

IBM

@server

iSeries

Zintegrowany system plików

Wersja 5 wydanie 3





@server

iSeries

Zintegrowany system plików

Wersja 5 wydanie 3

Uwaga

Przed użyciem niniejszych informacji i produktu, którego dotyczą, należy przeczytać informacje znajdujące się w sekcji “Uwagi”, na stronie 119.

Wydanie szóste (sierpień 2005)

| Wydanie to dotyczy wersji 5, wydania 3, modyfikacji 0 systemu operacyjnego IBM Operating System/400 (numer produktu
| 5722–SS1) i wszystkich kolejnych wydań i modyfikacji, jeśli w nowych wydaniach nie zostanie stwierdzone inaczej. Wersja ta nie
| działa we wszystkich modelach komputerów typu RISC ani modelach CISC.

© Copyright International Business Machines Corporation 1999, 2005. Wszelkie prawa zastrzeżone.

Spis treści

| | | | |
|---|----------|--|------------|
| Zintegrowany system plików | 1 | Przeгляд kronikowania | 82 |
| Informacje dotyczące kodu | 1 | Uruchomienie kronikowania | 86 |
| Co nowego w V5R3 | 2 | Zmiana kronikowania | 86 |
| Drukowanie tego dokumentu | 3 | Zakończenie kronikowania | 86 |
| Przeгляд zintegrowanego systemu plików | 3 | Obsługa programowania | 86 |
| Czym jest zintegrowany system plików | 3 | Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych | 87 |
| Dlaczego warto korzystać ze zintegrowanego systemu plików | 4 | Kopiowanie danych pomiędzy plikami strumieniowymi a zbiorami składowania | 90 |
| Pojęcia | 5 | Wykonywanie operacji za pomocą funkcji API | 91 |
| Katalog | 5 | Obsługa gniazd | 100 |
| Dowiązanie | 10 | Nazewnictwo i obsługa różnych języków | 100 |
| Nazwa ścieżki | 14 | Konwersja danych | 100 |
| Plik strumieniowy | 14 | Przykład: funkcje C zintegrowanego systemu plików | 101 |
| Niezmienność nazw | 16 | Praca z plikami i folderami z wykorzystaniem produktu iSeries Navigator | 106 |
| Atrybuty rozszerzone | 16 | Odblokowanie pliku | 107 |
| Obsługa skanowania | 17 | Zablokowanie pliku | 107 |
| Praca z systemami plików | 22 | Tworzenie folderu | 107 |
| Porównanie systemów plików | 23 | Usuwanie folderu | 107 |
| Bazowy system plików "root" (/) | 27 | Przenoszenie plików lub folderów do innego systemu plików | 108 |
| System plików systemów otwartych (QOpenSys) | 29 | Nadawanie uprawnień | 109 |
| System plików definiowany przez użytkownika (UDFS) | 30 | Konfigurowanie konwersji plików tekstowych | 109 |
| System plików bibliotek (QSYS.LIB) | 35 | Wysyłanie pliku lub folderu do innego systemu | 109 |
| System plików usług biblioteki dokumentów (QDLS) | 40 | Zmiana opcji definicji pakietów | 110 |
| System plików nośników optycznych (QOPT) | 42 | Planowanie daty i godziny wysyłki pliku lub folderu | 110 |
| System plików NetWare (QNetWare) | 44 | Tworzenie współużytkowanego zasobu plikowego | 110 |
| System plików iSeries NetClient (QNTC) | 47 | Zmiana współużytkowanego zasobu plikowego | 110 |
| System plików serwera plików OS/400 (QFileSvr.400) | 50 | Włączanie i wyłączanie skanowania obiektów | 111 |
| Sieciowy system plików (NFS) | 54 | Informacje pokrewne dotyczące zintegrowanego systemu plików | 111 |
| Dostęp do zintegrowanego systemu plików | 56 | Wywołanie TI-RPC (Transport Independent Remote Procedure Call) | 112 |
| Dostęp przy użyciu pozycji menu i ekranów | 57 | Odniesienia | 116 |
| Dostęp przy użyciu komend języka CL | 58 | Raporty doświadczenia | 117 |
| Dostęp przy użyciu funkcji API | 74 | Dodatek. Uwagi | 119 |
| Dostęp przy użyciu iSeries Navigator | 74 | Informacje o interfejsie programistycznym | 121 |
| Dostęp przy użyciu iSeries NetServer | 75 | Znaki handlowe | 121 |
| Dostęp przy użyciu programów przesyłania plików | 76 | Warunki dotyczące pobierania i drukowania publikacji | 121 |
| Dostęp przy użyciu komputera PC | 77 | | |
| Przekształcanie katalogów z formatu *TYPE1 na *TYPE2 | 77 | | |
| Przeгляд konwersji formatu *TYPE1 do *TYPE2 | 78 | | |
| Wskazówki: konwersje | 78 | | |
| Kronikowanie obiektów | 81 | | |

Zintegrowany system plików

Zintegrowany system plików jest częścią systemu OS/400, 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.

Szczegółowe informacje na temat zintegrowanego systemu plików w systemie OS/400 zawierają następujące tematy.

Co nowego w V5R3

Zmiany wprowadzone od ostatniego wydania.

Drukowanie tego dokumentu

Jak wyświetlać i drukować wersję PDF tego tematu.

Przegląd zintegrowanego systemu plików

Podstawy zintegrowanego systemu plików.

Pojęcia

Informacje o różnych pojęciach dotyczących zintegrowanego systemu plików.

Praca z systemami plików

Jak zarządzać różnymi systemami plików istniejącymi w systemie OS/400.

Dostęp do zintegrowanego systemu plików

Różne opcje dostępu do zintegrowanego systemu plików. Obejmują: używanie menu i ekranów, komend języka CL, funkcji API iSeries NetServer, iSeries Navigator, i programów do przesyłania plików.

Przekształcanie katalogów z formatu *TYPE1 na *TYPE2

Opis automatycznego przekształcania katalogów z formatu *TYPE1 do *TYPE2.

Kronikowanie obiektów

Jak uruchomić, zakończyć i zmienić kronikowanie obiektów zintegrowanego systemu plików.

Obsługa programowania

Jak kopiować dane, używać funkcji API i wykonywać inne funkcje obsługi.

Praca z plikami i folderami z wykorzystaniem produktu iSeries Navigator

Przeprowadzanie zadań na plikach i folderach, takich jak wprowadzanie i usuwanie plików oraz tworzenie i usuwanie folderów.

Informacje pokrewne dotyczące zintegrowanego systemu plików

Jak znaleźć informacje w Centrum informacyjnym iSeries i sieci Internet opisujące dodatkowe zagadnienia dotyczące zintegrowanego systemu plików.

Uwaga: Temat "Informacje dotyczące kodu" zawiera istotne informacje prawne.

Informacje dotyczące kodu

Ten dokument zawiera przykłady programów.

IBM udziela niewyłącznej licencji na prawa autorskie do używania wszystkich przykładów kodów programowania, z których można generować podobne funkcje dostosowane do konkretnych potrzeb użytkownika.

Cały kod przykładowy jest udostępniony przez firmę IBM tylko w celu zilustrowania przykładu. Przykłady te nie zostały gruntownie przetestowane. Dlatego firma IBM nie gwarantuje ani nie sugeruje niezawodności, przydatności ani funkcjonalności tych programów.

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

Co nowego w V5R3

Poniższe informacje zostały dodane lub zaktualizowane w niniejszym wydaniu dokumentacji:

Funkcje API zintegrowanego systemu plików

Zintegrowany system plików zawiera nowe funkcje i obsługę następujących funkcji API:

- Usuwanie zawartości pliku (Clear a file - fclear())
- Usuwanie zawartości pliku dla dużych plików (Clear a file - fclear64())
- Zmiana sygnatury skanowania (Change Scan Signature - QP0LCHSG)
- Pobranie obiektów odniesienia (Retrieve Referenced Objects - QP0LRRO)
- Pobranie sygnatury skanowania (Retrieve Scan Signature - QP0LRTSG)

Komendy CL zintegrowanego systemu plików

W wersji V5R3 dodano funkcje i obsługę następujących komend CL w zintegrowanym systemie plików:

- Komenda Zmiana kronikowanego obiektu (Change Journalled Object - CHGJRNOBJ)
- Komenda Wydruk informacji katalogowych (Print Directory Information - PRDIRINF)
- Komenda Pobranie informacji katalogowych (Retrieve Directory Information - RTVDIRINF)

Informacje o danych wyjściowych komend PRDIRINF i RTVDIRINF zawiera sekcja "Praca z danymi wyjściowymi komend RTVDIRINF i PRDIRINF" na stronie 63.

Dwa sterowniki urządzeń

System plików "root" (/) w katalogu /dev/xti zawiera teraz dwa sterowniki urządzeń o nazwie udp i tcp. Sterowniki urządzeń opisane są w następującym rozdziale:

- "Urządzenia UDP i TCP w systemie plików "root" (/)." na stronie 29

Konfigurowanie protokołu Kerberos dla systemu plików QNTC

W poprzednich wydaniach system plików QNTC używał hasła do uwierzytelniania się w serwerach zdalnych. Aby uprościć administrowanie hasłami, do uwierzytelnień w serwerach zdalnych system plików QNTC będzie używał biletów serwera kerberos. Więcej informacji na temat konfigurowania tego protokołu w systemie plików QNTC zawiera sekcja "Włączanie korzystania z protokołu Kerberos w systemie plików QNTC" na stronie 50.

Automatyczna konwersja katalogów do formatu *TYPE2

Bezpośrednio po zainstalowaniu systemu w wersji V5R3 system rozpocznie automatyczną konwersję katalogów do formatu *TYPE2 dla wszystkich systemów plików, które aktualnie obsługują format *TYPE1. Więcej informacji znajduje się w sekcji "Przekształcanie katalogów z formatu *TYPE1 na *TYPE2" na stronie 77.

Obsługa skanowania

iSeries wyposażony jest teraz w opcję zezwalania na skanowanie obiektów zintegrowanego systemu plików. Skanowanie może wystąpić w trakcie operacji otwarcia i zamknięcia obiektu. Obiekty można skanować z wielu

- | powodów, takich jak kontrola antywirusowa. Można także wybrać, kiedy skanowanie będzie przeprowadzane, zależnie od potrzeb. Dodatkowe informacje na temat obsługi skanowania i wymagań zawiera następujący temat:
- | • “Obsługa skanowania” na stronie 17

Drukowanie tego dokumentu


Aby przejrzeć lub pobrać ten dokument w formacie PDF, kliknij Zintegrowany system plików  (około 1455 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. Jeśli używany jest Internet Explorer, kliknąć opcję **Zapisz jako....** Jeśli używany jest Netscape Communicator, kliknąć opcję **Zapisz odsyłacz jako....**
3. Wybrać katalog, w którym plik PDF zostanie zapisany.
4. Kliknąć opcję **Zapisz.**

Pobranie aplikacji Adobe Acrobat Reader

Do przeglądania i drukowania plików PDF niezbędna jest aplikacja Adobe Acrobat Reader. Kopię tej aplikacji można pobrać z serwisu firmy Adobe (www.adobe.com/products/acrobat/readstep.html). 

Przegląd zintegrowanego systemu plików

Następujące tematy opisują zintegrowany system plików w serwerze iSeries i ilustrują, jak można go używać.

- “Czym jest zintegrowany system plików”
- “Dlaczego warto korzystać ze zintegrowanego systemu plików” na stronie 4

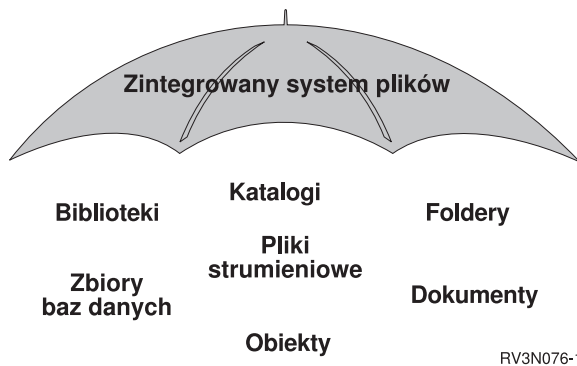
Czym jest zintegrowany system plików

Zintegrowany system plików jest częścią systemu OS/400 i 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. Więcej informacji znajduje się w temacie Praca z systemami plików.

