



Systemy IBM - iSeries

Pliki i systemy plików

Zintegrowany system plików

*Wersja 5 Wydanie 4*







Systemy IBM - iSeries

Pliki i systemy plików

Zintegrowany system plików

*Wersja 5 Wydanie 4*

**Uwaga**

Przed korzystaniem z poniższych informacji oraz produktu, którego dotyczą, należy przeczytać informacje znajdujące się w dodatku “Uwagi”, na stronie 141.

**Wydanie siódme (luty 2006)**

Niniejsze wydanie dotyczy Wersji 5, Wydania 4, Modyfikacji 0 systemu operacyjnego IBM i5/OS (numer produktu 5722–SS1) oraz wszelkich kolejnych wersji i modyfikacji tego produktu, o ile nowe wydania nie wskazują inaczej. Wersja ta nie działa na wszystkich modelach komputerów z procesorem RISC ani na modelach z procesorem CISC.

© Copyright International Business Machines Corporation 1999, 2006. Wszelkie prawa zastrzeżone.

# Spis treści

<b>Zintegrowany system plików . . . . .</b>	<b>1</b>		Ponowne tworzenie obiektów zintegrowanego systemu plików . . . . .	103
Co nowego w wersji V5R4 . . . . .	1		Przykłady: komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) . . . . .	104
Drukowalne pliki PDF . . . . .	2		Obsługa programistyczna . . . . .	105
Przegląd zintegrowanego systemu plików . . . . .	2		Kopiowanie danych między plikami strumieniowymi i zbiorami baz danych . . . . .	105
Czym jest zintegrowany system plików . . . . .	2		Kopiowanie danych między plikami strumieniowymi a zbiorami składowania . . . . .	109
Dlaczego należy używać zintegrowanego systemu plików . . . . .	3		Wykonywanie operacji za pomocą funkcji API . . . . .	110
Pojęcia dotyczące zintegrowanego systemu plików . . . . .	4		Obsługa gniazd . . . . .	120
Katalog . . . . .	4		Nazewnictwo i obsługa różnych języków . . . . .	120
Dowiązanie . . . . .	12		Konwersja danych . . . . .	121
Nazwa ścieżki . . . . .	16		Przykład: funkcje C zintegrowanego systemu plików . . . . .	121
Plik strumieniowy . . . . .	17		Praca z plikami i folderami z wykorzystaniem produktu iSeries Navigator . . . . .	127
Niezmienność nazw . . . . .	19		Odblokowanie pliku . . . . .	127
Atrybuty rozszerzone . . . . .	20		Zablokowanie pliku . . . . .	127
Obsługa skanowania . . . . .	21		Tworzenie folderu . . . . .	127
Praca z systemami plików . . . . .	26		Usuwanie folderu . . . . .	127
Porównanie systemów plików . . . . .	28		Przenoszenie plików lub folderów do innego systemu plików . . . . .	128
System plików "root" (/) . . . . .	32		Konfigurowanie uprawnień . . . . .	129
System plików systemów otwartych (QOpenSys) . . . . .	34		Konfigurowanie konwersji plików tekstowych . . . . .	129
Systemy plików użytkownika (UDFS) . . . . .	36		Wysyłanie pliku lub folderu do innego systemu . . . . .	129
System plików bibliotek (QSYS.LIB) . . . . .	43		Zmiana opcji definicji pakietów . . . . .	130
Niezależne pule ASP systemu QSYS.LIB . . . . .	46		Planowanie daty i godziny wysyłki pliku lub folderu . . . . .	130
System plików usług biblioteki dokumentów (QDLS) . . . . .	49		Tworzenie współużytkowanego zasobu plikowego . . . . .	131
system plików nośników optycznych (QOPT) . . . . .	51		Zmiana współużytkowanego zasobu plikowego . . . . .	131
System plików NetWare (QNetWare) . . . . .	54		Tworzenie nowego systemu plików użytkownika . . . . .	131
system plików iSeries NetClient (QNTC) . . . . .	57		Podłączanie systemu plików utworzonego użytkownika . . . . .	131
System plików serwera plików i5/OS (QFileSvr.400) . . . . .	61		Odłączanie systemu plików definiowanego przez użytkownika . . . . .	132
sieciowy system plików (Network File System - NFS) . . . . .	65		Włączanie i wyłączanie skanowania obiektów . . . . .	132
Dostęp do zintegrowanego systemu plików . . . . .	68		zdalne wywołanie procedury niezależne od transportów . . . . .	133
Dostęp przy użyciu pozycji menu i ekranów . . . . .	68		Funkcje API wyboru sieci . . . . .	133
Dostęp przy użyciu komend języka CL . . . . .	69		Funkcje API konwersji nazwy na adres . . . . .	133
Dostęp przy użyciu funkcji API . . . . .	86		Zewnętrzna reprezentacja danych (XDR) . . . . .	134
Dostęp przy użyciu programu iSeries Navigator . . . . .	86		Uwierzytelnianie funkcji API . . . . .	135
Dostęp przy użyciu serwera iSeries NetServer . . . . .	87		TI-RPC (Transport Independent RPC) . . . . .	136
Dostęp przy użyciu protokołu FTP . . . . .	88		Informacje pokrewne dotyczące zintegrowanego systemu plików . . . . .	137
Dostęp przy użyciu komputera PC . . . . .	89			
Przekształcanie katalogów z formatu *TYPE1 na *TYPE2 . . . . .	90			
Przegląd konwersji formatu *TYPE1 do *TYPE2 . . . . .	90			
Uwagi dotyczące konwersji . . . . .	91			
Kronikowanie obiektów . . . . .	95			
Przegląd kronikowania . . . . .	95			
Uruchomienie kronikowania . . . . .	100			
Zmiana kronikowania . . . . .	100			
Zakończenie kronikowania . . . . .	100			
Odzyskiwanie systemów plików "root" (/), QOpenSys i UDFS . . . . .	101		<b>Dodatek. Uwagi . . . . .</b>	<b>141</b>
Porównanie komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) . . . . .	101		Informacje o interfejsie programistycznym . . . . .	142
Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) . . . . .	103		Znaki handlowe . . . . .	143
			Warunki . . . . .	143



---

## Zintegrowany system plików

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

**Uwaga:** Użycie poniższych przykładów kodu jest równoznaczne z zaakceptowaniem warunków “Informacje dotyczące licencji na kod” na stronie 138.

---

## Co nowego w wersji V5R4

W poniższym temacie opisane są zmiany dokonane w bieżącej kolekcji tematycznej w wersji V5R4.

### QNTC



- | QNTC obsługuje port TCP/IP 445 i większe pliki.
- | • QNTC obsługuje port TCP/IP 445
- | System plików QNTC umożliwia teraz komunikację między serwerami przy użyciu portu 445 protokołu TCP/IP.
- | Konfigurowanie protokołu NetBios przez TCP/IP na serwerach Windows nie jest konieczne.
- | • QNTC obsługuje większe pliki
- | Począwszy od wersji V5R4 system plików QNTC obsługuje odczyt i zapis do plików o wielkościach do 1 TB (około 1 099 511 627 776 bajtów).

### Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)

- | Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) umożliwia zidentyfikowanie i naprawienie uszkodzonych obiektów w systemach plików “root” (/), QOpenSts i i podłączonych systemach plików użytkownika; system nie musi znajdować się w stanie zastrzeżonym. Umożliwia to usunięcie problemów z systemami plików bez obniżania efektywności. Może być on w wielu sytuacjach używana jako alternatywa do komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG). Użycie komendy RCLLNK jest na przykład dobrym sposobem identyfikowania i usuwania problemów w następujących sytuacjach:
- | • Problemy są związane z jednym obiektem.
- | • Problemy są związane z grupą obiektów.
- | • Zachodzi potrzeba zidentyfikowania lub usunięcia uszkodzonych obiektów.
- | • System nie może znajdować się w stanie zastrzeżonym podczas operacji odzyskiwania.
- | • Niezależne pule ASP muszą być dostępne podczas operacji odzyskiwania.

### Jak zobaczyć, co jest nowe lub zmienione

Aby zmiany techniczne były lepiej widoczne, w niniejszych informacjach zastosowano następujące znaki:

- Symbol  oznacza początek nowych lub zmienionych informacji.
- Symbol  oznacza koniec tych informacji.

Więcej informacji dotyczących nowych lub zmienionych funkcji systemu można znaleźć w temacie Informacje dla użytkowników.

---

## Drukowalne pliki PDF

Poniższy temat umożliwia wyświetlenie i wydrukowanie pliku PDF zawierającego poniższe informacje.


Aby przejrzeć lub pobrać ten dokument w formacie PDF, kliknij Zintegrowane systemy plików  (około 1855 kB).

## Zapisywanie plików PDF

Aby zapisać plik PDF na stacji roboczej do przeglądania i wydruku, należy:

1. Kliknąć prawym przyciskiem myszy plik PDF widoczny w przeglądarce (kliknąć prawym przyciskiem powyższy odsyłacz do pliku).
2. Kliknij opcję zapisania pliku PDF lokalnie.
3. Przejdź do katalogu, w ma zostać zapisany plik PDF.
4. Kliknąć opcję **Zapisz**.

## Pobieranie programu Adobe Reader

- 1 Do wyświetlenia lub wydrukowania zawartości plików PDF niezbędny jest program Adobe Reader. Jego kopię można
- 1 pobrać z serwisu WWW firmy Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

## Przegląd zintegrowanego systemu plików

Opis zintegrowanego systemu plików na serwerze iSeries oraz możliwości jego wykorzystania.

## Czym jest zintegrowany system plików

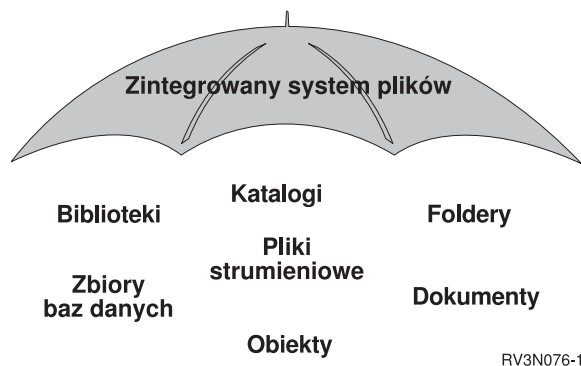
*Zintegrowany system plików* jest częścią systemu i5/OS, która umożliwia obsługę strumieniowych danych wejścia/wyjścia i zarządzanie pamięcią podobne do komputerów PC i systemów operacyjnych UNIX zapewniając strukturę integrującą wszystkie informacje przechowywane w serwerze.

Zintegrowany system plików składa się z 11 systemów plików, każdy z własnym zestawem struktur logicznych i reguł obsługi informacji w przestrzeni pamięci.

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.
- Ogólny przegląd plików strumieniowych przechowywanych lokalnie w serwerze, serwerze Integrated xSeries Server for iSeries lub zdalnym serwerze Windows NT. Pliki strumieniowe można przechowywać zdalnie w serwerze LAN, Novell NetWare, innym zdalnym serwerze iSeries lub serwerze NFS.





Rysunek 1. Ogólne informacje o strukturze informacji przechowywanych w serwerze iSeries

### Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

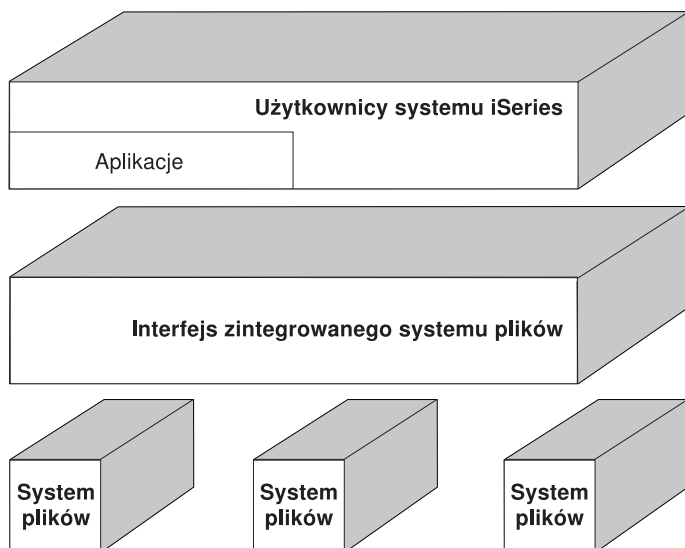
*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

## Dlaczego należy używać zintegrowanego systemu plików

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

Zintegrowany system plików umożliwia:

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



Rysunek 2. Wspólny interfejs dla różnych systemów plików.

- lepsze wykorzystanie graficznego interfejsu użytkownika przez użytkowników komputerów osobistych. Na przykład, użytkownicy systemu Windows mogą użyć narzędzi interfejsu graficznego Windows do obsługi pliku strumieniowego serwera iSeries i innych obiektów tak samo, jak obsługują pliki przechowywane w ich komputerach PC;
- zapewnienie spójności nazw obiektów i związanych z nimi informacji niezależnie od języka narodowego. Na przykład przy przejściu ze strony kodowej jednego języka na stronę kodową innego języka znaki specyficzne dla danego języka nie zostaną zmienione.

#### Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

## Pojęcia dotyczące zintegrowanego systemu plików

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

### Katalog

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

Zintegrowany system plików tworzy hierarchiczną strukturę katalogów umożliwiającą dostęp do wszystkich obiektów 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.

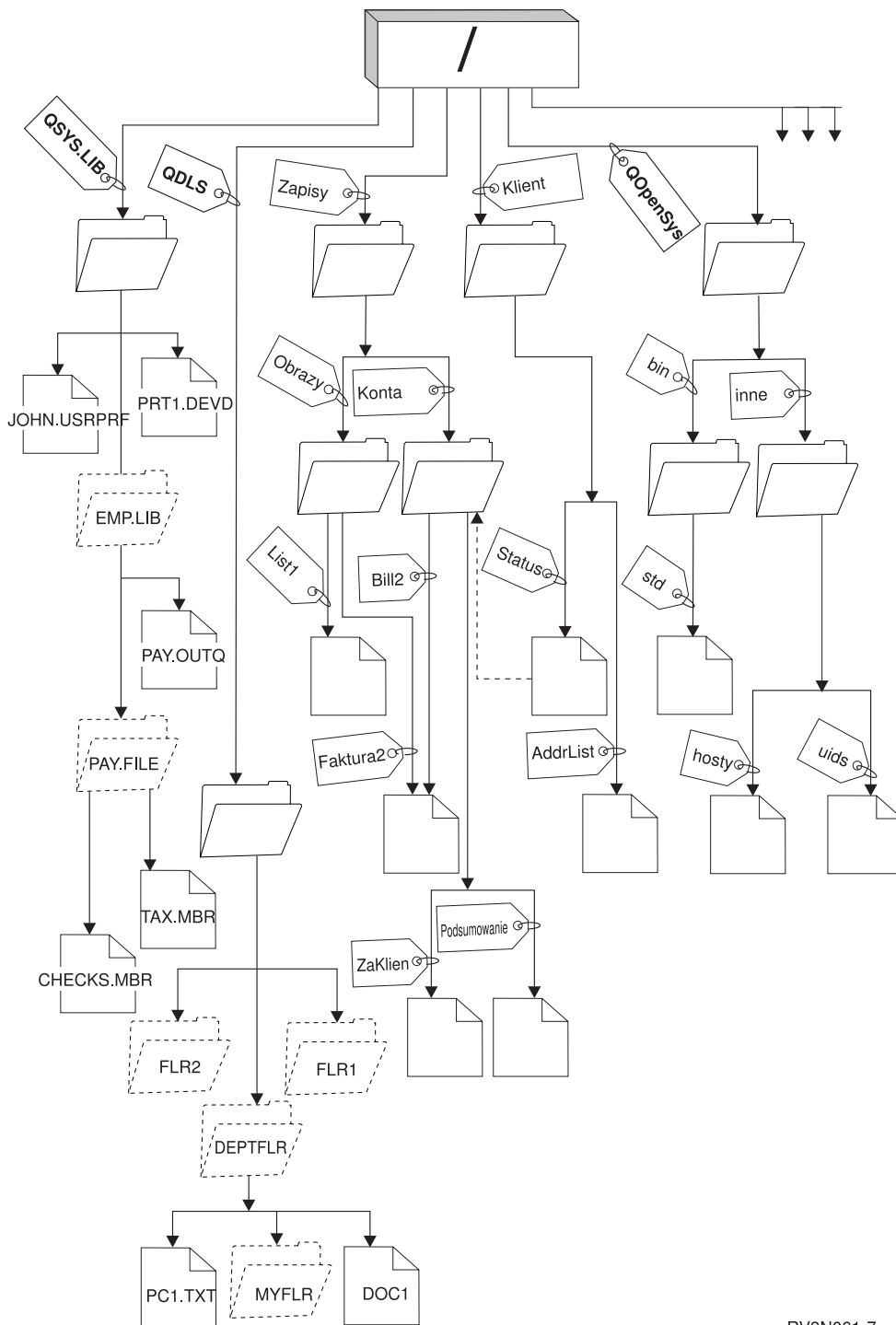
Pojedyncza gałąź katalogu, razem ze wszystkimi swoimi podrzędnymi gałęziami (podkatalogami) i ze wszystkimi obiektami dołączonymi do tych gałęzi, jest nazywana *poddrzewem*. Każdy system plików jest głównym poddrzewem w strukturze katalogów zintegrowanego systemu plików. W bibliotece QSYS.LIB oraz w poddrzewach systemu plików 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 (poddziewo) foldery obsługiwane są podobnie jak podkatalogi, a dokumenty w folderach podobnie jak obiekty w podkatalogu.

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

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

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



RV3N061-7

Rysunek 3. Przykładowe drzewo katalogów zintegrowanego systemu plików

## Katalog bieżący

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

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

## Katalog osobisty

*Katalog osobisty (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 `fhdir()`.

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

### Pojęcia pokrewne

"Utworzone katalogi"

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, poniższe katalogi.

### Informacje pokrewne

Komenda Zmiana bieżącego katalogu (Change current directory - CHGCURDIR)

`chdir()`

`fhdir()`

Aplikacyjne interfejsy programistyczne (API)

## Utworzone katalogi

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, poniższe katalogi.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**/home** Administratorzy systemu tworzą w katalogu /home katalogi osobiste wszystkich użytkowników. Administrator systemu zwykle zakłada użytkownikowi katalog osobisty o nazwie zgodnej z profilem użytkownika, na przykład /home/jasio.

**/etc** W katalogu `/etc` przechowywane są pliki używane do administrowania systemem, pliki konfiguracyjne i inne pliki systemowe.

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

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

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

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

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

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

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

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

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

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

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

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

### Pojęcia pokrewne

“Katalog osobisty” na stronie 7

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

### Odsyłacze pokrewne

“Urządzenia UDP i TCP w systemie plików `root` (`/`)” na stronie 34

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

“System plików systemów otwartych (`QOpenSys`)” na stronie 34

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

“Ponowne tworzenie obiektów zintegrowanego systemu plików” na stronie 103

Poniższa tabela zawiera obiekty zintegrowanego systemu plików, które są ponownie tworzone przez komendę `Odzyskiwanie dowiązań do obiektów (Reclaim Object Links - RCLLNK)`, jeśli nie istnieją. Są one zazwyczaj



tworzone podczas ładowania programu początkowego (IPL). Istnieje również możliwość ponownego utworzenia w razie potrzeby niektórych z tych obiektów przy użyciu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

### Informacje pokrewne

Komenda WRKLNK

## Katalogi \*TYPE2

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

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

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

Zalety katalogów \*TYPE2 są następujące:

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

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

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

Katalogi \*TYPE2 oferują dodatkowo następujące funkcje:

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

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

Istnieje kilka sposobów tworzenia katalogów \*TYPE2 w systemie:

- Nowe serwery iSeries o fabrycznie zainstalowanych systemach OS/400 w wersji V5R2, i5/OS w wersji V5R3 lub nowsze mają katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.
- Pełna instalacja systemu OS/400 w wersji V5R2, systemu i5/OS w wersji V5R3, nowszego lub serwera iSeries zawiera katalogi \*TYPE2. Konwersja nie jest wymagana dla bazowego systemu plików "root" (/), QOpenSys i UDFS w pulach ASP z zakresu 1-32.
- Do konwersji systemów plików używane jest narzędzie konwersji V5R1 lub V5R2.



- Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2, i5/OS w wersji V5R3 lub nowszy.
- Pozostałe obsługiwane systemy plików z wyjątkiem UDFS w niezależnej puli ASP, które używają katalogów \*TYPE1, zostaną automatycznie poddane konwersji. Konwersja rozpoczyna się po zainstalowaniu systemu i5/OS w wersji V5R3M0 lub późniejszej. Nie powinna ona znacząco wpłynąć na działanie systemu.

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 w wersji V5R2, i5/OS w wersji V5R3 lub nowszego, ale zawierają pewne różnice względem normalnej obsługi katalogów \*TYPE2.

## Używanie katalogów \*TYPE2 w systemach OS/400 w wersji V5R1 lub V5R2

Systemy plików "root" (/), QOpenSys, i systemy plików użytkownika UDFS w zintegrowanym systemie plików w systemie OS/400 w wersjach V5R1, V5R2, i późniejszych 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.


Jeśli system OS/400 działa w wersji V5R1 lub V5R2, można poddać katalogi konwersji do formatu katalogów \*TYPE2 używając odpowiedniego narzędzia do konwersji. Bezpośrednio po zainstalowaniu systemu i5/OS w wersji V5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Dlatego można rozważyć przeprowadzenie konwersji katalogów do formatu \*TYPE2 przed zainstalowaniem systemu w wersji V5R3, co umożliwi uniknięcie automatycznej konwersji.

Obsługa katalogów \*TYPE2 w wersji V5R2 systemu OS/400 jest dostępna w wersji V5R2 Centrum informacyjnego iSeries poprzez komendę Konwersja katalogu (Convert Directory - CVTDIR).

Obsługa katalogów \*TYPE2 w wersji V5R1 systemu OS/400 jest dostępna po zainstalowaniu odpowiednich poprawek (PTF). Program narzędziowy do konwersji różni się nieznacznie od programu w wersji V5R2 systemu OS/400. 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:

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

```
SNDPTFORD PTFID((II13161))
DSPPTFCVR LICPGM(INFOAS4) SELECT(II13161)
```

- APAR informacyjny znajduje się [www.ibm.com/eserver/series/support/supporthome.nsf/document/10000045](http://www.ibm.com/eserver/series/support/supporthome.nsf/document/10000045) . Należy wybrać pozycje: **Problem Solving** → **Technical Databases** → **Authorized Program Analysis Reports (APARs)** → **V5R1 APARs** → **APAR number II13161**.

### Pojęcia pokrewne

“Niezmienność nazw” na stronie 19

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

“Przegląd konwersji formatu \*TYPE1 do \*TYPE2” na stronie 90

Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

### Odsyłacze pokrewne

“Przekształcanie katalogów z formatu \*TYPE1 na \*TYPE2” na stronie 90  
Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

“Nazwy ścieżek w bazowym systemie plików "root" (/)” na stronie 33  
Nazwy ścieżek w systemie plików "root" (/) mają określoną postać.

“Nazwy ścieżek w systemie plików użytkownika zintegrowanego systemu plików” na stronie 38  
W momencie, gdy występuje konieczność manipulacji całym systemem plików użytkownika (UDFS) wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania systemu plików UDFS służy specjalny plik blokowy (\*BLKSF).

### Informacje pokrewne

Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komenda Konwersja katalogu (Convert Directory - CVTDIR)

## Dowiązanie

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

Użytkownikom systemów plików o strukturze katalogowej wygodnie jest traktować obiekt (taki jak plik) jak coś, co ma nazwę służącą do identyfikacji przez 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.

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

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

Tabela 1. Porównanie dowiązań stałych i symbolicznych

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 statyczne (atrybuty 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	Ograniczone. Dowiązania stałe nie mogą przekraczać granic systemów plików.	Nieograniczone. Dowiązania symboliczne mogą przekraczać granice systemów plików.

## Odsyłacze pokrewne

“Dowiązania w bazowym systemie plików "root" (/) na stronie 33

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

“Dowiązania w systemie plików QOpenSys” na stronie 36

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

“Dowiązania w systemie plików użytkownika zintegrowanego systemu plików” na stronie 39

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

“Dowiązania w systemie plików QSYS.LIB” na stronie 45

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

“Dowiązania w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 48

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

“Dowiązania w systemie plików QDLS” na stronie 50

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

“Dowiązania w systemie plików QOPT” na stronie 53

System plików QOPT dopuszcza tylko jedno dowiązanie do obiektu. Nie można w nim tworzyć ani przechowywać dowiązań symbolicznych.

“Dowiązania w systemie plików QNTC” na stronie 58

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

“Dowiązania w systemie plików QFileSvr.400” na stronie 64

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

“Dowiązania w systemie plików NFS” na stronie 66

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

“Wskazówki: dowiązanie symboliczne” na stronie 93

Dowiązania symboliczne są obiektami w zintegrowanym systemie plików zawierającymi ścieżkę do innego obiektu.

“Dowiązania w systemie plików QNetWare” na stronie 55

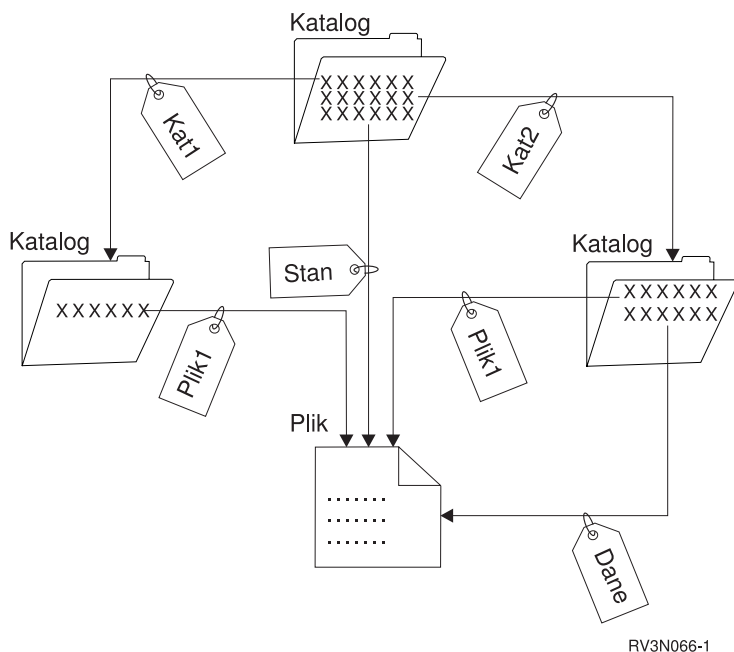
System plików QNetWare dopuszcza tylko jedno dowiązanie do obiektu. W systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych. systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych.

## Dowiązanie stałe

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

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

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



Rysunek 4. Każde dowiązanie stałe definiuje pozycję katalogu

Dowiązania stałe można usuwać bez wpływu na istnienie obiektu, dopóki pozostaje przynajmniej jedno dowiązanie do tego obiektu. Po usunięciu ostatniego dowiązania stałego, obiekt zostanie usunięty z 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 systemu QSYS.LIB niezależnej puli ASP i do systemu plików obsługującego obiekty biblioteki dokumentów (QDLS). Tak więc biblioteka ma jedno dowiązanie stałe do każdego zawartego w niej obiektu. Podobnie folder ma jedno dowiązanie stałe do każdego dokumentu. Wiele stałych dowiązań do *tego samego obiektu* nie jest jednak dozwolonych w systemach plików QSYS.LIB, systemach QSYS.LIB niezależnej puli ASP i w systemach QDLS.

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

#### Pojęcia pokrewne

“Zagadnienia dotyczące używania wielu napędów dysków i kronikowania” na stronie 99

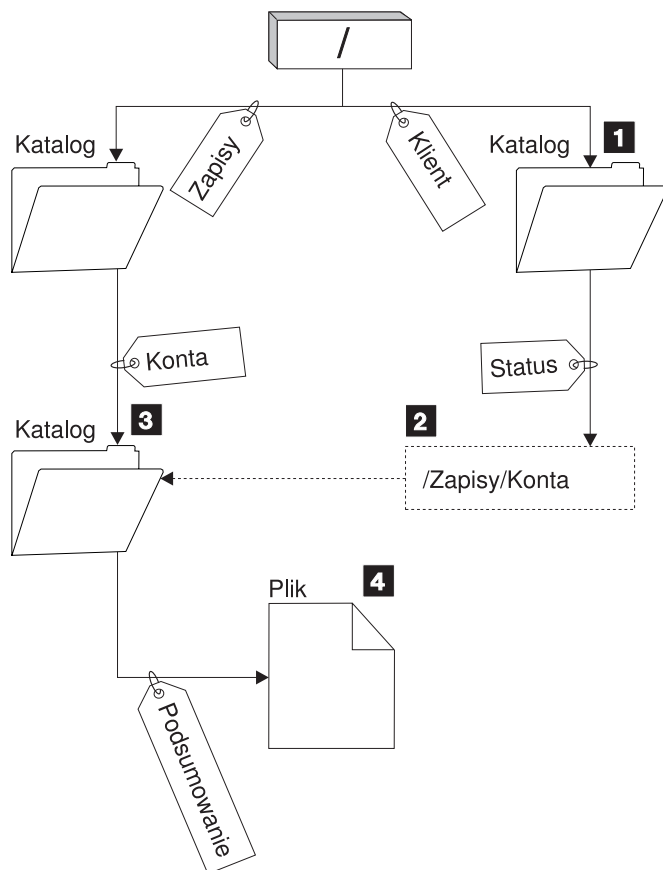
Jeśli istnieje kilka dowiązań stałych do kronikowanego obiektu zintegrowanego systemu plików, należy wszystkie te dowiązania zeszkładować i odtworzyć razem, aby zachowana została definicja dowiązania razem z odpowiednimi informacjami kroniki.

### Dowiązanie symboliczne

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

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

Rozważmy poniższy przykład sposobu użycia dowiązania symbolicznego:



RV3N068-1

Rysunek 5. Przykład 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. Dowiązania można użyć na przykład do utworzenia ścieżki do pliku, który zostanie dodany lub zastąpiony w późniejszym czasie.

