych dla pozycji kronik zintegrowanego systemu plików, a także informacje o różnych układach pozycji kronik zawiera plik w języku C o nazwie qp0ljrn1.h, dostarczany w podzbiorze QSYSINC/H (QP0LJRN1).

Pełną listę wszystkich pozycji kroniki dla obiektów zintegrowanego systemu plików zawiera publikacja Journal code finder w Centrum informacyjnym iSeries.

---

## 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,
- 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,
- zakończenie kronikowania,
- uruchomienie komendy Zastosowanie kronikowanych zmian (Apply Journalled Changes - APYJRNCHG),
- zakończenie komendy Zastosowanie kronikowanych zmian (Apply Journalled Changes - APYJRNCHG),
- składowanie,
- odtworzenie.

Szczegółowe informacje na temat kronikowania obiektów zintegrowanego systemu plików można znaleźć w sekcji Zarządzanie kronikami w Centrum informacyjnym iSeries. Pełną listę wszystkich pozycji kroniki dla obiektów zintegrowanego systemu plików zawiera publikacja Journal code finder w sekcji **Zarządzanie systemem**.

---

## Specjalne zagadnienia dotyczące pozycji kronik

Wiele kronikowanych operacji w systemie zintegrowany system 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/Przesunięcie (Rename/Move),
- B4 — Usunięcie dowiązania (Katalog nadrzędny) (Unlink (Parent Directory)),
- B5 — Usunięcie dowiązania (Dowiązanie) (Unlink (Link)),
- 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 replikują obiekty lub ponawiają operację, są zwykle zainteresowane tylko pozycjami zawierającymi podsumowania.

Dla operacji tworzenia w katalogu kronikowanym pozycją zawierającą podsumowanie jest pozycja kroniki B1 (Utworzenie podsumowania - Create Summary).

Niektóre operacje kronikowane muszą wysłać 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.

---

## Dodatek A. Wywołanie TI-RPC (Transport Independent Remote Procedure Call)

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 wielu typom maszyn udostępnia przesyłanie 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 pomiędzy protokołami. Jedynymi protokołami dostępnymi obecnie na serwerze iSeries 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 interfejsy pobierania danych.

---

### Wybór 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.
freenetconfigent()	Zwalnia strukturę netconfig, która została zwrócona na skutek wywołania do funkcji getnetconfigent().
getnetconfig()	Zwraca wskaźnik do bieżącego rekordu w pliku netconfig i ustawia wskaźnik na następny rekord.
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().

---

### Konwersja 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.

<b>Funkcja API</b>	<b>Opis</b>
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).

## Zewnętrzna reprezentacja danych (XDR)

Poniższe funkcje API umożliwiają aplikacjom RPC (Remote Procedure Call) obsługę dowolnych struktur danych, bez względu na różnice pomiędzy hostami dotyczące uporządkowania bajtów lub sposobu rozmieszczenia składowych struktury.

<b>Funkcja API</b>	<b>Opis</b>
xdr_array()	Filtr, który dokonuje konwersji pomiędzy 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 pomiędzy zmiennymi typu Boolean (zmienne całkowite w języku C) i ich zewnętrznymi reprezentacjami. Podczas kodowania danych filtr ten tworzy wartości równe 1 lub 0.
xdr_bytes()	Filtr, który dokonuje konwersji pomiędzy 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 pomiędzy znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_double()	Filtr, który dokonuje konwersji pomiędzy liczbami o podwójnej precyzji języka C i ich zewnętrznymi reprezentacjami.
xdr_double_char()	Filtr, który dokonuje konwersji pomiędzy dwubajtowymi znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_enum()	Filtr, który dokonuje konwersji pomiędzy 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 pomiędzy liczbami zmiennoprzecinkowymi (znormalizowane liczby zmiennoprzecinkowe pojedynczej precyzji) języka C i ich zewnętrznymi reprezentacjami.
xdr_int()	Filtr, który dokonuje konwersji pomiędzy liczbami całkowitymi języka C i ich zewnętrznymi reprezentacjami.
xdr_long()	Filtr, który dokonuje konwersji pomiędzy długimi liczbami całkowitymi (long integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_netobj()	Filtr, który dokonuje konwersji pomiędzy nieprzezroczystymi (opaque) danymi o zmiennej długości i ich zewnętrznymi reprezentacjami.
xdr_opaque()	Filtr, który dokonuje konwersji pomiędzy 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.