Oto najważniejsze cechy zintegrowanego systemu plików:

- Obsługa informacji przechowywanych w plikach strumieniowych; mogą to być długie, nieprzerwane ciągi danych. Takim ciągiem może być, na przykład, tekst dokumentu lub elementy obrazu. Obsługa plików strumieniowych służy do lepszego wykorzystania aplikacji klient/serwer.
- Hierarchiczna struktura katalogów, w której obiekty mogą być przedstawione jako owoce na gałęziach drzewa. Obiekt jest dostępny, kiedy zostanie podana ścieżka do niego zawierająca wszystkie katalogi.
- Wspólny interfejs umożliwiający użytkownikom i aplikacjom dostęp nie tylko do plików strumieniowych, ale także do zbiorów baz danych, dokumentów i innych obiektów przechowywanych na serwerze.
- 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 Local Area Network (LAN), Novell NetWare, innym zdalnym serwerze iSeries lub serwerze Network File System.



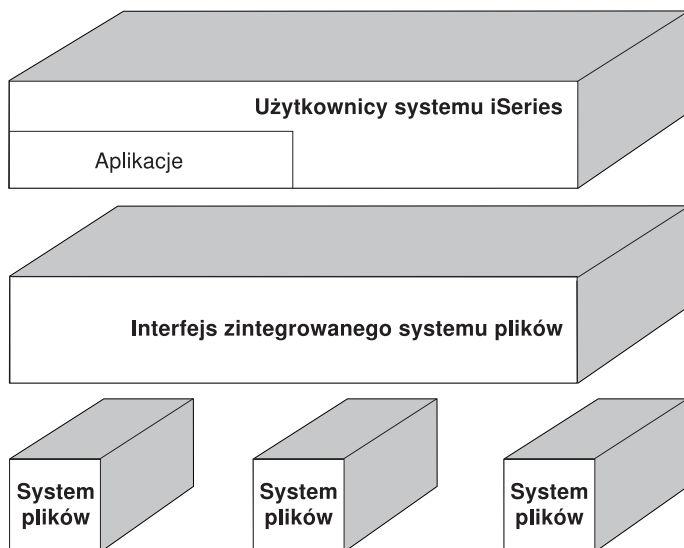
Rysunek 1. Ogólne informacje o strukturze informacji przechowywanych w serwerze iSeries

Dlaczego warto korzystać ze zintegrowanego systemu plików

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

Zintegrowany system plików pozwala na:

- Szybszy dostęp do danych w systemie OS/400, szczególnie dla takich aplikacji, jak Client Access, które wykorzystują serwer plików OS/400.
- Efektywniejszą obsługę danych strumieniowych, takich jak obrazy, dźwięk i wideo.
- Udostępnienie platformy systemu plików i katalogów dla obsługi standardu systemów otwartych bazujących na systemie UNIX, takich jak POSIX i XPG. Ta struktura plików i katalogów 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

W sekcji tej omówione zostały pojęcia dotyczące zintegrowanego systemu plików. Aby zrozumieć, jak serwer zintegrowanego systemu plików działa w danym systemie, należy przeczytać odpowiedni rozdział.

- “Katalog”
- “Dowiązanie” na stronie 10
- “Nazwa ścieżki” na stronie 14
- “Plik strumieniowy” na stronie 14
- “Niezmienność nazw” na stronie 16
- “Atrybuty rozszerzone” na stronie 16
- “Obsługa skanowania” na stronie 17

Katalog

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

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

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

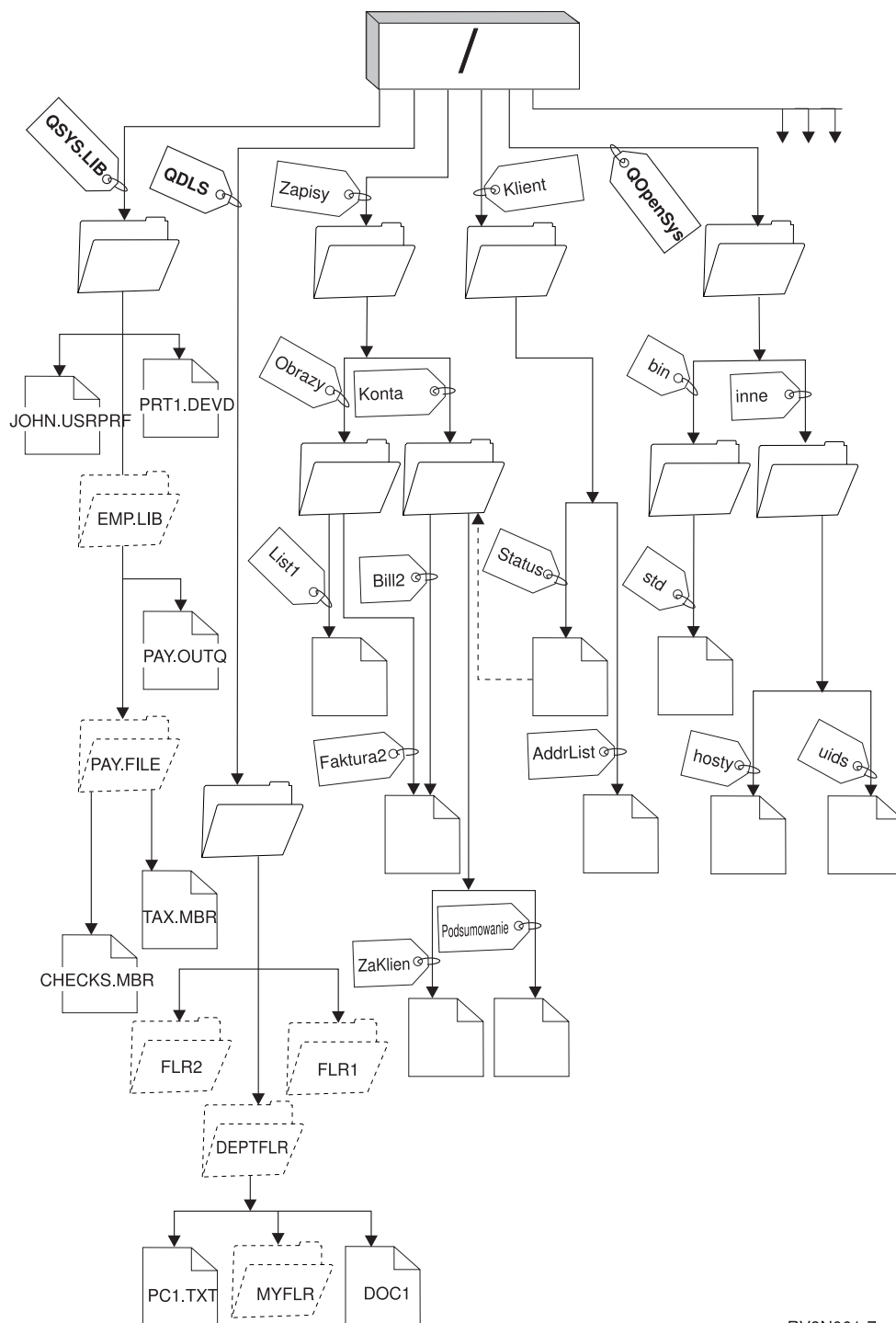
Ze względu na różnice w systemach plików pewne operacje można wykonywać tylko w niektórych poddrzewach struktury katalogów.

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

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

- “Katalog bieżący” na stronie 6
- “Katalog osobisty” na stronie 7
- “Utworzone katalogi” na stronie 7

- “Katalogi *TYPE2” na stronie 9



RV3N061-7

Rysunek 3. Systemy plików i ich obiekty stanowią gałęzie drzewa katalogów zintegrowanego systemu plików.

Katalog bieżący

Katalog bieżący (current directory) jest pierwszym katalogiem, w którym system operacyjny szuka programów oraz plików i w którym przechowuje tymczasowe pliki i wydruki. Gdy użytkownik żąda wykonania czynności na obiekcie

takim jak zbiór, system szuka obiektu w bieżącym katalogu użytkownika, chyba że użytkownik poda inną ścieżkę katalogów. Katalog bieżący pełni podobną rolę jak biblioteka bieżąca. Nazywa go się także **bieżącym katalogiem roboczym** lub po prostu **katalogiem roboczym**.

Katalog osobisty

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. 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.

Aby przejrzeć szczegóły dotyczące funkcji API QWTCHGJB, należy przeczytać rozdział Aplikacyjne interfejsy programistyczne (API).

Utworzone katalogi

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

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

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

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

Aplikacje mogą używać do zarządzania katalogiem /tmp i zawartymi w nim plikami funkcji API, które obsługują zintegrowany system plików (patrz sekcja "Dostęp przy użyciu funkcji API" na stronie 74). Na przykład program użytkowy może usunąć plik z katalogu /tmp używając funkcji API `unlink()`.

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

- Katalog `/tmp` tworzony jest teraz z włączonym atrybutem "ograniczenie zmiany nazwy i usunięcia dowiązania" (`restrict rename and unlink`). Do zmiany tego atrybutu można użyć komendy Zmiana atrybutu (`Change Attribute`) `CHGATR`. Użytkownicy mogą wyłączyć ten atrybut, ale po następnym uruchomieniu systemu jest on automatycznie włączany. Gdy atrybut ten jest włączony, katalog `/tmp` zawiera następujące ograniczenie:
- Zmiany nazw i usuwanie dowiązań są ograniczone do obiektów znajdujących się w tym katalogu. Obiekty można dowiązać do katalogu, ale nie można zmienić ich nazw, ani odłączyć ich, dopóki jeden z następujących warunków nie zostanie spełniony dla użytkownika wykonującego operację:
 1. Użytkownik jest właścicielem obiektu.
 2. Użytkownik jest właścicielem katalogu.
 3. Użytkownik posiada uprawnienia specjalne `*ALLOBJ`.
- /home** Administratorzy systemu tworzą w katalogu `/home` katalogi osobiste wszystkich użytkowników. Administrator systemu zwykle zakłada użytkownikowi katalog osobisty o nazwie zgodnej z profilem użytkownika, na przykład `/home/jasio`. Więcej informacji znajduje się w sekcji "Katalog osobisty" na stronie 7.
- /etc** W katalogu `/etc` przechowywane są pliki używane do administrowania systemem, pliki konfiguracyjne i inne pliki systemowe.
- /usr** Katalog `/usr` zawiera podkatalogi z informacjami używanymi przez system. Zazwyczaj pliki w katalogu `/usr` są rzadko zmieniane.
- /usr/bin**
Katalog `/usr/bin` zawiera standardowe programy narzędziowe.
- /QIBM**
Katalog `/QIBM` jest katalogiem systemowym i jest oferowany z systemem.
- /QIBM/ProdData**
Katalog `/QIBM/ProdData` jest katalogiem systemowym 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. Więcej informacji znajduje się w sekcji
| “Urządzenia UDP i TCP w systemie plików ”root” (/).” na stronie 29.

Katalogi *TYPE2

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

Zalety katalogów *TYPE2 są następujące:

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

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

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

Katalogi *TYPE2 oferują dodatkowo następujące funkcje:

1. Katalogi *TYPE2 obsługują zmianę wielkości liter w nazwie zapisanej w systemie plików stosującym wielkie litery (na przykład zmianę A na a).
2. Obiekt w katalogu *TYPE2 może mieć do miliona dowiązań, natomiast w katalogach *TYPE1 liczba ta wynosi 32767 dowiązań. Oznacza to, że dla pliku strumieniowego można zdefiniować do 1 miliona dowiązań stałych, a katalog *TYPE2 może zawierać do 999 998 podkatalogów.
3. Podczas korzystania z programu iSeries Navigator, podczas otwierania katalogu w formacie *TYPE2 lista pozycji jest automatycznie sortowana w porządku binarnym.
4. 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 mają fabrycznie instalowane katalogi *TYPE2 OS/400 w wersji V5R2 lub nowszej. 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 lub nowszej w serwerowi 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 użyto narzędzia 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 lub nowszej.
- Pozostałe obsługiwane systemy plików z wyjątkiem UDFS w niezależnej puli ASP, które używają katalogów *TYPE1, zostaną automatycznie poddane konwersji. Rozpocznie się ona po zainstalowaniu wersji V5R3 i nie powinna poważnie wpłynąć na działanie systemu. Dodatkowe informacje zawiera temat “Przekształcanie katalogów z formatu *TYPE1 na *TYPE2” na stronie 77.