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

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

## Nazwa ścieżki

Nazwa ścieżki (*path name*) (nazywana również *pathname* w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

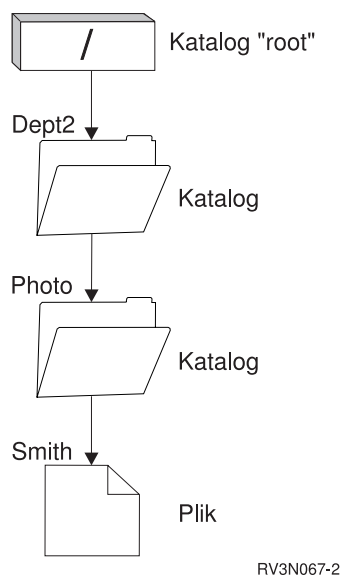
Jest to sekwencja nazw katalogów zakończona nazwą obiektu. Poszczególne nazwy katalogów i obiektów rozdziela znak ukośnika (/); na przykład:

```
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.

### Odsyłacze pokrewne

“Nazwy ścieżek w bazowym systemie plików "root" (/) na stronie 33

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

“Nazwy ścieżek w systemie plików QOpenSys” na stronie 35

Nazwy ścieżek w systemie plików QOpenSys mają określoną postać.

“Nazwy ścieżek w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 47

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

“Nazwy ścieżek w systemie plików QNTC” na stronie 57

Ścieżka składa się z nazwy systemu plików, nazwy serwera, nazwy udziału, nazw katalogów i podkatalogów oraz nazwy obiektu.

“Nazwy ścieżek w systemie plików QFileSvr.400” na stronie 62

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

“Nazwy ścieżek w systemie plików QOPT” na stronie 52

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

“Reguły dotyczące ścieżek dla funkcji API” na stronie 118

Gdy do wykonywania działań na obiekcie używane są zintegrowany system plików lub funkcja API C/400, obiekt identyfikowany jest przez podanie ścieżki do katalogu. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API.

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 73

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

“Nazwy ścieżek w systemie plików QSYS.LIB” na stronie 44

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

“Nazwy ścieżek w systemie plików QDLS” na stronie 50

Każdy składnik ścieżki może być po prostu nazwą.

“Nazwy ścieżek w systemie plików użytkownika zintegrowanego systemu plików” na stronie 38

W momencie, gdy występuje konieczność manipulacji całym systemem plików użytkownika (UDFS) wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania systemu plików UDFS służy specjalny plik blokowy (\*BLKSF).

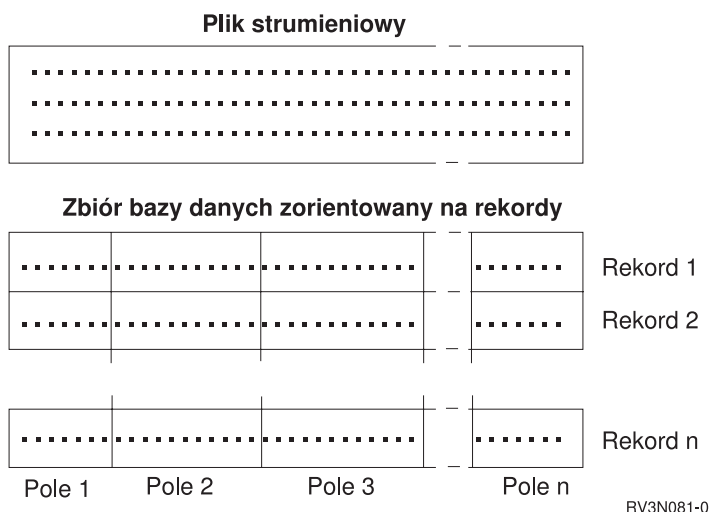
## Plik strumieniowy

*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 komputerów PC i pliki w systemach UNIX. Plik strumieniowy zintegrowanego systemu plików jest obiektem systemowym typu \*STMF.

Aby lepiej zrozumieć pliki strumieniowe, najlepiej porównać je ze zbiorami baz danych systemu 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 7. Porównanie pliku strumieniowego z plikiem opartym na rekordach.

Różna struktura plików strumieniowych i plików opartych na rekordach wpływa na ich wykorzystanie, na sposób pisania aplikacji, które mają z nimi współpracować, oraz na wybór typu pliku do użycia w aplikacji. Przykładowo plik oparty na rekordach nadaje się do przechowywania danych statystycznych klienta, takich jak nazwisko, adres i bilans rachunku. Za pomocą 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.

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

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

## Pliki strumieniowe \*TYPE1

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

- | Minimalny rozmiar pliku strumieniowego typu \*TYPE1 wynosi 4096 bajtów. Pliki strumieniowe \*TYPE1 mają przypisaną maksymalną wielkość obiektu około 256 GB (1 GB jest równy około 1 073 741 824 bajtom).

## Pliki strumieniowe \*TYPE2

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

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

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



### Odsyłacze pokrewne

“Systemy plików użytkownika (UDFS)” na stronie 36

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

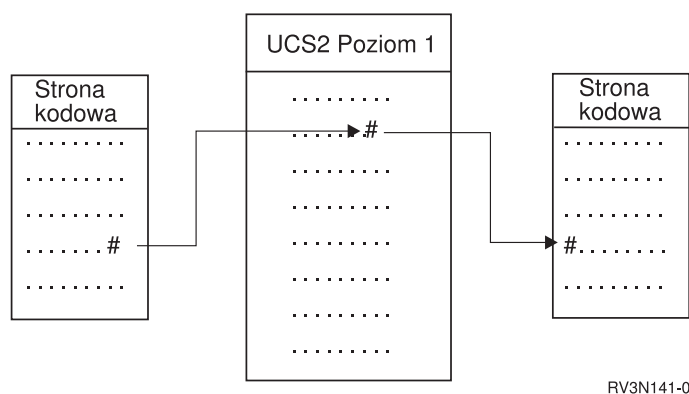
“Kopiowanie danych między plikami strumieniowymi i zbiorami baz danych” na stronie 105

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.

## Niezmiennosc nazw

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

W szczególności dotyczy to użycia tych systemów plików na serwerze 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. UCS2 Poziom 1 i UTF-16 są podzbiórmi 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.



Rysunek 8. Utrzymywanie niezmiennosci znaków w różnych schematach kodowania

Obsługa ta ułatwia współdziałanie serwera z urządzeniami używającymi innych stron kodowych. Na przykład, użytkownicy komputerów PC mogą uzyskać dostęp do pliku serwera iSeries używając tej samej nazwy pliku, pomimo że ich komputery PC nie używają tej samej strony kodowej, co serwer. Konwersja stron kodowych jest wykonywana automatycznie przez serwer. Oczywiście, urządzenie musi używać strony kodowej zawierającej znaki wykorzystywane w danej nazwie.

### Pojęcia pokrewne

“Katalogi \*TYPE2” na stronie 10

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

### Odsyłacze pokrewne

“Nazwy ścieżek w bazowym systemie plików “root” (/)” na stronie 33

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

“Nazwy ścieżek w systemie plików użytkownika zintegrowanego systemu plików” na stronie 38

W momencie, gdy występuje konieczność manipulacji całym systemem plików użytkownika (UDFS) wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania systemu plików UDFS służy specjalny plik blokowy (\*BLKSF).

“Nazwy ścieżek w systemie plików QFileSvr.400” na stronie 62

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

“Nazewnictwo i obsługa różnych języków” na stronie 120

Obsługa systemów plików “root” (/) i zapewnia niezmiennosc znaków w nazwach obiektów przy przechodzeniu między schematami kodowania używanymi w różnych językach narodowych i w różnych urządzeniach.

## Atrybuty rozszerzone

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

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

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

### **.SUBJECT**

Krótki opis zawartości i przeznaczenia obiektu.

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

### **.CODEPAGE**

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

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

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

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

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

Jeśli komputer PC klienta jest połączony z serwerem iSeries poprzez system OS/2 lub Windows, interfejsy programistyczne jego systemu operacyjnego (jak DosQueryFileInfo i DosSetFileInfo) mogą posłużyć do wydania zapytania i skonfigurowania atrybutów rozszerzonych dowolnego obiektu pliku. Użytkownicy systemu OS/2 mogą także zmienić atrybuty rozszerzone obiektu na pulpicie używając notatnika ustawień, to znaczy wybierając opcję Ustawienia menu podręcznego 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:

NazwaFirmyNazwaProduktu.Nazwa\_Atrybutu

## Obsługa skanowania

System iSeries, umożliwia skanowanie obiektów zintegrowanego systemu plików.

Ułatwia to pracę użytkownikom systemu iSeries przez umożliwienie skanowania różnych pozycji. Użytkownicy decydują, kiedy skanowanie zostanie przeprowadzone oraz jakie działanie zostanie podjęte w zależności od wyników skanowania.

Dwoma nowymi punktami wyjścia związanymi z obsługą są:

- QIBM\_QP0L\_SCAN\_OPEN - Skanowanie zintegrowanego systemu plików po otwarciu programu obsługi wyjścia. Dla tego punktu wyjścia wywoływane jest skanowanie zintegrowanego systemu plików dla otwartych programów obsługi wyjścia, które wykonują przetwarzanie skanowania po otwarciu obiektu zintegrowanego systemu plików w określonych warunkach.
- QIBM\_QP0L\_SCAN\_CLOSE - Skanowanie zintegrowanego systemu plików po zamknięciu programu obsługi wyjścia. Dla tego punktu wyjścia wywoływane jest skanowanie zintegrowanego systemu plików dla zamkniętych programów obsługi wyjścia, które wykonują przetwarzanie skanowania po zamknięciu obiektu zintegrowanego systemu plików w określonych warunkach.

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

### Zadania pokrewne

“Włączanie i wyłączanie skanowania obiektów” na stronie 132

Czynności, które należy wykonać, aby włączyć lub wyłączyć skanowanie obiektów.

### Odsyłacze pokrewne

“Skanowanie zintegrowanego systemu plików” na stronie 95

Obiekty w systemie plików "root" (/), QOpenSys i UDFS ASP użytkownika nie będą skanowane z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików do czasu zakończenia konwersji systemów do formatu katalogów \*TYPE2.

### Informacje pokrewne

QIBM\_QP0L\_SCAN\_OPEN

QIBM\_QP0L\_SCAN\_CLOSE

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

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

- Wirusy  
Program obsługi wyjścia może wykonać skanowanie w poszukiwaniu wirusów. Jeśli wirus zostanie znaleziony w pliku, program antywirusowy może naprawić problem lub spróbować poddać wirusa kwarantannie. Ponieważ sam serwer iSeries nie zostanie zainfekowany, kwarantanna dąży do zmniejszenia możliwości przeniesienia wirusa do innych serwerów.
- Wywołania w celu sprawdzenia, czy plik został otwarty  
Można także wykonać skanowanie, aby sprawdzić, kiedy plik został otwarty. Uruchamiając skanowanie można sprawdzić datę i godzinę dostępu do pewnych plików. Opcja ta jest użyteczna, gdy wskazane jest śledzenie działań wybranych użytkowników.

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

#### 1. Skanowanie w czasie wykonywania

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

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

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

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

Główną wadą tej metody jest zwiększenie czasu oczekiwania na zasoby o czas wykonywania skanowania.

Użytkownicy, którzy usiłują uzyskać dostęp do pliku, muszą czekać, aż skanowanie zostanie zakończone. System zapewnia, że skanowanie jest przeprowadzane tylko gdy jest to wymagane, a nie podczas każdego dostępu.

#### 2. Skanowanie masowe lub uruchamiane ręcznie

Opcji tej można użyć, aby jednocześnie skanować wiele pozycji. W takim przypadku można skonfigurować uruchamianie skanowania gdy serwer nie jest obciążony pracą, na przykład podczas dni wolnych. Spowoduje to minimalny wpływ skanowania na czas oczekiwania na dostęp do pliku w trakcie normalnej codziennej pracy. Skanowanie jest przeprowadzane bez łączenia się z siecią. Może więc zmniejszyć czas oczekiwania na skanowanie w czasie wykonywania dla plików, które nie uległy zmianie od czasu ostatniego skanowania masowego. Kolejne skanowanie nie jest wymagane podczas ponownego dostępu do tych plików.

##### **Pojęcia pokrewne**

"Pokrewne wartości systemowe"

Z obsługą skanowania związane są dwie wartości systemowe. Służą one do tworzenia środowiska skanowania w serwerze.

##### **Informacje pokrewne**

QIBM\_QP0L\_SCAN\_OPEN

QIBM\_QP0L\_SCAN\_CLOSE

## **Pokrewne wartości systemowe**

Z obsługą skanowania związane są dwie wartości systemowe. Służą one do tworzenia środowiska skanowania w serwerze.

Poniżej znajdują się nazwy tych dwóch wartości systemowych i ich opisy. Wartości te i ich opcje zostały opisane dla narzędzia iSeries Navigator. Porównywalne wartości dla interfejsu znakowego zostały umieszczone w nawiasach, po nazwach iSeries Navigator. Na przykład, dla wartości systemowej QSCANFCTL, gdy opcja sterowania produktem iSeries Navigator 'Skanowanie tylko dostępu poprzez serwery plików' zostanie wybrana, wynik będzie taki sam, jak po wpisaniu opcji \*FSVROONLY w trybie znakowym.

Nazwy i opisy wartości systemowych:

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

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

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

## 2. Sterowanie skanowaniem (QSCANFSCTL)

Dla tej wartości systemowej można użyć domyślnych opcji sterowania lub użyć zdefiniowanych opcji. Krótki opis różnych zdefiniowanych opcji sterowania w oparciu o wartości systemowe produktu iSeries Navigator znajduje się poniżej.

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

### Odsyłacze pokrewne

"Skanowanie zintegrowanego systemu plików" na stronie 95

Obiekty w systemie plików "root" (/), QOpenSys i UDFS ASP użytkownika nie będą skanowane z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików do czasu zakończenia konwersji systemów do formatu katalogów \*TYPE2.

### Informacje pokrewne

QSCANFS

QSCANFSCTL

## Uruchamianie operacji skanowania

Skanowanie może być uruchomione z wielu powodów. Poniższa lista zawiera zestaw powodów do uruchomienia skanowania.

Aby przejrzeć aktualny status skanowania, można użyć komend Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK), Wyświetlenie dowiązań obiektów (Display Object Links - DSPLNK), funkcji API Pobranie atrybutów (Get attributes - Qp01GetAttr()) lub strony Właściwości programu iSeries Navigator.

### Informacje pokrewne

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

Komenda Wyświetlenie dowiązań obiektów (Display Object Links - DSPLNK)

Funkcja API Pobranie atrybutów (Get attributes - QP01GetAttr())

### Zmiana obiektu:

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

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

### Zmiana sygnatury:

Skanowanie nastąpi, gdy użytkownik uzyska dostęp do obiektu lub globalna sygnatura jest inna niż sygnatura obiektu.

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

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

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

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

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 0000000000

Globalna sygnatura skanowania to: 0000 i nie jest aktualizowana.

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

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 1111111111

Globalna sygnatura skanowania jest aktualizowana do wartości: 0001.

3. Następnie otwierany jest plik, który aktualnie ma sygnaturę 0000. Istnienie programów obsługi wyjścia wraz z różnicą globalnych sygnatur skanowania (od 0000 do 0001) powoduje zainicjowanie operacji skanowania. Po pomyślnym zakończeniu skanowania sygnatura pliku jest aktualizowana do wartości 0001.



4. Jeśli tabela zostanie otwarta przez innego użytkownika, skanowanie nie zostanie powtórzone, ponieważ sygnatura obiektu i globalna sygnatura są takie same.

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

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

Klucz skanowania: XXXXXX  
Sygnatura klucza skanowania: 222222222

Globalna sygnatura klucza skanowania jest aktualizowana do wartości: 0002.

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

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

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

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

#### **Informacje pokrewne**

QIBM\_QP0L\_SCAN\_OPEN

QIBM\_QP0L\_SCAN\_CLOSE

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

#### **Inny CCSID:**

Dostęp do obiektu z innym identyfikatorem CCSID niż poprzednio skanowany spowoduje uruchomienie skanowania.

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

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

#### **W trakcie operacji składowania:**

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

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

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

#### **Informacje pokrewne**

Komenda Składowanie obiektu (Save Object - SAV)

#### **Sprawdzenie integralności obiektu:**

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

Opcji tej należy użyć, aby sprawdzić integralność pliku bez jego otwierania. Jeśli podany zostanie parametr SCANFS (\*STATUS), wszystkie obiekty, które nie przeszły pomyślnie poprzedniego skanowania otrzymają w protokole zapis o niepowodzeniu skanowania.

#### **Informacje pokrewne**

Komenda Zmiana integralności obiektu (Change Object Integrity - CHGOBJITG)

---

## **Praca z systemami plików**

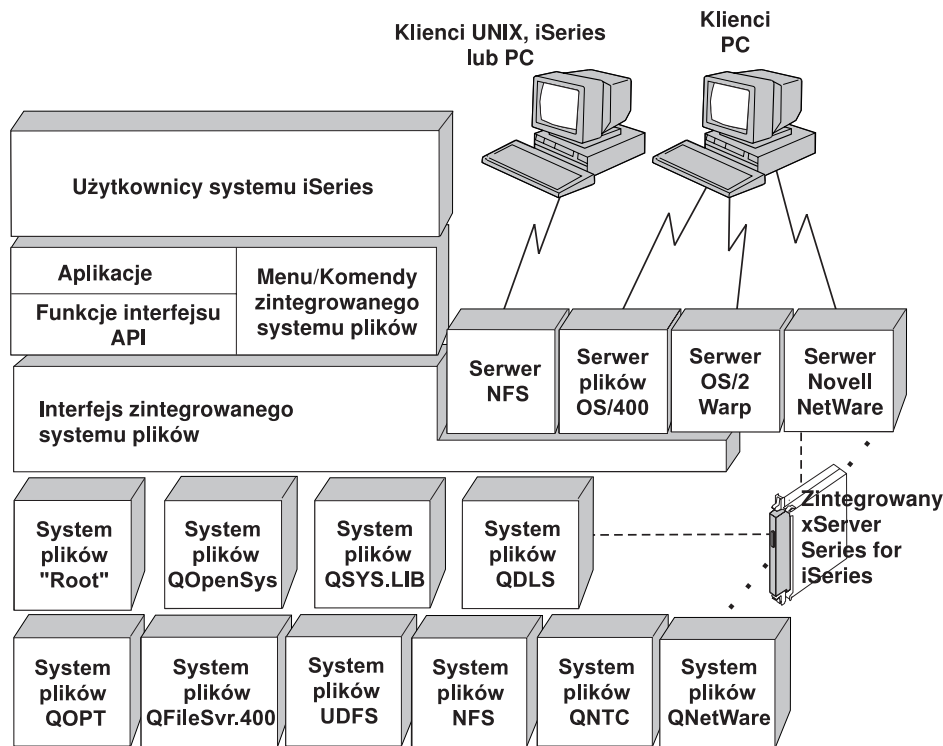
*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

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 i5/OS dostępu do zbiorów baz danych i obiektów innych typów poprzez biblioteki może być traktowane jako system plików. Podobnie obsługiwane przez i5/OS dostępu do dokumentów (będących w rzeczywistości plikami strumieniowymi) poprzez strukturę folderów może być traktowane jak oddzielny system plików.

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

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





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

## Używanie sieciowego systemu plików poprzez interfejs zintegrowanego systemu plików

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

### Pojęcia pokrewne

“Czym jest zintegrowany system plików” na stronie 2

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

### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 68

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

“Przenoszenie plików lub folderów do innego systemu plików” na stronie 128

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.

### Odsyłacze pokrewne

“Dostęp przy użyciu komend języka CL” na stronie 69

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

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 73

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

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

### Informacje pokrewne

Obsługa optyczna

Obsługa sieciowego systemu plików

## Porównanie systemów plików

Poniższe tabele zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

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

Cecha	"root" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
Standardowa część i5/OS	Tak	Tak	Tak	Tak	Tak
Typ zbioru	Strumieniowy	Strumieniowy	Rekord <sup>12</sup>	Strumieniowy	Strumieniowy
Limit wielkości zbioru	T2=1 TB; T1=128 GB	T2=1 TB; T1=128 GB	Wielkości zbiorów baz danych	4 GB	Różne <sup>17</sup>
Zintegrowany z OfficeVision (na przykład plik można wysłać pocztą elektroniczną)	Nie	Nie	Nie	Tak	Nie
Dostęp poprzez serwer plików i5/OS	Tak	Tak	Tak	Tak	Tak
Bezpośredni dostęp poprzez procesor we/wy serwera plików <sup>1</sup>	Nie	Nie	Nie	Nie	Tak
Względna szybkość operacji otwarcia/zamknięcia	Średnia <sup>2</sup>	Średnia <sup>2</sup>	Niska <sup>2</sup>	Niska <sup>2</sup>	Średnia <sup>2</sup>
Wyszukiwanie nazw z rozróżnianiem małych i wielkich liter	Nie	Tak	Nie <sup>4</sup>	Nie <sup>5</sup>	Nie
Maksymalna długość każdego składnika ścieżki	255 zn <sup>19</sup>	255 zn <sup>19</sup>	10.6 zn <sup>6</sup>	8.3 zn <sup>7</sup>	255 zn <sup>19</sup>
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	0 <sup>18</sup>
Maksymalna liczba poziomów hierarchii katalogów w systemie plików	Nieograniczona <sup>10</sup>	Nieograniczona <sup>10</sup>	3	32	127
Maksymalna liczba dowiązań do jednego obiektu <sup>11</sup>	Zmienia się <sup>15</sup>	Zmienia się <sup>15</sup>	1	1	1
Obsługa dowiązań symbolicznych	Tak	Tak	Nie	Nie	Nie
Obiekt/zbiór może mieć właściciela	Tak	Tak	Tak	Tak	Nie
Obsługa komend zintegrowanego systemu plików	Tak	Tak	Tak	Tak	Tak
Obsługa funkcji API zintegrowanego systemu plików	Tak	Tak	Tak	Tak	Tak
Obsługa funkcji API hierarchicznego systemu plików (HFS)	Nie	Nie	Nie	Tak	Nie
Ochrona wątków <sup>13</sup>	Tak	Tak	Tak	Nie	Tak
Obsługa kronikowania obiektów	Tak	Tak	Yes <sup>14</sup>	Nie	Nie

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

Cecha	"root" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
<b>Uwagi:</b>					
<ol style="list-style-type: none"> <li>1. Sprzęt procesor we/wy serwera plików używany jest przez Serwer LAN.</li> <li>2. Przy dostępie przez serwer plików i5/OS.</li> <li>3. Przy dostępie do klienta PC przez Serwer LAN. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny.</li> <li>4. Maksymalna długość ścieżki w systemie plików QSYS.LIB jest równa 55 znaków. Maksymalna długość ścieżki w systemie niezależnych plików QSYS.LIB jest równa 66 znaków.</li> <li>5. Szczegóły zawiera temat "System plików usług biblioteki dokumentów (QDLS)" na stronie 49.</li> <li>6. Do 10 znaków nazwy obiektu i do 6 znaków typu obiektu.</li> <li>7. Do 8 znaków nazwy i od 1 do 3 znaków rozszerzenia określającego typ pliku (jeśli ma występować).</li> <li>8. Przy założeniu, że ścieżka bezwzględna rozpoczyna się od znaku, po którym następuje nazwa systemu plików (taka jak /QDLS...).</li> <li>9. Systemy plików QSYS.LIB i QSYS.LIB niezależnej puli ASP obsługują trzy predefiniowane atrybuty rozszerzone: .SUBJECT, TYPE i CODEPAGE. Maksymalna długość jest określana przez złożenie długości tych trzech atrybutów rozszerzonych.</li> <li>10. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu.</li> <li>11. Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu.</li> <li>12. Obszary użytkowników w systemie plików SYS.LIB i niezależnej puli ASP obsługują operacje wejścia/wyjścia plików strumieniowych.</li> <li>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.</li> <li>14. Systemy plików QSYS.LIB i niezależnej puli ASP obsługują kronikowanie innych typów obiektów niż obiekty systemów plików "root" (/), UDFS, i QOpenSys.</li> <li>15. Katalogi w formacie *TYPE2 mają ograniczenie miliona dowiązań dla jednego obiektu i 999 998 podkatalogów. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt.</li> <li>16. Dane w tej kolumnie odnoszą się zarówno do systemu plików QSYS.LIB jak i do systemu plików QSYS.LIB niezależnej puli ASP.</li> <li>17. W zależności od systemu, do którego uzyskiwany jest dostęp.</li> <li>18. QNTC nie obsługuje atrybutów rozszerzonych.</li> <li>19. W przypadku niektórych wartości CCSID maksymalna długość może wynosić mniej niż 255 znaków.</li> </ol>					
<b>Skróty</b>					
<ul style="list-style-type: none"> <li>• zn = znaki</li> <li>• T1 = *TYPE1 *STMF</li> <li>• T2 = *TYPE2 *STMF</li> <li>• B = bajty    kB = kilobajty    MB = megabajty    GB = gigabajty    TB = terabajty</li> </ul>					

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

Cecha	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Standardowa część i5/OS	Tak	Tak	Tak	Tak	Nie
Typ zbioru	Strumieniowy	Strumieniowy	Strumieniowy	Strumieniowy	Strumieniowy
Limit wielkości zbioru	4 GB	2 GB - 1	T2 = 1 TB; T1=128 GB	Zmienne <sup>16</sup>	2 GB
Zintegrowany z OfficeVision (na przykład plik można wysłać pocztą elektroniczną)	Nie	Nie	Nie	Nie	Nie
Dostęp poprzez serwer plików i5/OS	Tak	Tak	Tak	Tak	Tak

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

Cecha	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Bezpośredni dostęp poprzez serwer Integrated PC Server <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 <sup>17</sup>	Zmienia się <sup>2</sup>	255 zn <sup>13, 17</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	Tak	Tak	Tak	Nie
Obsługa kronikowania obiektów	Nie	Nie	Tak	Nie	Nie

Tabela 3. 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>1. Sprzęt procesor we/wy serwera plików używany jest przez Serwer LAN.</li> <li>2. Zależy od tego, który zdalny system plików jest dostępny.</li> <li>3. Przy dostępie przez serwer plików i5/OS.</li> <li>4. Szczegóły zawiera temat "system plików nośników optycznych (QOPT)," na stronie 51.</li> <li>5. Przy założeniu ścieżki bezwzględnej rozpoczynającej się od znaku po którym następuje nazwa systemu plików.</li> <li>6. System plików QFileSvr.400 nie zwraca atrybutów rozszerzonych, nawet jeśli używany właśnie system plików je obsługuje.</li> <li>7. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu.</li> <li>8. Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu.</li> <li>9. Dostępny system plików może obsługiwać prawa własności do obiektów.</li> <li>10. Maksymalna długość atrybutów rozszerzonych całego UDFS nie może przekroczyć 40 B.</li> <li>11. Przy dostępie poprzez klienta Novell NetWare. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny.</li> <li>12. 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>13. Obiekty NDS (NetWare Directory Services) mogą mieć maksymalnie 255 znaków. Nazwy plików i katalogów są ograniczone wg konwencji DOS: 8.3.</li> <li>14. 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>15. Katalogi *TYPE2 posiadają limit miliona dowiązań na obiekt. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt.</li> <li>16. W zależności od systemu, do którego uzyskiwany jest dostęp.</li> <li>17. W przypadku niektórych wartości CCSID maksymalna długość może wynosić mniej niż 255 znaków.</li> </ol>					
<b>Skróty</b>					
<ul style="list-style-type: none"> <li>• zn = znaki</li> <li>• T1 = *TYPE1 *STMF</li> <li>• T2 = *TYPE2 *STMF</li> <li>• B = bajty    kB = kilobajty    MB = megabajty    GB = gigabajty    TB = terabajty</li> </ul>					

### Zadania pokrewne

"Przenoszenie plików lub folderów do innego systemu plików" na stronie 128

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 przenieść obiekty z jednego systemu plików do innego, aby wykorzystywać zalety jego charakterystyki.

### Odsyłacze pokrewne

"System plików "root" (/)" na stronie 32

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

"System plików systemów otwartych (QOpenSys)" na stronie 34

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

"Systemy plików użytkownika (UDFS)" na stronie 36

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

"System plików bibliotek (QSYS.LIB)" na stronie 43

System plików QSYS.LIB obsługuje strukturę bibliotek serwera iSeries.

"Niezależne pule ASP systemu QSYS.LIB" na stronie 46

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.

“System plików usług biblioteki dokumentów (QDLS)” na stronie 49

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

“system plików nośników optycznych (QOPT),” na stronie 51

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

“System plików NetWare (QNetWare)” na stronie 54

System plików QNetWare daje dostęp do danych na autonomicznych serwerach PC działających w systemie Novell NetWare 5.1 lub 6.0.

“system plików iSeries NetClient (QNTC),” na stronie 57

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych na serwerze Integrated xSeries Server for iSeries, na którym działa system Linux lub system Windows NT 4.0 Server lub nowszy. System plików QNTC zapewnia również dostęp do danych i obiektów przechowywanych na serwerach zdalnych, na których działa system Windows NT 4.0 lub nowszy, protokół Samba 3.0 lub nowszy systemu Linux lub obsługiwana wersja programu iSeries NetServer.

“System plików serwera plików i5/OS (QFileSvr.400)” na stronie 61

System plików QFileSvr.400 zapewnia przejrzysty dostęp do innych systemów plików znajdujących się w zdalnych serwerach iSeries. Jest to dostęp poprzez hierarchiczną strukturę katalogów.

“sieciowy system plików (Network File System - NFS),” na stronie 65

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

### **Informacje pokrewne**

Zarządzanie kronikami

## **System plików "root" (/)**

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

Dodatkowo, system ten:

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

Bazowy system plików "root" (/) obsługuje znakowe pliki specjalne (\*CHRSF) o nazwach /dev/null oraz /dev/zero. Znakowe pliki specjalne są powiązane z urządzeniami lub zasobami systemu komputerowego. Mają one nazwy ścieżek wyświetlane w katalogach i taką samą ochronę dostępu, jak zwykle 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.

## Używanie systemu plików "root" (/)

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

### Rozróżnianie wielkości znaków w 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" (/)

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

/Katalog/Katalog . . . /Obiekt

- Każdy komponent nazwy ścieżki może zawierać do 255 znaków, znacznie więcej niż w systemach plików QSYS.LIB lub QDLS. Pełna nazwa ścieżki może być bardzo długa i zawierać do 16 megabajtów znaków.
- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu lub pamięci serwera.
- Podczas zapisywania nazwy składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów \*TYPE1) i UTF-16 (dla katalogów \*TYPE2).

#### Pojęcia pokrewne

"Niezmienność nazw" na stronie 19

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

"Katalogi \*TYPE2" na stronie 10

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

"Nazwa ścieżki" na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

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

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

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

#### Pojęcia pokrewne

"Dowiązanie" na stronie 12

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

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

Wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL i na ekranach opisanych w temacie Dostęp za pomocą menu i ekranów mogą być uruchamiane w systemie plików "root" (/). Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

#### Zadania pokrewne

"Dostęp przy użyciu pozycji menu i ekranów" na stronie 68

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

#### Odsyłacze pokrewne

"Dostęp przy użyciu komend języka CL" na stronie 69

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików można także



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

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

Wszystkie funkcje API wymienione w temacie Używanie interfejsów API mogą być uruchamiane w systemie plików "root" (/).

### Odsyłacze pokrewne

"Wykonywanie operacji za pomocą funkcji API" na stronie 110

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

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

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

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

### Pojęcia pokrewne

"Kronikowanie obiektów" na stronie 95

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

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

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

Oba sterowniki są znakowymi plikami specjalnymi (\*CHRSF) i zostaną utworzone w trakcie IPL. Sterowniki UDP i TCP służą do otwierania obsługi transportowej UDP i TCP. Oba sterowniki będą urządzeniami użytkownika i otrzymają nowy numer główny urządzenia. Otrzymają także powielone funkcje open, co oznacza, że każda funkcja otrzyma unikalną instancję urządzenia. Używanie tych urządzeń jest obsługiwane tylko w środowisku i5/OS PASE. Tabela wyświetlona poniżej zawiera obiekty, które będą utworzone i ich właściwości.

Tabela 4. Obiekty sterowników urządzeń i ich właściwości

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

### Pojęcia pokrewne

"Utworzone katalogi" na stronie 7

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, poniższe katalogi.

### Informacje pokrewne

Środowisko PASE

## 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 główny system plików "root" (/), ten system plików wykorzystuje możliwości plików strumieniowych i katalogów dostarczane przez zintegrowany system plików.

Dodatkowo, system ten:



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

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

## Używanie systemu plików QOpenSys

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

### Pojęcia pokrewne

"Utworzone katalogi" na stronie 7

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, poniższe katalogi.

## 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 jakakolwiek 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 systemie plików QOpenSys w tym samym katalogu można przechowywać obiekt o nazwie Payroll, obiekt o nazwie PayRoll i obiekt o nazwie PAYROLL.

## Nazwy ścieżek w systemie plików QOpenSys

Nazwy ścieżek w systemie plików QOpenSys mają określoną postać.

/QOpenSys/Katalog/Katalog/ . . . /obiekt

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

### Pojęcia pokrewne

"Niezmienność nazw" na stronie 19

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

"Katalogi \*TYPE2" na stronie 10

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

"Nazwa ścieżki" na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Dowiązania w systemie plików QOpenSys

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

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

### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

Wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL i na ekranach opisanych w temacie Dostęp za pomocą menu i ekranów mogą być uruchamiane w systemie plików QOpenSys. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 68

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

### Odsyłacze pokrewne

“Dostęp przy użyciu komend języka CL” na stronie 69

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

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

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

### Odsyłacze pokrewne

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Kronikowanie zmian obiektów w systemie plików QOpenSys

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

### Pojęcia pokrewne

“Kronikowanie obiektów” na stronie 95

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

## Systemy plików użytkownika (UDFS)

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

Dodatkowo system ten:

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

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

- numer puli ASP lub nazwę niezależnej puli ASP, w których przechowywane są obiekty systemu plików UDFS,
- rozróżnianie małych i wielkich liter w nazwach obiektów umiejscowionych w obrębie systemu plików UDFS. Można w ten sposób określić, czy w trakcie wyszukiwania obiektu w obrębie systemu UDFS w nazwach obiektów będą rozróżniane małe i wielkie litery.
- Atrybut tworzenia skanowania obiektu definiuje, czym powinien być atrybut skanowania dla obiektów utworzonych w systemie plików UDFS.
- Atrybut ograniczenia zmiany nazwy i usuwania dowiązań.
- Wartość kontroli dla systemu plików UDFS.
- Różne formaty plików strumieniowych, \*TYPE1 i \*TYPE2.

## System plików użytkownika - pojęcia


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

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

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. Podczas ładowania IPL wszystkie systemy UDFS są odłączane. Dlatego muszą zostać ponownie podłączone po IPL.

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

Więcej informacji dotyczących podłączania systemów plików można znaleźć w dokumencie about mounting file systems, see OS/400 Network File System Support  .

## Używanie systemów plików UDFS poprzez interfejs zintegrowanego systemu plików

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

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

### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

“Plik strumieniowy” na stronie 17

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

### Informacje pokrewne

Komenda Tworzenie systemu plików UDFS (Create UDFS - CRTUDFS)

## Rozróżnianie wielkości znaków w systemie plików UDFS zintegrowanego systemu plików

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

Jeśli użytkownik tego zażąda, 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 jakakolwiek 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.

### Informacje pokrewne

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

## Nazwy ścieżek w systemie plików użytkownika zintegrowanego systemu plików

W momencie, gdy występuje konieczność manipulacji całym systemem plików użytkownika (UDFS) wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania systemu plików 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 MB długości.
- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu lub pamięci serwera.
- Podczas zapisywania nazwy składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów \*TYPE1) i UTF-16 (dla katalogów \*TYPE2).

#### Pojęcia pokrewne

“Niezmienność nazw” na stronie 19

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

“Katalogi \*TYPE2” na stronie 10

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

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Dowiązania w systemie plików użytkownika zintegrowanego systemu plików

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

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

#### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

Wszystkie komendy wymienione w temacie Dostęp za pomocą komend CL i na ekranach opisanych w temacie Dostęp za pomocą menu i ekranów mogą być uruchamiane w systemie plików użytkownika.

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

Tabela 5. Komendy CL systemu plików użytkownika

Komenda	Opis
ADDMFS	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.
CRTUDFS	Tworzenie systemu plików UDFS (Create UDFS). Tworzy system plików użytkownika.
DLTUDFS	Usunięcie systemu plików UDFS (Delete UDFS). Usuwa system plików użytkownika.
DSPMFSINF	Wyświetlenie danych podłączonego systemu plików (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.
DSPUDFS	Wyświetlenie systemu plików UDFS (Display UDFS). Wyświetla informacje o systemie plików użytkownika.

Tabela 5. Komendy CL systemu plików użytkownika (kontynuacja)

Komenda	Opis
MOUNT	Podłączenie systemu plików. Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. Jest to alternatywna postać komendy ADDMFS.
RMVMFS	Usuwanie podłączonego FSS (Remove Mounted File System). Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego.
UNMOUNT	Odłączenie systemu plików. Usuwa wyeksportowane systemy plików serwerów zdalnych z przestrzeni nazw klienta lokalnego. Jest to alternatywna postać komendy RMVMFS.

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

#### Zadania pokrewne

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 68

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

#### Odsyłacze pokrewne

“Dostęp przy użyciu komend języka CL” na stronie 69

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

## Corzystanie z funkcji API zintegrowanego systemu plików w systemie plików użytkownika

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

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

#### Odsyłacze pokrewne

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

#### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Graficzny interfejs użytkownika dla systemu plików użytkownika

Program iSeries Navigator, graficzny interfejs użytkownika w komputerze PC, zapewnia łatwy i dogodny dostęp do systemu plików UDFS.

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

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

- “Tworzenie nowego systemu plików użytkownika” na stronie 131
- “Podłączanie systemu plików utworzonego użytkownika” na stronie 131
- “Odłączanie systemu plików definiowanego przez użytkownika” na stronie 132



## Tworzenie zintegrowanego systemu plików użytkownika

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

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

Dla systemu plików UDFS można także określić następujące pozycje:

- Rozróżnianie wielkości znaków
- Określa, czy obiekty utworzone w systemie UDFS powinny być skanowane
- Wartość kontroli dla obiektów utworzonych w systemie plików UDFS
- Wartość atrybutu ograniczone, zmiana nazwy i usunięcie dowiązania

### Informacje pokrewne

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

Komenda Dodanie podłączonego systemu plików (Add Mounted File System - ADDMFS)

## Usuwanie zintegrowanego systemu plików użytkownika

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

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

### Informacje pokrewne

Komenda Usunięcie systemu plików użytkownika (Delete User-Defined File System - DLTUDFS)

## Wyświetlanie zintegrowanego systemu plików użytkownika

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

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

### Informacje pokrewne

Komenda Wyświetlenie systemu plików użytkownika (Display User-Defined File System - DSPUDFS)

Komenda Wyświetlenie informacji o podłączonym systemie plików (Display Mounted File System - DSPMFSINF)

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

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

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

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

### Informacje pokrewne

Komenda Dodanie podłączonego systemu plików (Add Mounted File System - ADDMFS)

## Odłączanie systemu plików użytkownika (UDFS) zintegrowanego systemu plików

Komenda odłączająca system plików użytkownika (UDFS) powoduje, że zawartość systemu plików UDFS staje się niedostępna dla interfejsu zintegrowanego systemu plików.

Obiekty znajdujące się w UDFS stają się niedostępne w momencie odłączenia UDFS. Komenda Usuwanie podłączonego systemu plików (Remove Mounted File System - RMVMFS) lub komenda UNMOUNT spowodują, że

podłączone systemy plików będą niedostępne dla przestrzeni nazw zintegrowanego systemu plików. Jeśli któryś z obiektów w systemie plików jest używany (na przykład plik jest otwarty) w trakcie wykonywania komendy, to użytkownik otrzyma komunikat o błędzie. System UDFS pozostanie wówczas podłączony. Jeśli użytkownik podłączył inny UDFS, przesłaniając którąkolwiek część starego UDFS, to starego UDFS nie można będzie odłączyć, dopóki nie zostanie on odsłonięty.

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

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

#### **Informacje pokrewne**

Komenda Usunięcie podłączonego systemu plików (Remove Mounted File System - RMVMFS)

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

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

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

#### **Informacje pokrewne**

Komenda Składowanie obiektu (Save Object - SAV)

Komenda Odtworzenie obiektu (Restore Object - RST)

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

Obiekty w systemach plików użytkownika (UDFS) mogą być kronikowane. Umożliwia to odtworzenie zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu.

#### **Pojęcia pokrewne**

“Kronikowanie obiektów” na stronie 95

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

## **System plików użytkownika i niezależne pule pamięci dyskowej (ASP)**

Gdy niezależna pula pamięci dyskowej ASP jest podłączana, w systemie plików "root" (/) następuje kilka zmian.

Są one następujące:

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



domyślny system plików UDFS nie zostanie podłączony. Należy wówczas zmienić nazwę katalogu lub usunąć jego zawartość i zablokować, a następnie udostępnić pulę ASP lub użyć komendy MOUNT do podłączenia domyślnego systemu plików UDFS.