<b>Funkcja API</b>	<b>Opis</b>
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 pomiędzy krótkimi liczbami całkowitymi (short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_string()	Filtr, który dokonuje konwersji pomiędzy ciągami znaków języka C i ich zewnętrznymi reprezentacjami.
xdr_u_char()	Filtr, który dokonuje konwersji pomiędzy znakami języka C (unsigned char) i ich zewnętrznymi reprezentacjami.
xdr_u_int()	Filtr, który dokonuje konwersji pomiędzy liczbami całkowitymi bez znaku (unsigned integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_long()	Filtr, który dokonuje konwersji pomiędzy długimi liczbami całkowitymi bez znaku (unsigned long integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_short()	Filtr, który dokonuje konwersji pomiędzy krótkimi liczbami całkowitymi bez znaku (unsigned short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_union()	Filtr, który dokonuje konwersji pomiędzy uniami języka C i ich zewnętrznymi reprezentacjami.
xdr_vector()	Filtr, który dokonuje konwersji pomiędzy 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.

## Uwierzytelnienie

Poniższe funkcje API zapewniają uwierzytelnienie aplikacji TI-RPC (Transport Independent Remote Procedure Call).

<b>Funkcja API</b>	<b>Opis</b>
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.

## TI-RPC (Transport Independent RPC)

Poniższe funkcje API udostępniają środowisko projektowania aplikacji rozproszonych poprzez odizolowanie aplikacji od specyfiki funkcji transportu. Ułatwia to korzystanie z transportu.

## Uproszczone funkcje API aplikacji TI-RPC

Poniższe uproszczone funkcje API określają typ transportu, którego należy użyć. Aplikacje używające tego poziomu nie muszą jawnie tworzyć uchwytów.

<b>Funkcja API</b>	<b>Opis</b>
rpc_call()	Wywołuje zdalną procedurę w podanym systemie.
rpc_reg()	Rejestruje procedurę z pakietem usług RPC.

## Funkcje API poziomu najwyższego aplikacji TI-RPC

Poniższe funkcje API umożliwiają aplikacji podanie typu transportu.

<b>Funkcja API</b>	<b>Opis</b>
clnt_call()	Wywołuje zdalną procedurę związaną z klientem.
clnt_control()	Zmienia informacje na temat obiektu klienta.
clnt_create()	Tworzy uniwersalny uchwyt klienta.
clnt_destroy()	Niszczy uchwyt RPC klienta.
svc_create()	Tworzy uchwyt serwera.
svc_destroy()	Niszczy uchwyt transportu usługi RPC.

## Funkcje API poziomu średniego 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.

## Funkcje API poziomu eksperckiego 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().

## Inne funkcje API aplikacji 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>
<code>clnt_freeres()</code>	Zwalnia dane przydzielone przez system RPC lub XDR.
<code>clnt_geterr()</code>	Pobiera strukturę błędu z uchwytu klienta.
<code>svc_freeargs()</code>	Zwalnia dane przydzielone przez system RPC lub XDR.
<code>svc_getargs()</code>	Dekoduje argumenty żądania RPC.
<code>svc_getrpccaller()</code>	Pobiera adres sieciowy wywołującego.
<code>svc_run()</code>	Czeka na żądania RPC.
<code>svc_sendreply()</code>	Wysyła wyniki wywołania procedury do zdalnego klienta.
<code>svcerr_decode()</code>	Wysyła informacje do klienta związane z błędem dekodowania.
<code>svcerr_noproc()</code>	Wysyła do klienta informacje związane z błędem numerycznym procedury.
<code>svcerr_systemerr()</code>	Wysyła do klienta informacje związane z błędem systemowym.