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 V5R1 lub nowszej, ale zawierają pewne różnice względem normalnej obsługi katalogów *TYPE2. Więcej informacji na ten temat zawiera sekcja Używanie katalogów *TYPE2 w systemach OS/400 w wersji V5R1 lub V5R2.

Więcej informacji na temat katalogów *TYPE2 zawiera sekcja “Przekształcanie katalogów z formatu *TYPE1 na *TYPE2” na stronie 77.

l **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 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.

l Jeżeli system działa w wersji V5R1 lub V5R2, można poddać katalogi konwersji do formatu katalogów *TYPE2 używając odpowiedniego narzędzia do konwersji. Bezpośrednio po zainstalowaniu systemu w wersji V5R3 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. Więcej informacji można znaleźć w następujących tematach: “Przekształcanie katalogów z formatu *TYPE1 na *TYPE2” na stronie 77.

l Obsługa katalogów *TYPE2 w wersji V5R2 jest dostępna w Centrum informacyjne iSeries V5R2 poprzez komendę Konwersja katalogu (Convert Directory - CVTDIR).

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

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

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

l 2. Aby przejrzeć APAR informacyjny elektronicznie, należy sprawdzić serwis WWW

l <http://www.ibm.com/eserver/iseries/support/supporthome.nsf/document/1000045>  . Następnie należy wybrać pozycje: **Problem Solving** —> **Technical Databases** —> **Authorized Program Analysis Reports (APARs)** —> **V5R1 APARs** —> **APAR number II13161**.

Dowiązanie

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

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

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 sekcja “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 “Dowiązanie symboliczne” na stronie 12. 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. |

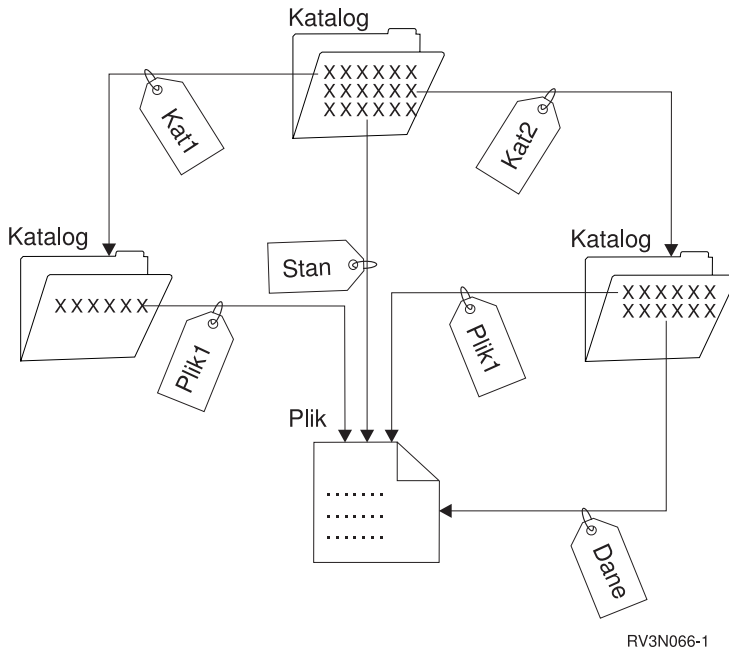
Informacje o dowiązaniach zawierają następujące sekcje zamieszczone w dalszej części tego dokumentu:

- Dowiązanie stałe
- Dowiązanie symboliczne

Dowiązanie stałe

Dowiązanie stałe (hard link), zwane po prostu dowiązaniem, nie może istnieć bez przyłączenia do rzeczywistego obiektu. Gdy obiekt zostanie utworzony w katalogu (na przykład poprzez skopiowanie pliku do katalogu), zostanie ustalone pierwsze dowiązanie stałe pomiędzy katalogiem a obiektem. Użytkownicy i programy użytkowe mogą dodawać inne dowiązania stałe. Każde dowiązanie stałe jest wskazywane przez osobną pozycję w katalogu. Dowiązania z tego samego katalogu nie mogą mieć tej samej nazwy, ale dowiązania z różnych katalogów - mogą.

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



Rysunek 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 zostaje usunięty z serwera. Obiektu nie można otworzyć po usunięciu ostatniego dowiązania stałego.

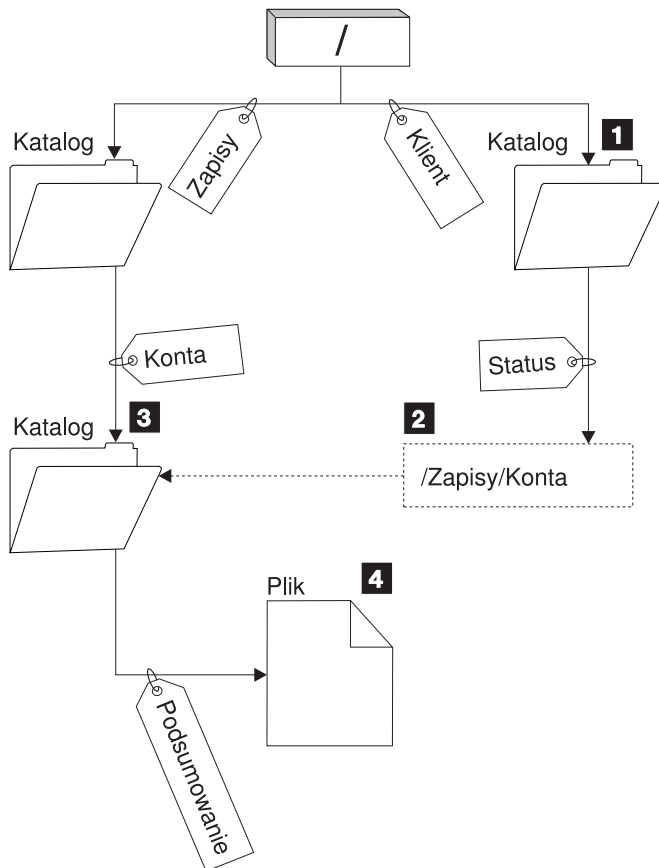
Pojęcie dowiązania stałego, z pewnymi ograniczeniami, może się także odnosić do systemu plików QSYS.LIB lub QSYS.LIB niezależnej puli ASP i do systemu plików obsługującego obiekty biblioteki dokumentów (QDLS). Tak więc biblioteka ma jedno dowiązanie stałe do każdego zawartego w niej obiektu. Podobnie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

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

Dowiązanie symboliczne

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

Rozważmy poniższy przykład sposobu użycia dowiązania symbolicznego:



RV3N068-1

Rysunek 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, QSYS.LIB w niezależnej puli ASP i 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 QDLS.

Patrz także sekcja Tabela 1 na stronie 11.

Nazwa ścieżki

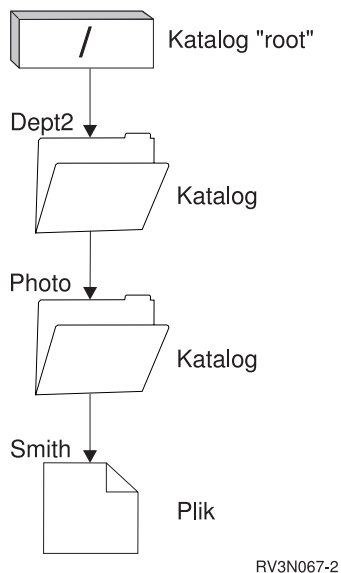
Nazwa ścieżki (path name) informuje serwer, jak znaleźć dany obiekt. Jest to sekwencja nazw katalogów zakończona nazwą obiektu. Poszczególne nazwy katalogów i obiektów rozdziela znak ukośnika (/); na przykład:

katalog1/katalog2/plik

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

Ścieżkę określa się na dwa sposoby:

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



Rysunek 6. Składniki ścieżki

Ścieżka bezwzględna do pliku Smith wygląda następująco:

/Dept2/Photo/Smith

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

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

Photo/Smith

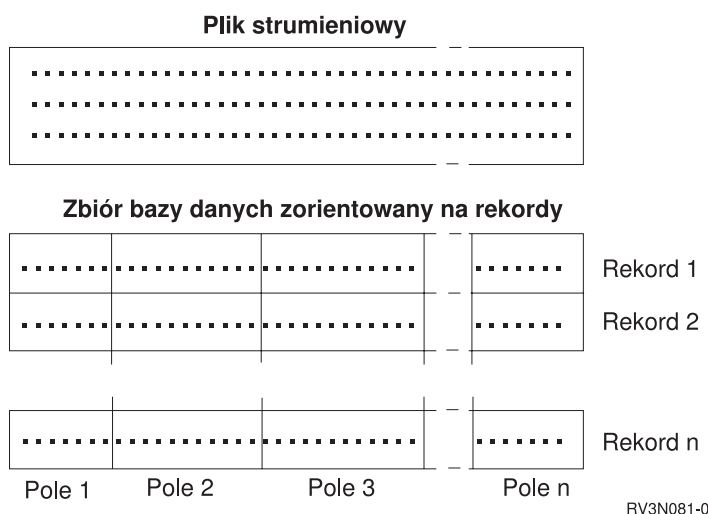
Zauważ, że ścieżka nie zawiera nazwy bieżącego katalogu. Pierwszą pozycję nazwy stanowi katalog lub obiekt znajdujący się na *kolejnym niższym poziomie* względem katalogu bieżącego.

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.

Więcej informacji o plikach strumieniowych w zintegrowanym systemie plików zawierają tematy:

- “Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych” na stronie 87.
- “Pliki strumieniowe *TYPE1”.
- “Pliki strumieniowe *TYPE2”.

Pliki strumieniowe *TYPE1

Plik strumieniowy *TYPE1 ma ten sam format, co pliki strumieniowe utworzone w wydaniach poprzedzających wersję 4, wydanie 4 systemu OS/400. Jego minimalna wielkość to 4096 bajtów. Pliki strumieniowe *TYPE1 mają przypisaną maksymalną wielkość obiektu około 256 gigabajtów.

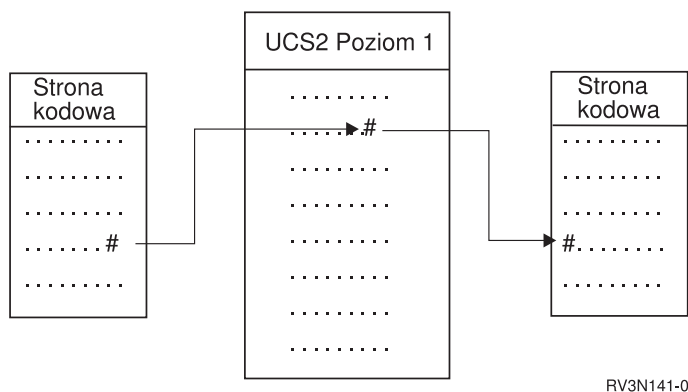
Pliki strumieniowe *TYPE2

Plik strumieniowy *TYPE2 charakteryzuje się wysoką wydajnością dostępu do zbioru i jest nowością w wersji 4 wydaniu 4 systemu OS/400. W wersji V5R3 pliki strumieniowe *TYPE2 mają maksymalną wielkość obiektu około jednego terabajta w systemach plików "root" (/), QOpenSys i użytkownika. W pozostałych przypadkach wielkość maksymalna wynosi około 256 gigabajtów. Możliwe jest także odwzorowywanie pamięci i określanie atrybutów optymalizujących przydzielanie pamięci głównej. Wszystkie pliki utworzone w systemach w wersji V4R4 i nowszych 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 starszych niż V5R3.

Niezmiennosc nazw

Uzytkownicy "bazowego" (/) systemu plikow, systemu plikow QOpenSys i systemu plikow uzytkownika korzystaja z funkcji systemowej zapewniajacej niezmiennosc znakow w nazwach obiektow. W szczegolnosci dotyczy to uzycia tych systemow plikow na serwerach iSeries i podlaczonych urzadzeniach o roznych schematach kodowania znakow (stronach kodowych). Serwer przechowuje znaki w nazwach w formacie 16-bitowym znanym jako UCS2 Poziom 1 (nazywanym takze **Unicode**) dla katalogow *TYPE1 i UTF-16 dla katalogow *TYPE2. Wiecej informacji o formatach katalogow mozna znalezc w sekcji Katalogi *TYPE2. UCS2 Poziom 1 i UTF-16 sa podzbiorem standardu ISO 10646. Gdy jest uzywana nazwa, system przekszalca przechowywany format znakow do odpowiedniej reprezentacji znakow w aktualnie uzywanej stronie kodowej. Nazwy rozszerzonych atrybutow zwiazanych z kazdym obiektem traktowane sa w taki sam sposob.