- Jeśli niezależna pula ASP jest podstawową lub dodatkową ASP, a domyślny system UDFS został pomyślnie podłączony, podłączony zostanie dodatkowy system plików. System plików QSYS.LIB niezależnej ASP zostanie podłączony w ścieżce /nazwa\_asp/QSYS.LIB.

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

#### **Odsyłacze pokrewne**

“Niezależne pule ASP systemu QSYS.LIB” na stronie 46

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.

## **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,
- zapewnia pełną obsługę administracyjną do zarządzania obiektami serwera iSeries,
- obsługuje operacje strumienia wejścia/wyjścia na podzbiórach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

Przed wersją 3 systemu i5/OS system plików QSYS.LIB określany był jako *właściwy* system plików serwera iSeries 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 systemu plików QSYS.LIB poprzez zintegrowany system plików.

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

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

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

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

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

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

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

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

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

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

## Obsługa przestrzeni użytkowników w systemie plików QSYS.LIB

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

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

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

## Obsługa zbiorów składowania w systemie plików QSYS.LIB

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

Na przykład, w zbiorze składowania mogą znajdować się dane, które można odczytać lub skopiować do innego zbioru, jeśli trzeba je umieścić w istniejącym, pustym obiekcie zbioru składowania. Kiedy zbiór składowania jest otwarty, nie są dozwolone żadne inne otwarte instancje tego pliku. Zbiór składowania **może** mieć wiele instancji otwartych do odczytu, pod warunkiem, że każde zadanie ma tylko jedną taką instancję. Zbiór 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 oraz poprzez system plików QSYS.LIB.

## Rozróżnianie wielkości znaków w systemie plików QSYS.LIB

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

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

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

## Nazwy ścieżek w systemie plików QSYS.LIB

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

- Na przykład:

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

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

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

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

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

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

Więcej informacji dotyczących identyfikatorów CCSID można znaleźć w temacie Centrum informacyjnego omawiającym globalizację serwera i5/OS.

#### Pojęcia pokrewne

"Nazwa ścieżki" na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Dowiązania w systemie plików QSYS.LIB

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

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

#### Pojęcia pokrewne

"Dowiązanie" na stronie 12

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

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

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

Komendy wymienione w temacie "Dostęp przy użyciu komend języka CL" na stronie 69, z wyjątkiem następujących, działają w systemie plików QSYS.LIB:

- 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 można użyć dla zbiorów fizycznych baz danych,
- komenda RCLLNK nie jest obsługiwana.

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie "Dostęp przy użyciu pozycji menu i ekranów" na stronie 68.

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

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110, z wyjątkiem następujących, działają w systemie plików QSYS.LIB:

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

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **Niezależne pule ASP systemu QSYS.LIB**

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.

## **Używanie systemu plików QSYS.LIB w niezależnej puli ASP poprzez interfejs zintegrowanego systemu plików**

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

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

#### **Odsyłacze pokrewne**

“System plików użytkownika i niezależne pule pamięci dyskowej (ASP)” na stronie 42

Gdy niezależna pula pamięci dyskowej ASP jest podłączana, w systemie plików "root" (/) następuje kilka zmian.

## **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 QSYS.LIB niezależnej puli ASP ze zdalnych klientów.

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

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

- Zbiory logiczne nie są obsługiwane.

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

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

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

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

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

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

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

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

Operacje strumieniowego wejścia/wyjścia dla zbioru składowania nie są również obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez serwer systemu Network File System. Są one jednak dostępne z klientów PC oraz poprzez system plików QSYS.LIB.

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

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

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

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

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

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

- Na przykład:

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

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

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

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



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

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

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

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

#### Pojęcia pokrewne

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

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

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

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

#### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

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

Większość komend wymienionych w temacie “Dostęp przy użyciu komend języka CL” na stronie 69 działa w systemie plików QSYS.LIB niezależnej puli ASP. Istnieje jednak kilka wyjątków:

- 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,
  - komend STRJRN i ENDJRN nie można użyć dla zbiorów fizycznych baz danych,
  - komendą MOV nie można przenosić bibliotek w systemie plików QSYS.LIB niezależnej ASP do podstawowych ASP; można jednak przenosić biblioteki z systemu plików QSYS.LIB niezależnej puli ASP do systemowej ASP lub innych niezależnych pul ASP,
  - jeśli do składowania i odtwarzania obiektów biblioteki w niezależnej puli ASP wykorzystywane są komendy SAV lub RST, to niezależna pula ASP musi być skojarzona z zadaniem wykonującym komendę SAV or RST lub musi być określona niezależna pula ASP w parametrze ASPDEV; konwencja nazw ścieżek /nazwa\_asp/QSYS.LIB/obiekt.typ nie jest obsługiwana w SAV i RST.
- l • Komenda RCLLNK nie jest obsługiwana.

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 68.

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

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110 działają w systemie plików QSYS.LIB niezależnej puli ASP z wyjątkiem następujących sytuacji:

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

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## System plików usług biblioteki dokumentów (QDLS)

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

Dodatkowo, system ten:

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

## Używanie systemu plików QDLS poprzez interfejs zintegrowanego systemu plików

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

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

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

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

Gdy zintegrowany system plików opiera się na modelu programu środowiska `Integrated Language Environment (ILE)`, system plików HFS opiera się na oryginalnym modelu programu 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.

### Informacje pokrewne

Funkcje API hierarchicznego systemu plików

## Rejestrowanie użytkowników w systemie plików QDLS

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



## Rozróżnianie wielkości znaków w systemie plików QDLS

System plików QDLS zamienia małe litery alfabetu angielskiego (od **a** do **z**) 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 plików QDLS jako różne od wielkich.

### Informacje pokrewne

Nazwa dokumentu i folderu

## Nazwy ścieżek w systemie plików QDLS

Każdy składnik ścieżki może być po prostu nazwą.

- Na przykład:

`/QDLS/FLR1/DOC1`

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

`/QDLS/FLR1/DOC1.TXT`

- Nazwa w każdym ze składników może mieć do 8 znaków, a rozszerzenie (o ile występuje) może mieć do 3 znaków. Maksymalna długość ścieżki bezwzględnej zaczynającej się od `/QDLS` wynosi 82 znaki.
- Hierarchia katalogów w ramach systemu plików 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 dotyczących stron kodowych można znaleźć w temacie Centrum informacyjnego iSeries omawiającym globalizację serwera i5/OS.

### Pojęcia pokrewne

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również *pathname* w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Dowiązania w systemie plików QDLS

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

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

### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

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

Komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 69, z następującymi wyjątkami, działają w systemie plików QDLS:

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

- następujące komendy nie są obsługiwane:
  - APYJRNCHG
  - CHGJRNOBJ
  - | – DSPJRN
  - ENDJRN
  - | – RCLLNK
  - | – RCVJRNE
  - | – RTVJRNE
  - SNDJRNE
  - STRJRN

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 68.

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

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110, z wyjątkiem następujących, działają w systemie plików QDLS:

- funkcji `symlink()` można użyć tylko do tworzenia dowiązania do obiektu w systemie plików QDLS z innego systemu plików, który obsługuje dowiązania symboliczne,
- poniższe funkcje nie są obsługiwane:
  - `givedescriptor()`
  - `ioctl()`
  - `link()`
  - `QjoEndJournal()`
  - | – `QjoRetrieveJournalEntries()`
  - `QjoRetrieveJournalInformation()`
  - `QJORJIDI()`
  - `QJOSJRNE()`
  - `QjoStartJournal()`
  - `Qp0lGetPathFromFileID()`
  - `readlink()`
  - `takedescriptor()`

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## system plików nośników optycznych (QOPT),

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

Dodatkowo, system ten:

- zawiera hierarchiczną strukturę katalogów podobną do struktury systemów operacyjnych komputerów PC, takich jak DOS i OS/2,
- jest zoptymalizowany do obsługi danych wejściowych i wyjściowych plików strumieniowych,
- obsługuje dane przechowywane w plikach strumieniowych.

## Używanie systemu plików QOPT poprzez zintegrowany system plików

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

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

### Informacje pokrewne


Optical device programming

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

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

Gdy zintegrowany system plików opiera się na modelu programu środowiska Integrated Language Environment (ILE), system plików HFS opiera się na oryginalnym modelu programu 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 zintegrowany system plików.

Więcej informacji na temat funkcji API HFS znaleźć można w podręczniku Optical device programming .

### Informacje pokrewne

Funkcje API hierarchicznego systemu plików

## Rozróżnianie wielkości znaków w systemie plików QOPT

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

## Nazwy ścieżek w systemie plików QOPT

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

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

Więcej informacji o regułach nazw ścieżek w systemie plików zawiera temat "Path Name Rules" w publikacji Optical device programming .

### Pojęcia pokrewne

"Nazwa ścieżki" na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Dowiązania w systemie plików QOPT

System plików QOPT dopuszcza tylko jedno dowiązanie do obiektu. Nie można w nim tworzyć ani przechowywać dowiązań symbolicznych.

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

### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

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

Większość komend wymienionych w temacie “Dostęp przy użyciu komend języka CL” na stronie 69 działa w systemie plików QOPT. Istnieje jednak kilka wyjątków. Należy pamiętać, że korzystanie z tych komend CL w przypadku procesu wielowątkowego może nie być bezpieczne. Zależnie od formatu nośnika optycznego mogą obowiązywać pewne ograniczenia. Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 68.

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

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

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

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

Wszystkie funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110, z wyjątkiem następujących, działają w systemie plików “root”:

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

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## System plików NetWare (QNetWare)

System plików QNetWare daje dostęp do danych na autonomicznych serwerach PC działających w systemie Novell NetWare 5.1 lub 6.0.

System plików QNetWare oferuje dodatkowo następujące funkcje:

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

### Uwagi:

1. System plików QNetWare jest dostępny tylko wtedy, gdy w systemie zainstalowano produkt NetWare Enhanced Integration, opcję BOSS 25. Po jego instalacji i wykonaniu IPL katalog i podkatalogi systemu plików /QNetWare zostaną wyświetlone jako część struktury katalogów zintegrowanego systemu plików.
2. Produkt NetWare Enhanced Integration nie obsługuje usług Novell Storage Services (NSS), toteż dostęp do danych zawartych na partycji może być ograniczony.

## 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 systemu plików (Add Mounted File System - ADDMFS), na przykład podłączenia systemu plików z możliwym zapisem (read-write) w trybie tylko do odczytu (read-only). Więcej informacji znaleźć można w temacie Komenda Dodanie podłączonego systemu plików (Add Mounted File System - ADDMFS).

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/SERWER.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.O/ORG\_UNIT.OU/SVR1\_VOL.CN  
Rozszerzenie .TRE służy do oznaczenia drzew NDS, .C oznacza kraje, .O organizacje, .OU jednostki organizacyjne, a .CN nazwy zwykłe. 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.

## Używanie systemu plików QNetWare poprzez interfejs zintegrowanego systemu plików

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

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.

Gdy użyte zostaną komendy i funkcje API do pobrania lub przypisania uprawnień właścicieli lub użytkowników, QNetWare odwzorowuje użytkowników NetWare w grupie użytkowników serwera iSeries w oparciu o ich nazwy. Jeśli nazwa NetWare jest dłuższa niż dziesięć znaków lub odpowiedni użytkownik serwera iSeries nie istnieje, 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. Gdy uprawnienia są przenoszone do i z serwera, odwzorowywane są na uprawnienia serwera 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. Komenda Zmiana wartości kontroli (Change Auditing Value - CHGAUD) nie jest więc 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 także obsługuje przestrzenie nazw wielu 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 NetWare Directory Services w systemie plików QNetWare

System plików QNetWare obsługuje wyświetlanie nazw usługi Network Directory Services (NDS) przy użyciu 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 w katalogach "root" (/) lub QOpenSys, które wskazują plik lub katalog QNetWare.

#### Pojęcia pokrewne

"Dowiązanie" na stronie 12

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

### Korzystanie z komend i terminali zintegrowanego systemu p

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

Komendy, które zawiera temat "Dostęp przy użyciu komend języka CL" na stronie 69, działają w systemie plików QNetWare; z następującymi wyjątkami:

- ADDLINK

- APYJRNCHG
- CHGAUD
- CHGJRNOBJ
- CHGPGP
- CHKIN
- CHKOUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- SNDJRNE
- 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**

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110 działają w systemie plików QNetWare z wyjątkiem następujących:

- givedescriptor()
- link()
- QjoEndJournal()
- | • QjoRetrieveJournalEntries()
- 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()

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **system plików iSeries NetClient (QNTC).**

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych na serwerze Integrated xSeries Server for iSeries, na którym działa system Linux lub system Windows NT 4.0 Server lub nowszy. System plików QNTC zapewnia również dostęp do danych i obiektów przechowywanych na serwerach zdalnych, na których działa system Windows NT 4.0 lub nowszy, protokół Samba 3.0 lub nowszy systemu Linux lub obsługiwana wersja programu iSeries NetServer.

System plików QNTC umożliwia aplikacjom serwera iSeries używanie danych zapisanych na serwerach systemów Windows i Linux.

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

## **Używanie systemu plików QNTC poprzez interfejs zintegrowanego systemu plików**

Używając produktu iSeries NetServer, iSeries Navigator, komend , ekranów użytkownika lub funkcji API można uzyskać dostęp do systemu plików QNTC poprzez interfejs zintegrowanego systemu plików.

Należy pamiętać o poniższych uwagach i ograniczeniach.

### **Uprawnienia i prawa własności w systemie plików QNTC**

System plików QNTC nie obsługuje pojęcia własności pliku lub katalogu.

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

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

### **Rozróżnianie wielkości znaków w systemie plików QNTC**

System plików QNTC zachowuje małe i wielkie litery we 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 składa się z nazwy systemu plików, nazwy serwera, nazwy udziału, nazw katalogu i podkatalogów oraz nazwy obiektu.

Wymagania dotyczące nazwy ścieżki są następujące:

- Ścieżka musi zaczynać się od ukośnika (/) i może zawierać co najwyżej 255 znaków. Ścieżki dostępu mają następującą postać:  
/QNTC/NazwaSerwera/NazwaWspółużytkowana/Katalog/  
. . . /Obiekt  
(QNTC stanowi wymaganą część nazwy ścieżki.)
- W nazwach ścieżek rozróżniane są wielkie i małe litery.
- Nazwa serwera może mieć do 15 znaków. Musi być ona częścią ścieżki.
- Nazwa zasobu współużytkowanego może mieć do 12 znaków.
- Każdy składnik ścieżki (występujący po nazwie zasobu współużytkowanego) może mieć do 255 znaków.
- W ramach QNTC dostępnych jest w zasadzie 130 poziomów hierarchii. Jeśli wszystkie składniki ścieżki zostały dodane do ścieżki jako poziomy hierarchii, hierarchia katalogów może mieć aż 132 poziomy.
- Nazwy są przechowywane w identyfikatorze CCSID Unicode.
- Domyślnie każdy działający serwer w lokalnej podsieci automatycznie będzie wyświetlany w systemie plików /QNTC jako katalog. Do dodania dostępnych dla użytkowników serwerów znajdujących się poza lokalną podsiecią można użyć komendy Tworzenie katalogu (Make Directory - MKDIR) lub funkcji API mkdir().

#### Pojęcia pokrewne

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

#### Informacje pokrewne

Komenda Utworzenie katalogu (Make Directory - MKDIR)

Funkcja API Utworzenie katalogu (Make Directory - mkdir())

## Dowiązania w systemie plików QNTC

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

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

#### Pojęcia pokrewne

“Dowiązanie” na stronie 12

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

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

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

Komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 69, z następującymi wyjątkami, działają w systemie plików QNTC:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHGOWN
- CHGAUT
- CHGPGP
- CHKIN
- CHKOUT

- DSPAUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- RST (dostępna w serwerach Integrated xSeries Servers)
- SAV (dostępna w serwerach Integrated xSeries Servers)
- SNDJRNE
- STRJRN
- WRKAUT
- WRKOBJOWN
- WRKOBJPGP

Te same ograniczenia dotyczą ekranów użytkownika opisanych w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 68.

## | **Zmienne środowiskowe systemu plików QNTC**

| Funkcjonowanie systemu plików QNTC podczas przeglądania sieci jest zależne od 2 zmiennych środowiskowych.  
| Obsługa tych zmiennych została wprowadzona w wersji V5R4 systemu i5/OS. Można je utworzyć za pomocą komendy  
| CL ADDENVVAR.

## | **QZLC\_SERVERLIST**

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

## | **QIBM\_ZLC\_NO\_BROWSE**

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

## **Tworzenie katalogów w systemie plików QNTC**

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

| Domyślnie katalog QNTC tworzony jest automatycznie dla wszystkich działających serwerów w domenie serwera  
| iSeries NetServer i lokalnej podsięci. Serwery znajdujące się poza lokalną podsięcią lub domeną serwera iSeries  
| NetServer należy dodać używając 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:**

- Konfigurując serwer iSeries NetServer dla protokołu WINS można automatycznie utworzyć katalogi dla serwerów znajdujących się poza lokalną podsiecią.
- Użycie funkcji API mkdir() lub komendy CL MKDIR w celu dodania katalogów do struktury katalogów spowoduje, że katalogi te przestaną być widoczne po ponownym wykonaniu IPL. Komenda MKDIR lub funkcja API mkdir() musi być ponownie użyta po każdym IPL systemu.

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

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

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

#### Informacje pokrewne

Komenda Utworzenie katalogu (Make Directory - MKDIR)

Funkcja API Utworzenie katalogu (Make Directory - mkdir())

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

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110, z wyjątkiem następujących, działają w systemie plików QNTC:

- użycie funkcji chmod(), fchmod(), utime() lub umask() do obiektów w QNTC nie będzie miało żadnego skutku, ale próba ich użycia nie spowoduje błęd;
- system plików QNTC nie obsługuje poniższych funkcji:
  - chown()
  - fchown()
  - fclear()
  - fclear64()
  - givedescriptor()
  - link()
  - QjoEndJournal()
  - QjoRetrieveJournalEntries()
  - QjoRetrieveJournalInformation()
  - QJORJIDI()
  - QJOSJRNE()
  - QjoStartJournal()
  - Qp0lGetPathFromFileID()
  - readlink()
  - symlink()
  - takedescriptor()

#### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

## Aktywowanie usługi uwierzytelniania sieciowego w systemie plików QNTC

System plików QNTC umożliwia w systemie iSeries dostęp do serwerów CIFS, które obsługują protokół uwierzytelniania Kerberos V5.

Zamiast używać hasła zarządzania siecią LAN do uwierzytelniania w każdym serwerze, poprawnie skonfigurowany serwer iSeries będzie mógł uzyskać dostęp do obsługiwanych serwerów CIFS w pojedynczej transakcji logowania.

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

- Usługę uwierzytelniania sieciowego (NAS)
- Odwzorowanie tożsamości dla przedsiębiorstwa (EIM)

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

- Profil użytkowników iSeries musi mieć przypisany parametr lokalnego zarządzania hasłami, LCLPMDMGT, z wartością **\*NO**. Podanie wartości **\*NO**, tego parametru spowoduje, że użytkownicy nie będą mieli hasła do systemu i nie będą mogli wpisać się w sesji 5250. Jedyny dostęp do systemu będzie możliwy poprzez aplikacje obsługujące uwierzytelnianie NAS, takie jak iSeries Navigator lub emulator terminalu iSeries Access.

Jeśli użytkownik poda wartość **\*YES**, hasło będzie obsługiwane przez system i użytkownik zostanie pomyślnie uwierzytelniony bez wykorzystania usługi NAS.

- Wymagany jest bilet protokołu kerberos i połączenie z aplikacją iSeries Navigator.
- Używany bilet protokołu kerberos dla systemu iSeries musi być przekazywany. Aby bilet mógł być przekazywany, należy wykonać następujące kroki:
  - Uzyskać dostęp do narzędzia 'Active Directory Users and Computers' w KDC dla dziedziny usługi NAS.
  - Wybrać użytkowników.
  - Wybrać nazwę odpowiadającą głównej nazwie usługi.
  - Wybrać opcję Properties (Właściwości).
  - Wybrać zakładkę Account (Konto).
  - W polu Account options (Opcje konta) zaznaczyć opcję 'Account is trusted for delegation' (Konto zaufane do przekazania)

#### **Informacje pokrewne**

Usługa uwierzytelniania sieciowego

Odwzorowanie tożsamości dla przedsiębiorstwa (EIM)

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

System plików QFileSvr.400 zapewnia przejrzysty dostęp do innych systemów plików znajdujących się w 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. System plików QFileSvr.400 współdziała z serwerem plików i5/OS w systemie docelowym w celu wykonywania bieżących operacji na zbiorach.

## **Używanie systemu plików QFileSvr.400 poprzez interfejs zintegrowanego systemu plików**

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

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

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

## **Rozróżnianie wielkości znaków w systemie plików QFileSvr.400**

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

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

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

## Nazwy ścieżek w systemie plików QFileSvr.400

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

- Poprawną postacią jest:

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

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.

### Pojęcia pokrewne

“Niezmienność nazw” na stronie 19

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

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również *pathname* w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

## Komunikacja w systemie plików QFileSvr.400

Komunikacja w QFileSvr.400 odbywa się na poniższe sposoby.

- Połączenia TCP z serwerem plików serwera docelowego mogą zostać ustanowione tylko wtedy, gdy jest w nim aktywny podsystem QSERVER.
- Próby połączenia SNA LU 6.2 wykonywane są tylko w przypadku, gdy istnieje nieużywana sesja obsługiwana lokalnie (na przykład sesja ustanowiona specjalnie do użycia przez połączenie LU 6.2). Podczas nawiązywania połączeń LU 6.2 system plików QFileSvr.400 używa trybu BLANK. W systemie docelowym zadanie QPWFSERV jest wpisywane do kolejki w podsystemie QSERVER. Profil użytkownika tego zadania jest zdefiniowany przez pozycję dotyczącą komunikacji dla trybu BLANK. Więcej informacji o 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 i5/OS.
- Jeśli nie zostało ustanowione połączenie z serwerem docelowym, system plików QFileSvr400 zakłada, że katalog pierwszego poziomu reprezentuje nazwę hosta TCP/IP. Aby ustanowić połączenie z serwerem docelowym, system plików QFileSrv.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 dwojaka, w zależności od sytuacji:
  - Jeśli usługa "as-file" znajduje się w tabeli usług serwera docelowego, program odwzorowujący serwera zwraca numer portu, na którym nasłuchuje demon serwera plików i5/OS.
  - Jeśli program odwzorowujący serwera docelowego nie jest aktywny, używany jest domyślny numer portu dla usługi "as-file" (8473).

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

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

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

Remote1 jest w tym przypadku systemem lokalnym. Gdy ścieżka zawiera zapętlenie, po krótkim czasie system plików QFileSvr.400 zgłosi błąd przekroczenia czasu oczekiwania (time-out).

Przy komunikacji poprzez SNA, system plików QFileSvr.400 użyje istniejącej wolnej sesji. Aby prawidłowo podłączyć się do zdalnego systemu komunikacyjnego, konieczne jest uruchomienie trybu i nawiązanie sesji dla QFileSvr.400.

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

- | Jeśli w obu systemach skonfigurowane są usługi NAS i EMI, a 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.

- 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.

### Informacje pokrewne

Usługa uwierzytelniania sieciowego

Odwzorowanie tożsamości dla przedsiębiorstwa (EIM)



## Dowiązania w systemie plików QFileSvr.400

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

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

### Pojęcia pokrewne

"Dowiązanie" na stronie 12

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

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

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

Komendy wymienione w temacie "Dostęp przy użyciu komend języka CL" na stronie 69, z wyjątkiem następujących, działają w systemie plików QFileSvr.400:

- ADDLNK
- APYJRNCHG
- CHGAUT
- CHGJRNOBJ
- CHGOWN
- DSPAUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- RST
- | • RTVJRNE
- SAV
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Te same ograniczenia odnoszą się do ekranów użytkownika opisanych w temacie "Dostęp przy użyciu pozycji menu i ekranów" na stronie 68.

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

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

Funkcje API wymienione w temacie "Wykonywanie operacji za pomocą funkcji API" na stronie 110, z wyjątkiem następujących, działają w systemie plików QFileSvr.400:

- chown()
- fchown()
- fclear()
- fclear64()
- givedescriptor()

- link()
- QjoEndJournal()
- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE
- QjoStartJournal
- Qp0lGetPathFromFileID()
- symlink()
- takedescriptor()

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

## **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.

## **Używanie systemów plików NFS poprzez interfejs zintegrowanego system plików**

System plików NFS jest dostępny poprzez interfejs zintegrowanego system plików. Należy pamiętać o poniższych 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 sieciowy system plików (NFS) funkcjonuje inaczej i inna jest jego charakterystyka.

Możliwościami tymi są:

1. Użytkownik podłącza system plików serwera iSeries w kliencie.

2. Użytkownik podłącza system plików serwera UNIX w kliencie.
3. Użytkownik podłącza system plików serwera nie będącego serwerem iSeries ani serwerem UNIX w kliencie.

W pierwszym przypadku podłączony system plików zachowuje się na kliencie podobnie, jak zachowywałby się na serwerowi 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 on miał charakterystyki i ograniczenia systemu plików QDLS. Na przykład w systemie plików QDLS komponenty nazwy ścieżki mają ograniczenie 8 znaków plus 3 znakowe rozszerzenie. Jednak podłączony system plików będzie miał także ograniczenia i charakterystykę systemu plików NFS. Na przykład, nie można użyć komendy CHGAUD do zmiany wartości kontroli obiektu systemu plików NFS.

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

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

#### **Odsyłacze pokrewne**

“System plików systemów otwartych (QOpenSys)” na stronie 34

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

## **Dowiązania w systemie plików NFS**

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

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

#### **Pojęcia pokrewne**

“Dowiązanie” na stronie 12

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

## **Komendy zintegrowanego systemu plików w systemie plików NFS**

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

Wszystkie komendy wymienione w temacie “Dostęp przy użyciu komend języka CL” na stronie 69 i ekrany opisane w temacie “Dostęp przy użyciu pozycji menu i ekranów” na stronie 68, z wyjątkiem następujących, działają również w sieciowym systemie plików (NFS):

- APYJRNCHG
- CHGJRNOBJ
- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP
- CHKIN
- CHKOUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK

- | • RCVJRNE
- | • RTVJRNE
- SNDJRNE
- STRJRN

Istnieją jednak komendy odnoszące się tylko do pewnych podłączonych systemów plików, w tym do NFS. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego. Poniższa tabela opisuje te komendy. Pełny opis komend i ekranów związanych głównie z systemem plików NFS zawiera publikacja OS/400

Network File System Support  .

Tabela 6. Komendy CL sieciowego systemu plików

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


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

## Funkcje API zintegrowanego systemu plików w systemie plików NFS

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

Funkcje API wymienione w temacie “Wykonywanie operacji za pomocą funkcji API” na stronie 110, z wyjątkiem następujących, działają w systemie plików NFS:

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

Pełny opis funkcji języka C związanych głównie z systemem plików NFS zawiera publikacja OS/400 Network File System Support .

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

### Informacje pokrewne

Aplikacyjne interfejsy programistyczne (API)

---

## Dostęp do zintegrowanego systemu plików

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

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

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

Z poziomu 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 można także obsługiwać przy użyciu narzędzia iSeries Navigator, graficznego interfejsu użytkownika (GUI), służącego do zarządzania i administrowania serwerem z pulpitu systemu Windows.

## Dostęp przy użyciu pozycji menu i ekranów

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

Aby wyświetlić menu zintegrowanego systemu plików:

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

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

Zintegrowany system plików stanie się również dostępny z dowolnego menu 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**.

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

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

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

Niektóre systemy plików nie obsługują wszystkich powyższych operacji.

#### Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

#### Odsyłacze pokrewne

“Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 73

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

“Dostęp przy użyciu komend języka CL”

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

“Używanie komend zintegrowanego systemu plików w systemie plików "root" (/)” na stronie 33

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

## Dostęp przy użyciu komend języka CL

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

Tabela 1 zawiera spis komend zintegrowanego systemu plików. Więcej informacji o komendach CL związanych z systemami plików użytkownika, systemem Network File System i podłączonym systemem plików zawierają tematy “Systemy plików użytkownika (UDFS)” na stronie 36 i “sieciowy system plików (Network File System - NFS),” na stronie 65. Jeśli komenda wykonuje tę samą operację, co komenda systemu OS/2 lub DOS, utworzono alias (alternatywną nazwę komendy) dla wygody użytkowników systemów OS/2 i DOS.

Tabela 7. Komendy zintegrowanego systemu plików

Komenda	Opis	Alias
ADDLNK	Dodanie dowiązania (Add Link). Dodaje dowiązanie między katalogiem i obiektem.	
ADDMFS	Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta.	MOUNT

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
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
CHGJRNOBJ <sup>2</sup>	Zmiana obiektów kronikowanych. Zmienia atrybuty kronikowania dla obiektu lub listy obiektów bez konieczności zakończenia i ponownego uruchomienia kronikowania dla obiektu.	
CHGNFSEXP	Zmiana eksportu NFS (Change Network File System Export). Dodaje drzewa katalogów lub usuwa je z tabeli eksportu systemów plików, które są eksportowane do klientów NFS.	EXPORTFS
CHGOWN	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). Dostarcza informacji o konwersji katalogów zintegrowanego systemu plików z formatu *TYPE1 do formatu *TYPE2.	
CVTRPCSRC	Konwersja kodu źródłowego RPC (Convert RPC Source). Generuje kod w języku C na podstawie pliku wejściowego napisanego w języku RPC.	RPCGEN
DLTUDFS	Usunięcie systemu plików UDFS (Delete UDFS). Usuwa system plików definiowany przez użytkownika (UDFS).	
DSPAUT	Wyświetlenie uprawnień (Display Authority). Wyświetla listę użytkowników mających uprawnienia do obiektu i ich uprawnienia do tego obiektu.	
DSPCURDIR	Wyświetlenie bieżącego katalogu (Display Current Directory). Wyświetla nazwę bieżącego katalogu.	



Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
DSPJRN <sup>2</sup>	Wyświetlenie kroniki (Display Journal). Konwertuje pozycje kroniki (zawarte w jednym lub wielu dziennikach) na postać umożliwiającą zewnętrzną reprezentację.	
DSPLNK	Wyświetlenie dowiązań obiektu (Display Object Links). Wyświetla listę obiektów w katalogu, zawiera opcje umożliwiające wyświetlenie informacji o obiektach.	
DSPF	Wyświetlenie pliku strumieniowego (Display Stream File). Wyświetla plik strumieniowy lub zbiór bazy danych.	
DSPMFSINF	Wyświetlenie danych podłączonego systemu plików (Display Mounted File System Information). Wyświetla informacje na temat dołączonego systemu plików.	STATFS
DSPUDFS	Wyświetlenie systemu plików UDFS (Display UDFS). Wyświetla system plików definiowany przez użytkownika (UDFS).	
EDTF	Edycja zbioru strumieniowego (Edit Stream File). Przeprowadza edycję pliku strumieniowego lub zbioru bazy danych.	
ENDJRN <sup>2</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
PRTDIRINF	Wydruk informacji katalogu. Dla obiektów w zintegrowanym systemie plików służy do drukowania informacji katalogu, zebranych przez komendę Pobranie informacji katalogu (Retrieve Directory Information - RTVDIRINF).	
RCLLNK	Odzyskanie dowiązań obiektów (Reclaim Object Links). Identyfikuje i, jeśli jest to możliwe, usuwa problemy z używanymi podłączonymi systemami plików.	
RCVJRNE <sup>2</sup>	Pobranie pozycji kroniki (Receive Journal Entry). Umożliwia określonemu programowi obsługi wyjścia użytkownika otrzymywanie pozycji kroniki w sposób ciągły.	
RLSIFSLCK	Zwolnienie blokad zintegrowanego systemu plików (Release in Integrated File System Locks). Zwalnia wszystkie blokady zakresu bitów nałożone na klienta NFS 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).	

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

Komenda	Opis	Alias
RTVDIRINF	Pobranie informacji katalogu. Służy do zbierania atrybutów obiektów w zintegrowanym systemie plików.	
RTVJRNE <sup>2</sup>	Odtworzenie pozycji kroniki (Retrieve Journal Entry). Umożliwia odtworzenie określonej pozycji kroniki i umieszczenie wyników w zmiennych 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).	
STRNFSSVR	Uruchomienie serwera NFS (Start Network File System Server). Uruchamia jeden lub wszystkie demony NFS w serwerze.	
WRKAUT	Praca z uprawnieniami (Work with Authority). Wyświetla listę użytkowników i ich uprawnienia, a także zawiera opcje dodawania użytkowników, zmiany ich uprawnień i usuwania ich.	
WRKLNK	Praca z dowiązaniem obiektów (Work with Object Links). Wyświetla listę obiektów w katalogu i zawiera opcje przeprowadzania działań na obiektach.	
WRKOBJOWN <sup>1</sup>	Praca z obiektami według właścicieli (Work with Objects by Owner). Wyświetla listę obiektów należących do profilu użytkownika i zawiera opcje przeprowadzania działań na obiektach.	
WRKOBJJGP <sup>1</sup>	Praca z obiektami według grupy głównej (Work with Objects by Primary Group). Wyświetla listę obiektów kontrolowanych przez grupę podstawową i zawiera opcje przeprowadzania działań na obiektach.	

**Uwagi:**

1. Komendy WRKOBJOWN i WRKOBJJGP mogą wyświetlać wszystkie typy obiektów, ale jest możliwe, że nie będą w pełni funkcjonować we wszystkich systemach plików.
2. Więcej informacji można znaleźć w temacie Zarządzanie kronikami w Centrum informacyjnym iSeries.

**Pojęcia pokrewne**

“Praca z systemami plików” na stronie 26

System plików (*file system*) umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

**Zadania pokrewne**

“Dostęp przy użyciu pozycji menu i ekranów” na stronie 68

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

**Odsyłacze pokrewne**

“Używanie komend zintegrowanego systemu plików w systemie plików *“root” (/)*” na stronie 33

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

**Informacje pokrewne**

Język CL

## Reguły dotyczące nazw ścieżek dla komend CL i ekranów

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

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

- Nazwy obiektów wewnątrz każdego katalogu muszą być unikalne.
- Nazwa ścieżki przekazana do komendy CL zintegrowanego systemu plików musi być przedstawiona za pomocą identyfikatora 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 znakach pojedynczego cudzysłowu (\*). Gdy ścieżki wpisuje się na ekranach, apostrofy nie są obowiązkowe. Jeśli ścieżka zawiera ciągi znaków w cudzysłowie, należy ją ująć w znaki ' '.
- Ścieżkę należy wprowadzać od lewej do prawej strony, poczynając od najwyższego poziomu katalogu i kończąc na obiekcie, na którym ma być wykonana komenda. Nazwy składników ścieżki oddzielane są ukośnikiem (/).

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

Na przykład:

```
'Kat1/Kat2/Kat3/ZbiórUż'
```

lub

```
'Dir1\Dir2\Dir3\UsrFile'
```

- Znaków ukośnika (/), ukośnika odwrotnego (\) oraz znaków pustych nie można używać w poszczególnych składnikach ścieżki, gdy znaki te pełnią funkcję znaków rozdzielających. Komendy nie zamieniają małych liter na wielkie. Nazwa może zostać zamieniona na pisaną wielkimi literami lub nie, w zależności od tego, czy system plików zawierający obiekt uwzględnia wielkość liter i czy obiekt jest tworzony, czy poszukiwany.
- Długość nazwy obiektu jest ograniczona przez system plików, w którym znajduje się obiekt, i przez maksymalną długość ciągu komendy. Komendy akceptują nazwy obiektów o maksymalnej długości 255 znaków i ścieżki o maksymalnej długości 5000 znaków.
- Znak rozdzielający (na przykład /) na początku ścieżki oznacza, że ścieżka rozpoczyna się od najwyższego katalogu, na przykład: "root" (/).  

```
'/Kat1/Kat2/Kat3/ZbiórUż'
```
- Jeśli ścieżka nie rozpoczyna się znakiem rozdzielającym (na przykład: /), to przyjmuje się, że rozpoczyna się w bieżącym katalogu użytkownika, który wprowadził komendę, na przykład:  

```
'MójKat/MójZbiór'
```

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

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

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

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

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

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

- Aby uniknąć pomyłek związanych z wartościami specjalnymi 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'
```

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

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

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

- gwiazdka (\*),

**Uwaga:** Aby uniknąć zamieszania związanego z użyciem wartości specjalnych serwera iSeries, nie należy posługiwać się nazwami ścieżek rozpoczynającymi się od znaku pojedynczej gwiazdki (\*).

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

Na przykład:

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

lub

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

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

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

```
crtdir dir(X'02')
```

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

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

### Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

“Nazwa ścieżki” na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

### Odsyłacze pokrewne

“Reguły dotyczące ścieżek dla funkcji API” na stronie 118

Gdy do wykonywania działań na obiekcie używane są zintegrowany system plików lub funkcja API C/400, obiekt identyfikowany jest przez podanie ścieżki do katalogu. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API.

### Informacje pokrewne

Język CL

## Praca z danymi wyjściowymi komend RTVDIRINF i PRDIRINF

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

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

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

Tabela 8. QAEZDxxxxO (plik zawierający atrybuty obiektów)

Nazwa pola	Typ pola	Opis pola
QEZACCTIM	TIMESTAMP	Data i godzina ostatniego dostępu do danych obiektu.
QEZALCSIZE <sup>1</sup>	BIGINT	Liczba bajtów przydzielonych dla obiektu.
QEZALWCKPW	SMALLINT	Określa, czy plik strumieniowy (*STMF) może być współużytkowany przez programy czytające i piszące w trakcie przetwarzania punktu kontrolnego składowania podczas użycia. Poprawne wartości:  0 - Obiekt może być współużytkowany tylko przez programy czytające.  1 - Obiekt może być współużytkowany przez programy czytające i piszące.
QEZASP	SMALLINT	Puła pamięci dyskowej, w której obiekt jest przechowywany.
QEZAUDT	GRAPHIC (10)	Wartość kontroli przypisana do obiektu. Poprawne wartości:  *NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.  *USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.  *CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.  *ALL - Kontrola obiektu jest przeprowadzana dla każdego dostępu do obiektu, uzyskanego przez dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.

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

Nazwa pola	Typ pola	Opis pola
QEZAUTLST	GRAPHIC (10)	Nazwa listy autoryzacji używanej do zabezpieczania wskazanego obiektu. Wartość *NONE oznacza, że do określenia uprawnień do obiektu nie jest używana żadna lista autoryzacji.
QEZBLKSIZ	INTEGER	Wielkość bloku obiektu.
QEZCASE	SMALLINT	Wskazuje rozróżnianie wielkości znaków systemu plików, w którym znajduje się obiekt.  0 - System plików nie rozróżnia wielkości znaków.  1 - System plików rozróżnia wielkość znaków.
QEZCCSID	INTEGER	Identyfikator CCSID danych i atrybutów rozszerzonych obiektu.
QEZCEAS	BIGINT	Liczba krytycznych atrybutów rozszerzonych przypisanych do obiektu.
QEZCHGTIMA <sup>1</sup>	TIMESTAMP	Data i godzina ostatniej zmiany atrybutów obiektu.
QEZCHGTIMD	TIMESTAMP	Data i godzina ostatniej zmiany danych obiektu.
QEZCHKOUT <sup>1</sup>	SMALLINT	Wskaźnik sprawdzenia obiektu. Poprawne wartości:  0 - Obiekt nie jest sprawdzany.  1 - Obiekt jest sprawdzany.
QEZCHKOWN	GRAPHIC (10)	Użytkownik, który uruchomił sprawdzenie obiektu. Pole to jest puste, jeśli sprawdzenie nie zostało zaznaczone.
QEZCHKTIM	TIMESTAMP	Data i godzina ostatniego sprawdzenia obiektu. Jeśli obiekt nie jest sprawdzany, pole to zawiera wartość NULL.
QEZCLSTRSP	SMALLINT	Obiekt należy do pamięci dyskowej przydzielonej dla zintegrowanych serwerów xSeries jako wirtualne napędy dysków serwerów xSeries. Z perspektywy serwera iSeries wirtualne dyski widziane są jako pliki strumieniowe w zintegrowanym systemie plików.  0 - Obiekt nie jest pamięcią dysku wirtualnego.  1 - Obiekt jest pamięcią dysku wirtualnego.
QEZCRTAUD	GRAPHIC (10)	Wartość kontroli przypisana do obiektu utworzonego w katalogu. Poprawne wartości:  *NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.  *USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.  *CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.  *ALL - Kontrola obiektu jest przeprowadzana po każdym dostępie do obiektu dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.
QEZCRTTIM	TIMESTAMP	Data i godzina utworzenia obiektu.
QEZDIRIDX	INTEGER	Indeks katalogu nadrzędnego.



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

Nazwa pola	Typ pola	Opis pola
QEZDIRTYP2	SMALLINT	Format podanego obiektu katalogu. Poprawne wartości:  0 - Format katalogu to *TYPE1.  1 - Format katalogu to *TYPE2.
QEZDOM	GRAPHIC (10)	Domena obiektu. Poprawne wartości:  *SYSTEM - Obiekt istnieje w domenie systemowej.  *USER - Obiekt istnieje w domenie użytkownika.
QEZDSTGOPT	SMALLINT	Opcja ta służy do określenia sposobu przydzielenia pamięci dyskowej w systemie dla wskazanego obiektu. Opcję tę można podać tylko dla plików strumieniowych w systemach plików "root" (/), QOpenSys i użytkownika (UDFS). opcja ta zostanie zignorowana dla bajtowych plików strumieniowych *TYPE1. Poprawne wartości:  0 - Pamięć dyskowa zostanie przydzielona normalnie. Oznacza to, że gdy wymagana będzie dodatkowa pamięć dyskowa, zostanie przydzielona w obszarach o logicznych wielkościach odpowiednich do pomieszczenia bieżących i przewidywanych wymagań pamięci z jednoczesnym minimalizowaniem liczby operacji dyskowych we/wy.  1 - Pamięć dyskowa będzie przydzielana tak, aby zminimalizować przestrzeń zajmowaną przez obiekt. Oznacza to, że w miarę potrzeb dodatkowa pamięć dyskowa będzie przydzielana w niewielkich ilościach, aby zaspokoić bieżące zapotrzebowanie. Dostęp do obiektu o wielu małych przydziałach może zwiększyć liczbę dyskowych operacji wejścia/wyjścia dla danego obiektu.  2 - System dynamicznie określi optymalną wielkość pamięci dyskowej dla obiektu, usiłując zrównoważyć ilość pamięci i dyskowych operacji we/wy. Na przykład, jeśli plik zajmuje dużo małych rozszerzeń pamięci dyskowej, a jednocześnie często występują dla niego operacje odczytu i zapisu, przyszłe operacje przydziału pamięci dyskowej przypiszą większe przestrzenie pamięci w celu zminimalizowania liczby operacji we/wy. Jeśli plik jest często obcinany, przyszłe operacje przydziału pamięci dyskowej będą przypisywały małe jednostki pamięci, w celu zminimalizowania zajętej przestrzeni dyskowej. Dodatkowo, informacje dla tego systemu i jego działań będą obsługiwane według wielkości pliku strumieniowego. Informacje o wielkości tego pliku posłużą także do określenia optymalnej wielkości przydziału pamięci dyskowej dla obiektu względem wielkości innych obiektów.
QEZDTASIZE	BIGINT	Wielkość danych w obiekcie podana w bajtach. Wielkość ta nie obejmuje nagłówek obiektów ani wielkości atrybutów rozszerzonych przypisanych do obiektu.
QEZEAS	BIGINT	Liczba atrybutów rozszerzonych przypisanych do obiektu.
QEZEEXTATRS	BIGINT	Całkowita liczba bajtów dla wszystkich danych atrybutów rozszerzonych.
QEZFILEID <sup>1</sup>	GRAPHIC (16)	ID zbioru dla obiektu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu.
QEZFILEIDS	INTEGER	4-bajtowy ID zbioru. Numer ten jednoznacznie identyfikuje obiekt w systemie plików. Nie może on identyfikować obiektu w całym systemie.
QEZFILTYP2 <sup>1</sup>	SMALLINT	Format pliku strumieniowego (*STMF). Poprawne wartości:  0 - Plik strumieniowy ma format *TYPE1.  1 - Plik strumieniowy ma format *TYPE2.
QEZFSID	BIGINT	ID systemu plików, w którym znajduje się obiekt. Numer ten jednoznacznie identyfikuje system plików, w którym znajduje się obiekt.
QEZGENID	BIGINT	ID generacji przypisany do ID zbioru.



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

Nazwa pola	Typ pola	Opis pola
QEZGID	INTEGER	Profile grup identyfikowane są według unikalnych numerów identyfikacyjnych grup (GID).
QEZINHSCN	GRAPHIC (1)	<p>Określa, czy obiekty utworzone w katalogu mają podlegać skanowaniu po zarejestrowaniu programów obsługi wyjścia w jednym ze zintegrowanych punktów wyjścia systemu plików związanych ze skanowaniem.</p> <p>Poprawne wartości:</p> <p>x'00' - po utworzeniu w katalogu obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>x'01' - po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>x'02' - po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu od momentu ostatniego skanowania obiektu. Nie będzie podlegać skanowaniu w przypadku aktualizacji oprogramowania do skanowania. Atrybut ten obowiązuje tylko wtedy, gdy podano wartość *USEOCOATR wartości systemowej Sterowanie skanowaniem systemów plików (QSCANFCTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p>
QEZJAFTERI	SMALLINT	<p>Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu po wprowadzeniu zmiany.</p> <p>0 - Obiekt nie jest kronikowany z obrazami po zmianach.  1 - Obiekt jest kronikowany z obrazami po zmianach.</p>
QEZJBEFORI	SMALLINT	<p>Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu przed wprowadzeniem zmian.</p> <p>0 - Obiekt nie jest kronikowany z obrazami przed wprowadzeniem zmian.  0 - Obiekt jest kronikowany z obrazami przed wprowadzeniem zmian.</p>
QEZJOPTENT	SMALLINT	<p>Gdy kronikowanie jest aktywne, kronikowane są pozycje opcjonalne. Lista pozycji opcjonalnych różni się zależnie od typu obiektu.</p> <p>0 - Obiekt nie jest kronikowany z pozycjami opcjonalnymi.  1 - Obiekt jest kronikowany z pozycjami opcjonalnymi.</p>
QEZJRCVASP	GRAPHIC(10)	<p>Nazwa puli ASP, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. Poprawne wartości:</p> <p>*SYSBAS - Dziennik znajduje się w puli ASP systemowej lub użytkownika.</p> <p>Urządzenie ASP - Nazwa urządzenia ASP, które zawiera dziennik.</p>
QEZJRCVLIB	GRAPHIC(10)	<p>Nazwa biblioteki, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.</p>

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

Nazwa pola	Typ pola	Opis pola
QEZRJVNAM	GRAPHIC(10)	Najstarszy dziennik wymagany do wprowadzenia zmian w dzienniku. Gdy pole informacji o wprowadzeniu zmian zawiera wartość PARTIAL_TRANSACTION, dziennik zawiera dane początku transakcji częściowej. W przeciwnym przypadku dziennik zawiera rozpoczęcie operacji składowania. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZRJNID	GRAPHIC (10)	Pole to przypisuje do kronikowanego obiektu identyfikator, którego można użyć w różnych komendach i funkcjach API związanych z kronikowaniem. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZRJNLIB	GRAPHIC (10)	Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się aktualnie używana kronika. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się kronika ostatnio używana dla obiektu. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZRJNNAM	GRAPHIC (10)	Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę aktualnie używanej kroniki. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę kroniki ostatnio używanej dla obiektu. Pole to jest puste jeśli obiekt nigdy nie był kronikowany.
QEZRJNSTR	TIMESTAMP	Liczba sekund, które upłynęły od epoki odpowiadającej dacie i godzinie uruchomienia kronikowania obiektu. Pole to ma wartość NULL jeśli obiekt nigdy nie był kronikowany.
QEZRJNSTS <sup>1</sup>	SMALLINT	Aktualny status kronikowania obiektu. W polu będzie jedna z następujących wartości:  0 (NOT_JOURNALED) - Obiekt nie jest aktualnie kronikowany.  1 (JOURNALED) - Obiekt jest aktualnie kronikowany.
QEZRJSUBTRE	SMALLINT	Gdy flaga ta zostanie zwrócona, oznacza, że obiekt jest katalogiem z semantyką poddrzewa kronikowania zintegrowanego systemu plików.  0 - Obiekt nie jest kronikowany z semantyką poddrzewa.  1 - Obiekt jest kronikowany z semantyką poddrzewa. Nowe obiekty utworzone w poddrzewie tego katalogu odziedziczają atrybuty kronikowania i opcje katalogu.
QEZRJTRNI	GRAPHIC(1)	Pole to zawiera informacje o aktualnym stanie obiektu i związane jest z granicami kontroli transakcji. Poprawne wartości:  x'00' (NONE) - Brak transakcji częściowych.  x'01' (PARTIAL_TRANSACTION) - Obiekt został odtworzony z transakcjami częściowymi. Nie można użyć obiektu do czasu, gdy użyta zostanie komenda Zastosowanie kronikowanych zmian (Apply Journaled Changes - APYJRNCHG) lub Usuwanie kronikowanych zmian (Remove Journaled Changes - RMVJRNCHG) w celu zastosowania lub wycofania zmian w transakcji częściowej.  x'02' (ROLLBACK_ENDED) - Operacja wycofania dla obiektu została zakończona przy użyciu opcji "Zakończenie wycofania (End Rollback)" na ekranie Praca z definicjami transakcji (Work with Commitment Definition - WRKCMDFN). Zalecane jest odzyskanie obiektu, ponieważ nie można go użyć. Ostatecznie, można użyć komendy Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ) w celu udostępnienia obiektu do użycia. Wykonanie tej czynności może spowodować brak spójności obiektu.
QEZLANGID	GRAPHIC (3)	Trzyznakowy ID oznaczający język nazwy obiektu (pole QEZOBJNAM).

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

Nazwa pola	Typ pola	Opis pola
QEZLOCAL	SMALLINT	Określa, czy obiekt jest przechowywany lokalnie, czy w systemie zdalnym. Decyzja o miejscu przechowywania obiektu zależy od reguł konkretnego systemu plików. Obiekty w systemach plików, które nie zawierają indykatora lokalnego lub zdalnego, traktowane są jak obiekty zdalne. Poprawne wartości:  1 - Dane obiektu przechowywane są lokalnie.  2 - Dane obiektu przechowywane są w systemie zdalnym.
QEZMLTSIG	SMALLINT	Określa, czy obiekt zawiera więcej niż jeden podpis cyfrowy systemu i5/OS. Poprawne wartości:  0 - Obiekt zawiera tylko jeden podpis cyfrowy.  1 - Obiekt zawiera więcej niż jeden podpis cyfrowy. Jeśli w polu QEZYSSIG znajduje się wartość 1, przynajmniej jeden z podpisów pochodzi ze źródła zaufanego w systemie.
QEZMODE	INTEGER	Tryb i rodzaj dostępu do pliku. Więcej informacji o trybie zawiera temat dotyczący funkcji API open().
QEZMSTGOPT	SMALLINT	Opcja ta służy do określenia sposobu przydzielenia i użycia pamięci głównej w systemie dla wskazanego obiektu. Opcję tę można podać tylko dla plików strumieniowych w systemach plików "root" (/), QOpenSys i użytkownika (UDFS). Poprawne wartości:  0 - Pamięć główna zostanie przydzielona normalnie. Przydzielone i użyte zostanie jak największą pamięć główną. Spowoduje to zminimalizowanie liczby operacji we/wy dla dysku, ponieważ informacje będą przechowywane w pamięci głównej.  1 - Pamięć główna będzie przydzielana tak, aby zminimalizować przestrzeń zajmowaną przez obiekt. Przydzielone i użyte zostanie jak najmniej pamięci głównej. Spowoduje to zminimalizowanie zużycia pamięci głównej i zwiększenie liczby operacji we/wy dla dysku, ponieważ mniej informacji będzie przechowywane w pamięci głównej.  2 - System dynamicznie określi optymalny przydział pamięci głównej dla obiektu w oparciu o pozostałe działania systemu i rywalizację o pamięć główną. Oznacza to, że gdy rywalizacja o pamięć główną jest niewielka, system przydzieli jak największą pamięć w celu zminimalizowania liczby operacji dyskowych wejścia/wyjścia. Kiedy rywalizacja o pamięć główną jest znaczna, system przydzieli mniej pamięci głównej w celu zminimalizowania rywalizacji. Opcja ta ma znaczenie tylko wtedy, gdy opcja stronicowania puli pamięci ma wartość *CALC. Gdy opcja stronicowania puli pamięci ma wartość *FIXED, system działa tak samo, jak dla opcji STG_NORMAL. Gdy dostęp do obiektu uzyskiwany jest poprzez serwer plików, opcja ta nie ma znaczenia. Jej działanie jest takie samo, jak dla opcji STG_NORMAL.
QEZNLNK	INTEGER	Liczba dowiązań stałych do obiektu.
QEZNMCCSID	INTEGER	Identyfikator CCSID oznaczający nazwę obiektu (pole QEZOBJNAM).
QEZNONSAV	SMALLINT	Określa, czy obiekt może być zeskładowany, czy nie. Poprawne wartości:  0 - Obiekt zostanie zeskładowany.  1 - Obiekt nie zostanie zeskładowany. Dodatkowo, jeśli obiekt jest katalogiem, żaden z obiektów w poddrzewie katalogu nie zostanie zeskładowany, jeśli nie zostanie osobno określony jako obiekt do składowania. Poddrzewo obejmuje wszystkie podkatalogi i obiekty w tych podkatalogach.
QEZOBJLEN	INTEGER	Liczba bajtów zawartych w nazwie obiektu (pole QEZOBJNAM).
QEZOBJNAM <sup>1</sup>	VARGRAPHIC (1024)	Nazwa obiektu. <sup>2</sup>

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

Nazwa pola	Typ pola	Opis pola
QEZOBJTYPE <sup>1</sup>	GRAPHIC (10)	Typ obiektu.
QEZOFLOW	SMALLINT	Wskazuje, czy obiekt spowodował przepełnienie puli pamięci, w której się znajduje. Poprawne wartości:  0 - Pula pamięci dyskowej nie jest przepełniona.  0 - Pula pamięci dyskowej jest przepełniona.
QEZOWN <sup>1</sup>	GRAPHIC (10)	Nazwa profilu użytkownika będącego właścicielem obiektu lub następująca wartość specjalna:  *NOUSRPRF - Ta wartość specjalna używana jest przez system plików Network File System do wskazania, że w lokalnym serwerze iSeries nie ma profilu użytkownika z ID użytkownika (UID) odpowiadającym UID obiektu zdalnego.
QEZOWNPGP	GRAPHIC (10)	Nazwa profilu użytkownika będącego grupą podstawową obiektu lub następujące wartości specjalne:  *NONE - Obiekt nie posiada skojarzonej z nim grupy podstawowej.  *NOUSRPRF - Ta wartość specjalna używana jest przez system plików Network File System do wskazania, że w lokalnym serwerze nie ma profilu użytkownika z ID grupy (GID) odpowiadającym GID obiektu zdalnego.
QEZPCARC	SMALLINT	Określa, czy obiekt zmienił się od ostatniego sprawdzenia obiektu.  0 - Obiekt nie zmienił się.  1 - Obiekt zmienił się.
QEZPCHID <sup>1</sup>	SMALLINT	Określa, czy obiekt można wyświetlić przy użyciu zwykłego listingu katalogów.  0 - Obiekt nie jest ukryty.  1 - Obiekt jest ukryty.
QEZPCREAD	SMALLINT	Określa, czy obiekt może być zapisany lub usunięty, jeśli jego atrybuty rozszerzone zostały zapisane lub usunięte albo zmieniono jego wielkość. Poprawne wartości:  0 - Obiekt można zmienić.  1 - Obiektu nie można zmienić.
QEZPCSYS	SMALLINT	Określa, czy obiekt jest plikiem systemowym i jest wyłączony z normalnego wyszukiwania katalogów.  0 - Obiekt nie jest plikiem systemowym.  1 - Obiekt jest plikiem systemowym.
QEZPRMLNK	SMALLINT	Gdy obiekt ma kilka nazw, pole to otrzyma wartość tylko dla pierwszej znalezionej nazwy.
QEZRDEV	BIGINT	Jeśli obiekt reprezentuje plik specjalny urządzenia, jest to rzeczywiste urządzenie reprezentowane przez ten obiekt.
QEZREGION	GRAPHIC (2)	Dwuznakowy ID kraju dla nazwy obiektu (pole QEZOBJNAM). Decyduje on o działaniach definiowanych według lokalizacji, takich jak kolejność zestawiania.

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

Nazwa pola	Typ pola	Opis pola
QEZSBINARY	GRAPHIC (1)	<p>Określa, czy podczas ostatniego skanowania obiekt podlegał skanowaniu w trybie binarnym. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Obiekt nie podlegał skanowaniu w trybie binarnym.</p> <p>x'01' - Obiekt podlegał skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_FAILURE, skanowanie obiektu w trybie binarnym nie powiodło się.</p>
QEZSCCSID1	INTEGER	<p>Określa, czy obiekt podlegał skanowaniu z wymienionym identyfikatorem CCSID podczas poprzedniego skanowania. Jeśli status skanowania obiektu wskazuje wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu z identyfikatorem CCSID. Jeśli status skanowania obiektu wskazuje wartość SCAN_FAILURE, skanowanie obiektu z identyfikatorem CCSID nie powiodło się. Wartość 0 oznacza, że pole to nie obowiązuje.</p>
QEZSCCSID2	INTEGER	<p>Określa, czy obiekt podlegał skanowaniu z wymienionym identyfikatorem CCSID podczas poprzedniego skanowania. Jeśli status skanowania obiektu wskazuje wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu z identyfikatorem CCSID. Jeśli status skanowania obiektu wskazuje wartość SCAN_FAILURE, w polu tym będzie wartość 0. Wartość 0 oznacza, że pole nie obowiązuje.</p>
QEZSCN	GRAPHIC (1)	<p>Określa, czy obiekt ma podlegać skanowaniu po zarejestrowaniu programów obsługi wyjścia w jednym ze zintegrowanych punktów wyjściowych systemu plików związanych ze skanowaniem.</p> <p>Poprawne wartości:</p> <p>x'00' (SCANNING_NO) - obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>x'01' - obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>x'02' (SCANNING_CHGONLY) - obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach obsługi wyjścia dotyczących skanowania tylko w przypadku modyfikacji od obiektu momentu ostatniego skanowania obiektu. Nie będzie podlegać skanowaniu w przypadku aktualizacji oprogramowania do skanowania. Atrybut ten obowiązuje tylko wtedy, gdy podano wartość *USEOCOATR wartości systemowej Sterowanie skanowaniem systemów plików (QSCANFCTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES.  <b>Uwaga:</b> Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p>
QEZSIG <sup>1</sup>	SMALLINT	<p>Określa, czy obiekt zawiera podpis cyfrowy systemu i5/OS. Poprawne wartości:</p> <p>0 - obiekt nie zawiera podpisu cyfrowego systemu i5/OS.</p> <p>1 - obiekt zawiera podpis cyfrowego systemu i5/OS.</p>

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

Nazwa pola	Typ pola	Opis pola
QEZSSIGDF	GRAPHIC (1)	<p>Sygnatury skanowania oznaczają poziom obsługi oprogramowania do skanowania.</p> <p>Gdy obiekt jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z powiązaną sygnaturą skanowania grupy niezależnych pul ASP. Gdy obiekt nie jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z wartością globalnej sygnatury skanowania. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Porównywane sygnatury nie różnią się.</p> <p>x'01' - Porównywane sygnatury różnią się.</p>
QEZSSTATUS	GRAPHIC (1)	<p>Status skanowania przypisany do obiektu. W polu będzie jedna z następujących wartości:</p> <p>x'00' (SCAN_REQUIRED) - Skanowanie jest wymagane dla obiektu, ponieważ nie podlegał on jeszcze skanowaniu przez programy obsługi wyjścia dotyczące skanowania lub dane obiektu bądź identyfikator CCSID uległ modyfikacji od ostatniego skanowania. Przykłady modyfikacji danych obiektu lub identyfikatora CCSID to: zapis do obiektu, bezpośrednio lub przez odwzorowanie pamięci, obcięcie obiektu, usunięcie obiektu i zmiana atrybutu identyfikatora CCSID obiektu.</p> <p>x'01' (SCAN_SUCCESS) - Obiekt podlegał skanowaniu przez program obsługi wyjścia dotyczący skanowania i dla ostatniego żądania skanowania operacja skanowania obiektu zakończyła się pomyślnie.</p> <p>x'02' (SCAN_FAILURE) - Obiekt podlegał skanowaniu przez program obsługi wyjścia dotyczący skanowania i dla ostatniego żądania skanowania operacja skanowania obiektu nie powiodła się i nie została zakończona. Po oznaczeniu obiektu jako błędny, nie będzie on podlegał kolejnemu skanowaniu do czasu zmiany sygnatury skanowania obiektu na inną niż sygnatura klucza globalnego skanowania lub sygnatura klucza skanowania grupowego niezależnej ASP. Dlatego kolejne żądania pracy z obiektem będą powodować błąd i wyświetlenie informacji o niepowodzeniu skanowania. Przykłady żądań, które spowodują błąd, to otwarcie obiektu, zmiana identyfikatora CCSID lub kopiowanie obiektu.</p> <p>x'05' (SCAN_PENDING_CVN) - Obiekt nie znajduje się w katalogu *TYPE2 i dlatego nie będzie poddany skanowaniu do czasu przeprowadzenia konwersji katalogu.</p> <p>x'06' (SCAN_NOT_REQUIRED) - Obiekt nie wymaga skanowania, ponieważ jest oznaczony jako nie do skanowania.</p>
QEZSTGFREE <sup>1</sup>	SMALLINT	<p>Informacje o tym, czy dane obiektu zostały odłączone ze zwolnieniem pamięci. Poprawne wartości:</p> <p>0 - Dane obiektu nie zostały odłączone.</p> <p>1 - Dane obiektu zostały odłączone.</p>
QEZSYSARC	SMALLINT	<p>Określa, czy obiekt zmienił się i powinien być składowany. Gdy czas zmiany obiektu jest aktualizowany, atrybut ten jest włączany, a po składowaniu obiektu atrybut jest wyłączany.</p> <p>0 - Obiekt nie zmienił się i nie musi być składowany.</p> <p>1 - Obiekt zmienił się i musi być składowany.</p>



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

Nazwa pola	Typ pola	Opis pola
QEZSYSSIG	SMALLINT	Określa, czy obiekt został podpisany przez źródło zaufane w systemie. Poprawne wartości:  0 - Żaden z podpisów nie pochodzi ze źródła zaufanego w systemie.  1 - Obiekt został podpisany przez źródło zaufane w systemie. Jeśli obiekt zawiera kilka podpisów, przynajmniej jeden podpis pochodzi ze źródła zaufanego w systemie.
QEZUDATE	TIMESTAMP	Liczba sekund, które upłynęły od epoki odpowiadającej dacie i godzinie ostatniego użycia obiektu. Gdy obiekt zostanie utworzony, pole to jest zerowe. Jeśli informacje o użyciu obiektu nie są obsługiwane dla typu systemu i5/OS lub systemu plików, do którego należy obiekt, pole to jest zerowe.
QEZUDCOUNT	INTEGER	Liczba dni, przez które obiekt był używany. Użycie obiektu ma różne znaczenie, w zależności od systemu plików i typów obiektów obsługiwanych w systemach plików. Użycie może oznaczać otwarcie lub zamknięcie zbioru lub dodanie dowiązań, zmianę nazwy, odtworzenie lub sprawdzenie obiektu. Licznik ten jest zwiększany każdego dnia, w którym obiekt jest używany i zerowany po wywołaniu funkcji API Qp0lSetAttr().
QEZUDFTYP2	SMALLINT	Domyślny format pliku dla plików strumieniowych (*STMF) utworzonych w systemie plików użytkownika. Poprawne wartości:  0 - Plik strumieniowy ma format *TYPE1.  1 - Plik strumieniowy ma format *TYPE2.
QEZUID	INTEGER	Każdy użytkownik w systemie musi mieć unikalny numer identyfikacyjny użytkownika (UID).
QEZURESET	INTEGER	Liczba sekund, które upłynęły od epoki odpowiadającej dacie ostatniego zerowania licznika dni używania obiektu. Data ta otrzymuje wartość bieżącej daty, gdy wywołana zostanie funkcja API Qp0lSetAttr() do wyzerowania licznika dni używania.
<b>Uwagi:</b>		
<ol style="list-style-type: none"> <li>1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRDIRINF.</li> <li>2. W polu tym przechowywana jest tylko nazwa obiektu. Pozostałą część nazwy ścieżki zawiera pole QEZDIRNAM1, jeśli długość nazwy katalogu jest mniejsza niż 1 kB (1024 bajty), lub pole QEZDIRNAM2, jeśli długości nazw przekraczają 1 kB (1024 bajty).</li> </ol>		

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

Tabela 9. QAEZDxxxD (składowanie atrybutów katalogów)

Nazwa pola	Typ pola	Opis pola
QEZDFID	INTEGER	Identyfikator zbioru katalogu.
QEZDIRFID	GRAPHIC (16)	Identyfikator zbioru katalogu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu.
QEZDIRFSID	BIGINT	Systemowy identyfikator zbioru katalogu.
QEZDIRGID	BIGINT	Identyfikator pokolenia.
QEZDIRIDX	INTEGER	Identyfikator nazwy ścieżki (dotyczy tylko katalogów).
QEZDIRLEN <sup>1</sup>	INTEGER	Długość nazwy ścieżki katalogu.
QEZDIRNAM1 <sup>1</sup>	VARGRAPHIC (1024)	Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki nie przekracza 1 kB (1024 bajtów).
QEZDIRNAM2 <sup>1</sup>	DBCLOB (16M)	Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki przekracza 1 kB (1024 bajty). Może pomieścić ścieżki do 16 MB długości.



Tabela 9. QAEZDxxxD (składowanie atrybutów katalogów) (kontynuacja)

Nazwa pola	Typ pola	Opis pola
QEZRCCSID	INTEGER	Identyfikator CCSID katalogu.
QEZDREGION	GRAPHIC (2)	Identyfikator regionu ścieżki katalogu.
QEZLANGID	GRAPHIC (3)	Identyfikator języka ścieżki katalogu.
<b>Uwaga:</b>		
<ul style="list-style-type: none"> <li>• Pole to jest dołączane do podzestawu pól używanych przez komendę PRTDIRINF.</li> </ul>		

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

Tabela 10. QUSRSYS/QAEZDBFILE (składowanie utworzonych zbiorów)

Nazwa pola	Typ pola	Opis pola
QEZDIRFILE <sup>1</sup>	VARGRAPHIC (20)	Nazwa zbioru wygenerowanego do zapisania indeksów katalogu.
QEZDIRSRC	VARGRAPHIC (5000)	Ścieżka podana w parametrze DIR (RTVDIRINF).
QEZENDTIME	TIMESTAMP	Data/godzina zakończenia komendy RTVDIRINF.
QEZLIB <sup>1</sup>	VARGRAPHIC (20)	Biblioteka, w której znajdują się oba wygenerowane zbiory.
QEZOBJFILE <sup>1</sup>	VARGRAPHIC (20)	Nazwa zbioru wygenerowanego do zapisania atrybutów obiektu.
QEZPLANGID	GRAPHIC (3)	ID języka ścieżki.
QEZPRCCSID	INTEGER	CCSID ścieżki.
QEZPREGION	GRAPHIC (2)	ID regionu ścieżki.
QEZSTRTIME	TIMESTAMP	Data/godzina wywołania komendy RTVDIRINF.
<b>Uwaga:</b>		
<ul style="list-style-type: none"> <li>• Pole to jest dołączane do podzestawu pól używanych przez komendę PRTDIRINF.</li> </ul>		

### Informacje pokrewne

Komenda Pobranie informacji katalogowych (Retrieve Directory Information - RTVDIRINF)

Funkcja API Qp01GetPathFromID()

Funkcja API Qp01SetAttr()

Komenda Zastosowanie kronikowanych zmian (Apply Journalled Changed - APYJRNCHG)

Komenda Usuwanie kronikowanych zmian (Remove Journalled Changes - RMVJRNCHG)

Komenda Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ)

Komenda Wydruk informacji katalogowych (Print Directory Information - PRTDIRINF)

### Dostęp do danych RTVDIRINF:

Dostęp do danych tabeli można uzyskać na kilka sposobów.