---

## Dodatek B. Przykładowy program wykorzystujący 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 zbiór (**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.

Zastrzeżenia te odnoszą się do przykładów kodu.

```
#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
/* Tworzenie 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;
}

```

---

## Dodatek C. Przykład programu w języku RPG dla zintegrowanego systemu plików

Code Snippets  zawiera przykład kodu zintegrowanego systemu plików RPG. Aby obejrzeć ten przykład, wykonaj poniższe czynności:








1. Z rozwijanej listy w kategorii Search wybierz **ILE RPG Source**.
2. Kliknij **Search**.
3. Przewiń listę, aż pojawi się pozycja **Using IFS from RPG**.
4. Kliknij **Code for using IFS from RPG**.

Zastrzeżenia te odnoszą się do przykładów kodu.



## Bibliografia

Bibliografia ta obejmuje publikacje dotyczące serwera iSeries zawierające podstawowe wiadomości oraz dodatkowe szczegóły związane z zagadnieniami poruszonymi w niniejszej książce.

- Sekcja Język CL w kategorii **Programowanie**, w Centrum informacyjnym iSeries zawiera opis języka CL serwera iSeries i jego komend. Opis każdej komendy obejmuje diagram składniowy, parametry, wartości domyślne, słowa kluczowe i przykład.
- Temat Globalizacja w Centrum informacyjnym iSeries wyjaśnia pojęcia związane z obsługą języków narodowych, takie jak zestaw znaków i strona kodowa, oraz dostarcza informacji potrzebnych do oceny, planowania i korzystania z NLS i możliwości obsługi wielu języków na serwerze iSeries.
- Sekcja API w kategorii **Programowanie** w Centrum informacyjnym iSeries zawiera opis każdej funkcji OS/400 API, włączając w to funkcje API zintegrowanego systemu plików.
- Sekcja Journal management w kategorii **Zarządzanie systemem** w Centrum informacyjnym iSeries zawiera informacje o sposobie konfiguracji, zarządzania i rozwiązywania problemów z ochroną SMAPP (system-managed access-path protection), lokalnymi kronikami i kronikami zdalnymi na serwerze iSeries.
- Sekcja Kontrola transakcji w kategorii **Baza danych** w Centrum informacyjnym iSeries wyjaśnia, jak zdefiniować i przetwarzać grupę zmian w zasobach, takich jak zbiory baz danych, pliki zintegrowanego systemu plików, jako logiczną jednostkę roboczą.
- OS/400 Network File System Support  Książka ta opisuje sieciowy system plików (Network File System) posługując się przykładami prawdziwych zastosowań. Zawarto w niej informacje na temat eksportu, dołączania, blokowania plików i na temat ochrony. Można się z niej dowiedzieć, jak używać systemu NFS do zbudowania i utrzymywania bezpiecznego sieciowego systemu plików.
- Optical Support  Książka ta jest jednocześnie przewodnikiem użytkownika i instrukcją obsługi do nośników optycznych IBM w systemie OS/400. Zawarte w niej informacje powinny pomóc użytkownikom zrozumieć koncepcję serwera danych biblioteki optycznej, zaplanować administrowanie biblioteką optyczną i serwerem danych biblioteki optycznej oraz rozwiązywać dotyczące ich problemy.
- WebSphere Development Studio: ILE C/C++  
Programmers Guide  Książka ta zawiera informacje niezbędne do projektowania, edycji, kompilacji, uruchamiania oraz debugowania programów ILE C/400 na serwerze iSeries.
- WebSphere Development Studio: C/C++  
Language Reference  Książka ta zawiera informacje o strukturze programów ILE C/400, jak również szczegóły dotyczące bibliotek funkcji i plików nagłówkowych (include).
- Security — Reference  Książka ta zawiera szczegółowe informacje techniczne o ochronie w systemie OS/400.
- APPC Programming  W książce tej opisano obsługę zaawansowanej komunikacji program-program (APPC) na serwerze iSeries. Jest to przewodnik po programach do rozwoju aplikacji korzystających z APPC podczas definiowania środowiska komunikacyjnego dla APPC.
- Składowanie i odtwarzanie  Ta książka zawiera ogólne informacje o opcjach odzyskiwania i dostępności dla serwera iSeries.