Rysunek 8. Utrzymywanie niezmiennosci znakow w roznych schematach kodowania

Obsluga ta ulatwia wspoldzialanie serwera z urzadzeniami uzywajacymi innych stron kodowych. na przyklad, uzytkownicy komputerow PC moga uzyskac dostep do pliku serwera iSeries uzywajac tej samej nazwy pliku, pomimo ze ich komputery PC nie uzywaja tej samej strony kodowej, co serwer. Konwersja stron kodowych jest wykonywana automatycznie przez serwer. Oczywiscie, urzadzenie musi uzywac strony kodowej zawierajacej znaki wykorzystywane w danej nazwie.

Atrybuty rozszerzone

Atrybut rozszerzony (extended attribute) jest informacja zwiazana z obiektem, ktora zawiera dodatkowe szczegoly na temat tego obiektu. Atrybut rozszerzony sklada sie z nazwy, budej odnoznikiem do atrybutu, i wartosci. Wartością moze byc tekst, dane binarne lub dane innego typu.

Atrybuty rozszerzone dla obiektu istnieja tylko tak dlugo, jak dlugo istnieje dany obiekt.

Atrybuty rozszerzone (EA) wystepuja w wielu odmianach i moga byc uzywane do przechowywania roznych informacji. Nalezzy znac w szczegolnosci nastepujace trzy atrybuty rozszerzone:

.SUBJECT

Krotki opis zawartosci i przeznaczenia obiektu.

.TYPE Typ danych w obiekcie. Typem tym moze byc tekst, dane binarne, kod zrodlowy programu, skompilowany program lub inne informacje.

.CODEPAGE

Strona kodowa, ktora ma zostac uzyta przy pracy z obiektem. Strona kodowa uzywana dla obiektu jest wykorzystywana rowniez dla atrybutow rozszerzonych zwiazanych z obiektem.

Kropka (.) jako pierwszy znak nazwy oznacza, ze atrybuty te sa 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 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 komendy Praca z dowiązaniami obiektów (Work with Object Links - WRKLNK). W zintegrowanym systemie plików nie istnieje żadna inna metoda dająca aplikacjom i użytkownikom dostęp do atrybutów rozszerzonych. Jedynym odstępstwem od tej reguły są następujące komendy CL: Wyświetlenie danych podłączonego FS (Display a UDFS - DSPUDFS) i Wyświetlenie informacji o dołączonym systemie plików (Display Mounted File System Information - DSPMFSINF), które wyświetlają atrybuty rozszerzone.

Atrybuty rozszerzone - związane z pewnymi obiektami w systemie plików QDLS - mogą być zmieniane poprzez interfejsy dostarczane przez hierarchiczny system plików (HFS). Więcej informacji o tych systemach plików zawiera temat "System plików usług biblioteki dokumentów (QDLS)" na stronie 40 i "System plików nośników optycznych (QOPT)" na stronie 42.

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 uwzględniać następujące zasady dotyczące nazewnictwa:

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

NazwaFirmyNazwaProduktu.Nazwa_Atrybutu

Obsługa skanowania

System iSeries wyposażony jest teraz w opcję zezwalania na skanowanie obiektów zintegrowanego systemu plików. Ta nowa obsługa ułatwia pracę użytkownikom systemu iSeries przez: umożliwienie skanowania różnych pozycji.

Użytkownicy decydują, kiedy skanowanie zostanie przeprowadzone. Użytkownicy decydują, jakie działanie zostanie podjęte w zależności od wyników skanowania. Więcej informacji o skanowaniu zawierają następujące tematy:

- "Pokrewne wartości systemowe" na stronie 18
- "Uruchamianie operacji skanowania" na stronie 20
- "Przykłady skanowania" na stronie 18

Dwa nowe punkty wyjścia związane z tą obsługą to:

- 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.

Więcej informacji o konfigurowaniu skanowania obiektów zawiera temat “Włączanie i wyłączanie skanowania obiektów” na stronie 111.

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

Przykłady skanowania

Poniżej opisano przykładowe sytuacje, w których program obsługi wyjścia może używać operacji skanowania:

- **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 system 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 przypadkach, zależnie od konfiguracji wartości systemowych i środowiska skanowania. Poniższa lista opisuje te dwa przypadki.

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

W środę uzyskano dostęp do pliku w zintegrowanym systemie plików z komputera PC. Gdy plik został otwarty z komputera PC, uruchomiono skanowanie, ponieważ zarejestrowany został otwarty program obsługi wyjścia, a wartość systemowa QSCANFS wskazuje na skanowanie plików w systemach "root" (/), QOpenSys i UDFS. Skanowanie wykryło jeden wirus, a program antywirusowy przystąpił do naprawy problemu. Ponieważ program obsługi wyjścia naprawił plik tak, że nie zawiera już wirusów, dostęp z komputera PC nie może być zainfekowany, ani wirus nie może się rozprzestrzeniać.

Teraz założmy, że zamiast skanowania pod kątem wirusów zdecydowano nie przeprowadzać skanowania w czasie wykonywania. Po uzyskaniu dostępu do zainfekowanego pliku z komputera PC wirus został przeniesiony do tego komputera. Uruchamiając skanowanie w czasie wykonywania wykryto wirusa zanim zainfekował komputer PC.

Głównym mankamentem 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 sprawdza, czy skanowanie jest wykonywane tylko wtedy, gdy jest niezbędne, a nie przy każdej próbie dostępu. Więcej informacji zawierają sekcje QIBM_QP0L_SCAN_OPEN i QIBM_QP0L_SCAN_CLOSE.

2. Skanowanie masowe lub uruchamiane ręcznie

Opcji tej należy użyć, aby jednocześnie skanować wiele pozycji. W takim przypadku można skonfigurować uruchamianie skanowania w dniach wolnych, gdy serwer nie jest obciążony pracą. Spowoduje to minimalny wpływ skanowania na czas oczekiwania na dostęp do pliku w trakcie normalnej codziennej pracy. Ponieważ skanowanie wykonywane jest w trybie bez połączenia, może to zminimalizować nakład pracy przy skanowaniu plików, które nie zostały zmienione od ostatniego skanowania masowego, gdy ponowne skanowanie ograniczyłoby dostępność plików.

Pokrewne wartości systemowe

Z nową 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 QSCANFSCTL, gdy opcja sterowania

l produktu iSeries Navigator 'Skanowanie tylko dostępu poprzez serwery plików' zostanie wybrana, wynik będzie taki
l sam, jak po wpisaniu opcji *FSVRONLY w trybie znakowym.

l Nazwy i opisy:

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

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

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

l 2. Sterowanie skanowaniem (QSCANFCTL)

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

l • Dostęp skanowania tylko przez serwery plików - (z opcją *FSVRONLY)

l Skanowanie zostanie wykonane tylko wtedy, gdy dostęp do serwera iSeries uzyskano z serwera plików. Jeśli
l opcja ta nie zostanie wybrana, skanowane będą wszystkie próby dostępu.

l • Przerwij zgłoszenie, jeśli program obsługi wyjścia nie powiedzie się - (z opcją *ERRFAIL)

l Jeśli występują błędy podczas wywoływania programu obsługi wyjścia, zgłoszenie lub operacja, która
l wyzwoliła wywołanie programu obsługi wyjścia, zakończy się niepowodzeniem. Gdy opcja ta nie jest wybrana,
l system pominie błędny program obsługi wyjścia i będzie traktował obiekt, jakby nie był skanowany.

l • Wykonaj aktualizację dostępu z prawami zapisu - (bez opcji *NOWRTUPG)

l Aktualizacja dostępu zostanie wykonana dla deskryptora skanowania przekazanego do programu obsługi wyjścia
l w celu uwzględnienia dostępu z prawami zapisu. Jeśli nie wybrano opcji *NOWRTUPG, system nie będzie
l próbował przeprowadzić aktualizacji dostępu z prawami zapisu.

l Jeśli podano opcję *NOWRTUPG, system **nie** będzie próbował zaktualizować dostępu do deskryptora
l skanowania przekazanego do programu obsługi wyjścia w celu uwzględnienia dostępu z prawami zapisu. Jeśli
l nie podano opcji *NOWRTUPG, system będzie próbował zaktualizować dostęp z prawami zapisu.

l • Do sterowania skanowaniem służy atrybut 'Tylko, gdy obiekty zostały zmienione' - (z opcją *USEOCOATR)

l Użyty zostanie atrybut 'tylko zmiana obiektu' (skanowanie obiektu tylko, gdy został zmieniony). Jeśli opcja ta
l nie została wybrana, atrybut nie zostanie użyty, a obiekt zostanie przeskanowany po wprowadzeniu zmian i gdy
l oprogramowanie skanowania wykaże aktualizację.

l • Przerwij zgłoszenia zamknięcia, jeśli skanowanie nie powiedzie się w trakcie operacji zamykania - (bez opcji
l *NOFAILCLO)

l Jeśli skanowanie obiektu w trakcie przetwarzania zamknięcia zakończy się niepowodzeniem, zgłoszenie
l zamknięcia zwróci błąd. Jeśli opcja ta nie zostanie wybrana, zgłoszenie zamknięcia nie zwróci błędu. Nie
l wybranie tej wartości powoduje przesłonięcie podanej wartości 'niepowodzenie zgłoszenia w przypadku
l niepowodzenia programu obsługi wyjścia'.

l Jeśli podano opcję *NOFAILCLO, system **nie** zwróci błędu dla zgłoszenia zamknięcia ze wskazaniem
l przyczyny niepowodzenia nawet wtedy, gdy skanowanie obiektu w ramach przetwarzania zamknięcia nie
l powiedzie się.

l • Skanuj przy kolejnym dostępie, gdy obiekt został odtworzony - (bez opcji *NOPOSTRST)

l Obiekty będą skanowane po odtworzeniu. Jeśli podano atrybut 'obiekt nie będzie skanowany', obiekt zostanie
l przeskanowany raz, po odtworzeniu. Jeśli podano atrybut 'tylko zmiana obiektu', obiekt zostanie
l przeskanowany po odtworzeniu.

l Jeśli podano opcję *NOPOSTRST, gdy obiekt jest odtwarzany, nie zostanie przeskanowany, ponieważ został
l właśnie odtworzony. Jeśli atrybut obiektu ma wartość 'obiekt nie będzie skanowany', obiekt nie będzie
l skanowany ani razu. Jeśli atrybut obiektu ma wartość 'tylko zmiana obiektu', obiekt będzie skanowany, gdy
l zostanie zmieniony po odtworzeniu.

| Więcej informacji o tych dwóch wartościach systemowych zawierają tematy QSCANFS i QSCANFSCTL.

| **Uruchamianie operacji skanowania**

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

- | • “Zmiana obiektu”
- | • “Zmiana sygnatury”
- | • “Różnica identyfikatorów CCSID” na stronie 21
- | • “W trakcie operacji składowania” na stronie 21
- | • “Sprawdzenie integralności obiektu” na stronie 21

| Aby przejrzeć aktualny status skanowania i atrybuty obiektu, można użyć komendy Praca z dowiązaniem obiektów (Work with Object Links - WRKLNK), funkcji API Pobranie atrybutów (Qp0lGetAttr()) lub strony Właściwości w produkcie iSeries Navigator.

| **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. Więcej szczegółów zawiera temat QIBM_QP0L_SCAN_OPEN. 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. Więcej szczegółów zawiera temat QIBM_QP0L_SCAN_OPEN.

| 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 Change Scan Signature, QPOLCHSG, 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.

Różnica identyfikatorów 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.

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.

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 OS/400 dostępu do zbiorów baz danych i obiektów innych typów poprzez biblioteki może być traktowane jako system plików. Podobnie obsługiwane przez OS/400 dostępu do dokumentów (będących w rzeczywistości plikami strumieniowymi) poprzez strukturę folderów może być traktowane jak oddzielny system plików.