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

- Korzystanie z komendy Wydruk informacji katalogowych (Print Directory Information - PRTDIRINF)  
Komenda ta służy do drukowania informacji katalogowych o obiektach i w zintegrowanym systemie plików. Drukowane informacje przechowywane są w zbiorze bazy danych wskazanym przez użytkownika w komendzie RTVDIRINF.

- Użycie programu lub komendy IBM, która może uruchamiać zapytania do tabeli bazy danych DB2 w systemie iSeries.

Niektóre bardziej popularne narzędzia to komenda Uruchomienie interaktywnej sesji SQL (Start SQL Interactive Session - STRSQL) i narzędzie iSeries Navigator.

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

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

- Można także utworzyć własne programy i uzyskiwać dostęp do tabel bazy danych używając dowolnych poprawnych metod DB.

#### **Informacje pokrewne**

Komenda Wydruk informacji katalogowych (Print Directory Information - PRDIRINF)

Komenda Uruchomienie SQL (Start SQL - STRSQL)

Wbudowany SQL

Interfejsy CLI języka SQL

### **Dostęp do danych komendy RTVDIRINF:**

Poniżej znajdują się przykłady ilustrujące, dlaczego dane są ważne i jak można użyć danych utworzonych z każdej z trzech tabeli.

- Tabela 8 na stronie 75 dopuszcza tworzenie zapytań w celu utworzenia raportów statystyk w oparciu o dowolne pozycje tej tabeli. Komenda PRDIRINF nie obejmuje raportów opartych na wszystkich polach tabeli. Użyty zostanie podzestaw pól.
- Tabela 9 na stronie 84 zawiera dane, które dotyczą wszystkich katalogów w ścieżce podanej w parametrze DIR komendy RTVDIRINF. Dane te będą użyteczne do poznania konkretnych atrybutów nazwy ścieżki, jak CCSID, ID języka lub długość. Każdy katalog przechowywany w tej tabeli zawiera unikalną wartość lub indeks, który go identyfikuje. Tabela 8 na stronie 75 umożliwia wyszukanie tego samego pola, QEZDIRIDX, zawierającego informacje, które obiekty należą do którego katalogu. Aby sprawdzić, który obiekt należy do którego katalogu, można użyć zapytań z połączeniami. Na przykład, następująca instrukcja wybiera nazwy wszystkich obiektów istniejących w katalogu "/MYDIR":

```
SELECT QEZOBJNAM FROM nazwa_biblioteki/QAEZxxxx0,
nazwa_biblioteki/QAEZxxxxD WHERE QEZDIRNAM1 = "/MYDIR" AND
nazwa_biblioteki/QAEZxxxx0.QEZDIRIDX=nazwa_biblioteki/QAEZxxxxD.QEZDIRIDX
```

- Tabela 10 na stronie 85 używana jest głównie przez komendę PRDIRINF do pobrania konkretnych danych o uruchomieniach RTVDIRINF. Przykładami są: nazwy utworzonych tabel, biblioteka, w której znajduje się tabela, i czas rozpoczęcia oraz zakończenia przetwarzania. Tabeli tej można użyć do sprawdzenia, kiedy uruchomiono komendę RTVDIRINF lub które tabele należy wyszukać, aby wydać zapytanie.

## **Dostęp przy użyciu funkcji API**

Funkcji API można użyć, aby uzyskać dostęp do zintegrowanego systemu plików.

#### **Odsyłacze pokrewne**

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

## **Dostęp przy użyciu programu iSeries Navigator**

iSeries Navigator jest graficznym interfejsem użytkownika służącym do zarządzania i administrowania systemami z pulpitu systemu 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 tematy opisujące niektóre najczęściej wykonywane zadania systemu plików:

### **Praca z plikami i folderami**

- “Tworzenie folderu” na stronie 127
- “Usuwanie folderu” na stronie 127
- “Odblokowanie pliku” na stronie 127
- “Zablokowanie pliku” na stronie 127
- “Konfigurowanie uprawnień” na stronie 129
- “Konfigurowanie konwersji plików tekstowych” na stronie 129
- “Wysyłanie pliku lub folderu do innego systemu” na stronie 129
- “Zmiana opcji definicji pakietów” na stronie 130
- “Planowanie daty i godziny wysyłki pliku lub folderu” na stronie 130
- “Włączanie i wyłączanie skanowania obiektów” na stronie 132

### **Praca z współużytkowanymi zasobami plikowymi**

- “Tworzenie współużytkowanego zasobu plikowego” na stronie 131
- “Zmiana współużytkowanego zasobu plikowego” na stronie 131

### **Praca z systemami plików użytkownika**

- “Tworzenie nowego systemu plików użytkownika” na stronie 131
- “Podłączanie systemu plików utworzonego użytkownika” na stronie 131
- “Odłączanie systemu plików definiowanego przez użytkownika” na stronie 132

### **Kronikowanie obiektów**

- “Uruchomienie kronikowania” na stronie 100
- “Zakończenie kronikowania” na stronie 100

#### **Odsyłacze pokrewne**

“Dostęp przy użyciu komputera PC” na stronie 89

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

## **Dostęp przy użyciu serwera iSeries NetServer**

iSeriesSupport for Windows Network Neighborhood (iSeries NetServer) jest funkcją systemu i5/OS, która umożliwia klientom systemu Windows dostęp do ścieżek współużytkowanych katalogów systemu i5/OSi współużytkowanych kolejek wyjściowych. Serwer iSeries NetServer umożliwia komputerom PC z oprogramowaniem Windows bezkonfliktowy dostęp do danych i drukarek zarządzanych w systemie iSeries.

Klienci PC w sieci używają funkcji współużytkowania plików i drukarek wbudowanych we własne systemy operacyjne. Oznacza to, że aby skorzystać z iSeries NetServer, nie trzeba na komputerze PC instalować żadnego dodatkowego oprogramowania.

Klienci systemu Linux z zainstalowanym oprogramowaniem klienckim Samba również mogą uzyskać dostęp do danych i drukarek za pomocą iSeries NetServer. Współużytkowane katalogi serwera iSeries NetServer można podłączać w klientach systemu Linux jako systemy plików Samba (smbfs) podobnie, jak podłącza się systemy plików NFS wyeksportowane z systemu iSeries.

Współużytkowany zasób plikowy serwera iSeries NetServer jest ścieżką do katalogu, którą serwer iSeries NetServer udostępnia klientom PC w sieci iSeries. Plik współużytkowany może zawierać dowolny katalog ze zintegrowanego systemu plików systemu iSeries. Zanim możliwe będzie korzystanie z funkcji współużytkowania plików z użyciem serwera iSeries NetServer, należy utworzyć współużytkowany zasób plikowy serwera iSeries NetServer i jeśli to konieczne, zmienić współużytkowany zasób plikowy serwera iSeries NetServer używając narzędzia iSeries Navigator.

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

1. Kliknij prawym przyciskiem myszy pozycję **Start**, i wybierz opcję **Eksploruj**, aby otworzyć okno programu Windows Explorer w komputerze PC z systemem Windows.
2. Otwórz menu Narzędzia i wybierz **Mapuj dysk sieciowy**.
3. Wybierz dla współużytkowanego zasobu literę oznaczającą wolny napęd (na przykład napęd I:).
4. Podaj nazwę współużytkowanego zasobu plikowego serwera iSeries NetServer. Na przykład, można wpisać następującą składnię: \\QSYSTEM1\Sharename

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

5. Kliknij przycisk **OK**.

**Uwaga:** Jeśli korzysta się z iSeries NetServer, nazwa serwera może różnić się od nazwy używanej przez program iSeries Access Family. 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 Family może być AS400X, a ścieżka umożliwiająca pracę z plikami może mieć postać \\AS400X\QDLS\MÓJFOLDER.FLR\MÓJPLIK.DOC.

Za pomocą serwera 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 systemu iSeries zapewnia bazowy system plików "root" (/). Użytkownik może przenosić pliki z bazowego systemu plików "root" (/). Więcej informacji zawiera temat "Przenoszenie plików lub folderów do innego systemu plików" na stronie 128.

#### **Odsyłacze pokrewne**

"Dostęp przy użyciu komputera PC" na stronie 89

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

#### **Informacje pokrewne**

iSeries NetServer

Współużytkowane zasoby plikowe serwera iSeries NetServer

Dostęp do współużytkowanych zasobów plikowych serwera iSeries NetServer z użyciem klienta Windows na komputerze PC

## **Dostęp przy użyciu protokołu FTP**

Klient protokołu File Transfer Protocol (FTP) umożliwia przesłanie plików znajdujących się na serwerze iSeries, łącznie z plikami z systemu plików "root" (/), QOpenSys, QSYS.LIB, niezależnej puli ASP systemu QSYS.LIB, 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 usług biblioteki dokumentów (QDLS)
- system plików nośników optycznych (QOPT),
- sieciowy system plików (Network File System - NFS),
- System plików NetWare (QNetWare)
- system plików iSeries NetClient (QNTC).

Należy jednak pamiętać o następujących ograniczeniach:

- Zintegrowany system plików zawęża 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 zawęża 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.

#### **Odsyłacze pokrewne**

“Dostęp przy użyciu komputera PC”

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

#### **Informacje pokrewne**

FTP

Przesyłanie plików za pomocą FTP

## **Dostęp przy użyciu komputera PC**

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

Obiekty można kopiować między katalogami używając funkcji przeciągnij i upuść narzędzia Windows Explorer. 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.

Dowolne obiekty skopiowane między serwerem iSeries a komputerami PC za pomocą interfejsu Windows mogą być automatycznie poddane konwersji z EBCDIC and 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 Family może zostać tak skonfigurowany, aby wykonywał konwersję automatycznie, lub też aby wykonywał konwersję na plikach z konkretnymi rozszerzeniami.

W zależności od typu obiektu, użytkownik może używać do pracy z nim interfejsów i aplikacji komputera PC. Na przykład, plik strumieniowy zawierający tekst można poddać edycji z użyciem edytora komputera PC.

Po podłączeniu klienta PC do serwera iSeries, zintegrowany system plików udostępnia katalogi i obiekty serwera na kliencie PC. Komputery PC mogą pracować z plikami, w zintegrowanym systemie plików z użyciem klientów współużytkownika zasobów plikowych wbudowanych w systemy operacyjne Windows, klientów FTP lub iSeries Navigator (część iSeries Access Family). Komputer PC używa klientów współużytkownika zasobów plikowych systemu Windows w celu uzyskania dostępu do systemu iSeries NetServer, działającego na serwerze iSeries.

#### **Pojęcia pokrewne**

“Dostęp przy użyciu programu iSeries Navigator” na stronie 86

iSeries Navigator jest graficznym interfejsem użytkownika służącym do zarządzania i administrowania systemami z pulpitu systemu Windows. iSeries Navigator ułatwia eksploatację systemu i zarządzanie nim oraz przyczynia się do zwiększenia produktywności.

#### **Zadania pokrewne**

“Dostęp przy użyciu serwera iSeries NetServer” na stronie 87

iSeriesSupport for Windows Network Neighborhood (iSeries NetServer) jest funkcją systemu i5/OS, która umożliwia klientom systemu Windows dostęp do ścieżek współużytkowanych katalogów systemu i5/OSi współużytkowanych kolejek wyjściowych. Serwer iSeries NetServer umożliwia komputerom PC z oprogramowaniem Windows bezkonfliktowy dostęp do danych i drukarek zarządzanych w systemie iSeries.

#### **Odsyłacze pokrewne**

“Dostęp przy użyciu protokołu FTP” na stronie 88

Klient protokołu File Transfer Protocol (FTP) umożliwia przesłanie plików znajdujących się na serwerze iSeries, łącznie z plikami z systemu plików "root" (/), QOpenSys, QSYS.LIB, niezależnej puli ASP systemu QSYS.LIB, QOPT oraz QFileSvr.400.

---

## **Przekształcanie katalogów z formatu \*TYPE1 na \*TYPE2**

Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1. Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1.

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Nie powinna ona poważnie wpłynąć na działanie systemu.

#### **Pojęcia pokrewne**

“Katalogi \*TYPE2” na stronie 10

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

## **Przegląd konwersji formatu \*TYPE1 do \*TYPE2**

Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

Format katalogu \*TYPE2 stanowi ulepszenie oryginalnego formatu katalogu \*TYPE1. Katalogi \*TYPE2 mają inną strukturę wewnętrzną i inną implementację niż katalogi \*TYPE1. Oprócz lepszej wydajności i niezawodności, niektóre nowe funkcje, takie jak obsługa skanowania zintegrowanego systemu plików, dostępne są tylko dla obiektów w katalogach \*TYPE2. Więcej informacji znajduje się w temacie “Obsługa skanowania” na stronie 21.

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Konwersja ta nie powinna znacząco wpłynąć na działanie systemu, ponieważ będzie działać w zadaniu w tle o niskim priorytecie.

Jeśli konwersja nie została jeszcze zakończona, a system wykona normalny lub nieprawidłowy IPL, funkcja konwersji podejmie działanie po zakończeniu IPL. Konwersja będzie ponownie uruchamiana do czasu, gdy wszystkie odpowiednie systemy plików zostaną w pełni przekształcone.

Systemy plików, które można poddać automatycznej konwersji, to "root" (/), QOpenSys i system plików użytkownika dla pul ASP od 1 do 32.



**Uwaga:** Można uniknąć automatycznej konwersji do formatu katalogów \*TYPE2, jeśli konwersja ta zostanie przeprowadzona ręcznie przed zainstalowaniem systemu operacyjnego w wersji V5R3M0 lub nowszej.

#### **Pojęcia pokrewne**

“Katalogi \*TYPE2” na stronie 10

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

#### **Odsyłacze pokrewne**

“Określanie statusu konwersji”

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. To przetwarzanie konwersji nastąpi w dodatkowym wątku zadania systemowego QFILESYS1.

“Wskazówki: niezależne ASP” na stronie 94

Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2 lub nowszej.

## **Uwagi dotyczące konwersji**

Podczas przeprowadzania konwersji należy wziąć pod uwagę następujące zagadnienia.

### **Określanie statusu konwersji**

Bezpośrednio po zainstalowaniu systemu w wersji i5/OSV5R3M0 system rozpocznie automatyczną konwersję katalogów do formatu \*TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. To przetwarzanie konwersji nastąpi w dodatkowym wątku zadania systemowego QFILESYS1.

Aby określić status konwersji, można użyć komendy Konwersja katalogu (Convert Directory - CVTDIR) w następujący sposób:

```
CVTDIR OPTION(*CHECK)
```

Ta składnia komendy CVTDIR powoduje wyświetlenie listy aktualnego formatu katalogów systemów plików "root" (/), QOpenSys i UDFS oraz informacje o wykonywanej konwersji systemu plików. Dodatkowo, zawiera bieżący priorytet funkcji konwersji, aktualnie konwertowany system plików w systemie, liczbę dowiązań dotychczas przetworzonych dla tego systemu plików i procent katalogów, które zostały już przetworzone dla tego systemu plików. System uruchamia funkcję konwersji z bardzo niskim priorytetem (99), aby funkcja nie wpływała znacząco na działanie systemu. Można jednak zmienić ten priorytet używając wartości \*CHGPTY parametru OPTION w komendzie CVTDIR. Dodatkowe informacje o specyfikacji tego parametru zawiera temat CVTDIR.

Ponieważ zadanie QFILESYS1 przetwarza konwersję, można wyświetlić protokół zadania QFILESYS1 i przejrzeć ewentualne komunikaty o błędach w procesie konwersji. Dodatkowo, do powyższego protokołu wysyłane będą różne komunikaty informujące o postępie operacji konwersji. Komunikaty te zawierają następujące informacje: przetwarzany system plików, liczbę dowiązań dotychczas przetworzonych dla tego systemu plików, procent katalogów, które zostały już przetworzone dla tego systemu plików itp. Wszystkie komunikaty o błędach i wiele komunikatów o postępie operacji wysyłane jest do kolejki komunikatów QSYSOPR. Dlatego, na przyszłość warto sprawdzić, czy protokoły QHST lub protokoły zadania QFILESYS1 są zapisywane, ponieważ zawierają one te komunikaty. Gdy systemy plików zostaną w pełni przekształcone, a zintegrowany system plików działa poprawnie, można usunąć te informacje.

#### **Pojęcia pokrewne**

“Przegląd konwersji formatu \*TYPE1 do \*TYPE2” na stronie 90

Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

#### **Informacje pokrewne**

Komenda Konwersja katalogu (Convert Directory - CVTDIR)



## Tworzenie profili użytkowników

Funkcja konwersji tworzy profil użytkownika, który używany jest w trakcie działania funkcji. Ten profil ma nazwę QP0FCWA. Jest wykorzystywany przez funkcję konwersji jako właściciel przetwarzanych katalogów w systemie plików, jeśli nie można im przypisać oryginalnego właściciela.

Profil ten jest usuwany po zakończeniu konwersji, o ile jest to możliwe. Komunikat CPIA08B jest wysyłany do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR, jeśli temu użytkownikowi zostaną przydzielone prawa własności katalogu.

### Informacje pokrewne

“Zmiana właściciela katalogu” na stronie 93

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika.

## Zmiana nazw obiektów

Katalogi \*TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16.

Reguła nazewnictwa dla katalogów \*TYPE2 różni się od reguły dla katalogów \*TYPE1, które mają nazwy UCS2 Poziom 1. Dlatego podczas konwersji katalogów mogą wystąpić nieprawidłowe lub zduplikowane nazwy. Gdy znaleziona zostanie niepoprawna lub powielona nazwa, zmieniana jest ona na unikalną, poprawną nazwę UTF-16, a do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08A podający oryginalną i nową nazwę. Znaki złożone lub nieprawidłowe pary znaków odpowiedników znajdujące się w nazwie mogą spowodować zmianę nazwy obiektu.

Więcej informacji o kodzie UTF-16 można znaleźć na stronie głównej Unicode (<http://www.unicode.org> ).

### Znaki złożone:

Niektóre znaki mogą składać się z więcej niż jednego znaku Unicode.

Znaki posiadające akcent (na przykład é lub à) lub umlaut (na przykład ä lub ö) muszą być zmienione lub *znormalizowane* do wspólnego formatu zanim zostaną umieszczone w katalogu, tak aby wszystkie obiekty miały unikalne nazwy. Normalizacja znaków złożonych jest procesem, w którym znak jest przekształcany do nowego i przewidywalnego formatu. Format wybrany dla katalogu \*TYPE2 to *złożona postać kanoniczna*. Jeśli w katalogu \*TYPE1 znajdują się dwa obiekty zawierające te same znaki złożone, są one normalizowane do tej samej nazwy. Powoduje to kolizję, nawet jeśli jeden obiekt zawiera połączone złożone znaki, a drugi zawiera rozłączone złożone znaki. Dlatego jeden z nich musi mieć zmienioną nazwę zanim zostanie dowiązany w katalogu \*TYPE2.

### Znaki odpowiedników:

Niektóre znaki nie mają prawidłowych odpowiedników w systemie Unicode.

Znaki te mają specjalne wartości, które składają się z dwóch znaków Unicode, w dwóch określonych przedziałach, z których pierwszy znak znajduje się w pierwszym zakresie (na przykład 0xD800-0xD8FF), a drugi w drugim zakresie (na przykład 0xDC00-0xDCFF). Nazywa się to parą odpowiedników.

Jeśli jeden ze znaków Unicode zostanie usunięty lub są one podane w złej kolejności (tylko jeden znak), nazwa jest nieprawidłowa. Nazwy tego typu są dozwolone w katalogach \*TYPE1, ale nie są dozwolone w katalogach \*TYPE2. Aby konwersja mogła być kontynuowana, jeśli nazwa zawiera jeden z nieprawidłowych znaków, jest zmieniana zanim obiekt zostanie dowiązany w katalogu \*TYPE2.

## Uwagi na temat 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 właścicielem pewnej liczby katalogów istniejących w obu formatach jednocześnie jest ten sam profil użytkownika.

Jeśli w trakcie konwersji zostanie przekroczony maksymalny limit pamięci dla profilu użytkownika, limit ten jest zwiększany. Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08C.

### **Zmiana właściciela katalogu:**

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika.

Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08B, a konwersja jest kontynuowana.

Jeśli profil użytkownika, który jest właścicielem katalogu \*TYPE1, nie może być właścicielem tworzonego katalogu \*TYPE2, właścicielem nowego katalogu \*TYPE2 staje się alternatywny profil użytkownika. Do protokołu zadania QFILESYS1 i kolejki komunikatów QSYSOPR wysyłany jest komunikat CPIA08B, a konwersja jest kontynuowana.

#### **Odsyłacze pokrewne**

“Tworzenie profili użytkowników” na stronie 92

Funkcja konwersji tworzy profil użytkownika, który używany jest w trakcie działania funkcji. Ten profil ma nazwę QPOFCWA. Jest wykorzystywany przez funkcję konwersji jako właściciel przetwarzanych katalogów w systemie plików, jeśli nie można im przypisać oryginalnego właściciela.

## **Wymagania pamięci dyskowej**

Wymagania pamięci dyskowej należy rozważyć, gdy w systemie rozpocznie się konwersja katalogów w systemie plików do formatu \*TYPE2.

Poniżej znajduje się kilka uwag odnoszących się do wymaganej przestrzeni dyskowej:

- końcowa wielkość katalogów po konwersji do formatu \*TYPE2,
- dodatkowa przestrzeń dyskowa wymagana, gdy funkcja konwersji jest uruchomiona.

W wielu przypadkach, końcowa wielkość katalogu \*TYPE2 jest mniejsza niż katalogu \*TYPE1. Zazwyczaj katalogi \*TYPE2 zawierające mniej niż 350 obiektów wymagają mniej pamięci dyskowej niż katalogi \*TYPE1 zawierające tę samą liczbę obiektów. Katalogi \*TYPE2 zawierające więcej niż 350 obiektów są o 10 procent większe (średnio) niż katalogi \*TYPE1.

Gdy funkcja konwersji jest uruchomiona, wymagana jest dodatkowa przestrzeń dyskowa. Funkcja konwersji wymaga, aby obie wersje katalogów, w formacie \*TYPE1 i \*TYPE2 istniały jednocześnie.

**Uwaga:** Zanim rozpocznie się instalacja wersji i5/OS V5R3MO lub nowszej wersji systemu operacyjnego, należy rozważyć uruchomienie komendy systemu OS/400 V5R2 (CVTDIR) z opcją \*ESTIMATE, która dostarczy dokładnych danych szacunkowych ilości wolnej pamięci dyskowej wymaganej do przeprowadzenia operacji konwersji.

#### **Informacje pokrewne**

Komenda Konwersja katalogu (Convert Directory - CVTDIR)

## **Wskazówki: dowiązanie symboliczne**

Dowiązania symboliczne są obiektami w zintegrowanym systemie plików zawierającymi ścieżkę do innego obiektu.

W trakcie konwersji istnieje kilka sytuacji, w których nazwa obiektu może zostać zmieniona. Jeśli podczas konwersji zostanie zmieniona nazwa jednego z elementów ścieżki w dowiązaniu symbolicznym, to zawartość dowiązania symbolicznego nie będzie już wskazywać na obiekt.

#### **Pojęcia pokrewne**

“Dowiązanie” na stronie 12

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

#### **Odsyłacze pokrewne**

“Zmiana nazw obiektów” na stronie 92

Katalogi \*TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16.

#### **Informacje pokrewne**

Utworzenie dowiązania symbolicznego (Make symbolic link - symlink())

### **Wskazówki: niezależne ASP**

Jeśli systemy plików UDFS w niezależnej puli ASP nie zostały jeszcze poddane konwersji do formatu \*TYPE2, zostaną jej poddane, gdy tylko niezależna ASP zostanie udostępniona w systemie, w którym zainstalowano system operacyjny OS/400 w wersji V5R2 lub nowszej.

Do celów planowania utworzono funkcję, która w systemie OS/400 w wersji V5R1 udostępnia informacje o przewidywanym czasie trwania konwersji. Zanim niezależne ASP zostaną udostępnione w serwerze w wersji V5R2 lub nowszej, należy uruchomić następującą funkcję API w systemie w wersji V5R1 z udostępnioną i aktywną niezależną ASP (o nazwie NAZWA\_ASP):

```
CALL QP0FCVT2 (*ESTIMATE ASP_NAME *TYPE2)
```

#### **Pojęcia pokrewne**

“Przegląd konwersji formatu \*TYPE1 do \*TYPE2” na stronie 90

Systemy plików "root" (/), QOpenSys i systemy plików użytkownika UDFS w zintegrowanym systemie plików obsługują format katalogów \*TYPE2 od wersji OS/400 V5R1.

### **Wskazówki: składowanie i odtwarzanie**

Katalogi \*TYPE1 mogą być zapisane i odtworzone w systemie plików, który został przekonwertowany do \*TYPE2.

Podobnie, katalogi \*TYPE2 mogą być zapisane i odtworzone w systemie plików o formacie \*TYPE1, pod warunkiem, że nie zostało przekroczone żadne ograniczenie formatu \*TYPE1, gdy katalog istniał jako katalog \*TYPE2.

### **| Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików**

| Gdy system przetwarza systemy plików "root" (/), QOpenSys i UDFS w puli ASP użytkownika do obsługi formatu katalogów \*TYPE2, nie można uruchomić komend Odzyskiwanie pamięci (RCLSTG) oraz Odzyskiwanie dowiązań obiektu (RCLLNK) w żadnym katalogu zintegrowanego systemu plików, również w katalogach niezależnej ASP.

| Można jednak użyć parametru OMIT(\*DIR) w komendzie RCLSTG, aby pominąć katalogi zintegrowanego systemu plików i umożliwić odzyskanie obiektów niezwiązanych ze zintegrowanym systemem plików.

#### **| Pojęcia pokrewne**

| “Odzyskiwanie systemów plików "root" (/), QOpenSys i UDFS” na stronie 101

| Odzyskanie systemów plików "root" (/), QOpenSys i UDFS jest możliwe przy użyciu komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

#### **| Informacje pokrewne**

| Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

| Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)

## **Skanywanie zintegrowanego systemu plików**

Obiekty w systemie plików "root" (/), QOpenSys i UDFS ASP użytkownika nie będą skanowane z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików do czasu zakończenia konwersji systemów do formatu katalogów \*TYPE2.

Atrybuty związane ze skanowaniem można przypisać do obiektów w katalogach \*TYPE1 i \*TYPE2 w celu określenia, czy obiekty mają być skanowane, czy nie, nawet jeśli system plików nie został w pełni przetworzony.

Gdy system przetwarza obiekty z formatu \*TYPE1 do \*TYPE2, uwzględnia wartość systemową sterowania skanowaniem ('Skanowanie przy następnym dostępie po odtworzeniu obiektu') tak, jakby obiekty były odtwarzane. Na przykład, jeśli wartość Skanowanie przy następnym dostępie po odtworzeniu obiektu została podana w trakcie konwersji, wówczas obiekt z katalogu \*TYPE1 mający atrybut obiekt nie będzie skanowany zostanie poddany skanowaniu przynajmniej raz po zakończeniu pełnej konwersji.

### **Pojęcia pokrewne**

"Obsługa skanowania" na stronie 21

System iSeries, umożliwia skanowanie obiektów zintegrowanego systemu plików.

"Pokrewne wartości systemowe" na stronie 22

Z obsługą skanowania związane są dwie wartości systemowe. Służą one do tworzenia środowiska skanowania w serwerze.

---

## **Kronikowanie obiektów**

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

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

### **Informacje pokrewne**

Zarządzanie kronikami

## **Przegląd kronikowania**

Poniższe tematy opisują obsługę kronikowania obiektów zintegrowanego systemu plików.

### **Informacje pokrewne**

Zarządzanie kronikami

## **Zarządzanie kronikami**

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

Zarządzanie kronikami może również służyć do:

- zapisów kontrolnych dla działań dotyczących obiektów w systemie;
- rejestrowania innych niż kronikowane działań dotyczących obiektów;
- szybszego odtwarzania z nośników, na których przeprowadza się składowanie podczas użycia;
- asysty w powielaniu zmian w obiektach w innych systemach w celu zwiększenia dostępności lub wyrównania obciążenia;
- asysty przy testowaniu programów użytkowych.

Za pomocą kroniki można zdefiniować, które obiekty zostaną zabezpieczone poprzez kronikowanie. W zintegrowanym systemie plików można kronikować pliki strumieniowe, katalogi i dowiązania symboliczne. Obsługiwane są tylko obiekty następujących systemów plików: bazowego (root, "/"), QOpenSys i UDFS.

### **Pojęcia pokrewne**

“Jakie obiekty należy uwzględnić w kronikowaniu”

Przed podjęciem decyzji o kronikowaniu obiektu zintegrowanego systemu plików należy rozważyć pewne kwestie.

## Jakie obiekty należy uwzględnić w kronikowaniu

Przed podjęciem decyzji o kronikowaniu obiektu zintegrowanego systemu plików należy rozważyć pewne kwestie.

Aby określić, które obiekty powinny być kronikowane, należy wziąć pod uwagę następujące czynniki:

- Jak bardzo obiekt się zmienia? Dobrymi kandydatami do kronikowania są obiekty, które ulegają znacznym zmianom między 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 opóźnienia 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.

### Pojęcia pokrewne

“Zarządzanie kronikami” na stronie 95

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

## Kronikowane obiekty zintegrowanego systemu plików

Obsługa kronikowania w systemie i5/OS umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików.

Obsługiwanymi typami obiektów są pliki strumieniowe, katalogi i dowiązania symboliczne. Jedynymi systemami plików obsługującymi kronikowanie obiektów tych typów są bazy 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. Operacje Uruchomienie kronikowania i Zakończenie kronikowania można wykonać w aplikacji iSeries Navigator, podobnie jak można wyświetlić informacje o statusie kronikowania.

- | **Uwaga:** Pliki strumieniowe mapowania pamięci, pliki woluminów wirtualnych i pliki strumieniowe używane jako przestrzeń pamięci programu Integrated xSeries Server for iSeries (IXS) nie mogą być kronikowane. Katalogi, które mogą zawierać blokowe obiekty plików specjalnych, nie mogą być kronikowane. Przykładami są: /dev/QASP01, /dev/QASP22 i /dev/IASPNAM.

Poniższa lista zawiera podsumowanie obsługi kronikowania w zintegrowanym systemie plików:

- Do wykonywania operacji kronikowania na obsługiwanych typach obiektów można używać komend ogólnych i funkcji API. Interfejsy te akceptują identyfikację obiektów poprzez nazwę ścieżki, identyfikator zbioru lub obie te metody jednocześnie.
- Niektóre komendy operacji kronikowania, takie jak Uruchomienie kronikowania, Zakończenie kronikowania, Zmiana kronikowania i Zastosowanie kronikowanych zmian można wykonać dla całych poddrzew obiektów zintegrowanego systemu plików. Opcjonalnie możliwe jest zastosowanie list włączających i wyłączających z wzorcami nazw obiektów wykorzystującymi znaki zastępcze. Można na przykład użyć komendy Uruchomienie kronikowania (Start 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:** Zagadnienia związane z atrybutem dziedziczenia kronikowania:

- Jeśli nazwa obiektu zostanie zmieniona w tym samym katalogu, w którym obecnie obiekt się znajduje, kronikowanie nie zostanie uruchomione dla obiektu, nawet jeśli dla katalogu włączono atrybut dziedziczenia stanu kronikowania.
  - Gdy katalog zostanie przeniesiony do katalogu z włączonym atrybutem dziedziczenia kronikowania, kronikowanie zostanie uruchomione tylko dla przeniesionego katalogu, jeśli jest to możliwe. Obiekty w przeniesionym katalogu nie będą objęte kronikowaniem.
  - Jeśli obiekt zostanie odtworzony w katalogu z włączonym atrybutem dziedziczenia kronikowania, kronikowanie nie zostanie uruchomione dla tego obiektu, jeśli obiekt był już kronikowany.
  - Użycie komendy Zastosowanie kronikowanych zmian (Apply Journalled Changes - APYJRNCHG) powoduje, że bieżąca wartość atrybutu dziedziczenia kronikowania nie zostanie użyta dla żadnego katalogu. Zamiast niej, dla obiektów utworzonych jako część działania komendy kronikowanie zostanie uruchomione w zależności od przebiegu działania uruchomionej komendy.
- Nazwy obiektów i pełne nazwy ścieżek są zapisane w kilku pozycjach kronik obiektów zintegrowanego systemu plików. Dla nazwy obiektów i nazw ścieżek włączona jest obsługa języków narodowych (NLS).
  - W przypadku nieprawidłowego zakończenia pracy systemu kronikowane obiekty zintegrowanego systemu plików objęte zostaną odtwarzaniem przy IPL.
  - Maksymalna wielkość pozycji kroniki obsługiwana przez różne interfejsy zapisu zawiera się w przedziale 2 GB - 1; jeśli podano wartość RCVSIZOPT (\*MAXOPT2 lub \*MAXOPT3), wartość ta wynosi 4 000 000 000 bajtów. W przeciwnym razie maksymalna wielkość pozycji kroniki wynosi 15 761 440 bajtów. Jeśli kronikowany jest plik strumieniowy, a programy piszące przekraczają wielkość 15 761 bajtów, należy użyć obsługi \*MAXOPT2 lub \*MAXOPT3, aby uniknąć wystąpienia błędów.

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

#### **Pojęcia pokrewne**

“Plik strumieniowy” na stronie 17

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

“Katalog” na stronie 4

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

“Dowiązanie symboliczne” na stronie 14

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

#### **Zadania pokrewne**

“Uruchomienie kronikowania” na stronie 100

Czynności, które należy wykonać, aby uruchomić kronikowanie obiektu przy użyciu iSeries Navigator.

“Zakończenie kronikowania” na stronie 100

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

“Zmiana kronikowania” na stronie 100

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ).

#### **Informacje pokrewne**

Zarządzanie kronikami

Narzędzie wyszukiwania informacji o pozycjach kroniki

## Kronikowane operacje

Wymienione niżej operacje są kronikowane tylko wtedy, kiedy dotyczą obiektów lub dowiązań obsługiwanych przy kronikowaniu.

- utworzenie obiektu,
- dodanie dowiązania do istniejącego obiektu,
- usunięcie dowiązania,
- zmiana nazwy dowiązania,
- zmiana identyfikatora pliku,
- przeniesienie dowiązania do katalogu lub z katalogu.

W przypadku plików strumieniowych kronikowane są następujące działania:

- zapis danych lub usunięcie zawartości,
- obcięcie/rozszerzenie pliku,
- wymuszenie pliku danych,
- składowanie z opróżnieniem pamięci.