## Bibliografia

# Indeks

## A

- atrybuty rozszerzone
  - ciągłość pomiędzy językami narodowymi 22, 55
  - definicja 21
  - zasady nazewnictwa 21

## B

- bezwzględna ścieżka 16
- bibliografia 107
- biblioteki, system plików (QSYS.LIB)
  - charakterystyka i ograniczenia 67
  - opis 4
- bieżący katalog 7
- binarny tryb otwarcia pliku 55

## C

- Client Access 31

## D

- dane, konwersja 55
- definiowany przez użytkownika, system plików (UDFS) 4
  - charakterystyka i ograniczenia 63
  - opis 4
- deskryptor pliku 53
- dostęp, tryb 54
- dowiązanie
  - definicja 17
  - komendy 24
  - menu i ekrany 24
  - porównanie 20
  - stałe 18
  - symboliczne 19
  - w programie przykładowym 99
  - zastosowanie 17
  - zastosowanie w systemie plików / (root) 61
  - zastosowanie w systemie plików QDLS 74
  - zastosowanie w systemie plików QFileSvr.400 84
  - zastosowanie w systemie plików QNTC 80
  - zastosowanie w systemie plików QOpenSys 62
  - zastosowanie w systemie plików QOPT 75
  - zastosowanie w systemie plików QSYS.LIB 69
  - zastosowanie w systemie plików QSYS.LIB niezależnej puli ASP 72
- dowiązanie symboliczne
  - definicja 19
  - porównanie z dowiązaniem stałym 20
  - przykład zastosowanie 19

## E

- ekrany
  - reguły dotyczące ścieżek 27

- ekrany (*kontynuacja*)
  - zastosowanie 23

## F

- foldery
  - system plików QDLS 4, 72
- FTP 30
- funkcja API
  - ILE C/400 51
  - przykładowy program 99
  - reguły dotyczące ścieżek 52
  - zastosowanie w programach w języku C 41, 45
- funkcje
  - ILE C/400 51
  - reguły dotyczące ścieżek 52
  - w programie przykładowym 99
  - zastosowanie w programach w języku C 41, 45

## G

- gniazdo 54

## H

- hierarchiczny system plików (HFS)
  - korzystanie z funkcji API dla systemu plików QDLS 73
  - korzystanie z funkcji API dla systemu plików QOPT 75

## I

- ILE C/400
  - alternatywy API 45
  - funkcje ANSI 51
- interfejs użytkownika
  - komendy 24
  - menu i ekrany 23
  - widok z PC 32
- interfejs zintegrowanego systemu plików 1, 2, 5

## K

- katalog
  - bieżący 7
  - definicja 6
  - komendy 24
  - menu i ekrany 24
  - osobisty 7
  - w programie przykładowym 99
  - zalety 1
  - zintegrowany system plików 33
- katalog osobisty 7
- komendy
  - lista 24
  - reguły dotyczące ścieżek 27

komendy (*kontynuacja*)  
zastosowanie 24  
komputer osobisty typu klient  
praca ze zintegrowanym systemem plików 29  
sposób prezentacji systemów plików 32  
konwersja  
danych 55  
nazwy obiektów 22, 55  
konwersja znaków 2, 22, 55  
Kronikowanie  
Start 40  
Zakończenie 40  
Kronikowanie obiektów zintegrowanego systemu  
plików 89