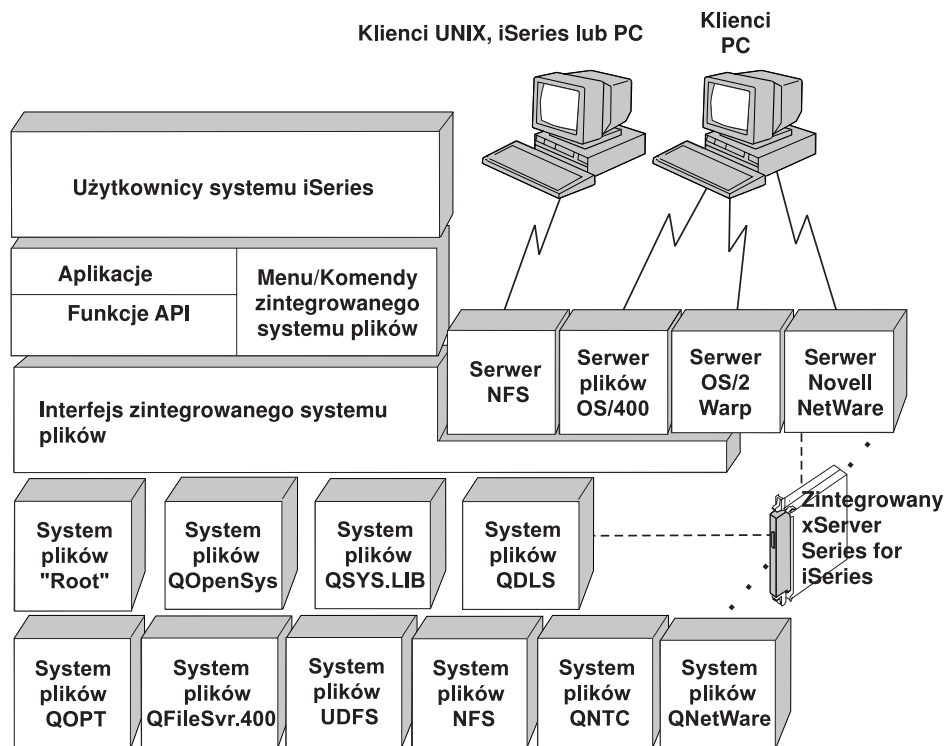
W rzeczywistości zintegrowany system plików traktuje obsługę bibliotek i folderów jak oddzielne systemy plików. Inne rodzaje obsługi zarządzania plikami, mające inne cechy i możliwości, są także traktowane jak oddzielne systemy plików.

Porównanie właściwości i ograniczeń każdego z systemów plików znajduje się w sekcji “Porównanie systemów plików” na stronie 23.

Zintegrowany system plików obsługuje następujące systemy plików:



- “root” (/)
- System plików systemów otwartych (QOpenSys)
- System plików definiowany przez użytkownika (UDFS)
- System plików bibliotek (QSYS.LIB)
- System plików QSYS.LIB niezależnej puli ASP
- System plików usług biblioteki dokumentów (QDLS)
- System plików nośnika optycznego (QOPT)
- System plików NetWare (QNetWare)
- System plików iSeries NetClient (QNTC)
- OS/400 System plików serwera plików (QFileSvr.400)
- Network file system (NFS)

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

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

- Optical Support 
- Obsługa systemu plików OS/400 Network File System 

Porównanie systemów plików

Tabela 2 i Tabela 3 na stronie 25 zawierają podsumowanie właściwości i ograniczeń każdego systemu plików.

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

| Cecha | "root" (/) | QOpenSys | QSYS.LIB ¹⁶ | QDLS | QNTC |
|--|-----------------------|-----------------------|------------------------------|--------------------|----------------------|
| Standardowa część OS/400 | Tak | Tak | Tak | Tak | Tak |
| Typ zbioru | Strumieniowy | Strumieniowy | Rekord ¹² | Strumieniowy | Strumieniowy |
| Limit wielkości zbioru | T2=1 TB; T1=256 GB | T2=1 TB; T1=256 GB | Wielkości zbiorów baz danych | 4 GB | Różne ¹⁷ |
| Zintegrowany z OfficeVision (na przykład plik można wysłać pocztą elektroniczną) | Nie | Nie | Nie | Tak | Nie |
| Dostęp poprzez serwer plików OS/400 | Tak | Tak | Tak | Tak | Tak |
| Bezpośredni dostęp poprzez procesor wejścia/wyjścia serwera plików ¹ | Nie | Nie | Nie | Nie | Tak |
| Względna szybkość operacji otwarcia/zamknięcia | Średnia ² | Średnia ² | Niska ² | Niska ² | Średnia ² |
| Wyszukiwanie nazw z rozróżnianiem małych i wielkich liter | Nie | Tak | Nie ⁴ | Nie ⁵ | Nie |

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

| Cecha | "root" (/) | QOpenSys | QSYS.LIB ¹⁶ | QDLS | QNTC |
|---|------------------------------|------------------------------|--------------------------|---------------------|-----------------|
| Maksymalna długość każdego składnika ścieżki | 255 zn | 255 zn | 10.6 zn ⁶ | 8.3 zn ⁷ | 255 zn |
| Maksymalna długość ścieżki ⁸ | 16 MB | 16 MB | 55 – 66 zn ⁴ | 82 zn | 255 zn |
| Maksymalna długość atrybutów rozszerzonych obiektu | 2 GB | 2 GB | Zmienia się ⁹ | 32 kB | 0 ¹⁸ |
| Maksymalna liczba poziomów hierarchii katalogów w systemie plików | Nieograniczona ¹⁰ | Nieograniczona ¹⁰ | 3 | 32 | 127 |
| Maksymalna liczba dowiązań do jednego obiektu ¹¹ | Zmienia się ¹⁵ | Zmienia się ¹⁵ | 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 ¹³ | Tak | Tak | Tak | Nie | Tak |
| Obsługa kronikowania obiektów | Tak | Tak | Tak ¹⁴ | Nie | Nie |

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

| Cecha | "root" (/) | QOpenSys | QSYS.LIB ¹⁶ | QDLS | QNTC |
|---|------------|----------|------------------------|------|------|
| Uwagi: | | | | | |
| 1. LAN Server używa procesora we/wy serwera plików. | | | | | |
| 2. Przy dostępie przez serwer plików OS/400. | | | | | |
| 3. Przy dostępie do klienta PC przez LAN Server. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny. | | | | | |
| 4. Maksymalna długość ścieżki w systemie plików QSYS.LIB jest równa 55 znaków. Szczegóły zawiera sekcja "System plików bibliotek (QSYS.LIB)" na stronie 35. Maksymalna długość ścieżki w systemie plików QSYS.LIB niezależnej puli ASP jest równa 66 znaków. Szczegóły zawiera sekcja "System plików QSYS.LIB niezależnej ASP" na stronie 38. | | | | | |
| 5. Szczegóły zawiera sekcja "System plików usług biblioteki dokumentów (QDLS)" na stronie 40. | | | | | |
| 6. Do 10 znaków nazwy obiektu i do 6 znaków typu obiektu. Więcej szczegółów zawiera sekcja: "System plików bibliotek (QSYS.LIB)" na stronie 35. | | | | | |
| 7. Do 8 znaków nazwy i od 1 do 3 znaków rozszerzenia określającego typ pliku (jeśli ma występować). Szczegóły zawiera sekcja "System plików usług biblioteki dokumentów (QDLS)" na stronie 40. | | | | | |
| 8. Przy założeniu, że ścieżka bezwzględna rozpoczyna się od znaku /, po którym następuje nazwa systemu plików (taka jak /QDLS...). | | | | | |
| 9. Systemy plików QSYS.LIB i QSYS.LIB niezależnej puli ASP obsługują trzy predefiniowane atrybuty rozszerzone: .SUBJECT, .CODEPAGE i .TYPE. Maksymalna długość jest określana przez złożenie długości tych trzech atrybutów rozszerzonych. | | | | | |
| 10. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu. | | | | | |
| 11. Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu. | | | | | |
| 12. Obszary użytkowników w systemie plików SYS.LIB i QSYS.LIB niezależnej puli ASP obsługują operacje wejścia/wyjścia plików strumieniowych. | | | | | |
| 13. W przypadku, gdy operacja jest kierowana do obiektu w systemie plików z ochroną wątków, funkcje API zintegrowanego systemu plików zapewniają taką ochronę. Jeśli działają one na obiektach w systemach plików bez ochrony wątków, po uruchomieniu kilku wątków w jednym zadaniu funkcje te kończą się niepomyślnie. | | | | | |
| 14. Systemy plików QSYS.LIB i QSYS.LIB niezależnej ASP obsługują kronikowanie innych typów obiektów niż obiekty systemów plików "root" (/), UDFS, i QOpenSys. Temat Zarządzanie kronikowaniem w Centrum informacyjnym iSeries zawiera więcej informacji o kronikowaniu obiektów, które znajdują się w systemach plików QSYS.LIB lub QSYS.LIB niezależnej ASP. | | | | | |
| 15. Katalogi w formacie *TYPE2 mają ograniczenie miliona dowiązań dla jednego obiektu i 999 998 podkatalogów. Katalogi *TYPE1 mają limit 32 767 dowiązań dla obiektu. Więcej informacji zawiera sekcja Katalogi *TYPE2. | | | | | |
| 16. Dane w tej kolumnie odnoszą się zarówno do systemu plików QSYS.LIB jak i do systemu plików QSYS.LIB niezależnej puli ASP. | | | | | |
| 17. W zależności od systemu, do którego uzyskiwany jest dostęp. | | | | | |
| 18. QNTC nie obsługuje atrybutów rozszerzonych. | | | | | |
| Skróty | | | | | |
| zn = znaki | | | | | |
| T1 = *TYPE1 *STMF | | | | | |
| T2 = *TYPE2 *STMF | | | | | |
| B = bajty kB = kilobajty MB = megabajty GB = gigabajty TB = terabajty | | | | | |

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

| Cecha | QOPT | QFileSvr.400 | UDFS | NFS | QNetWare |
|--------------------------|--------------|--------------|-------------------------|-----------------------|--------------|
| Standardowa część OS/400 | Tak | Tak | Tak | Tak | Nie |
| Typ zbioru | Strumieniowy | Strumieniowy | Strumieniowy | Strumieniowy | Strumieniowy |
| Limit wielkości zbioru | 4 GB | 4 GB | T2 = 1 TB; T1=256 GB | Zmienne ¹⁶ | |