Dla wszystkich kronikowanych obiektów kronikowane są następujące operacje:

- Zmiany atrybutów (w tym zmiany ochrony, np. zmiany uprawnień i praw własności)
- Otwarcie
- Zamknięcie
- Uruchomienie kronikowania
- Komenda Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ)
- Zakończenie kronikowania
- Komenda Rozpoczęcie zastosowania kronikowanych zmian (Start the Apply Journaled Changes - APYJRNCHG)
- Komenda Zakończenie zastosowania kronikowanych zmian (End the Apply Journaled Changes - APYJRNCHG)
- Składowanie
- Odtwarzanie

### Informacje pokrewne

Zarządzanie kronikami

Narzędzie wyszukiwania informacji o pozycjach kroniki

## Specjalne zagadnienia dotyczące pozycji kronik

Wiele kronikowanych operacji w zintegrowanym systemie plików korzysta z kontroli transakcji w celu uformowania pojedynczej transakcji z wielu funkcji wykonywanych podczas działania.

Tych kronikowanych operacji nie można uznać za kompletne, dopóki w cyklu kontroli transakcji nie pojawi się pozycja kroniki Zatwierdzenie (Commit) (Kod kroniki C, Typ CM). Operacje kronikowane zawierające pozycję kroniki Wycofanie (Rollback) (Kod kroniki C, Typ RB) w cyklu kontroli transakcji to operacje, które się nie powiodły, a pozycje kronik dla tych operacji nie powinny być powtarzane ani replikowane.

Kronikowane pozycje zintegrowanego systemu plików (Kod kroniki B) używające w ten sposób kontroli transakcji obejmują:

- AA - Zmianę wartości kontroli (Change Audit Value),
- B0 - Rozpoczęcie tworzenia (Begin Create),
- B1 - Utworzenie podsumowania (Create Summary),
- B2 - Dodanie dowiązania (Add Link),



- B3 — Zmianę nazwy/Przeniesienie (Rename/Move)
- B4 - Usunięcie dowiązania (katalog nadrzędny) (Unlink - Parent Directory),
- B5 Usunięcie dowiązania (dowiązanie) (Unlink - link),
- B7 — Informacje o uprawnieniach utworzonego obiektu (Created Object Authority Information),
- FA - Zmianę atrybutu (Attribute Change),
- JT - Uruchomienie kronikowania (Start Journal) (tylko gdy kronikowanie się rozpoczyna, ponieważ operacja została wykonana w katalogu z włączonym atrybutem kronikowania),
- OA - Zmianę uprawnień (Authority Change),
- OG - Zmianę grupy głównej obiektu (Object Primary Group Change),
- OO - Zmianę właściciela obiektu (Object Owner Change).

Kilka pozycji kronik dla zintegrowanego systemu plików ma specyficzne pola wskazujące, czy dana pozycja jest pozycją podsumowania. Operacje wysyłające pozycje kronik zawierające podsumowania, przesyłają dwie pozycje do kroniki. Pierwsza z nich zawiera podzbiór danych właściwych dla danej pozycji. Druga pozycja zawiera pełny zbiór danych właściwych dla pozycji i jest oznaczona jako pozycja podsumowania. Programy, które powielają obiekt lub ponownie wykonują operacje, będą głównie zainteresowane pozycjami podsumowań.

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

Niektóre operacje kronikowane muszą wysyłać pozycje kronik, które są przeciwne do tych operacji. Na przykład cykl kontroli transakcji zawierający pozycje kroniki B4 (Usunięcie dowiązania - Unlink) może również zawierać pozycję B2 (Dodanie dowiązania - Add Link). Jednak taki przypadek wystąpi tylko podczas operacji, w wyniku których pojawi się pozycja kroniki Wycofanie (Rollback) (C - RB) .

Może się tak zdarzyć z dwóch powodów:

1. Wystąpiły problemy podczas operacji i pozycja ta była potrzebna wewnętrznie w procedurze czyszczącej błąd ścieżki,
2. Operacja została przerwana przez wyłączenie systemu i podczas procedury IPL wykonano odtwarzanie konieczne do wysłania pozycji w celu wycofania przerwanej operacji.

#### **Informacje pokrewne**

Narzędzie wyszukiwania informacji o pozycjach kroniki

## **Zagadnienia dotyczące używania wielu napędów dysków i kronikowania**

Jeśli istnieje kilka dowiązań stałych do kronikowanego obiektu zintegrowanego systemu plików, należy wszystkie te dowiązania zeszkładować i odtworzyć razem, aby zachowana została definicja dowiązania razem z odpowiednimi informacjami kroniki.

Podanie - w niektórych komendach związanych z kroniką - nazw, które są wieloma dowiązaniem stałymi spowoduje, że obiekt będzie przetwarzany jednorazowo dla wszystkich nazw. Pozostałe dowiązania stałe zostaną praktycznie zignorowane.

Ponieważ wiele dowiązań stałych wskazuje ten sam obiekt, a pozycja kroniki zawiera tylko identyfikator pliku (File ID), który jest ten sam dla obiektu, wszystkie interfejsy kroniki wyświetlą nazwę ścieżki. Na przykład komenda Wyświetlenie kroniki (Display Journal - DSPJRN) spowoduje wyświetlenie nazwy tylko jednego dowiązania dla obiektu. Nie powinno to jednak powodować błędów, ponieważ obiekt można obsługiwać z użyciem dowolnej nazwy i uzyskać ten sam rezultat.

#### **Pojęcia pokrewne**

“Dowiązanie stałe” na stronie 13

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

## Uruchomienie kronikowania

Czynności, które należy wykonać, aby uruchomić kronikowanie obiektu przy użyciu iSeries Navigator.

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**.

Aby uruchomić kronikowanie obiektu przy użyciu interfejsu znakowego, można użyć komendy Uruchomienie kroniki (Start Journal - STRJRN) lub funkcji API QjoStartJournal.

### Pojęcia pokrewne

“Kronikowane obiekty zintegrowanego systemu plików” na stronie 96

Obsługa kronikowania w systemie i5/OS umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików.

### Zadania pokrewne

“Zmiana kronikowania”

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ).

“Zakończenie kronikowania”

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

### Informacje pokrewne

Komenda Uruchomienie kroniki (Start Journal - STRJRN)

Funkcja API QjoStartJournal

Zarządzanie kronikami

## Zmiana kronikowania

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ).

### Pojęcia pokrewne

“Kronikowane obiekty zintegrowanego systemu plików” na stronie 96

Obsługa kronikowania w systemie i5/OS umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików.

### Zadania pokrewne

“Uruchomienie kronikowania”

Czynności, które należy wykonać, aby uruchomić kronikowanie obiektu przy użyciu iSeries Navigator.

“Zakończenie kronikowania”

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

### Informacje pokrewne

Komenda Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ)

## Zakończenie kronikowania

Gdy kronikowanie obiektu zostanie uruchomione, a użytkownik chce je przerwać, może dokonać tego, wykonując poniższe czynności.

Aby zatrzymać kronikowanie obiektu przy użyciu iSeries Navigator, wykonaj następujące czynności:

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**.

Aby zatrzymać kronikowanie obiektu przy użyciu interfejsu znakowego, można użyć komendy Zatrzymanie kroniki (End Journal - ENDJRN) lub funkcji API QjoEndJournal.

#### **Pojęcia pokrewne**

“Kronikowane obiekty zintegrowanego systemu plików” na stronie 96

Obsługa kronikowania w systemie i5/OS umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików.

#### **Zadania pokrewne**

“Zmiana kronikowania” na stronie 100

Gdy dla obiektu zostanie uruchomione kronikowanie, atrybuty kroniki dla danego obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania zmienić można za pomocą komendy Zmiana kronikowanego obiektu (Change Journaled Object - CHGJRNOBJ).

“Uruchomienie kronikowania” na stronie 100

Czynności, które należy wykonać, aby uruchomić kronikowanie obiektu przy użyciu iSeries Navigator.

#### **Informacje pokrewne**

Komenda Zakończenie kronikowania (End Journal - ENDJRN)

Funkcja API QjoEndJournal

Zarządzanie kronikami

## **Odzyskiwanie systemów plików "root" (/), QOpenSys i UDFS**

Odzyskanie systemów plików "root" (/), QOpenSys i UDFS jest możliwe przy użyciu komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

Komendy RCLLNK i RCLSTG umożliwiają wykonanie następujących zadań:

- Usunięcie problemów z profilami użytkownika
- Usunięcie problemów z systemami plików użytkownika
- Usunięcie problemów z obiektami wewnętrznymi
- Usunięcie niepoprawnych dowiązań obiektów
- Obsługę uszkodzonych obiektów
- Utworzenie brakujących obiektów systemowych
- Usunięcie wewnętrznych problemów systemów plików (tylko RCLSTG)
- Wyszukiwanie utraconych obiektów (tylko RCLSTG)

#### **Odsyłacze pokrewne**

“Wskazówki: odzyskiwanie obiektów zintegrowanego systemu plików” na stronie 94

Gdy system przetwarza systemy plików "root" (/), QOpenSys i UDFS w puli ASP użytkownika do obsługi formatu katalogów \*TYPE2, nie można uruchomić komend Odzyskiwanie pamięci (RCLSTG) oraz Odzyskiwanie dowiązań obiektu (RCLLNK) w żadnym katalogu zintegrowanego systemu plików, również w katalogach niezależnej ASP.

## **Porównanie komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)**

Komendy Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) umożliwiają usunięcie problemów z systemami plików "root" (/), QOpenSys i UDFS.

| Komenda RCLLNK identyfikuje i, jeśli jest to możliwe, usuwa problemy z używanymi podłączonymi systemami  
 | plików. Komenda RCLSTG nie ma takiej funkcjonalności. Komenda RCLSTG może jednak usunąć problemy, których  
 | komenda RCLLNK nie jest w stanie zidentyfikować i usunąć. Poniższa tabela zawiera bardziej szczegółowe  
 | porównanie obu komend.

| *Tabela 11. Porównanie komend RCLLNK i RCLSTG*

	RCLLNK OBJ('MyDir/MyObj')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
<b>Czy system musi być w stanie zastrzeżonym?</b>	Nie	Tak	Nie
<b>Czy podczas odzyskiwania możliwe jest użycie wszystkich systemów plików?</b>	Tak	Nie	Użycie systemów w niezależnej puli ASP jest niemożliwe.
<b>Obiekty z których puli ASP można odzyskać?</b>	Odzyskuje obiekty w systemowych i niezależnych pulach ASP oraz w pulach ASP użytkownika.	Odzyskuje obiekty w systemowych pulach ASP oraz w pulach ASP użytkownika.	Odzyskuje obiekty w niezależnych ASP.
<b>Jak odzyskiwane są obiekty?</b>	Obiekty są odzyskiwane pojedynczo lub podrzewami, zgodnie z parametrami komendy.	Odzyskiwane są obiekty w całym systemie.	Odzyskiwane są obiekty w niezależnych pulach ASP.
<b>Które ze znanych problemów systemu plików są identyfikowane i usuwane, jeśli możliwe?</b>	Większość (więcej informacji zawiera temat Odzyskiwanie systemów plików "root" (/), QOpenSys i UDFS).	Wszystkie	Wszystkie
<b>Czy utracone obiekty są odnajdywane?</b>	Nie	Tak	Tak
<b>Czy odzyskiwane są obiekty w odłączonych systemach plików?</b>	Nie	Tak	Tak
<b>Czy komenda jest wątkowo bezpieczna?</b>	Tak	Nie	Nie
<b>Ile instancji komendy może zostać wywołanych jednocześnie?</b>	Wiele	Jedna	Jedna
<b>Które z obiektów zintegrowanego systemu plików są ponownie tworzone w razie potrzeby?</b>	Wszystkie	Większość (więcej informacji zawiera temat Ponowne tworzenie obiektów zintegrowanego systemu plików).	Brak
<b>Czy możliwe jest zidentyfikowanie uszkodzonych obiektów bez ich odtwarzania?</b>	Tak	Nie	Nie

### Pojęcia pokrewne

| “Przykłady: komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)” na stronie 104  
 | Poniższe przykłady opisują sytuacje, w których komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object  
 | Links - RCLLNK) może zostać użyta do odzyskania obiektów w systemach plików "root" (/), QOpenSys i  
 | podłączonych systemach plików użytkownika.

### Odsyłacze pokrewne

| “Ponowne tworzenie obiektów zintegrowanego systemu plików” na stronie 103  
 | Poniższa tabela zawiera obiekty zintegrowanego systemu plików, które są ponownie tworzone przez komendę  
 | Odzyskiwanie dowiązań do obiektów (Reclaim Object Links - RCLLNK), jeśli nie istnieją. Są one zazwyczaj

tworzone podczas ładowania programu początkowego (IPL). Istnieje również możliwość ponownego utworzenia w razie potrzeby niektórych z tych obiektów przy użyciu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

#### Informacje pokrewne

Komenda Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)

Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)

## Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)

Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) umożliwia zidentyfikowanie i naprawienie uszkodzonych obiektów w systemach plików "root" (/), QOpenSts i i podłączonych systemach plików użytkownika; system nie musi znajdować się w stanie zastrzeżonym. Umożliwia to usunięcie problemów z systemami plików bez obniżania efektywności.

Komenda RCLLNK może być w wielu sytuacjach używana jako alternatywa do komendy Odzyskiwanie pamięci (RCLSTG). Użycie komendy RCLLNK jest na przykład dobrym sposobem identyfikowania i usuwania problemów w następujących sytuacjach:

- Problemy są związane z jednym obiektem.
- Problemy są związane z grupą obiektów.
- Zachodzi potrzeba zidentyfikowania lub usunięcia uszkodzonych obiektów.
- System nie może znajdować się w stanie zastrzeżonym podczas operacji odzyskiwania.
- Niezależne pule ASP muszą być dostępne podczas operacji odzyskiwania.

## Ponowne tworzenie obiektów zintegrowanego systemu plików

Poniższa tabela zawiera obiekty zintegrowanego systemu plików, które są ponownie tworzone przez komendę Odzyskiwanie dowiązań do obiektów (Reclaim Object Links - RCLLNK), jeśli nie istnieją. Są one zazwyczaj tworzone podczas ładowania programu początkowego (IPL). Istnieje również możliwość ponownego utworzenia w razie potrzeby niektórych z tych obiektów przy użyciu komendy Odzyskiwanie pamięci (Reclaim Storage - RCLSTG).

Tabela 12. Obiekty zintegrowanego systemu plików ponownie tworzone przez komendy RCLLNK i RCLSTG

Nazwa ścieżki	Typ	Ponownie tworzony przez komendę RCLLNK	Ponownie tworzony przez komendę RCLSTG ASPDEV(*SYSBASE)
/dev/zero	*CHRFS	Tak	Tak
/dev/null	*CHRFS	Tak	Tak
/dev/xti/tcp	*CHRFS	Tak	Nie
/dev/xti/udp	*CHRFS	Tak	Nie
/etc/vfs	*STMF	Tak	Nie

Aby możliwe było ponowne utworzenie obiektów zintegrowanego systemu plików przez komendę RCLLNK, należy ją wywołać, podając katalog nadrzędny, przy parametrze SUBTREE mającym wartość \*DIR lub \*ALL. Możliwe musi być pomyślne odzyskanie przez komendę katalogu nadrzędnego obiektu systemowego. Na przykład komenda RCLLNK OBJ('/dev') SUBTREE(\*DIR)

ponownie tworzy obiekty /dev/zero i /dev/null \*CHRFS, jeśli nie istnieją.

Aby możliwe było ponowne utworzenie obiektów zintegrowanego systemu plików przez komendę RCLSTG, należy ją wywołać przy parametrze ASPDEV mającym wartość \*SYSBASE; nie można pominąć fragmentu odzyskiwania odpowiedzialnego za odzyskiwanie katalogów.

#### Pojęcia pokrewne

“Utworzone katalogi” na stronie 7

Podczas restartu systemu zintegrowany system plików tworzy, o ile jeszcze nie istnieją, poniższe katalogi.

#### **Odsyłacze pokrewne**

“Porównanie komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)” na stronie 101

Komendy Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) umożliwiają usunięcie problemów z systemami plików “root” (/), QOpenSys i UDFS.

#### **Informacje pokrewne**

Komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)

## **Przykłady: komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK)**

Poniższe przykłady opisują sytuacje, w których komenda Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) może zostać użyta do odzyskania obiektów w systemach plików “root” (/), QOpenSys i podłączonych systemach plików użytkownika.

#### **Odsyłacze pokrewne**

“Porównanie komend Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG)” na stronie 101

Komendy Odzyskiwanie dowiązań do obiektu (Reclaim Object Links - RCLLNK) i Odzyskiwanie pamięci (Reclaim Storage - RCLSTG) umożliwiają usunięcie problemów z systemami plików “root” (/), QOpenSys i UDFS.

### **Przykład: usuwanie problemów z obiektem**

W tej sytuacji problemy są znane i dotyczą 1 obiektu. Obiekt jest uszkodzony, użycie go jest niemożliwe oraz niemożliwe jest odtworzenie wersji zapasowej z nośnika. Zachodzi potrzeba szybkiego usunięcia problemu bez zakłócania normalnych działań w systemie plików.

Aby odtworzyć obiekt, należy wywołać następującą komendę:

```
RCLLNK OBJ('/Kat/UszkObiekt') SUBTREE(*NONE)
```

, gdzie wartość /Kat/UszkObiekt odpowiada nazwie uszkodzonego obiektu.

### **Przykład: usuwanie problemów występujących w strukturze katalogów**

W tej sytuacji problemy są znane i dotyczą grupy obiektów w poddrzewie katalogów. Z ich powodu występuje błąd aplikacji. Zachodzi potrzeba szybkiego usunięcia problemu bez zakłócania normalnych działań w systemie plików.

Aby odzyskać obiekty w poddrzewie katalogów, należy wywołać następującą komendę:

```
RCLLNK OBJ('/KatInstalacyjnyAplikacji') SUBTREE(*ALL)
```

, gdzie KatInstalacyjnyAplikacji jest nazwą katalogu zawierającego obiekty, z którymi występują problemy.

### **Przykład: wyszukanie wszystkich uszkodzonych obiektów w systemach plików “root” (/), QOpenSys i UDFS**

W tej sytuacji awaria dysku powoduje uszkodzenie pewnej liczby obiektów. Zachodzi potrzeba zidentyfikowania uszkodzonych obiektów przed określeniem właściwego sposobu ich odzyskania.

Konieczne jest rozwiązanie umożliwiające wyszukanie uszkodzonych obiektów bez usuwania problemów. Operacja nie może mieć wpływu na normalne działania na systemie plików.

Aby zidentyfikować uszkodzone obiekty, należy wywołać następującą komendę:

```
RCLLNK OBJ('/') SUBTREE(*ALL) DMGOBJOPT(*KEEP *KEEP)
```



| Dodatkowo komenda usunie podczas wyszukiwania problemy nie dotyczące uszkodzonych obiektów.

| **Przykład: usunięcie wszystkich uszkodzonych obiektów w systemach plików "root" (/), QOpenSys i UDFS**

| W tej sytuacji awaria dysku powoduje uszkodzenie pewnej liczby obiektów. Uszkodzone obiekty należy usunąć, aby możliwe było odtworzenie kopii zapasowej z nośnika.

| Aby usunąć uszkodzone obiekty, należy wywołać następującą komendę:

```
RCLLNK OBJ('/ ') SUBTREE(*ALL) DMGOBJOPT(*DELETE *DELETE)
```

| Uszkodzone obiekty są usuwane; operacja ta nie ma wpływu na normalne działania na systemie plików. Dodatkowo podczas usuwania uszkodzonych obiektów usuwane są inne problemy.

| **Przykład: uruchomienie wielu instancji komendy RCLLNK w celu szybkiego odzyskania wszystkich obiektów w systemach plików "root" (/), QOpenSys i UDFS**

| W sytuacji wszystkie obiekty w systemach plików "root" (/), QOpenSys i podłączonych systemach plików użytkownika są odzyskiwane w trakcie rutynowej obsługi systemu. Użytkownik może pragnąć zakończyć operację odzyskiwania tak szybko, jak możliwe w celu umożliwienia dodatkowej obsługi.

| Podzielenie operacji odzyskiwania na grupy umożliwia jednoczesne wykonanie wielu instancji komendy RCLLNK, co skraca czas odzyskiwania.

| Aby uruchomić wiele operacji odzyskiwania katalogów systemowych i katalogów najwyższego poziomu, należy wywołać poniższe komendy (każdą w osobnym wątku lub zadaniu).

```
RCLLNK OBJ('/ ') SUBTREE(*DIR)
RCLLNK OBJ('/tmp') SUBTREE(*ALL)
RCLLNK OBJ('/home') SUBTREE(*ALL)
RCLLNK OBJ('/etc') SUBTREE(*ALL)
RCLLNK OBJ('/usr') SUBTREE(*ALL)
RCLLNK OBJ('/QIBM') SUBTREE(*ALL)
RCLLNK OBJ('/QOpenSys') SUBTREE(*ALL)
RCLLNK OBJ('/IaspName') SUBTREE(*ALL)
RCLLNK OBJ('/dev') SUBTREE(*ALL)
RCLLNK OBJ('/InneKatalogiNajwyzszegoPoziomu') SUBTREE(*ALL)
```

| , gdzie wartość InneKatalogiNajwyzszegoPoziomu oznacza pozostałe katalogi, które mają zostać odzyskane.

---

## Obsługa programistyczna

Dodanie do serwera iSeries w wersji V3R1M0 obsługi zintegrowanego systemu plików nie miało wpływu na istniejące aplikacje serwera iSeries. Języki programowania, narzędzia i obsługa systemowa (np. 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, należy używać zestawu funkcji API, które umożliwiają dostęp do funkcji dostarczanych przez zintegrowany system plików.

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

## Kopiowanie danych między 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.

### Pojęcia pokrewne

“Plik strumieniowy” na stronie 17

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

## Kopiowanie danych za pomocą komend CL

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

### Komendy CPYTOSTMF i CPYFRMSTMF

Do kopiowania danych między plikami strumieniowymi a podzbiorami baz danych można używać komendy Kopiowanie z pliku strumieniowego (Copy from Stream File - CPYFRMSTMF) i Kopiowanie do pliku strumieniowego (Copy to Stream File - CPYTOSTMF). Plik strumieniowy można utworzyć z podzbioru bazy danych za pomocą komendy CCPYTOSTMF. Można też utworzyć podzbiór bazy danych z pliku strumieniowego, używając komendy CPYFRMSTMF. Jeśli zbiór lub podzbiór, do którego są kopiowane dane, nie istnieje, to zostanie on utworzony.

Pojawiają się jednak pewne ograniczenia. Zbiór bazy danych musi być albo zbiorem fizycznym opisywanym przez program i zawierającym tylko jedno pole, albo źródłowym zbiorem fizycznym zawierającym tylko jedno pole tekstowe. Omawiane komendy obejmują wiele opcji konwersji i zmiany formatu kopiowanych danych.

Komendy CPYTOSTMF i CPYFRMSTMF mogą również zostać użyte do kopiowania danych między plikiem strumieniowym a zbiorem składowania.

### Komendy CPYTOIMPF i CPYFRMIMPF

Do kopiowania danych między plikami strumieniowymi a podzbiorami baz danych można również użyć komend Kopiowanie do zbioru importu (Copy to Import File - CPYTOIMPF) i Kopiowanie ze zbioru importu (Copy from Import File - CPYFRMIMPF). Komendy CPYTOSTMF i CPYFRMSTMF uniemożliwiają przenoszenie danych ze złożonych, 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 między heterogenicznymi bazami danych.

Podczas kopiowania z pliku strumieniowego (lub pliku importu) komenda CPYFRMIMPF pozwala użytkownikowi określić plik definicji pola (field definition file - FDF), który opisuje dane zawarte w tym pliku strumieniowym. Użytkownik może też określić, czy plik strumieniowy jest ograniczony i jakie znaki zostaną użyte do oznaczenia granic łańcuchów, pól i rekordów. Dostępne są też opcje przekształcania specjalnych typów danych, takich jak data i godzina.

Konwersja danych jest w tych komendach dostępna, jeśli docelowy plik strumieniowy lub podzbiór bazy danych już istnieje. Jeśli ten plik nie istnieje, można wykorzystać przedstawioną poniżej dwustopniową metodę uzyskiwania przekształconych danych.

1. Użyj komend CPYTOIMPF i CPYFRMIMPF do skopiowania danych z pliku zewnętrznie opisanego do źródłowego zbioru fizycznego.

2. Komendy CPYTOSTMF i CPYFRMSTMF (które przeprowadzają pełną konwersję danych bez względu na to, gdzie znajduje się plik docelowy) można użyć do skopiowania danych między fizycznym plikiem źródłowym, a plikiem strumieniowym.

A oto przykład:

```
| CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)  
|           FLDDLML(';') RCDDLML(X'07') STRDLML(*DBLQUOTE) DATFMT(*USA) TIMFMT(*USA)
```

Parametr DTAFMT określa, czy strumień wejściowy (plik importu) jest ograniczony; można też wybrać parametr DTAFMT(\*FIXED), który wymaga podania pliku definicji pola. Parametry FLDDLML, RCDDLML i STRDLML określają znaki, które działają jako ograniczniki lub separatory pól, rekordów i łańcuchów.

Parametry DATFMT i TIMFMT określają format wszelkich informacji dotyczących daty i godziny, które są kopiowane do pliku importu.

Komendy te są użyteczne, ponieważ mogą zostać umieszczone w programie i działają wyłącznie na serwerze. Jednak ich interfejsy są złożone.

#### **Informacje pokrewne**

Komenda Kopiowanie do pliku strumieniowego (Copy to Stream File - CPYTOSTMF)

Komenda Kopiowanie z pliku strumieniowego (Copy from Stream File - CPYFRMSTMF)

Komenda Kopiowanie do zbioru importu (Copy to Import File - CPYTOIMPF)

Kopiowanie ze zbioru importu (Copy from Import File - CPYFRMIMPF)

Język CL

## **Kopiowanie danych za pomocą funkcji API**

Aby skopiować podzbiory zbioru bazy danych do pliku strumieniowego w aplikacji, należy użyć funkcji `open()`, `read()` i `write()` w celu otwarcia podzbioru, odczytania danych i zapisania ich w nim lub w innym pliku.

#### **Odsyłacze pokrewne**

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

#### **Informacje pokrewne**

`open()`

`read()`

`write()`

Funkcje API zintegrowanego systemu plików

## **Kopiowanie danych za pomocą funkcji przesyłania danych**

Zaletami aplikacji przesyłania danych w programie licencjonowanym iSeries Access Family są: łatwy w użyciu interfejs graficzny, automatyczna numeracja i konwersja danych znakowych.

Funkcja przesyłania danych wymaga jednak zainstalowania produktu iSeries Access Family i używania zarówno zasobów komputera PC jak i zasobów serwera iSeries, a także korzystania z komunikacji między nimi.

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

#### **Transfer danych ze zbioru bazy danych do pliku strumieniowego:**

Aby przesłać dane ze zbioru bazy danych do pliku strumieniowego na serwerze, należy wykonać poniższe czynności.

1. Połącz się z serwerem.

2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. Kliknij opcję **Transfer danych z serwera iSeries** w oknie dialogowym iSeries Access for Windows .
4. Wybierz serwer, z którego chcesz przesyłać dane.
5. Używając baz danych biblioteki iSeries i nazwy pliku, wybierz nazwę pliku, z którego będą kopiowane dane, a także napęd sieciowy dla wynikowego pliku strumieniowego. Możesz też wybrać przycisk **Szczegóły** w temacie **PC** w celu wybrania formatu pliku PC dla pliku strumieniowego. Transfer danych obsługuje najczęściej spotykane typy plików PC, takie jak: tekstowy ASCII, BIFF3, CSV, DIF, tekstowy ograniczony znakami tabulacji oraz WK4.
6. Kliknij przycisk **Transfer danych z serwera 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 Family z wiersza komend można uruchomić wsadowo dwa programy:

- RTOPCB, którego parametrem jest plik .DTF lub plik .TTO
- RFROMPCB, którego parametrem jest plik .DTT lub plik .TFR

Używając programu planującego można w określonym terminie uruchomić dowolną z tych komend. Na przykład, można użyć narzędzia System Agent Tool (część pakietu Plus Pack firmy Microsoft) do wskazania programu, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), i godziny jego uruchomienia.

#### **Transfer danych z pliku strumieniowego do zbioru bazy danych:**

Aby przesłać dane z pliku strumieniowego do pliku bazy danych na serwerze, wykonaj poniższe czynności.

1. Połącz się z serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. Kliknij opcję **Transfer danych do serwera iSeries** w oknie dialogowym iSeries Access for Windows.
4. Na komputerze PC wybierz nazwę pliku, który ma być przesłany. 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 opcję **Transfer danych do serwera 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 pliku bazy danych, kliknij przycisk **Transfer danych do serwera 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 IBM eServer iSeries Access Family z wiersza komend można uruchomić wsadowo dwa programy:

- RTOPCB, którego parametrem jest plik .DTF lub plik .TTO
- RFROMPCB, którego parametrem jest plik .DTT lub plik .TFR

Używając programu planującego można w określonym terminie uruchomić dowolną z tych komend. Na przykład, można użyć narzędzia System Agent Tool (część pakietu Plus Pack firmy Microsoft) do wskazania programu, który ma być uruchomiony (na przykład RTOPCB MYFILE.TTO), i godziny jego uruchomienia.

#### **Odsyłacze pokrewne**

“Tworzenie pliku opisu formatu”

Jeśli dane są przesyłane do istniejącej definicji zbioru bazy danych, aplikacja Transfer danych do serwera iSeries wymaga użycia powiązanego pliku opisu formatu (FDF).

### **Transfer 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 poniższe czynności.

1. Połącz się z serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. Kliknij opcję **Transfer danych do serwera iSeries** w oknie dialogowym iSeries Access for Windows.
4. Otwórz menu Narzędzia aplikacji Transfer danych do serwera iSeries.
5. Kliknij polecenie **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 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 utworzyć plik .FDF, wykonaj następujące czynności:

1. Utwórz opisany zewnętrznie zbiór bazy danych o formacie odpowiadającym źródłowemu plikowi strumieniowemu (liczba pól, typy danych).
2. Utwórz jeden tymczasowy rekord w zbiorze bazy danych.
3. Funkcja Transfer danych z serwera iSeries służy do tworzenia pliku strumieniowego i przypisanego do niego pliku .FDF ze zbioru bazy danych.

Teraz można użyć funkcji Transfer danych do serwera iSeries. Określ plik .FDF oraz źródłowy plik strumieniowy, który ma być przesłany.

#### **Odsyłacze pokrewne**

“Transfer danych z pliku strumieniowego do zbioru bazy danych” na stronie 108

Aby przesłać dane z pliku strumieniowego do pliku bazy danych na serwerze, wykonaj poniższe czynności.

“Transfer danych ze zbioru bazy danych do pliku strumieniowego” na stronie 107

Aby przesłać dane ze zbioru bazy danych do pliku strumieniowego na serwerze, należy wykonać poniższe czynności.

## **Kopiowanie danych między plikami strumieniowymi a zbiorami składowania**

Zbiór składowania jest używany, wraz z komendami składowania i odtwarzania, do zachowywania danych, które w przeciwnym wypadku byłyby zapisywane na taśmie lub na dyskietce.

Zbioru tego można również używać jak zbioru bazy danych do odczytywania i zapisywania rekordów zawierających informacje o składowaniu/odtworzeniu. Zbioru składowania można ponadto używać do wysyłania obiektów do innych użytkowników w sieci SNADS.

Do kopiowania zbioru składowania do pliku strumieniowego i odwrotnie służy komenda CPY. Jednak przy kopiowaniu pliku strumieniowego z powrotem do zbioru składowania, dane muszą mieć poprawny format danych składowania (muszą to być dane pochodzące ze zbioru składowania i skopiowane do pliku strumieniowego).

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

### Informacje pokrewne

Komenda Kopiowanie obiektu (Copy Object - CPY)

## Wykonywanie operacji za pomocą funkcji API

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

Dostępne są dwa zestawy funkcji, których można użyć w programach tworzonych przy pomocy środowiska Integrated Language Environment (ILE) C/400:

- funkcje zintegrowanego systemu plików w języku C zawarte w systemie i5/OS,
- funkcje języka C dostępne w programie licencjonowanym ILE C/400.

Informacje o programach obsługi wyjścia obsługiwanych przez zintegrowany system plików można uzyskać po wybraniu odsyłacza Tabela 14 na stronie 115.