## L

LU 6.2 w systemie plików QFileSvr.400 82

## M

menu  
reguły dotyczące ścieżek 27  
zastosowanie 23  
migracja między systemami plików 32

## N

nazwy  
ciągłość pomiędzy językami narodowymi 55  
niezmiennosc pomiędzy schematami kodowania 22  
zastosowanie w systemie plików / (root) 61  
zastosowanie w systemie plików QDLS 73  
zastosowanie w systemie plików QFileSvr.400 82  
zastosowanie w systemie plików QNTC 80  
zastosowanie w systemie plików QOpenSys 62  
zastosowanie w systemie plików QOPT 75  
zastosowanie w systemie plików QSYS.LIB 69  
zastosowanie w systemie plików QSYS.LIB  
niezależnej puli ASP 71  
NetServer 31  
nośniki optyczne, system plików (QOPT)  
charakterystyka i ograniczenia 74  
opis 5

## O

obiekt  
migracja między systemami plików 32  
obsługa języka narodowego 2, 22, 55  
obszar użytkownika  
zastosowanie w systemie plików QSYS.LIB 68  
zastosowanie w systemie plików QSYS.LIB  
niezależnej puli ASP 71  
ochrona  
komendy 24  
ograniczenia systemu plików QNTC 79  
ograniczenia w systemie plików QFileSvr.400 83  
w programach używających zintegrowanego systemu  
plików 54

operacje (program przykładowy) 99  
otwieranie deskryptora pliku 53

## P

plik 99  
menu i ekrany 24  
przesyłanie 30  
tryby otwarcia 55  
plik strumieniowy  
definicja 3  
kopiowanie do/ze zbioru bazy danych 42  
porównanie z plikiem opartym na rekordach 3  
w programie przykładowym 99  
wskazówki dotyczące wykorzystania w ILE C/400 51  
zalety 1  
zastosowanie 3  
zastosowanie w programach 41  
procesor wejścia/wyjścia serwera plików 5  
program w języku C  
funkcje ILE C/400 51  
przykład 99  
protokół FTP 30  
przykład  
program używający API zintegrowanego systemu  
plików 99  
ścieżki 16, 27, 52  
użycie dowiązania symbolicznego 19

## Q

QFileSvr.400 5  
QFileSvr.400, system plików  
charakterystyka i ograniczenia 81  
opis 5  
QNetWare, system plików 5  
charakterystyka i ograniczenia 76  
opis 5  
QOpenSys, system plików  
charakterystyka i ograniczenia 62  
opis 4  
QOPT, system plików 5  
charakterystyka i ograniczenia 74  
opis 5  
QSYS.LIB niezależnej ASP  
charakterystyka i ograniczenia 70  
QSYS.LIB, system plików  
charakterystyka i ograniczenia 67  
opis 4

## R

roboczy katalog 7  
root (/), bazowy system plików  
charakterystyka i ograniczenia 60

## S

schematy kodowania 22, 55  
serwer plików 5



- serwer plików OS/400 5
- serwer plików OS/400, system plików (QFileSvr.400)
  - charakterystyka i ograniczenia 81
  - opis 5
- serwer plików PC 5
- sieciowy system plików (NFS) 5
  - charakterystyka i ograniczenia 85
  - opis 5
- stałe dowiązanie
  - definicja 18
  - porównanie z dowiązaniem symbolicznym 20
- strona kodowa 2, 22, 55
- system plików
  - bibliotek (QSYS.LIB)
    - charakterystyka i ograniczenia 67
    - opis 4
  - definicja 4
  - definiowany przez użytkownika, (UDFS)
    - charakterystyka i ograniczenia 63
  - definiowany przez użytkownika, system plików (UDFS)
    - opis 4
  - interfejs 5
  - migracja obiektów 32
  - NetWare (QNetWare)
    - charakterystyka i ograniczenia 76
  - nośników optycznych (QOPT)
    - charakterystyka i ograniczenia 74
    - opis 5
  - obsługujący biblioteki dokumentów (QDLS)
    - charakterystyka i ograniczenia 72
    - opis 4
  - porównanie 57
  - przesyłanie plików 30
  - QFileSvr.400
    - opis 5
  - QNetWare
    - opis 5
  - QNTC
    - charakterystyka i ograniczenia 79
  - QSYS.LIB niezależnej ASP
    - charakterystyka i ograniczenia 70
    - opis 4
  - root (/)
    - charakterystyka i ograniczenia 60
    - opis 4
  - serwera plików OS/400 (QFileSvr.400)
    - charakterystyka i ograniczenia 81
  - sieciowy system plików (NFS)
    - charakterystyka i ograniczenia 85
    - opis 5
  - system plików QNTC
    - opis 5
  - systemów otwartych (QOpenSys)
    - charakterystyka i ograniczenia 62
    - opis 4
  - zalety 2
- system plików NetWare (QNetWare)
  - charakterystyka i ograniczenia 76