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

| Cecha | QOPT | QFileSvr.400 | UDFS | NFS | QNetWare |
|--|-----------------------------|-----------------------------|-----------------------------|-----------------------------|----------------------|
| Zintegrowany z OfficeVision (na przykład plik można wysłać pocztą elektroniczną) | Nie | Nie | Nie | Nie | Nie |
| Dostęp poprzez serwer plików OS/400 | Tak | Tak | Tak | Tak | Tak |
| Bezpośredni dostęp poprzez serwer Integrated PC Server ¹ | Nie | Nie | Nie | Nie | Tak |
| Względna szybkość operacji otwarcia/zamknięcia | Niska | Niska ² | Średnia ² | Średnia ² | Wysoka ¹¹ |
| Wyszukiwanie nazw z rozróżnianiem małych i wielkich liter | Nie | Nie ² | Tak ¹² | Zmienia się ² | Nie |
| Maksymalna długość każdego składnika ścieżki | Zmienia się ⁴ | Zmienia się ² | 255 zn | Zmienia się ² | 255 zn ¹³ |
| Maksymalna długość ścieżki | 294 zn | Nieograniczona ² | 16 MB | Nieograniczona ² | 255 zn |
| Maksymalna długość atrybutów rozszerzonych obiektu | 8 MB | 0 ⁶ | 2 GB ¹⁰ | 0 ⁶ | 64 kB |
| Maksymalna liczba poziomów hierarchii katalogów w systemie plików | Nieograniczona ⁷ | Nieograniczona ² | Nieograniczona ⁷ | Nieograniczona ² | 100 |
| Maksymalna liczba dowiązań do jednego obiektu ⁷ | 1 | 1 | Zmienia się ¹⁵ | Zmienia się ² | 1 |
| Obsługa dowiązań symbolicznych | Nie | Nie | Tak | Tak ² | Nie |
| Obiekt/zbiór może mieć właściciela | Nie | Nie ⁹ | Tak | Tak ² | 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 ² | Nie |
| Ochrona wątków ¹⁴ | 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 |
|---|------|--------------|------|-----|----------|
| Uwagi: | | | | | |
| <ol style="list-style-type: none"> 1. LAN Server używa procesora we/wy serwera plików. 2. Zależy od tego, który zdalny system plików jest dostępny. 3. Przy dostępie przez serwer plików OS/400. 4. Szczegóły zawiera sekcja "System plików nośników optycznych (QOPT)" na stronie 42. 5. Przy założeniu ścieżki bezwzględnej rozpoczynającej się od znaku / po którym następuje nazwa systemu plików. 6. System plików QFileSvr.400 nie zwraca atrybutów rozszerzonych, nawet jeśli używany właśnie system plików je obsługuje. 7. W praktyce liczba poziomów katalogów limitowana jest przez ograniczenia programu i pamięci systemu. 8. Z wyjątkiem katalogu, który może zawierać tylko jedno dowiązanie do innego katalogu. 9. Dostępny system plików może obsługiwać prawa własności do obiektów. 10. Maksymalna długość atrybutów rozszerzonych całego UDFS nie może przekroczyć 40 B. 11. Przy dostępie poprzez klienta Novell NetWare. Dostęp za pomocą funkcji API iSeries jest stosunkowo wolny. 12. Rozróżnianie wielkości liter jest definiowane w momencie tworzenia UDFS. Jeśli w trakcie tworzenia UDFS zostanie użyty parametr *MIXED, spowoduje to rozróżnianie wielkości liter. 13. Obiekty NDS (NetWare Directory Services) mogą mieć maksymalnie 255 znaków. Nazwy plików i katalogów są ograniczone wg konwencji DOS: 8.3. 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. 15. Katalogi *TYPE2 posiadają limit miliona dowiązań na obiekt. Katalogi *TYPE1 posiadają limit 32767 dowiązań na obiekt. Więcej informacji zawiera sekcja Katalogi *TYPE2. 16. W zależności od systemu, do którego uzyskiwany jest dostęp. | | | | | |
| Skróty | | | | | |
| zn = znaki | | | | | |
| T1 = *TYPE1 *STMF | | | | | |
| T2 = *TYPE2 *STMF | | | | | |
| B = bajty kB = kilobajty MB = megabajty GB = gigabajty TB = terabajty | | | | | |

Bazowy system plików "root" (/)

Bazowy system plików ("root", /) wykorzystuje w pełni możliwość obsługi plików strumieniowych i hierarchiczną strukturę katalogów zintegrowanego systemu plików. System plików "root" (/) ma charakterystykę 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 funkcje API z ochroną wątków,
- obsługuje obiekty *FIFO,
- obsługuje obiekty /dev/null i /dev/zero *CHRSF a także obiekty *CHRSF,
- obsługuje kronikowanie zmian obiektów,
- obsługuje skanowanie obiektów z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików.

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

/dev/zero są zawsze puste, a wszelkie dane zapisywane w plikach /dev/null i /dev/zero są usuwane. Pliki /dev/null i /dev/zero są obiektami typu *CHRSF i mogą być wykorzystywane jak zwykcyjne pliki, z wyjątkiem tego, że dane z pliku /dev/null nie są nigdy odczytywane, a plik /dev/zero zawsze zwraca wartość pomyślną z danymi ustawionymi na zero.

Więcej informacji o systemie plików "root" (/) zawiera sekcja Korzystanie z systemu plików "root" (/).

Korzystanie z bazowego systemu plików "root" (/)

System plików "root" (/) jest dostępny poprzez interfejs zintegrowanego systemu plików, który wykorzystuje komendy serwera plików OS/400 lub zintegrowanego systemu plików, ekrany użytkownika i funkcje API.

- "Rozróżnianie wielkości znaków w bazowym systemie plików"
- "Nazwy ścieżek w bazowym systemie plików "root" (/)"
- "Dowiązania w bazowym systemie plików "root" (/)"
- "Korzystanie z komend zintegrowanego systemu plików w bazowym systemie plików"
- "Korzystanie z funkcji API zintegrowanego systemu plików w bazowym systemie plików"
- "Kronikowanie zmian obiektów w systemie plików "root" (/)." na stronie 29
- "Urządzenia UDP i TCP w systemie plików "root" (/)." na stronie 29

Rozróżnianie wielkości znaków w bazowym systemie plików: Ten system plików zachowuje małe i wielkie litery we wprowadzanej nazwie obiektu, lecz nie rozróżnia wielkości liter podczas wyszukiwania nazwy.

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

- Ścieżki dostępu mają następującą postać:

Katalog/Katalog . . . /Obiekt

- 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) (patrz "Niezmienność nazw" na stronie 16). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi *TYPE2.

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.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 10.

Korzystanie z komend zintegrowanego systemu plików w bazowym systemie plików: Wszystkie komendy wymienione w sekcji "Dostęp przy użyciu komend języka CL" na stronie 58 i na ekranach opisanych w sekcji "Dostęp przy użyciu pozycji menu i ekranów" na stronie 57 mogą być uruchamiane w systemie plików "root" (/). Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

Korzystanie z funkcji API zintegrowanego systemu plików w bazowym systemie plików: Wszystkie funkcje API wymienione w sekcji "Wykonywanie operacji za pomocą funkcji API" na stronie 91 mogą być uruchamiane w systemie plików "root" (/). Dodatkowe informacje na ten temat, takie jak bezpieczeństwo wątków, zawiera sekcja Funkcje API.

Kronikowanie zmian obiektów w systemie plików "root" (/): Obiekty w bazowym systemie plików "root" (/) mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o kronikowaniu zmian obiektów w systemie plików "root" (/) zawiera sekcja "Kronikowanie obiektów" na stronie 81.

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. Użycie tych urządzeń będzie obsługiwane tylko w środowisku PASE (Portable Application Solutions Environment). Tabela 4 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 |

System plików systemów otwartych (QOpenSys)

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

Dodatkowo, system ten:

- dostępny jest 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.

Więcej informacji na temat QOpenSys zawiera sekcja Korzystanie z systemu QOpenSys.

Używanie QOpenSys

QOpenSys jest dostępny z poprzez interfejs zintegrowanego systemu plików wykorzystujący serwer plików OS/400 lub komendy zintegrowanego systemu plików, ekrany użytkownika i funkcje API.

- "Rozróżnianie wielkości znaków w systemie plików QOpenSys" na stronie 30
- "Nazwy ścieżek w systemie plików QOpenSys" na stronie 30
- "Dowiązania w systemie plików QOpenSys" na stronie 30
- "Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOpenSys" na stronie 30
- "Używanie funkcji API zintegrowanego systemu plików w systemie plików QOpenSys" na stronie 30
- "Kronikowanie zmian obiektów w systemie plików QOpenSys" na stronie 30

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 tym samym katalogu QOpenSys można przechowywać obiekt o nazwie Payroll, obiekt o nazwie PayRoll i obiekt o nazwie PAYROLL.

Nazwy ścieżek w systemie plików QOpenSys:

- Ścieżki dostępu mają następującą postać:
Katalog/Katalog/ . . . /Obiekt
- Każdy składnik ścieżki może mieć do 255 znaków. Pełna ścieżka może mieć do 16 megabajtów długości.
- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu lub pamięci serwera.
- Podczas zapisywania nazwy, składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów *TYPE1) i UTF-16 (dla katalogów *TYPE2) (patrz “Niezmienność nazw” na stronie 16). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi *TYPE2.

Dowiązania w systemie plików QOpenSys: W systemie plików QOpenSys dozwolonych jest wiele dowiązań stałych do jednego obiektu. W pełni obsługiwane są dowiązania symboliczne. Można używać dowiązań symbolicznych z systemu plików QOpenSys do obiektów w innych systemach plików.

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOpenSys: Wszystkie komendy wymienione w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 i na ekranach opisanych w sekcji “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57 mogą być uruchamiane w systemie plików QOpenSys. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego.

| **Używanie funkcji API zintegrowanego systemu plików w systemie plików QOpenSys:** Wszystkie funkcje API
| wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 mogą być uruchamiane w
| systemie plików QOpenSys. Dodatkowe informacje na ten temat, takie jak bezpieczeństwo wątków, zawiera sekcja
| Funkcje API.

Kronikowanie zmian obiektów w systemie plików QOpenSys: Obiekty w systemie plików QOpenSys mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o kronikowaniu zmian obiektów w systemie plików QOpenSys zawiera sekcja “Kronikowanie obiektów” na stronie 81.

System plików definiowany przez użytkownika (UDFS)

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

Dodatkowo system ten:

- 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.

| Więcej informacji o tworzeniu systemów plików UDFS zawiera sekcja CRTUDFS. 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. Informacje o ustawieniu tego atrybutu zawiera sekcja “Włączanie i wyłączanie skanowania obiektów” na stronie 111.
- | • Atrybut ograniczenia zmiany nazwy i usuwania dowiązań.
- | • Wartość kontroli dla systemu plików UDFS.
- | • Różne formaty plików strumieniowych, *TYPE1 i *TYPE2. Dodatkowe informacje o tych dwóch formatach zawiera sekcja “Plik strumieniowy” na stronie 14.

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

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

Koncepcja UDFS

| W systemie plików UDFS, tak jak 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.

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

Aby poznać więcej szczegółów o podłączaniu systemów plików, należy przejrzeć publikację OS/400 Network File

System Support .

Korzystanie z UDFS poprzez interfejs zintegrowanego systemu plików

Do systemu plików UDFS można uzyskać dostęp poprzez interfejs zintegrowanego systemu plików używając serwera plików OS/400 lub komend zintegrowanego systemu plików, ekranów użytkownika i funkcji API. Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

- “Rozróżnianie wielkości znaków w zintegrowanym systemie plików UDFS” na stronie 32

- “Nazwy ścieżek w zintegrowanym systemie plików UDFS”
- “Dowiązania w zintegrowanym systemie plików UDFS” na stronie 33
- “Korzystanie z komend zintegrowanego systemu plików w systemie UDFS” na stronie 33
- “Korzystanie z funkcji API zintegrowanego systemu plików w systemie UDFS” na stronie 33
- “Graficzny interfejs użytkownika dla systemu plików UDFS” na stronie 33
- “Utworzenie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Usuwanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Wyświetlanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Podłączanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Odłączanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Składowanie i odtwarzanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Kronikowanie zmian obiektów w systemie plików UDFS” na stronie 35
- “UDFS i niezależne pule pamięci dyskowej (ASP)” na stronie 35

Rozróżnianie wielkości znaków w zintegrowanym systemie plików UDFS: Podczas tworzenia systemu UDFS można określić, czy ma on rozróżniać wielkość liter w nazwach obiektów.

Jeśli użytkownik 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 jakkolwiek litera jest mała. Dlatego `/home/MURPH/` i `/home/murph/` rozpoznane zostaną jako inne katalogi. Aby system UDFS rozróżniał wielkość liter, należy podczas użycia komendy `CRTUDFS` podać wartość `*MIXED` dla parametru `CASE`.

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

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

Nazwy ścieżek w zintegrowanym systemie plików UDFS: W momencie, gdy występuje konieczność manipulacji całym systemem plików UDFS wraz ze wszystkimi plikami w nim zawartymi, do reprezentowania UDFS służy specjalny plik blokowy (`*BLKSF`). Jeśli system UDFS znajduje się w podstawowej puli ASP, nazwa specjalnego pliku blokowego musi mieć format:

```
/dev/QASPXX/nazwa_udfs.udfs
```

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

Jeśli system UDFS znajduje się w niezależnej puli ASP, nazwa specjalnego pliku blokowego musi mieć format:

```
/dev/nazwa_asp/nazwa_udfs.udfs
```

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

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

Dodatkowe reguły nazwa ścieżek:

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

- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu lub pamięci serwera.
- Podczas zapisywania nazwy, składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (dla katalogów *TYPE1) i UTF-16 (dla katalogów *TYPE2) (patrz “Niezmienność nazw” na stronie 16). Więcej informacji o formatach katalogów można znaleźć w sekcji Katalogi *TYPE2.

Dowiązania w zintegrowanym systemie plików UDFS: 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.

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend zintegrowanego systemu plików w systemie UDFS: Wszystkie komendy przedstawione w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 i ekrany opisane w sekcji “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57 działają również w systemie plików UDFS. Istnieją jednak komendy CL przeznaczone specjalnie dla UDFS i innych dołączanych systemów plików. Są one opisane w poniższej tabeli.