Funkcje zintegrowanego systemu plików działają jedynie poprzez opcję obsługi wejścia/wyjścia strumieni zintegrowanego systemu plików. Następujące funkcje API są obsługiwane:

*Tabela 13. Funkcje API zintegrowanego systemu plików*

Funkeja	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
closedir()	Zamknięcie katalogu
creat()	Utworzenie nowego lub wyczyszczenie istniejącego pliku
creat64()	Utworzenie nowego lub wyczyszczenie istniejącego pliku (również dla dużych plików)
DosSetFileLocks()	Blokowanie i anulowanie blokady zakresu bajtów w pliku
DosSetFileLocks64()	Blokowanie i anulowanie blokady zakresu bajtów w pliku (obsługa wielkich plików)
DosSetRelMaxFH()	Zmiana maksymalnej liczby deskryptorów plików
dup()	Powielenie deskryptora otwartego pliku
dup2()	Powielenie deskryptora otwartego pliku do nowego deskryptora
faccessx()	Określenie dostępności pliku dla klasy użytkowników przez deskryptor
fchdir()	Zmiana bieżącego katalogu przez deskryptor
fchmod()	Zmiana uprawnień do pliku przez deskryptor
fchown()	Zmiana właściciela i grupy dla pliku przez deskryptor
fclear()	Usunięcie zawartości pliku



Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

<b>Funkcja</b>	<b>Opis</b>
fclear64()	Usunięcie zawartości pliku (obsługa wielkich plików)
fcntl()	Wykonanie kontroli pliku
fpathconf()	Pobranie konfigurowalnych zmiennych z nazwami ścieżek przez deskryptor
fstat()	Pobranie informacji o pliku przez deskryptor
fstat64()	Pobranie informacji o pliku przez deskryptor (również dla dużych plików)
fstatvfs()	Pobranie informacji przez deskryptor
fstatvfs64()	Pobranie informacji przez deskryptor (uaktywniony tryb 64-bitowy)
fsync()	Synchronizacja zmian w pliku
ftruncate()	Obcięcie pliku
ftruncate64()	Obcięcie pliku (również dla dużych plików)
getcwd()	Pobranie ścieżki bieżącego katalogu
getegid()	Pobranie obowiązującego ID grupy
geteuid()	Pobranie obowiązującego ID użytkownika
getgid()	Pobranie rzeczywistego ID grupy
getgrgid()	Pobranie informacji o grupie za pomocą ID grupy
getgrnam()	Pobranie informacji o grupie za pomocą nazwy grupy
getgroups()	Pobranie ID grupy
getpwnam()	Pobranie informacji o użytkowniku dla danej nazwy użytkownika
getpwuid()	Pobranie informacji o użytkowniku dla danego ID użytkownika
getuid()	Pobranie rzeczywistego ID użytkownika
givedescriptor()	Przekazanie dostępu do pliku do innego zadania
ioctl()	Wykonanie kontroli we/wy pliku
link()	Utworzenie dowiązania do pliku
lseek()	Ustawienie przesunięcia (offset) odczytu/zapisu dla pliku
lseek64()	Ustawienie przesunięcia (offset) odczytu/zapisu dla pliku (również dla dużych plików)
lstat()	Pobranie informacji o pliku lub dowiązaniu
lstat64()	Pobranie informacji o pliku lub dowiązaniu (również dla dużych plików)
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

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
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
QjoRetrieveJournalEntries()	Odtworzenie pozycji kroniki
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)
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)

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
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)
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)

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
QlgSetAttr()	Ustawienie atrybutów systemowych dla obiektu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStat()	Pobranie informacji o pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStat64()	Pobranie informacji o pliku (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStatvfs()	Pobranie informacji o systemie plików (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgStatvfs64()	Pobranie informacji o systemie plików (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgSymlink()	Utworzenie dowiązania symbolicznego (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgUnlink()	Usunięcie dowiązania do pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QlgUtime()	Ustawienie godziny dostępu do pliku i jego modyfikacji (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe)
QP0FPTOS()	Wykonanie różnych funkcji systemowych
QP0LCHSG()	Zmiana sygnatury skanowania
Qp0lCvtPathToSYSObjName()	Tłumaczenie nazwy ścieżki zintegrowanego systemu plików na nazwę obiektu QSYS
QP0LFLOP()	Wykonanie różnych operacji na obiektach
Qp0lGetAttr()	Pobranie atrybutów systemowych dla obiektu
Qp0lGetPathFromFileID()	Pobranie ścieżki obiektu z jego ID pliku
Qp0lOpen()	Otwarcie pliku z nazwą ścieżki dostępną dla NLS
Qp0lProcessSubtree()	Przetworzenie katalogów lub obiektów wewnątrz drzewa katalogów
Qp0lRenameKeep()	Zmiana nazwy pliku lub katalogu z zachowaniem <i>nowego</i> , jeśli istnieje
Qp0lRenameUnlink()	Zmiana nazwy pliku lub katalogu z usunięciem dowiązania do <i>nowego</i> , jeśli istnieje
QP0LROR()	Wczytanie referencji do obiektu
QP0LRRO()	Pobranie obiektów odniesienia
QP0LRTSG()	Pobranie sygnatury skanowania
Qp0lSaveStgFree()	Zeszkładowanie danych obiektów i zwolnienie ich pamięci
Qp0lSetAttr()	Ustawienie atrybutów systemowych dla obiektu
Qp0lUnlink()	Odlączenie pliku z nazwą ścieżki dostępną dla NLS
qsysetgid()	Ustawienie obowiązującego ID grupy
qsyseteuid()	Ustawienie obowiązującego ID użytkownika
qsysetgid()	Ustawienie ID grupy
qsysetregid()	Ustawienie rzeczywistego i obowiązującego ID grupy
qsysetreuid()	Ustawienie rzeczywistego i obowiązującego ID użytkownika
qsysetuid()	Ustawienie ID użytkownika

Tabela 13. Funkcje API zintegrowanego systemu plików (kontynuacja)

Funkcja	Opis
QZNFRTVE()	Odzyskanie informacji dotyczących eksportu NFS
read()	Wczytanie z pliku
readdir()	Wczytanie pozycji katalogu
readdir_r()	Wczytanie pozycji katalogu (bezpieczne przy wątkach)
readlink()	Wczytanie wartości dowiązania symbolicznego
readv()	Wczytanie z pliku (wektor)
rename()	Zmiana nazwy pliku lub katalogu. Można ją zdefiniować ze składnią Qp0lRenameKeep() lub Qp0lRenameUnlink()
rewinddir()	Ustawienie początkowe (reset) strumienia katalogu
rmdir()	Usunięcie katalogu
select()	Sprawdzenie statusu we/wy wielu deskryptorów plików
stat()	Pobranie informacji o pliku
stat64()	Pobranie informacji o pliku (również dla dużych plików)
statvfs()	Pobranie informacji o systemie plików
statvfs64()	Pobranie informacji o systemie plików (również dla dużych plików)
symlink()	Utworzenie dowiązania symbolicznego
sysconf()	Pobranie zmiennych konfiguracyjnych systemu
takedescriptor()	Przejęcie dostępu do pliku od innego zadania
umask()	Ustawienie maski uprawnień dla zadania
unlink()	Usunięcie dowiązania do pliku
utime()	Ustawienie czasów dostępu i modyfikacji pliku
write()	Zapis do pliku
writev()	Zapis do pliku (wektor)

**Uwaga:** Niektóre z tych funkcji używane są także dla gniazd systemu i5/OS.

Tabela 14. Programy obsługi wyjścia zintegrowanego systemu plików

Funkcja	Opis
Funkcja API skanowania zintegrowanego systemu po zamknięciu	Wywoływana w trakcie przetwarzania po zamknięciu, na przykład funkcją API close(). Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Funkcja API Skanowania zintegrowanego systemu plików po otwarciu	Wywoływana w trakcie przetwarzania po otwarciu, na przykład funkcją API open(). Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Przetworzenie nazwy ścieżki	Wywołane przez funkcję API Qp0lProcessSubtree() dla każdego obiektu wyszukanego przez funkcję API, który spełnia kryteria wyboru. Ten program obsługi wyjścia musi być utworzony przez użytkownika.
Zwolnienie pamięci składowania	Wywołane przez funkcję API Qp0lSaveStgFree() w celu zeskładowania obiektów typu *STMF iSeries. Ten program obsługi wyjścia musi być utworzony przez użytkownika.

## Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

#### **Odsyłacze pokrewne**

“Używanie funkcji API zintegrowanego systemu plików w systemie plików "root" (/)" na stronie 34

Wszystkie funkcje API wymienione w temacie Używanie interfejsów API mogą być uruchamiane w systemie plików "root" (/).

“Korzystanie z funkcji API zintegrowanego systemu plików w systemie plików użytkownika” na stronie 40

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

“Dostęp przy użyciu funkcji API” na stronie 86

Funkcji API można użyć, aby uzyskać dostęp do zintegrowanego systemu plików.

“Przykład: funkcje C zintegrowanego systemu plików” na stronie 121

Kolejne strony zawierają prosty program napisany w języku C, przedstawiający użycie pewnych funkcji zintegrowanego systemu plików.

“Kopiowanie danych za pomocą funkcji API” na stronie 107

Aby skopiować podzbiory zbioru bazy danych do pliku strumieniowego w aplikacji, należy użyć funkcji `open()`, `read()` i `write()` w celu otwarcia podzbioru, odczytania danych i zapisania ich w nim lub w innym pliku.

#### **Informacje pokrewne**

Aplikacyjne interfejsy programistyczne (API)

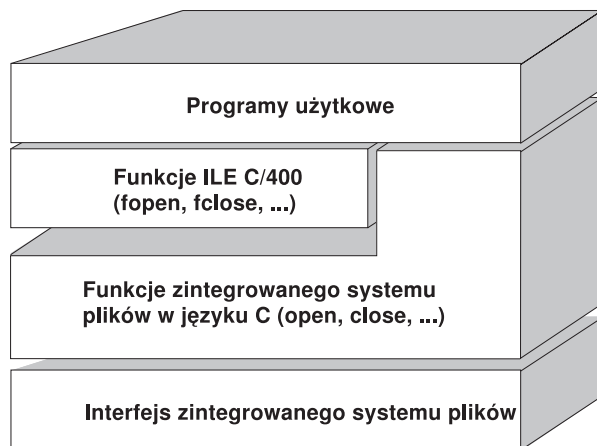
## **Funkcje ILE C/400.**

Funkcje ILE C/400 zawierają standardowe funkcje C zdefiniowane przez instytut American National Standards Institute (ANSI).

Funkcje te mogą obsługiwać zarządzanie danymi wejścia/wyjścia lub strumieniowe wejście/wyjście zintegrowanego systemu plików, zależnie od deklaracji przy tworzeniu programu w języku C. Domyślnie kompilator używa zarządzania danymi wejścia/wyjścia.

Aby wskazać programowi kompilującemu użycie strumieniowych danych we/wy zintegrowanego systemu plików, należy podać wartość `*IFSIO` parametru Interfejs systemowy (SYSIFCOPT) w komendzie `Utworzenie Modułu ILE C/400 (Create ILE C/400 Module - CRTCMOD)` lub `Utworzenie przypisanego programu C (Create Bound C Program - CRTBNDC)`. Po wpisaniu wartości `*IFSIO`, zamiast funkcji wejścia/wyjścia zarządzania danymi, dołączane są funkcje wejścia/wyjścia zintegrowanego systemu plików. W rezultacie funkcje C ILE C/400 używają funkcji zintegrowanego systemu plików do przeprowadzenia operacji wejścia/wyjścia.







RV3N070-3

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

zintegrowanego systemu plików.

Więcej informacji na temat używania funkcji C ILE C/400 z danymi strumieniowymi wejścia/wyjścia znaleźć można w publikacji WebSphere Development Studio: ILE C/C++ Programmers Guide . Szczegółowe informacje o każdej funkcji C ILE C/400 znaleźć można w publikacji WebSphere Development Studio: C/C++ Language Reference .

## Obsługa dużych plików

Funkcje API zintegrowanego systemu plików zostały tak rozbudowane, aby umożliwić aplikacjom przechowywanie w tym systemie bardzo dużych plików. Maksymalną wielkość obiektu wynosi dla nich około 1 TB (1 TB jest równy około 1 099 511 627 776 bajtom) w systemach plików “root” (/), QOpenSys i użytkownika.

Zintegrowany system plików wyposażony jest w zestaw 64-bitowych funkcji API typu UNIX i umożliwia łatwe odwzorowanie istniejących 32-bitowych funkcji API na funkcje API 64-bitowe, które mogą uzyskać dostęp do dużych plików za pomocą 8-bajtowych argumentów całkowitych.

Poniższe sytuacje umożliwiają aplikacjom użycie obsługi dużych plików:

- Jeśli w czasie kompilacji zostanie zdefiniowana etykieta makro `_LARGE_FILE_API`, aplikacje mają dostęp do 64-bitowych funkcji API i struktur danych. Na przykład aplikacja, która zamierza używać funkcji API `stat64()` i struktury `stat64`, będzie wymagała zdefiniowania podczas kompilacji etykiety `_LARGE_FILE_API`.
- Jeśli etykieta makro `_LARGE_FILES` zostanie zdefiniowana przez aplikację podczas kompilacji, istniejące funkcje API i struktury danych są odwzorowane na swoje wersje 64-bitowe. Na przykład, jeśli aplikacja definiuje podczas kompilacji etykiety `_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 etykiety `_LARGE_FILE_API` i bezpośrednio używać 64-bitowych funkcji API, albo zdefiniować podczas kompilacji etykiety `_LARGE_FILES`. Wszystkie odpowiednie funkcje API i struktury danych zostaną automatycznie odwzorowane na wersje 64-bitowe.

Aplikacje, które nie będą obsługiwały dużych plików, mogą używać funkcji API zintegrowanego systemu plików w dotychczasowej postaci.

### Informacje pokrewne

Funkcje API zintegrowanego systemu plików

`stat64()`

`stat()`

## Reguły dotyczące ścieżek dla funkcji API

Gdy do wykonywania działań na obiekcie używane są zintegrowany system plików lub funkcja API C/400, obiekt identyfikowany jest przez podanie ścieżki do katalogu. Poniżej przedstawiono kilka reguł, o których należy pamiętać, określając ścieżki w funkcjach API.

Określenie *obiekt* w tych zasadach odnosi się do dowolnego katalogu, pliku, dowiązania lub innego obiektu.

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

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

Ukośnik odwrotny (\) nie jest rozpoznawany jako separator. Jest on traktowany jak zwykły znak w nazwie.

- Nazwy obiektów w ramach katalogu nie mogą się powtarzać.
- Maksymalna długość każdego składnika ścieżki i maksymalna długość łańcucha określającego ścieżkę mogą się zmieniać w zależności od systemu plików.
- Znak / na początku ścieżki oznacza, że zaczyna się ona w katalogu głównym ("root"/), na przykład:

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

- Jeśli ścieżka nie zaczyna się od znaku /, przyjmuje się, że zaczyna się ona w bieżącym katalogu, na przykład:

```
MójKat/MójPlik
```

gdzie *MójKat* jest podkatalogiem bieżącego katalogu.

- Aby uniknąć pomyłek związanych z wartościami specjalnymi 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
```

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

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

- W nazwach ścieżek nie wolno używać dwukropka (:). Ma on specjalne znaczenie w systemie.
- W odróżnieniu od nazw ścieżek w komendach zintegrowanego systemu plików, gwiazdka (\*), znak zapytania (?), cudzysłów pojedynczy (') , cudzysłów (") oraz tylda (~) nie mają specjalnego znaczenia. Traktowane są one jak zwykłe znaki w nazwie. Aby uniknąć zamieszania związanego z użyciem wartości specjalnych serwera iSeries, nie należy posługiwać się nazwami ścieżek rozpoczynającymi się od znaku pojedynczej gwiazdki (\*). Jedynymi funkcjami API, których zasada ta nie dotyczy, są funkcje QjoEndJournal i QjoStartJournal.
- Gdy używane są interfejsy API Qlg (z użyciem nazw ścieżek obsługiwanych w systemie plików NLS), znaki null w nazwach ścieżek są niedozwolone, chyba że znak ten podany zostanie jako ogranicznik nazwy ścieżki.

### Pojęcia pokrewne

"Nazwa ścieżki" na stronie 16

*Nazwa ścieżki (path name)* (nazywana również pathname w niektórych systemach) informuje serwer, jak znaleźć dany obiekt.

### Odsyłacze pokrewne

"Reguły dotyczące nazw ścieżek dla komend CL i ekranów" na stronie 73

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

### Informacje pokrewne

Funkcja API QjoEndJournal

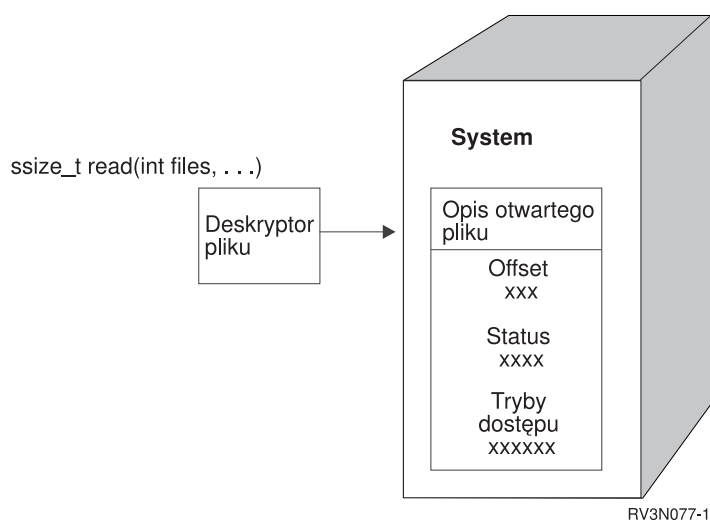
Funkcja API QjoStartJournal

## Deskryptor pliku

Gdy funkcje ILE obsługi strumieniowych danych wejścia/wyjścia C/400 zdefiniowane przez instytut American National Standards Institute (ANSI) używane są do operacji wykonywanych na pliku, plik identyfikowany jest przy użyciu wskaźników. Używając funkcji C zintegrowanego systemu plików, plik określa się podając deskryptor pliku. *Deskryptor pliku* jest dodatnią liczbą całkowitą, która musi być unikalna dla każdego zadania.

Zadanie używa deskryptora pliku do identyfikacji otwartego pliku podczas przeprowadzania na nim operacji. Deskryptor pliku reprezentowany jest w funkcjach w języku C działających na zintegrowanym systemie plików przez zmienną *files*, a w funkcjach w języku C działających na gniazdach przez zmienną *descriptor*.

Każdy deskryptor pliku powiązany jest z *opisem otwartego pliku*, który zawiera informacje, takie jak przesunięcie (offset), status pliku i tryby dostępu do niego. Z tym samym opisem otwartego pliku może być powiązanych wiele deskryptorów, ale deskryptor pliku może wskazywać tylko na jeden opis otwartego pliku.



Rysunek 11. Deskryptor pliku i opis otwartego pliku.

Gdy funkcje ILE C/400 obsługi strumieniowych danych wejścia/wyjścia używane są z zintegrowanym systemem plików, obsługa czasu wykonywania funkcji ILE C/400 zmienia wskaźnik pliku na deskryptor pliku.

Używając systemów plików "root" (/), QOpenSys lub UDFS, można przekazywać dostęp do opisu otwartego pliku z jednego zadania do innego, umożliwiając mu w ten sposób dostęp do tego pliku. Służą do tego funkcje `givedescriptor()` i `takedescriptor()` przekazujące deskryptory między zdaniami.

### Informacje pokrewne

- Funkcja API `givedescriptor()`
- Funkcja API `takedescriptor()`
- Programowanie z użyciem gniazd
- Interfejsy API gniazd

## Ochrona

Korzystając z funkcji API zintegrowanego systemu plików, podobnie jak w przypadku interfejsów zarządzania danymi, istnieje możliwość ograniczenia dostępu do obiektów. Trzeba jednak mieć świadomość, że przejmowanie uprawnień nie jest obsługiwane. Funkcja API zintegrowanego systemu plików korzysta z uprawnień pobieranych z profilu użytkownika, który jest właścicielem zadania.

Każdy system plików może mieć własne wymagania dotyczące uprawnień. 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 w serwerze są odpowiednikiem *praw* w systemach UNIX. Prawa dostępu mogą dotyczyć odczytu i zapisu (dla pliku lub katalogu), wykonywania (dla pliku) i przeszukiwania (dla katalogu). Prawa dostępu określa zestaw bitów praw dostępu składających się na “tryb dostępu” do pliku lub katalogu. Bity uprawnień można zmienić używając funkcji “zmiany” trybu `chmod()` lub `fchmod()`. Do sterowania ustawieniami bitów praw dostępu do plików tworzonych przez dane zadanie można również wykorzystywać funkcję `umask()`.

Szczegółowe informacje na temat ochrony danych i uprawnień opisane zostały w publikacji *Ochrona* .

#### **Informacje pokrewne**

Funkcja API `chmod()`

Funkcja API `fchmod()`

Funkcja API `umask()`

Funkcje API zintegrowanego systemu plików

## **Obsługa gniazd**

Jeśli aplikacja korzysta z systemu plików “root” (*/*), QOpenSys lub UDFS, można wykorzystać obsługę gniazd lokalnych (le systems, you can take advantage of the integrated file system *local sockets*) zintegrowanego systemu plików. Obiekt typu gniazdo lokalne (typ obiektu \*SOCKET) umożliwia nawiązanie komunikacji między dwoma zadaniami działającymi w tym samym systemie.

Jedno z zadań ustanawia punkt połączeniowy tworząc gniazdo lokalne za pomocą funkcji `bind()` w języku C. Drugie zadanie podaje nazwę gniazda lokalnego w funkcji `connect()`, `sendto()` lub `sendmsg()`.

Po nawiązaniu połączenia, zadania mogą wzajemnie wysłać i odebrać dane używając funkcji zintegrowanego systemu plików, takich jak `write()` i `read()`. Żadne z przesyłanych danych nie przechodzą przez gniazdo lokalne. Gniazdo jest jedynie miejscem spotkania, w którym jedno zadanie może znaleźć drugie.

Po zakończeniu komunikacji między zadaniami każde z nich używa funkcji `close()` w celu zamknięcia połączenia przez gniazdo. Lokalny obiekt gniazda pozostaje w systemie dopóki nie zostanie usunięty za pomocą funkcji `unlink()` lub komendy Usuwanie dowiązania (Remove Link - RMVLNK).

Gniazda lokalnego nie można zeszkładować.

#### **Informacje pokrewne**

Programowanie z użyciem gniazd

Funkcja API `write()`

Funkcja API `read()`

Funkcja API `close()`

Funkcja API `unlink()`

Komenda Usuwanie dowiązania (Remove link - RMVLNK)

## **Nazewnictwo i obsługa różnych języków**

Obsługa systemów plików “root” (*/*) i zapewnia niezmiennosc znaków w nazwach obiektów przy przechodzeniu między schematami kodowania używanymi w różnych językach narodowych i w różnych urządzeniach.

Podczas przekazywania nazwy obiektu do systemu, każdy jej znak jest poddawany konwersji do postaci 16-bitowej, w której wszystkie znaki mają standardowo kodowaną reprezentację. W chwili 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 niezmiennione 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.

### Pojęcia pokrewne

“Niezmienność nazw” na stronie 19

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

## 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.

Domyślna opcja *Binarne, wymuszone* użyta jest dla funkcji `open()`.

### Informacje pokrewne

Funkcja API `open()`

## Przykład: funkcje C zintegrowanego systemu plików

Kolejne strony zawierają prosty program napisany w języku C, przedstawiający użycie pewnych funkcji zintegrowanego systemu plików.

Program wykonuje następujące operacje:

- 1 Używa funkcji `getuid()` do określenia rzeczywistego identyfikatora użytkownika (`uid`).
- 2 Używa funkcji `getcwd()` do określenia bieżącego katalogu.
- 3 Używa funkcji `open()` do utworzenia pliku. Nadaje ona właścicielowi (osobie, która utworzyła zbiór) uprawnienia do zapisu, odczytu i wykonywania pliku.
- 4 Używa funkcji `write()` do zapisu ciągu bajtów do pliku. Plik identyfikuje deskryptor pliku, który został dostarczony podczas operacji otwarcia (3).
- 5 Używa funkcji `close()` do zamknięcia pliku.
- 6 Używa funkcji `mkdir()` do utworzenia nowego podkatalogu w katalogu bieżącym. Właściciel ma prawo do czytania, zapisu i wykonywania tego podkatalogu.
- 7 Używa funkcji `chdir()` do zmiany podkatalogu bieżącego na nowy podkatalog.
- 8 Używa funkcji `link()`, aby utworzyć dowiązanie do poprzednio utworzonego pliku (3).
- 9 Używa funkcji `open()` do otwarcia pliku tylko do odczytu. Dowiązanie, które zostało utworzone w (8), umożliwia dostęp do danego pliku.
- 10 Używa funkcji `read()` do odczytu ciągu bajtów z pliku. Plik identyfikuje deskryptor pliku, który został dostarczony podczas operacji otwarcia (9).
- 11 Używa funkcji `close()` do zamknięcia pliku.
- 12 Używa funkcji `unlink()`, aby usunąć dowiązanie do pliku.
- 13 Używa funkcji `chdir()`, aby zmienić bieżący katalog z powrotem na katalog nadrzędny, w którym został utworzony nowy podkatalog.
- 14 Używa funkcji `rmdir()`, aby usunąć poprzednio utworzony podkatalog (6).
- 15 Używa funkcji `unlink()`, aby usunąć poprzednio utworzony 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.

**Uwaga:** Użycie poniższych przykładów kodu jest równoznaczne z zaakceptowaniem warunków “Informacje dotyczące licencji na kod” na stronie 138.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
```

```
#define BUFFER_SIZE      2048
#define NEW_DIRECTORY    "testdir"
#define TEST_FILE        "test.file"
#define TEST_DATA        "Hello World!"
#define TEST_DATA        "Hello World!"
#define USER_ID          "user_id_"
#define PARENT_DIRECTORY ".."
```

```
char InitialFile[BUFFER_SIZE];
char LinkName[BUFFER_SIZE];
```



```

char InitialDirectory[BUFFER_SIZE] = ".";
char Buffer[32];
int FilDes = -1;
int BytesRead;
int BytesWritten;
uid_t UserID;

void CleanUpOnError(int level)
{
    printf("Napotkano błąd, trwa sprzątanie.\n");
    switch ( level )
    {
        case 1:
            printf("Nie można pobrać bieżącego katalogu roboczego.\n");
            break;
        case 2:
            printf("Nie można utworzyć pliku %s.\n",TEST_FILE);
            break;
        case 3:
            printf("Nie można pisać do pliku %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 4:
            printf("Nie można zamknąć pliku %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 5:
            printf("Nie można utworzyć katalogu %s.\n",NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 6:
            printf("Nie można zmienić na katalog %s.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 7:
            printf("Nie można utworzyć dowiązania %s to %s.\n",LinkName,InitialFile);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 8:
            printf("Nie można utworzyć dowiązania %s.\n",LinkName);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 9:
            printf("Nie można odczytać dowiązania %s.\n",LinkName);
            close(FilDes);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 10:
            printf("Nie można zamknąć dowiązania %s.\n",LinkName);
            close(FilDes);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
    }
}

```

```

        case 11:
            printf("Nie można zamknąć dowiązania %s.\n",LinkName);
            unlink(TEST_FILE);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 12:
            printf("Nie można zmienić katalogu %s.\n",PARENT_DIRECTORY);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 13:
            printf("Nie można usunąć katalogu %s.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 14:
            printf("Nie można usunąć dowiązania pliku %s.\n",TEST_FILE);
            unlink(TEST_FILE);
            break;
        default:
            break;
    }
    printf("Program zakończony błędem.\n\
        "Nie wszystkie testowe pliki i katalogi mogły zostać usunięte.\n");
}

int main ()
{
    1
    /* Pobranie i wydruk rzeczywistego id użytkownika przy wykorzystaniu funkcji getuid(). */
    UserID = getuid();
    printf("Rzeczywistym identyfikatorem użytkownika jest %u. \n",UserID);

    2
    /* Pobranie bieżącego katalogu roboczego i zachowanie go w InitialDirectory. */
    if ( NULL == getcwd(InitialDirectory,BUFFER_SIZE) )
    {
        perror("Błąd funkcji getcwd");
        CleanupOnError(1);
        return 0;
    }
    printf("Bieżącym katalogiem roboczym jest %s. \n",InitialDirectory);

    3
    /* Utworzenie pliku TEST_FILE do zapisu, jeśli taki nie istnieje.
        Nadanie właścicielowi uprawnień do odczytu, zapisu i wykonywania. */
    FilDes = open(TEST_FILE, O_WRONLY | O_CREAT | O_EXCL, S_IRWXU);
    if ( -1 == FilDes )
    {
        perror("Błąd funkcji open");
        CleanupOnError(2);
        return 0;
    }
    printf("Utworzono %s w katalogu %s.\n",TEST_FILE,InitialDirectory);

    4
    /* Wpisz TEST_DATA do TEST_FILE przez FilDes */
    BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
    if ( -1 == BytesWritten )
    {
        perror("Błąd funkcji write");
        CleanupOnError(3);
        return 0;
    }
}

```

```

        printf("Wpisano %s do pliku %s.\n",TEST_DATA,TEST_FILE);

5
/* Zamknięcie TEST_FILE przez FilDes */
    if ( -1 == close(FilDes) )
    {
        perror("Błąd funkcji close");
        CleanupOnError(4);
        return 0;
    }
    FilDes = -1;
    printf("Zamknięcie pliku %s.\n",TEST_FILE);

6
/* Utworzenie nowego katalogu w bieżącym katalogu roboczym
i nadanie właścicielowi uprawnień do odczytu, zapisu i wykonania */
    if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )
    {
        perror("Błąd funkcji mkdir");
        CleanupOnError(5);
        return 0;
    }
    printf("Utworzono katalog %s w katalogu %s.\n",NEW_DIRECTORY,InitialDirectory);

7
/* Zmiana bieżącego katalogu roboczego na
właśnie utworzony katalog NEW_DIRECTORY. */
    if ( -1 == chdir(NEW_DIRECTORY) )
    {
        perror("Błąd funkcji chdir");
        CleanupOnError(6);
        return 0;
    }
    printf("Zmiana na katalog %s/%s.\n",InitialDirectory,NEW_DIRECTORY);

/* Skopiowanie PARENT_DIRECTORY do InitialFile i
dodanie "/" i TEST_FILE do InitialFile. */
    strcpy(InitialFile,PARENT_DIRECTORY);
    strcat(InitialFile,"/");
    strcat(InitialFile,TEST_FILE);

/* Skopiowanie Copy USER_ID do LinkName, a następnie dołączenie
UserID jako łańcucha do LinkName. */
    strcpy(LinkName, USER_ID);
    sprintf(Buffer, "%d\0", (int)UserID);
    strcat(LinkName, Buffer);

8
/* Utworzenie dowiązania do nazwy InitialFile z LinkName. */
    if ( -1 == link(InitialFile,LinkName) )
    {
        perror("Błąd funkcji link");
        CleanupOnError(7);
        return 0;
    }
    printf("Utworzono dowiązanie %s do %s.\n",LinkName,InitialFile);

9
/* Otwarcie pliku LinkName tylko do odczytu. */
    if ( -1 == (FilDes = open(LinkName,O_RDONLY)) )
    {
        perror("Błąd funkcji open");
        CleanupOnError(8);
        return 0;
    }
    printf("Otwarto %s do odczytu.\n",LinkName);

```

```

10
/* Czytanie z pliku LinkName poprzez FilDes do Buffer. */
BytesRead = read(FilDes,Buffer,sizeof(Buffer));
if ( -1 == BytesRead )
{
    perror("Błąd funkcji read");
    CleanupOnError(9);
    return 0;
}
printf("Odczyt %s z %s.\n",Buffer,LinkName);
if ( BytesRead != BytesWritten )
{
    printf("OSTRZEŻENIE: liczba odczytanych bajtów "\
          "jest różna od liczby wpisanych bajtów.\n");
}

11
/* Zamknięcie pliku LinkName poprzez FilDes. */
if ( -1 == close(FilDes) )
{
    perror("Błąd funkcji close");
    CleanupOnError(10);
    return 0;
}
FilDes = -1;
printf("Zamknięto %s.\n",LinkName);

12
/* Odłączenie dowiązania LinkName do InitialFile. */
if ( -1 == unlink(LinkName) )
{
    perror ("Błąd funkcji unlink");
    CleanupOnError(11);
    return 0;
}
printf("Dowiązanie %s zostało usunięte.\n",LinkName);

13
/* Zmiana bieżącego katalogu roboczego
z powrotem na katalog początkowy. */
if ( -1 == chdir(PARENT_DIRECTORY) )
{
    perror("Błąd funkcji chdir");
    CleanupOnError(12);
    return 0;
}
printf("zmiana katalogu na %s.\n",InitialDirectory);

14
/* Usunięcie katalogu NEW_DIRECTORY */
if ( -1 == rmdir(NEW_DIRECTORY) )
{
    perror("Błąd funkcji rmdir");
    CleanupOnError(13);
    return 0;
}
printf("Usunięto katalog %s.\n",NEW_DIRECTORY);

15
/* Odłączenie pliku TEST_FILE */
if ( -1 == unlink(TEST_FILE) )
{
    perror ("Błąd funkcji unlink");
    CleanupOnError(14);
    return 0;
}
printf("Odłączono plik %s.\n",TEST_FILE);

```

```
    printf("Program został zakończony poprawnie.\n");  
    return 0;  
}
```

### Odsyłacze pokrewne

“Wykonywanie operacji za pomocą funkcji API” na stronie 110

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

---

## Praca z plikami i folderami z wykorzystaniem produktu iSeries Navigator

Dla plików i folderów można wykonać następujące zadania.

### Odblokowanie pliku

Czynności, które należy wykonać w celu zwrócenia pliku.

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

### Zablokowanie pliku

Czynności, które należy wykonać w celu pobrania pliku.

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

### Tworzenie folderu

Aby utworzyć folder, wykonaj następujące czynności.

1. Rozwiń okna **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** systemu, z którego chcesz skorzystać.
2. Kliknij prawym klawiszem myszy system plików, do którego chcesz dodać folder, i wybierz **Nowy folder**.
3. W oknie dialogowym **Nowy folder** wpisz nazwę nowego obiektu.
4. Kliknij przycisk **OK**.

Gdy w serwerze iSeries tworzony jest folder, należy zdecydować, czy nowy folder (lub obiekty) ma być chroniony przez kronikowanie. Należy także rozważyć, czy obiekty utworzone w tym folderze mają być skanowane, czy nie.

#### Zadania pokrewne

“Włączanie i wyłączanie skanowania obiektów” na stronie 132

Czynności, które należy wykonać, aby włączyć lub wyłączyć skanowanie obiektów.

#### Informacje pokrewne

Zarządzanie kronikami

### Usuwanie folderu

Aby usunąć folder, należy wykonać poniższe czynności.

1. W oknie **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** rozwiń system, który ma zostać uszyty. Rozwijaj kolejne gałęzie do momentu wyświetlenia odpowiedniego pliku lub folderu.

2. Kliknij prawym klawiszem myszy plik lub folder, a następnie wybierz opcję **Usuń**.

## Przenoszenie plików lub folderów do innego systemu plików

Każdy system plików ma swoją własną charakterystykę. Jednak przeniesienie obiektów do innego systemu plików może oznaczać utratę korzyści płynących z systemu plików, w którym obiekt jest aktualnie przechowywany. Można jednak tak przenosić obiekty z jednego systemu plików do innego, aby wykorzystywać zalety jego charakterystyki.

Przed przeniesieniem obiektów do innego systemu plików należy zatem zapoznać się z systemami plików w zintegrowanym systemie plików i ich charakterystykami.

Należy także rozważyć następujące kwestie:

- Czy używa się aplikacji wykorzystujących zalety systemu plików, w którym obecnie znajdują się obiekty?

Niektóre systemy plików obsługują interfejsy, których nie obsługuje zintegrowany system plików. Aplikacje używające tych interfejsów mogą nie mieć dostępu do obiektów przeniesionych do innego systemu. Na przykład systemy plików QDLS i QOPT obsługują hierarchiczny system plików (HFS). Nie można używać tych interfejsów w przypadku obiektów znajdujących się w innych systemach plików.

- Jakie cechy obiektów są ważne?

Nie wszystkie cechy są obsługiwane przez każdy system plików. Na przykład system plików QSYS.LIB lub QSYS.LIB niezależnej puli ASP obsługuje zapisywanie i odczytywanie tylko kilku atrybutów rozszerzonych, a systemy plików "root" (/) i QOpenSys obsługują zapisywanie i odczytywanie wszystkich atrybutów rozszerzonych. Dlatego systemów plików QSYS.LIB i QSYS.LIB niezależnej puli ASP nie należy używać do przechowywania obiektów posiadających atrybuty rozszerzone.

Dobrym rozwiązaniem jest przeniesienie plików PC znajdujących się w systemie QDLS. Większość aplikacji komputera PC powinna bez zakłóceń kontynuować pracę z plikami komputera PC przeniesionymi z systemu QDLS do innych systemów plików. Do przechowywania tych plików PC dobrze nadają się systemy plików: bazowy ("root", /), QOpenSys, QNetWare i QNTC. Ponieważ te systemy plików obsługują wiele charakterystyk systemu plików OS/2, mogą zapewnić szybszy dostęp do 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.