- system plików obsługujący biblioteki dokumentów (QDLS)
  - charakterystyka i ograniczenia 72
  - opis 4
- system plików QDLS
  - charakterystyka i ograniczenia 72
  - opis 4
- system plików QNTC 5
  - charakterystyka i ograniczenia 79
  - opis 5
- system plików QSYS.LIB niezależnej ASP
  - opis 4
- system plików root (/)
  - opis 4
- system plików serwera Windows NT (QNTC)
  - charakterystyka i ograniczenia 79
- system plików systemów otwartych (QOpenSys)
  - charakterystyka i ograniczenia 62
  - opis 4

## Ś

- ścieżka
  - bezwzględna ścieżka 16
  - definicja 16
  - reguły dotyczące API 52
  - reguły dotyczące komend i ekranów 27
  - względna ścieżka 17
  - zastosowanie w systemie plików / (root) 61
  - zastosowanie w systemie plików QDLS 74
  - zastosowanie w systemie plików QFileSvr.400 82
  - zastosowanie w systemie plików QNTC 80
  - zastosowanie w systemie plików QOpenSys 62
  - zastosowanie w systemie plików QOPT 75
  - zastosowanie w systemie plików QSYS.LIB 69
  - zastosowanie w systemie plików QSYS.LIB niezależnej puli ASP 72

## T

- TCP/IP w systemie plików QFileSvr.400 82
- tekstowy tryb otwarcia pliku 55
- tryb dostępu 54
- tryby otwarcia pliku 55

## U

- Unicode 22
- uprawnienia 54
  - komendy 24
  - ograniczenia systemu plików QNTC 79
  - ograniczenia w systemie plików QFileSvr.400 83
  - w programach używających zintegrowanego systemu plików 54
  - w programie przykładowym 99

## W

- wskaźniki 53
- względna ścieżka 17

## Z

- zbiory składowania
  - zastosowanie w systemie plików QSYS.LIB 68
  - zastosowanie w systemie plików QSYS.LIB niezależnej puli ASP 71
- zbiór bazy danych
  - kopiowanie do/z pliku strumieniowego 42
  - porównanie z plikiem strumieniowym 3
  - tworzenie z pliku strumieniowego 42
- zintegrowany system plików
  - definicja 1
  - interfejsy programistyczne
    - obsługa języka narodowego 55
    - ochrona 54
    - przykładowy program 99
    - reguły dotyczące ścieżek 52
    - wskaźniki i deskryptory plików 53
    - zastosowanie w programach w języku C 45
  - komendy
    - lista 24
    - reguły dotyczące ścieżek 27
    - zastosowanie 24
  - menu i ekrany
    - reguły dotyczące ścieżek 27
    - zastosowanie 23
  - praca na komputerze PC
    - sposób prezentacji systemów plików 32
  - praca z komputera PC
    - praca z katalogami i obiektami 29
    - zalety 1
- zintegrowany system plików, drukowane informacje pokrewne 107



**IBM**