Tabela 5. Komendy CL dotyczące systemu plików definiowanego przez użytkownika

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

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

Korzystanie z funkcji API zintegrowanego systemu plików w systemie UDFS: Wszystkie funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 mogą być uruchamiane w systemie plików użytkownika.

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

Graficzny interfejs użytkownika dla systemu plików UDFS: 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 iSeries Navigator. Podstawowe zadania opisuje sekcja:

- “Utworzenie systemu plików UDFS w zintegrowanym systemie plików”.
- “Podłączanie systemu plików UDFS w zintegrowanym systemie plików”.
- “Odłączanie systemu plików UDFS w zintegrowanym systemie plików”.

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

Usuwanie systemu plików UDFS w zintegrowanym systemie plików: Komenda Usunięcie systemu plików użytkownika (Delete User-Defined File System - DLTUDFS) powoduje usunięcie istniejącego, nie podłączonego 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.

Wyświetlanie systemu plików UDFS w zintegrowanym systemie plików: 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.

Podłączanie systemu plików UDFS w zintegrowanym systemie plików: Komenda Dodanie podłączonego systemu plików (Add Mounted File System - ADDMFS) i MOUNT powoduje udostępnienie obiektów znajdujących się w systemie plików w przestrzeni nazw zintegrowanego systemu plików. Aby dołączyć UDFS, należy w komendzie ADDMFS dla parametru TYPE podać wartość *UDFS.

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

Odłączanie systemu plików UDFS w zintegrowanym systemie plików: Komenda odłączająca UDFS powoduje, że zawartość UDFS staje się niedostępna dla interfejsu zintegrowanego systemu plików. Obiekty znajdujące się w UDFS stają się niedostępne w momencie odłączenia UDFS. Komenda Usuwanie podłączonego FS (Remove Mounted File System - RMVMFS) lub komenda UNMOUNT spowodują, że podłączone systemy plików będą niedostępne dla przestrzeni nazw zintegrowanego systemu plików. Jeśli któryś z obiektów w systemie plików jest używany (na przykład plik jest otwarty) w trakcie wykonywania komendy, to użytkownik otrzyma komunikat o błędzie. System UDFS pozostanie wówczas podłączony. Jeśli użytkownik podłączył inny UDFS, 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.

Składowanie i odtwarzanie systemu plików UDFS w zintegrowanym systemie plików: Użytkownik ma możliwość składowania i odtwarzania wszystkich obiektów UDFS wraz ze związanymi z nimi uprawnieniami. Komenda Składowanie (Save - SAV) pozwala składać obiekty UDFS, natomiast komenda Odtworzenie (Restore -

RST) pozwala użytkownikowi odtworzyć obiekty UDFS. Obydwie komendy funkcjonują bez względu na to, czy UDFS jest dołączony czy odłączony. Jednak aby poprawnie składować atrybuty UDFS, a nie tylko obiekty znajdujące się w UDFS, system plików powinien zostać odłączony.

Kronikowanie zmian obiektów w systemie plików UDFS: Obiekty w zdefiniowanych przez użytkowników systemach plików mogą być kronikowane. Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o kronikowaniu zmian obiektów w systemie plików UDFS zawiera sekcja “Kronikowanie obiektów” na stronie 81.

UDFS i niezależne pule pamięci dyskowej (ASP): Gdy niezależna pula pamięci dyskowej ASP jest podłączana, w systemie plików "root" (/) następuje kilka zmian:

- 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. Więcej informacji znajduje się w sekcji “System plików QSYS.LIB niezależnej ASP” na stronie 38.

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

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 podzbiorach zbiorów fizycznych, przestrzeniach użytkownika i zbiorach składowania.

Przed wersją 3 systemu OS/400, system plików QSYS.LIB określany był jako *system plików serwera* iSeries. Programiści, używający do pisania aplikacji języków programowania takich jak RPG lub COBOL i narzędzi takich jak DDS, korzystali z systemu plików QSYS.LIB. Operatorzy systemu używający komend, menu i ekranów do pracy z kolejkami wydruków korzystali z systemu plików QSYS.LIB, podobnie jak administratorzy systemu tworzący i zmieniający profile użytkowników.

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

Więcej informacji na temat QSYS.LIB zawiera sekcja Korzystanie z systemu plików QSYS.LIB przy użyciu interfejsu zintegrowanego systemu plików.

Korzystanie z QSYS.LIB poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QSYS.LIB jest możliwy poprzez interfejs zintegrowanego systemu plików po użyciu komend systemu serwera plików OS/400 lub zintegrowanego systemu plików, ekranów użytkownika i funkcji API. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

- “Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB”
- “Obsługa ograniczeń w systemie plików QSYS.LIB”
- “Obsługa przestrzeni użytkownika w systemie plików QSYS.LIB”
- “Obsługa zbiorów składowania w systemie plików QSYS.LIB”
- “Rozróżnianie wielkości znaków w systemie plików QSYS.LIB” na stronie 37
- “Nazwy ścieżek w systemie plików QSYS.LIB” na stronie 37
- “Dowiązania w systemie plików QSYS.LIB” na stronie 37
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB” na stronie 37
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB” na stronie 38

Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB: QPWFSERVER jest listą autoryzacji (typ obiektu *AUTL). Określa ona dodatkowe wymagania dotyczące dostępu dla wszystkich obiektów w systemie plików QSYS.LIB, do których istnieje dostęp poprzez klientów zdalnych. Uprawnienia podane w tej liście odnoszą się do wszystkich obiektów znajdujących się w systemie plików QSYS.LIB.

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

Obsługa ograniczeń w systemie plików QSYS.LIB:

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

Obsługa przestrzeni użytkownika w systemie plików QSYS.LIB: QSYS.LIB obsługuje operacje danych wejściowych i wyjściowych pliku strumieniowego dla obiektów przestrzeni użytkownika. Na przykład program może zapisać i odczytać dane strumieniowe w przestrzeni użytkownika. Obszar użytkownika może mieć maksymalnie 16 776 704 bajtów.

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

Obsługa zbiorów składowania w systemie plików QSYS.LIB: System plików QSYS.LIB obsługuje operacje strumieniowego wejścia/wyjścia dla obiektów zbioru składowania. Na przykład, w zbiorze składowania mogą znajdować się dane, które można odczytać lub skopiować do innego zbioru, jeśli trzeba je umieścić w istniejącym, pustym obiekcie zbioru składowania. Kiedy zbiór składowania jest otwarty, nie są dozwolone żadne inne otwarte instancje tego pliku. Zbiór składowania **może** mieć wiele instancji otwartych do odczytu, pod warunkiem, że każde

zadanie ma tylko jedną taką instancję. Zbioru składowania nie można otwierać w celu wykonania jednoczesnych operacji zapisu i odczytu. Operacje strumieniowego wejścia/wyjścia dla pliku składowania nie są dozwolone, jeśli w ramach zadania wykonywanych jest wiele wątków.

Operacje strumieniowego wejścia/wyjścia dla zbioru składowania nie są również obsługiwane, jeśli zbiór składowania lub jego katalog zostały wyeksportowane poprzez serwer systemu Network File System. Są one jednak dostępne z klientów PC i poprzez system plików QFileSvr.400.

Rozróżnianie wielkości znaków w systemie plików QSYS.LIB: System plików QSYS.LIB nie rozróżnia małych i wielkich liter w nazwach obiektów. Wyszukiwanie nazw obiektów daje ten sam wynik niezależnie od tego, czy litery w nazwie obiektu są wielkie, czy małe.

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

Nazwy ścieżek w systemie plików QSYS.LIB:

- Każdy składnik ścieżki musi zawierać nazwę obiektu i następujący po niej typ obiektu. Na przykład:

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

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

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

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

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

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

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

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

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

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB: Komendy wymienione w sekcji "Dostęp przy użyciu komend języka CL" na stronie 58 działają w systemie plików QSYS.LIB, z wyjątkiem następujących:

- 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,
- komend STRJRN i ENDJRN nie można użyć dla zbiorów fizycznych baz danych.

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB: Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QSYS.LIB, z wyjątkiem następujących:

- 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 z innego systemu plików, który obsługuje dowiązania symboliczne,
- funkcje API `QjoStartJournal()` i `QjoEndJournal()` nie mogą być użyte na fizycznych zbiorach bazy danych.

System plików QSYS.LIB niezależnej ASP

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

Dodatkowo, system ten:

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

Więcej informacji o systemie plików QSYS.LIB niezależnej puli ASP zawiera sekcja Używanie systemu plików QSYS.LIB niezależnej puli ASP z wykorzystaniem interfejsu zintegrowanego systemu plików.

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

System plików QSYS.LIB niezależnej puli ASP jest dostępny poprzez interfejs zintegrowanego systemu plików przy użyciu komend serwera plików OS/400 lub zintegrowanego systemu plików, ekranów użytkowników i funkcji API. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

- “Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB niezależnej puli ASP”
- “Obsługa ograniczeń w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 39
- “Obsługa przestrzeni użytkownika w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 39
- “Obsługa zbiorów składowania w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 39
- “Rozróżnianie wielkości znaków w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 39
- “Nazwy ścieżek w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 39
- “Dowiązania w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 40
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 40
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP” na stronie 40

Lista autoryzacji QPWFSERVER w systemie plików QSYS.LIB niezależnej puli ASP: QPWFSERVER jest listą autoryzacji (typ obiektu *AUTL). Określa ona dodatkowe wymagania dotyczące dostępu dla wszystkich obiektów w systemie plików QSYS.LIB niezależnej puli ASP, do których istnieje dostęp poprzez klientów zdalnych. Uprawnienia podane w tej liście odnoszą się do wszystkich obiektów znajdujących się w systemie plików QSYS.LIB niezależnej puli ASP.

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

Obsługa ograniczeń w systemie plików QSYS.LIB niezależnej puli ASP:

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

Obsługa przestrzeni użytkownika w systemie plików QSYS.LIB niezależnej puli ASP: System plików QSYS.LIB niezależnej 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 ASP obsługuje operacje strumieniowego wejścia/wyjścia dla obiektów zbioru składowania. Na przykład, w zbiorze składowania mogą znajdować się dane, które można odczytać lub skopiować do innego zbioru, jeśli trzeba je umieścić w istniejącym, pustym obiekcie zbioru składowania. Kiedy zbiór składowania jest otwarty, nie są dozwolone żadne inne otwarte instancje tego pliku. Zbiór składowania **może** mieć wiele instancji otwartych do odczytu, pod warunkiem, że każde zadanie ma tylko jedną taką instancję. 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 i poprzez system plików QFileSvr.400.

Rozróżnianie wielkości znaków w systemie plików QSYS.LIB niezależnej puli ASP: System plików QSYS.LIB niezależnej puli ASP nie rozróżnia małych i wielkich liter w nazwach obiektów. Wyszukiwanie nazw obiektów daje ten sam wynik niezależnie od tego, czy litery w nazwie obiektu są wielkie, czy małe.

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

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

- Każdy składnik ścieżki musi zawierać nazwę obiektu i następujący po niej typ obiektu. Na przykład:

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

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

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

- Nazwa obiektu w każdym składniku może mieć do 10 znaków, a typ obiektu - do 6 znaków.
- Hierarchia katalogów w ramach systemu plików QSYS.LIB niezależnej puli ASP może mieć dwa lub trzy poziomy (w ścieżce mogą występować dwa lub trzy składniki), zależnie od typu danego obiektu. Jeśli obiekt jest zbiorem

bazy danych, hierarchia może obejmować trzy poziomy (bibliotekę, zbiór, podzbiór); w innym przypadku mogą wystąpić jedynie dwa poziomy (biblioteka, obiekt). Maksymalna długość ścieżki określona jest przez kombinację długości składników i liczby poziomów katalogów.

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

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

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

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

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

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QSYS.LIB niezależnej puli ASP: Komendy wymienione w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 działają w systemie plików QSYS.LIB niezależnej ASP, z wyjątkiem następujących:

- 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 lub RST lub musi być określona niezależna pula ASP w parametrze ASPDEV; konwencja nazw ścieżek /nazwa_asp/QSYS.LIB/obiekt.typ nie jest obsługiwana w SAV i RST.