### Pojęcia pokrewne

“Praca z systemami plików” na stronie 26

*System plików (file system)* umożliwia dostęp do konkretnych segmentów pamięci zorganizowanych w logiczne jednostki. Jednostkami tymi są na serwerze zbiory, katalogi, biblioteki i obiekty.

### Odsyłacze pokrewne

“Porównanie systemów plików” na stronie 28

Poniższe tabele zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

### Informacje pokrewne

Komenda Utworzenie katalogu (Create Directory - CRTDIR)

Komenda Przeniesienie (Move - MOV)

Komenda Kopiowanie (Copy - CPY)

## Konfigurowanie uprawnień

Nadawanie uprawnień do obiektów pozwala na kontrolę manipulacji tymi obiektami przez innych użytkowników. Dzięki otrzymanym uprawnieniom niektórzy użytkownicy będą mogli jedynie wyświetlać obiekty, podczas gdy inni będą mogli poddawać je edycji.

Aby ustawić uprawnienia dla pliku lub folderu, wykonaj następujące czynności:

1. Rozwiń w oknie **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** system, który ma zostać użyty. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony obiekt, do którego chcesz nadać uprawnienia.
2. Kliknij obiekt prawym klawiszem myszy i wybierz **Uprawnienia**.
3. Kliknij przycisk **Dodaj** w oknie dialogowym **Uprawnienia**.
4. W oknie dialogowym **Dodaj** wybierz przynajmniej jednego użytkownika lub grupę albo wpisz odpowiednią nazwę w polu nazwy użytkownika lub grupy.
5. Kliknij przycisk **OK**. Spowoduje to dodanie użytkowników lub grup i umieszczenie ich na początku listy.
6. Kliknij przycisk **Szczegóły** aby wprowadzić szczegółowe uprawnienia.
7. Zaznaczenie odpowiedniego pola wyboru powoduje nadanie wymaganych uprawnień użytkownikowi.
8. 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 umożliwia automatyczną konwersję danych w pliku podczas przesyłania plików między serwerem iSeries a komputerem PC. W przypadku dostępu do pliku z komputera PC, jest on obsługiwany jak plik ASCII.

Aby skonfigurować konwersję tekstu pliku, wykonaj następujące czynności:

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

## Wysyłanie pliku lub folderu do innego systemu

Czynności, które należy wykonać, aby wysłać plik lub folder do innego systemu.

1. Rozwiń w oknie **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** system, który ma zostać użyty. Kontynuuj rozwijanie, dopóki nie zostanie wyświetlony plik lub folder, który ma zostać wysłany.

2. Kliknij plik lub folder prawym klawiszem myszy i wybierz polecenie **Wyślij**. Plik lub folder zostanie wyświetlony na liście Wybrane pliki i foldery w oknie dialogowym Wysyłanie plików z.
3. Rozwiń listę dostępnych systemów i grup.
4. Wybierz system i kliknij **Dodaj**, aby dodać go do listy **Systemy i grupy docelowe**. Powtórz tę czynność dla wszystkich systemów, do których chcesz wysłać wybrany plik lub folder.
5. Aby wysłać plik lub folder z bieżącymi definicjami pakietu i informacjami harmonogramu, kliknij **OK**.

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.

#### **Zadania pokrewne**

“Zmiana opcji definicji pakietów”

Zmiana opcji definicji pakietów umożliwia grupowanie zestawów obiektów i5/OS lub plików zintegrowanego systemu plików.

“Planowanie daty i godziny wysyłki pliku lub folderu”

Funkcja programu do planowania zadań zapewnia elastyczność i wykonywanie zadań w czasie najbardziej dogodnym dla użytkownika.

## **Zmiana opcji definicji pakietów**

Zmiana opcji definicji pakietów umożliwia grupowanie zestawów obiektów i5/OS 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. Wykonaj kroki opisane w temacie “Wysyłanie pliku lub folderu do innego systemu” na stronie 129.
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.

#### **Zadania pokrewne**

“Wysyłanie pliku lub folderu do innego systemu” na stronie 129

Czynności, które należy wykonać, aby wysłać plik lub folder do innego systemu.

“Planowanie daty i godziny wysyłki pliku lub folderu”

Funkcja programu do planowania zadań zapewnia elastyczność i wykonywanie zadań w czasie najbardziej dogodnym dla użytkownika.

## **Planowanie daty i godziny wysyłki pliku lub folderu**

Funkcja 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. Wykonaj kroki opisane w temacie “Wysyłanie pliku lub folderu do innego systemu” na stronie 129.
2. Kliknij **Harmonogram**.
3. Określ porę wysłania pliku lub folderu.

#### **Zadania pokrewne**

“Wysyłanie pliku lub folderu do innego systemu” na stronie 129

Czynności, które należy wykonać, aby wysłać plik lub folder do innego systemu.

“Zmiana opcji definicji pakietów” na stronie 130

Zmiana opcji definicji pakietów umożliwia grupowanie zestawów obiektów i5/OS lub plików zintegrowanego systemu plików.

## 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 w iSeries.

Aby utworzyć współużytkowany zasób plikowy, wykonaj następujące czynności.

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

## Zmiana współużytkowanego zasobu plikowego

*Współużytkowany zasób plikowy* jest ścieżką do katalogu, którą iSeries NetServer udostępnia klientom PC w sieci iSeries. Plik współużytkowany może zawierać dowolny katalog ze zintegrowanego systemu plików systemu iSeries.

Aby zmienić współużytkowany zasób plikowy, wykonaj następujące czynności.

1. Rozwiń system w **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików**.
2. Rozwiń folder ze zdefiniowanym zasobem współużytkowanym, który chcesz zmienić.
3. Prawym klawiszem myszy kliknij powyższy folder i wybierz opcję **Współużytkowanie**.
4. Wybierz **Nowy zasób współużytkowany**.

## Tworzenie nowego systemu plików użytkownika

System plików użytkownika (UDFS) jest systemem, dla którego należy utworzyć oraz zdefiniować atrybuty. Systemy UDFS znajdują się w pulach ASP systemu.

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

1. Rozwiń system w **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** → **Root** → **Dev**.
2. Kliknij ikonę puli ASP, w której chcesz umieścić nowy system plików użytkownika (UDFS).
3. Z menu folderu wybierz opcję **Nowy system plików użytkownika (UDFS)**.
4. Określ nazwę UDFS, opis (opcjonalnie), wartości nadzoru, domyślny format pliku, domyślny atrybut przeglądania. Określ również, czy w nazwach plików w nowym systemie UDFS będą rozróżniane wielkie litery.

## Podłączanie systemu plików utworzonego użytkownika

Dostęp do danych przechowywanych w systemie plików UDFS jest możliwy po podłączeniu go po ładowaniu IPL.

Po podłączeniu systemu plików UDFS następuje przesłonięcie systemów plików, katalogów i obiektów znajdującym się pod punktem podłączenia w hierarchii folderów. Wszystkie te systemy plików, katalogi lub obiekty będą niewidoczne aż do odłączenia UDFS. Aby zapewnić dostęp do wszystkich danych zintegrowanego systemu plików, należy podłączyć UDFS do pustego folderu. Po podłączeniu UDFS wszystkie pliki zawarte w nim będą dostępne w tym folderze. Wszelki zmiany dokonane w folderze spowodują zmianę w systemie UDFS, a nie w przysłoniętym folderze.

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

Aby podłączyć system plików zdefiniowany przez użytkownika (UDFS), należy wykonać poniższe czynności:

1. Rozwiń system w **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** → **Root** → **Dev**.
2. Kliknij pulę pamięci dyskowej (ASP) zawierającą system plików UDFS, który ma zostać podłączony.
3. W kolumnie **Nazwa UDFS** prawego panelu programu Operations Navigator kliknij prawym przyciskiem system UDFS, który ma zostać podłączony.
4. Wybierz polecenie **Podłącz**.

Jeśli używana jest technika przeciągnij i upuść, można podłączyć UDFS, przeciągając go do folderu 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 definiowanego przez użytkownika

Po podłączeniu systemu plików UDFS następuje przesłonięcie systemów plików, katalogów i obiektów znajdującym się pod punktem podłączenia w hierarchii folderów. Wszystkie te systemy plików, katalogi lub obiekty będą niewidoczne aż do odłączenia UDFS.

Aby odłączyć system plików zdefiniowany przez użytkownika (UDFS), należy wykonać poniższe czynności:

1. Rozwiń system w **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików** → **Root** → **Dev**.
2. Kliknij pulę pamięci dyskowej (ASP) zawierającą system plików UDFS, który ma zostać odłączony.
3. W kolumnie **Nazwa UDFS** prawego panelu programu iSeries Navigator kliknij prawym przyciskiem system UDFS, który ma zostać podłączony.
4. Wybierz polecenie **Odłącz**.

## Włączanie i wyłączanie skanowania obiektów

Czynności, które należy wykonać, aby włączyć lub wyłączyć skanowanie obiektów.

1. Rozwiń system w **iSeries Navigator** → **Systemy plików** → **Zintegrowany system plików**.
2. Rozwiń folder lub wybrany plik.
3. Kliknij prawym klawiszem myszy folder lub plik, a następnie wybierz opcję **Właściwości**
4. Kliknij zakładkę **Ochrona**.
5. Wybierz opcję w polu **Skanowanie obiektów**.

Więcej informacji o opcjach znajduje się w poniższych tematach. Opisy tych opcji dotyczą plików. Skanować można tylko pliki. Dla folderów i systemów plików użytkownika można zdefiniować przydzielanie atrybutu skanowania dla plików tworzonych w wybranym folderze lub systemie plików użytkownika.

- Tak

Obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.

- Nie

Obiekt nie będzie podlegać skanowaniu przez programy wyjściowe dotyczące skanowania.

**Uwaga:** Jeśli opcja Skanuj przy kolejnym dostępie po przywróceniu obiektu jest wybrana w wartościach systemowych, obiekt będzie podlegał skanowaniu co najmniej raz po przywróceniu.

- Tylko po zmianie obiektu

Obiekt będzie podlegać skanowaniu zgodnie z regułami opisanymi w programach wyjściowych dotyczących skanowania tylko w przypadku modyfikacji od momentu ostatniego skanowania obiektu. Nie będzie podlegać skanowaniu w przypadku aktualizacji oprogramowania do skanowania.

Jeśli wartość systemowa Używaj atrybutu "tylko w przypadku zmiany obiektów" do kontroli skanowania nie jest określona, atrybut tylko w przypadku zmiany obiektów nie będzie używany, a obiekt będzie podlegał skanowaniu po modyfikacji i gdy oprogramowanie do skanowania wskaże aktualizację.

#### Uwagi:

1. W zakładce tej dla plików można określić status skanowania obiektu.
2. Jeśli opcja Skanuj przy kolejnym dostępie po przywróceniu obiektu jest wybrana w wartościach systemowych, obiekt będzie podlegał skanowaniu co najmniej raz po przywróceniu.

#### Pojęcia pokrewne

“Obsługa skanowania” na stronie 21

System iSeries, umożliwia skanowanie obiektów zintegrowanego systemu plików.

#### Zadania pokrewne

“Tworzenie folderu” na stronie 127

Aby utworzyć folder, wykonaj następujące czynności.

---

## zdalne wywołanie procedury niezależne od transportów

Opracowane przez firmę Sun Microsystems wywołanie RPC (remote procedure call - zdalne wywołanie procedury) w prosty sposób oddziela aplikacje klienta od mechanizmu serwera.

Obejmuje ono standardową reprezentację danych o nazwie XDR (eXternal Data Representation)(która 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 między protokołami. Jednymi protokołami dostępnymi aktualnie w 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.

## Funkcje API wyboru sieci

Poniższe funkcje API zapewniają środki do wyboru transportu aplikacji.

Te funkcje API wymagają, aby w systemie istniał plik \*STMF /etc/netconfig. Jeśli w katalogu /etc nie ma pliku netconfig, użytkownik musi skopiować go z katalogu /QIBM/ProdData/OS400/RPC. Plik netconfig zawsze jest w katalogu /QIBM/ProdData/OS400/RPC.

funkcja API	Opis
endnetconfig()	Zwalnia wskaźnik do rekordów przechowywanych w pliku netconfig.
freenetconfig()	Zwalnia strukturę netconfig, która została zwrócona na skutek wywołania do funkcji getnetconfig().
getnetconfig()	Zwraca wskaźnik do bieżącego rekordu w pliku netconfig i ustawia wskaźnik na następny rekord.
getnetconfignt()	Zwraca wskaźnik do struktury netconfig, która odpowiada wejściowemu identyfikatorowi sieci (netid).
setnetconfig()	Ustawia wskaźnik rekordu na pierwszą pozycję pliku netconfig. Funkcji setnetconfig() należy używać przed pierwszym użyciem funkcji getnetconfig(). Funkcja setnetconfig() zwraca unikalny uchwyt (wskaźnik do rekordów przechowywanych w pliku netconfig), który ma zostać użyty przez funkcję getnetconfig().

#### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API konwersji nazwy na adres

Poniższe funkcje API umożliwiają aplikacji otrzymywanie adresu usługi lub podanego hosta w sposób niezależny od transportu.

<b>funkcja API</b>	<b>Opis</b>
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 netconfg.
netdir_options()	Stanowi interfejs dla funkcji transportu, takich jak adres rozgłośni i zarezerwowany port TCP i UDP.
netdir_spperror()	Zwraca komunikaty opisowe, określające dlaczego jedna z funkcji API konwersji nazwy na adres nie powiodła się.
taddr2uaddr()	Dokonuje konwersji adresu specyficznego dla danego transportu (lokalnego) na adres niezależny od danego transportu (uniwersalny).
uaddr2taddr()	Dokonuje konwersji adresu niezależnego od danego transportu (uniwersalnego) na adres specyficzny dla danego transportu (lokalny) (struktura netbuf).

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Zewnętrzna reprezentacja danych (XDR)

Poniższe funkcje API umożliwiają aplikacjom RPC obsługę dowolnych struktur danych, bez względu na różnice między hostami dotyczące uporządkowania bajtów lub sposobu rozmieszczenia składowych struktury.

<b>funkcja API</b>	<b>Opis</b>
xdr_array()	Filtr, który dokonuje konwersji między tablicami o zmiennej długości i odpowiadającymi im zewnętrznymi reprezentacjami. Funkcja ta jest wywoływana do kodowania i dekodowania każdego elementu tablicy.
xdr_bool()	Filtr, który dokonuje konwersji między zmiennymi typu Boolean (zmiennie całkowite w języku C) i ich zewnętrznymi reprezentacjami. Podczas kodowania danych filtr ten tworzy wartości równe 1 lub 0.
xdr_bytes()	Filtr, który dokonuje konwersji między tablicami bajtów i ich zewnętrznymi reprezentacjami. Funkcja ta operuje na tablicach, których elementy mają rozmiar 1 bajtu z wbudowanym zewnętrznym opisem każdego elementu. Długość ciągu bajtów jest jawnie podana jako liczba całkowita bez znaku (unsigned integer). Ciąg bajtów nie jest zakończony znakiem null. Zewnętrzna i wewnętrzna reprezentacja bajtów jest taka sama.
xdr_char()	Filtr, który dokonuje konwersji między znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_double()	Filtr, który dokonuje konwersji między liczbami o podwójnej precyzji języka C i ich zewnętrznymi reprezentacjami.
xdr_double_char()	Filtr, który dokonuje konwersji między dwubajtowymi znakami języka C i ich zewnętrznymi reprezentacjami.
xdr_enum()	Filtr, który dokonuje konwersji między liczbami typu wyliczeniowego (enum) języka C i ich zewnętrznymi reprezentacjami.
xdr_free()	Rekurencyjnie zwalnia obiekt wskazany przez przekazany wskaźnik.
xdr_float()	Filtr, który dokonuje konwersji między liczbami zmiennoprzecinkowymi (znormalizowane liczby zmiennoprzecinkowe pojedynczej precyzji) języka C i ich zewnętrznymi reprezentacjami.
xdr_int()	Filtr, który dokonuje konwersji między liczbami całkowitymi języka C i ich zewnętrznymi reprezentacjami.
xdr_long()	Filtr, który dokonuje konwersji między długimi liczbami całkowitymi (long integer) języka C i ich zewnętrznymi reprezentacjami.



<b>funkcja API</b>	<b>Opis</b>
xdr_netobj()	Filtr, który dokonuje konwersji między nieprzezroczystymi (opaque) danymi o zmiennej długości i ich zewnętrznymi reprezentacjami.
xdr_opaque()	Filtr podstawowy, który dokonuje konwersji między nieprzezroczystymi danymi o stałej długości i ich zewnętrznymi reprezentacjami.
xdr_pointer()	Zapewnia uchwycenie (chasing) wskaźnika wewnątrz struktur i szereguje wskaźniki null (dokonuje ich konwersji do postaci XDR). Może reprezentować rekurencyjne struktury danych, takie jak drzewa binarne lub listy.
xdr_reference()	Filtr, który zapewnia uchwycenie wskaźnika wewnątrz struktur. Umożliwia on szeregowanie, deszeregowanie i zwalnianie wskaźników wewnątrz jednej struktury, do której istnieją odniesienia z innej struktury. Funkcja xdr_reference() nie nadaje specjalnego znaczenia wskaźnikowi null podczas szeregowania i przesłanie adresu wskaźnika null może spowodować błąd pamięci. Dlatego programista musi opisywać dane za pomocą unii o dwóch składowych. Jedna składowa jest używana, gdy wskaźnik jest poprawny, druga, gdy wskazuje na wartość null.
xdr_short()	Filtr, który dokonuje konwersji między krótkimi liczbami całkowitymi (short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_string()	Filtr, który dokonuje konwersji między ciągami znaków języka C i ich zewnętrznymi reprezentacjami.
xdr_u_char()	Filtr, który dokonuje konwersji między znakami języka C (unsigned char) i ich zewnętrznymi reprezentacjami.
xdr_u_int()	Filtr, który dokonuje konwersji między liczbami całkowitymi bez znaku (unsigned integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_long()	Filtr, który dokonuje konwersji między długimi liczbami całkowitymi bez znaku (unsigned long integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_u_short()	Filtr, który dokonuje konwersji między krótkimi liczbami całkowitymi bez znaku (unsigned short integer) języka C i ich zewnętrznymi reprezentacjami.
xdr_union()	Filtr, który dokonuje konwersji między uniami języka C i ich zewnętrznymi reprezentacjami.
xdr_vector()	Filtr, który dokonuje konwersji między tablicami o stałej długości i odpowiadającymi im zewnętrznymi reprezentacjami.
xdr_void()	Bez parametrów. Jest przesyłana do innych funkcji RPC, które wymagają parametrów, ale nie przesyłają danych.
xdr_wrapstring()	Funkcja, która wywołuje funkcję API xdr_string(xdr, sp, maxuint), gdzie maxuint jest maksymalną wartością liczby całkowitej bez znaku. Funkcja xdr_wrapstring() jest użyteczna, ponieważ pakiet RPC przesyła maksymalnie dwie funkcje XDR jako parametry, a funkcja xdr_string() wymaga trzech.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Uwierzytelnianie funkcji API

Poniższe funkcje API uwierzytelniają aplikacje TI-RPC.

<b>funkcja API</b>	<b>Opis</b>
auth_destroy()	Niszczy strukturę informacji dotyczącej uwierzytelnienia, które zostało wskazane przez parametr auth.
authnone_create()	Tworzy i zwraca domyślny uchwyt zalegalizowania RPC, który za każdym zdalnym wywołaniem procedury przesyła puste (null) informacje dotyczące uwierzytelnienia.
authsys_create()	Tworzy i zwraca uchwyt uwierzytelnienia RPC, który zawiera informacje dotyczące uwierzytelnienia.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## 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.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Uprozczone funkcje API aplikacji TI-RPC

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

funkcja API	Opis
rpc_call()	Wywołuje zdalną procedurę w podanym systemie.
rpc_reg()	Rejestruje procedurę z pakietem usług RPC.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API poziomu najwyższego aplikacji TI-RPC

Poniższe funkcje API umożliwiają aplikacji podanie typu transportu.

funkcja API	Opis
clnt_call()	Wywołuje zdalną procedurę związaną z klientem.
clnt_control()	Zmienia informacje na temat obiektu klienta.
clnt_create()	Tworzy ogólny uchwyt klienta
clnt_destroy()	Niszczy uchwyt RPC klienta.
svc_create()	Tworzy uchwyt serwera.
svc_destroy()	Niszczy uchwyt transportu usługi RPC.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API poziomu pośredniego aplikacji TI-RPC

Poniższe funkcje API są podobne do funkcji API najwyższego poziomu, ale aplikacje użytkownika wybierają informacje specyficzne dla transportu za pomocą funkcji API służących do wyboru sieci.

funkcja API	Opis
clnt_tp_create()	Tworzy uchwyt klienta.
svc_tp_create()	Tworzy uchwyt serwera.

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Funkcje API poziomu najwyższego aplikacji TI-RPC

Poniższe funkcje API umożliwiają aplikacji wybór typu transportu. Oferują także zwiększony zakres sterowania szczegółami uchwytów CLIENT i SVCXPRT. Te funkcje API są podobne do funkcji API pośredniego poziomu z dodatkowym sterowaniem zapewnionym poprzez używanie funkcji API służących do konwersji nazwy na adres.

Używanie funkcji API służących do konwersji nazwy na adres zapewnia dodatkowe sterowanie.

funkcja API	Opis
clnt_tli_create()	Tworzy uchwyt klienta.
rpcb_getaddr()	Znajduje uniwersalny adres usługi.
rpcb_set()	Rejestruje adres serwera za pomocą RPCbind.
rpcb_unset()	Używana przez serwery do wyrejestrowania adresów.
svc_reg()	Wiąże program i jego wersję z programem szeregującym.
svc_tli_create()	Tworzy uchwyt serwera.
svc_unreg()	Usuwa dowiązanie ustawione za pomocą funkcji svc_reg().

### Informacje pokrewne

Narzędzie wyszukiwania funkcji API

## Inne funkcje API aplikacji TI-RPC

Te funkcje API w połączeniu z innymi funkcjami API (uproszczonymi, najwyższego, pośredniego i eksperckiego poziomu) umożliwiają różnym aplikacjom pracę.

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



### Informacje pokrewne





Narzędzie wyszukiwania funkcji API

## Informacje pokrewne dotyczące zintegrowanego systemu plików

Poniżej wymienione są podręczniki, serwisy WWW i tematy Centrum informacyjnego związane ze zintegrowanym systemem plików. Każdy z plików PSF można wyświetlić lub wydrukować.

### Podręczniki

-  i5/OS Network File System Support W podręczniku tym opisano system plików NFS z punktu widzenia serii powszechnie używanych aplikacji. Zawarto w nim 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 Publikacja ta służy jako przewodnik i podręcznik do obsługi nośników optycznych IBM w systemie i5/OS. 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: C/C++ Language Reference  Publikacja ta zawiera informacje niezbędne do projektowania, edycji, uzupełniania, uruchamiania i poprawiania programów ILE C/400 na serwerze iSeries.
- Security — Reference  Publikacja ta zawiera szczegółowe informacje techniczne o bezpieczeństwie systemu i5/OS, obejmujące wartości systemowe związane z bezpieczeństwem, które mają wpływ na przetwarzanie związane ze skanowaniem zintegrowanego systemu plików.
- APPC Programming  Publikacja ta zawiera informacje o zaawansowanej obsłudze 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  Podręcznik ten zawiera ogólne informacje o odtwarzaniu i opcjach dostępnych na serwerze IBM iSeries.

## Inne informacje

### • Raporty doświadczenia

Raporty te są pisane przez programistów firmy IBM, dokumentując ich bezpośrednie doświadczenia w realnym wprowadzaniu scenariuszy i rozwiązań. Służą do przekazywania doświadczeń programistów IBM z konkretnymi implementacjami rozwiązań serwera iSeries; zawierają instrukcje krok po kroku i wskazówki. Raport doświadczenia Tworzenie kopii zapasowej zintegrowanego systemu plików jest związany z plikami i systemami plików.


- Język CL
- Globalizacja systemu i5/OS
- Aplikacyjne interfejsy programistyczne (API)
- Zarządzanie kronikami
- Kontrola transakcji

## Zapisywanie plików PDF

Aby zapisać plik PDF na stacji roboczej do przeglądania i wydruku, należy:

1. Kliknąć prawym przyciskiem myszy plik PDF widoczny w przeglądarce (kliknąć prawym przyciskiem powyższy odsyłacz do pliku).
2. Kliknij opcję zapisania pliku PDF lokalnie.
3. Przejdź do katalogu, w ma zostać zapisany plik PDF.
4. Kliknąć opcję **Zapisz**.

## Pobieranie programu Adobe Reader

- Do wyświetlenia lub wydrukowania zawartości plików PDF niezbędny jest program Adobe Reader. Jego kopię można pobrać z serwisu WWW firmy Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

## Informacje dotyczące licencji na kod

IBM udziela niewyłącznej licencji na prawa autorskie, stosowanej przy używaniu wszelkich przykładowych kodów programów, na podstawie których można wygenerować podobne funkcje dostosowane do indywidualnych wymagań.

- Z UWZGLĘDNIENIEM BEZWZGLĘDNIEM OBOWIĄZUJĄCYCH GWARANCJI, KTÓRYCH NIE MOŻNA WYKLUCZYĆ, IBM, PROGRAMIŚCI I DOSTAWCY IBM NIE UDZIELAJĄ, W ZAKRESIE PROGRAMU CZY WSPARCIA TECHNICZNEGO, JAKIKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI) ANI NIE USTALAJĄ WARUNKÓW, ZARÓWNO WYRAŻNYCH JAK I DOMNIEMANYCH, A W SZCZEGÓLNOŚCI GWARANCJI CZY WARUNKÓW PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU CZY NIENARUSZANIA PRAW STRON TRZECICH.

| W ŻADNYM PRZYPADKU IBM, PROGRAMIŚCI CZY DOSTAWCY IBM NIE PONOSZĄ  
| ODPOWIEDZIALNOŚCI ZA NASTĘPUJĄCE STRATY LUB SZKODY, NAWET JEŚLI BYLIBY  
| POINFORMOWANI O MOŻLIWOŚCI ICH WYSTĄPIENIA:

- | 1. UTRATA LUB USZKODZENIE DANYCH;
- | 2. SZKODY BEZPOŚREDNIE, SZCZEGÓLNE, UBOCZNE, POŚREDNIE ORAZ SZKODY, KTÓRYCH NIE  
| MOŻNA BYŁO PRZEWIDZIEĆ PRZY ZAWIERANIU UMOWY; ORAZ
- | 3. UTRATA ZYSKÓW, KONTAKTÓW HANDLOWYCH, PRZYCHODÓW, REPUTACJI (GOODWILL) CZY  
| PRZEWIDYWANYCH OSZCZĘDNOŚCI.

| USTAWODAWSTWA NIEKTÓRYCH KRAJÓW NIE DOPUSZCZAJĄ WYŁĄCZENIA CZY OGRANICZENIA  
| ODPOWIEDZIALNOŚCI ZA SZKODY BEZPOŚREDNIE, UBOCZNE LUB SZKODY, KTÓRYCH NIE MOŻNA  
| BYŁO PRZEWIDZIEĆ PRZY ZAWIERANIU UMOWY, W ZWIĄZKU Z CZYM W ODNIESIENIU DO  
| NIEKTÓRYCH KLIENTÓW POWYŻSZE WYŁĄCZENIE LUB OGRANICZENIE (TAK W CAŁOŚCI JAK I W  
| CZĘŚCI) MOŻE NIE MIEĆ ZASTOSOWANIA.





---

## Dodatek. Uwagi

Niniejsze informacje zostały opracowane dla produktów i usług oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji, omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej IBM. Jednakże odpowiedzialność za weryfikację funkcjonalności wszelkich produktów, programów i usług innych dostawców spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Zapytania o licencje można wysyłać na adres:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Zapytania w sprawie licencji na informacje dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub zgłaszać na piśmie pod adresem:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokio 106-0032, Japonia

**Ponizszy paragraf nie dotyczy Wielkiej Brytanii, ani żadnego kraju, w którym zastrzeżenia te są niezgodne z lokalnym prawem:** FIRMA INTERNATIONAL BUSINESS MACHINES CORPORATION UDOSTĘPNIŁA NINIEJSZĄ PUBLIKACJĘ "W STANIE, W JAKIM SIĘ ZNAJDUJE", BEZ ŻADNYCH GWARANCJI, BEZPOŚREDNICH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI GWARANCJI NIENARUSZANIA PRAW, PRZYDATNOŚCI HANDLOWEJ LUB PRZYDATNOŚCI DO OKREŚLONEGO CELU. W niektórych krajach zastrzeżenia dotyczące wyraźnych lub domniemanych gwarancji w pewnych transakcjach są niezgodne z prawem, dlatego oświadczenie to nie musi dotyczyć użytkowników z tych krajów.

Niniejsze informacje mogą zawierać techniczne niedokładności lub błędy typograficzne. Okresowo wprowadzane są zmiany w tych informacjach. Zmiany te zostaną dołączone w nowych wydaniach publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie odsyłacze w niniejszych informacjach do serwisów WWW nie należących do firmy IBM zostały umieszczone jedynie dla wygody użytkownika i nie należy w żadnej mierze traktować ich jako akceptacji treści zamieszczonych w tych serwisach. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do korzystania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencje do programu dla osób, które szukają informacji dotyczących licencjonowania w celu: (i) wymiany informacji między niezależnie utworzonymi programami (włączając niniejszy program) i (ii) obustronnego wykorzystania wymienionych informacji, powinny nawiązać kontakt z następującą jednostką:

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Informacje te mogą być dostępne, z zastrzeżeniem odpowiednich warunków i uzgodnień, włączając w niektórych przypadkach także opłaty.

- | Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla
- | tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej
- | Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane o wydajności zawarte w niniejszym dokumencie zostały określone w kontrolowanym środowisku. Dlatego wyniki uzyskane w innych środowiskach pracy mogą się znacznie różnić. Niektóre pomiary mogły być wykonane w systemach na poziomie projektowym i nie gwarantuje się tożsamości pomiarów w systemach ogólnodostępnych. Co więcej, niektóre pomiary mogły być oszacowane przez ekstrapolację. Rzeczywiste wyniki mogą różnić się od podanych. Użytkownicy niniejszej dokumentacji powinni sprawdzić odpowiednie dane własnych środowisk.

Informacje dotyczące produktów firm innych niż IBM zostały uzyskane od dostawców tych produktów. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące parametrów produktów firm innych niż IBM należy kierować do odpowiednich producentów.

Wszystkie oświadczenia dotyczące przyszłych planów lub zamierzeń firmy IBM mogą się zmienić bez uprzedniego powiadomienia i są jedynie założonymi celami.

Niniejsze informacje zawierają przykłady danych i raportów używanych w codziennych operacjach. Aby w pełni je zilustrować, przykłady zawierają nazwiska osób, nazwy firm, marki i produkty. Wszystkie te nazwy są fikcyjne i jakiegokolwiek podobieństwo do nazw i adresów użytych w rzeczywistej działalności przedsiębiorstw są całkowicie przypadkowe.

#### LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsze informacje zawierają przykładowe programy aplikacji w języku źródłowym, które ilustrują techniki programistyczne dla różnych platform systemowych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programowym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Przykłady te nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów.

Każda kopia lub część tych przykładowych programów lub ich pochodna musi zawierać następujące uwagi dotyczące praw autorskich:

© (nazwa przedsiębiorstwa użytkownika, rok). Fragmenty tego kodu pochodzą z © Copyright IBM Corp. (wpisać rok lub lata). Wszelkie prawa zastrzeżone.

Jeśli informacje te są przeglądane elektronicznie, fotografie i kolorowe ilustracje mogą być niewidoczne.

---

## Informacje o interfejsie programistycznym

Niniejsza publikacja opisuje planowane interfejsy programistyczne, pozwalające na pisanie programów umożliwiających korzystanie z usług systemu operacyjnego IBM i5/OS.

---

## Znaki handlowe

Następujące nazwy są znakami towarowymi firmy International Business Machines Corporation w Stanach Zjednoczonych i/lub w innych krajach:

- | C/400
- | DB2
- | i5/OS
- | IBM
- | IBM (logo)
- | Integrated Language Environment
- | iSeries
- | NetServer
- | OfficeVision
- | OS/2
- | OS/400
- | WebSphere
- | xSeries

Microsoft, Windows, Windows NT oraz logo Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

- | Linux jest znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym Open Group w Stanach Zjednoczonych i w innych krajach.

Nazwy innych przedsiębiorstw, produktów i usług mogą być znakami towarowymi lub znakami usług innych podmiotów.

---

## Warunki

Zezwolenie na korzystanie z tych publikacji jest przyznawane na poniższych warunkach.

**Użytek osobisty:** Użytkownik ma prawo kopiować te publikacje do własnego, niekomercyjnego użytku pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa dystrybuować ani wyświetlać tych publikacji czy ich części, ani też wykonywać na ich podstawie prac pochodnych bez wyraźnej zgody IBM.

**Użytek służbowy:** Użytkownik ma prawo kopiować te publikacje, dystrybuować je i wyświetlać wyłącznie w ramach przedsiębiorstwa Użytkownika pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa wykonywać na podstawie tych publikacji ani ich fragmentów prac pochodnych, kopiować ich, dystrybuować ani wyświetlać poza przedsiębiorstwem Użytkownika bez wyraźnej zgody IBM.

Z wyjątkiem zezwoleń wyraźnie udzielonych w niniejszym dokumencie, nie udziela się jakichkolwiek innych zezwoleń, licencji ani praw, wyraźnych czy domniemanych, odnoszących się do tych publikacji czy jakichkolwiek informacji, danych, oprogramowania lub innej własności intelektualnej, o których mowa w niniejszym dokumencie.

IBM zastrzega sobie prawo do anulowania zezwolenia przyznanego w niniejszym dokumencie w każdej sytuacji, gdy, według uznania IBM, korzystanie z tych publikacji jest szkodliwe dla IBM lub jeśli IBM uzna, że warunki niniejszego dokumentu nie są przestrzegane.

Zabrania się pobierania, eksportowania lub reeksportowania niniejszych informacji inaczej niż w pełni zgodnie z wszelkimi prawami i regulacjami znajdującymi tu zastosowanie, włączając wszystkie prawa i regulacje eksportowe Stanów Zjednoczonych.

IBM NIE UDZIELA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, DOTYCZĄCYCH TREŚCI TYCH PUBLIKACJI. PUBLIKACJE TE SĄ DOSTARCZANE W STANIE, W JAKIM SIĘ ZNAJDUJĄ ("AS IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, WYRAŻNYCH CZY DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ NIENARUSZANIA PRAW STRON TRZECICH.



**IBM**