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

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

Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QSYS.LIB niezależnej ASP, z wyjątkiem następujących:

- 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 ASP z innego systemu plików, który obsługuje dowiązania symboliczne,
- funkcje API QjoStartJournal() i QjoEndJournal() nie mogą być użyte na fizycznych zbiorach bazy danych.

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.

Więcej informacji o systemie plików QDLS zawiera sekcja Korzystanie z systemu plików QDLS przy użyciu interfejsu zintegrowanego systemu plików.

Korzystanie z QDLS poprzez interfejs zintegrowanego systemu plików

Dostęp do systemu plików QDLS jest możliwy poprzez interfejs zintegrowanego systemu plików po użyciu komend systemu serwera plików OS/400 lub zintegrowanego systemu plików, ekranów użytkownika i funkcji API. Przy korzystaniu z interfejsu zintegrowanego systemu plików należy mieć na uwadze poniższe okoliczności i ograniczenia.

Więcej informacji o systemie plików QDLS zawierają następujące tematy:

- “Zintegrowany system plików i hierarchiczny system plików (HFS) w systemie plików QDLS”
- “Wpisanie użytkownika na listę w systemie plików QDLS”
- “Rozróżnianie wielkości znaków w systemie plików QDLS”
- “Nazwy ścieżek w systemie plików QDLS”
- “Dowiązania w systemie plików QDLS” na stronie 42
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QDLS” na stronie 42
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików QDLS” na stronie 42

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

Więcej informacji o systemie plików HFS zawiera temat Funkcje API hierarchicznego systemu plików w Centrum informacyjnym iSeries.

Wpisanie użytkownika na listę w systemie plików QDLS: Użytkownik pracujący z obiektami w QDLS musi zostać wpisany do katalogu dystrybucyjnego systemu.

Rozróżnianie wielkości znaków w systemie plików QDLS: System plików QDLS zamienia małe litery alfabetu angielskiego (od **a** do **z**) użyte w nazwach obiektów na wielkie. Przy wyszukiwaniu nazw obiektów zawierających jedynie takie znaki nie są rozróżniane małe i wielkie litery.

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

Więcej szczegółów zawiera temat Nazwa folderu i dokumentu w Centrum informacyjnym iSeries.

Nazwy ścieżek w systemie plików QDLS:

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

/QDLS/FLR1/DOC1

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

/QDLS/FLR1/DOC1.TXT

- Nazwa w każdym ze składników może mieć do 8 znaków, a rozszerzenie (o ile występuje) może mieć do 3 znaków. Maksymalna długość ścieżki bezwzględnej zaczynającej się od /QDLS to 82 znaki.

- Hierarchia katalogów w ramach systemu QDLS może mieć do 32 poziomów. Jeśli pierwszymi dwoma poziomami są / i QDLS, hierarchia katalogów może liczyć 34 poziomy.
- Dopóki w bibliotece QUSRSYS nie zostanie utworzony obszar danych Q0DEC500, przed zapamiętaniem nazw występujące w nich znaki są przekształcane do strony kodowej zadania. Jeśli obszar danych zostanie utworzony, to przed zachowaniem nazw występujące w nich znaki są przekształcane do strony kodowej 500. Dzięki tej funkcji zapewniona jest zgodność z systemem plików QDLS z poprzednich wersji. Nazwa może zostać odrzucona, jeśli nie można jej poddać konwersji do właściwej strony kodowej.

Więcej informacji o stronach kodowych zawiera temat Globalizacja w Centrum informacyjnym iSeries.

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

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

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QDLS: Komendy wymienione w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 działają w systemie plików QDLS, z wyjątkiem następujących:

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

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

Używanie funkcji API zintegrowanego systemu plików w systemie plików QDLS: Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QDLS, z wyjątkiem następujących:

- funkcja symlink() może być używana jedynie do tworzenia dowiązań do obiektu w QDLS z innego systemu plików obsługującego dowiązania symboliczne,
- poniższe funkcje nie są obsługiwane:
 - givedescriptor()
 - ioctl()
 - link()
 - QjoEndJournal()
 - QjoRetrieveJournalInformation()
 - QJORJIDI()
 - QJOSJRNE()
 - QjoStartJournal()
 - Qp0lGetPathFromFileID()
 - readlink()
 - takedescriptor()

System plików nośników optycznych (QOPT)

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

Dodatkowo, system ten:

- 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.

Więcej informacji o systemie plików QOPT zawiera sekcja Korzystanie z systemu plików QOPT przy użyciu zintegrowanego systemu plików.

Używanie systemu plików QOPT poprzez zintegrowany system plików

Dostęp do systemu plików QOPT jest możliwy poprzez interfejs zintegrowanego systemu plików po użyciu komend systemu serwera plików OS/400 lub zintegrowanego systemu plików, ekranów użytkownika i funkcji API. Korzystając z interfejsu zintegrowanego systemu plików, należy uwzględnić poniższe uwagi i ograniczenia.

Więcej informacji o systemie plików QOPT zawierają następujące tematy:

- “Zintegrowany system plików i hierarchiczny system plików (HFS) w systemie plików QOPT”
- “Rozróżnianie wielkości znaków w systemie plików QOPT”
- “Nazwy ścieżek w systemie plików QOPT”
- “Dowiązania w systemie plików QOPT” na stronie 44
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOPT” na stronie 44
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików QOPT” na stronie 44

Więcej szczegółów można znaleźć w publikacji Optical Support .

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

Więcej informacji o funkcjach API systemu plików HFS zawiera temat Funkcje API hierarchicznego systemu plików w Centrum informacyjnym iSeries lub publikacja Optical Support .

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

Nazwy ścieżek w systemie plików QOPT:

- Nazwa ścieżki musi zaczynać się od ukośnika (/). Ścieżka dostępu zbudowana jest z nazwy systemu plików, nazwy woluminu, nazwy katalogu, nazwy podkatalogu i nazwy obiektu. Na przykład:
/QOPT/WOLUMIN/KATALOG/PODKATALOG/OBIEKT
- Wymagane jest podanie nazwy systemu plików QOPT.
- Wolumin i długość nazwy ścieżki mogą zależeć od formatu nośnika optycznego.
- 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 znaku o kodzie X'FF'. Dodatkowe ograniczenia mogą wynikać z formatu nośnika optycznego.

- Ostatnim elementem ścieżki jest nazwa obiektu. Liczba znaków w nazwie obiektu jest ograniczona przez liczbę znaków w nazwie katalogu.

Więcej informacji o regułach nazw ścieżek w systemie plików QOPT zawiera temat “Path Name Rules” w publikacji Optical Support  .

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

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QOPT: Większość komend wymienionych w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 działa w systemie plików QOPT. Istnieje jednak kilka wyjątków. Należy pamiętać, że korzystanie z tych komend CL w przypadku procesu wielowątkowego może nie być bezpieczne. Zależnie od formatu nośnika optycznego mogą obowiązywać pewne ograniczenia. Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

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

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

Używanie funkcji API zintegrowanego systemu plików w systemie plików QOPT: Wszystkie funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików “root”, z wyjątkiem następujących:

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

System plików NetWare (QNetWare)

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

Dodatkowo, system ten:

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

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

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

- Podłączanie systemów plików NetWare
- Struktura katalogów QNetWare
- Korzystanie z QNetWare poprzez interfejs zintegrowanego systemu plików

Podłączanie systemów plików NetWare

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

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

```
DREYFUSS/NEST:doorway
```

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

Struktura katalogów QNetWare

Struktura katalogów /QNetWare reprezentuje różne złożone systemy plików.

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

Korzystanie z QNetWare poprzez interfejs zintegrowanego systemu plików

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

Więcej informacji o systemie plików QNetWare zawierają następujące pozycje:

- "Uprawnienia i prawa własności w systemie plików QNetWare" na stronie 46
- "Kontrola w systemie plików QNetWare" na stronie 46
- "Pliki i katalogi w systemie plików QNetWare" na stronie 46
- "Obiekty NDS w systemie plików QNetWare" na stronie 46
- "Dowiązania w systemie plików QNetWare" na stronie 46

- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QNetWare”
- “Korzystanie z funkcji API zintegrowanego systemu plików w systemie plików QNetWare” na stronie 47

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. Dlatego komenda CHGAUD nie jest obsługiwana.

Pliki i katalogi w systemie plików QNetWare: System plików QNetWare nie zachowuje wielkości liter w nazwach plików i katalogów wpisywanych w komendach lub funkcjach API. Podczas transmisji do serwera wszystkie litery w nazwach są zamieniane na wielkie. Novell NetWare 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 NDS w systemie plików QNetWare: System plików QNetWare obsługuje wyświetlanie nazw NDS używając małych i wielkich liter.

Dowiązania w systemie plików QNetWare: System plików QNetWare dopuszcza tylko jedno dowiązanie do obiektu. W systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych. W systemie QNetWare nie można tworzyć ani przechowywać dowiązań symbolicznych. Mogą jednak zostać utworzone dowiązania symboliczne w katalogach “root” (/) lub QOpenSys, które wskazują plik lub katalog QNetWare.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QNetWare: Komendy, które zawiera sekcja “Dostęp przy użyciu komend języka CL” na stronie 58, działają w systemie plików QNetWare; z następującymi wyjątkami:

ADDLINK
 APYJRNCHG
 CHGAUD
 CHGJRNOBJ
 CHGPGP
 CHKIN
 CHKOUT
 ENDJRN
 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: Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QNetWare z wyjątkiem następujących:

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

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

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

System plików iSeries NetClient (QNTC)

System plików QNTC zapewnia dostęp do danych i obiektów przechowywanych w lokalnym lub zdalnym serwerze Integrated xSeries Server for iSeries, w którym działa system Windows NT 4.0 Server lub nowszy, lub serwerze autonomicznym. Umożliwia on aplikacjom serwera iSeries używanie tych samych danych, których używają klienci systemu Windows.

- | System plików QNTC jest częścią podstawowego systemu operacyjnego OS/400. Instalacja opcji 29 systemu operacyjnego serwera iSeries 400 Integration with Windows NT Server nie jest wymagana do uzyskania dostępu do systemu plików /QNTC.

Więcej informacji na temat systemu plików QNTC, zawiera sekcja Użycie systemu plików QNTC w interfejsie zintegrowanego systemu plików.

Korzystanie z QNTC poprzez interfejs zintegrowanego systemu plików

- | Używając produktu iSeries NetServer, iSeries Navigator, komend zintegrowanego systemu plików, ekranów użytkownika lub funkcji API można uzyskać dostęp do systemu plików QNTC poprzez interfejs zintegrowanego systemu plików. Należy mieć na uwadze poniższe okoliczności i ograniczenia.

Więcej informacji o systemie plików QNTC zawierają następujące tematy.

- “Uprawnienia i prawa własności w systemie plików QNTC”
- “Rozróżnianie wielkości znaków w systemie plików QNTC”
- “Nazwy ścieżek w systemie plików QNTC”
- “Dowiązania w systemie plików QNTC”
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QNTC”
- “Korzystanie z komendy MKDIR w systemie plików QNTC” na stronie 49
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików QNTC” na stronie 49
- “Włączanie korzystania z protokołu Kerberos w systemie plików QNTC” na stronie 50

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 musi zaczynać się od ukośnika (/) i może zawierać co najwyżej 255 znaków.
- W nazwach ścieżek rozróżniane są wielkie i małe litery.
- Ścieżka składa się z nazwy systemu plików, nazwy serwera Windows NT, nazwy udziału, nazw katalogu i podkatalogów oraz nazwy obiektu. Ścieżki dostępu mają następującą postać:

```

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

```
- Nazwa serwera może mieć do 15 znaków. Musi być ona częścią ścieżki.
- Nazwa współużytkowana może mieć do 12 znaków.
- Każdy składnik ścieżki (występujący po nazwie współużytkowanej) może mieć do 255 znaków.
- W ramach QNTC dostępnych jest w zasadzie 130 poziomów hierarchii. Jeśli wszystkie składniki ścieżki zostały dodane do ścieżki jako poziomy hierarchii, hierarchia katalogów może mieć aż 132 poziomy.
- Nazwy są przechowywane w identyfikatorze CCSID Unicode.
- Każdy działający serwer Windows NT w lokalnej podsieci automatycznie pojawi się w systemie plików /QNTC jako katalog. Komendy Tworzenie katalogu (Make Directory - MKDIR) (patrz Tabela 7 na stronie 58) lub funkcji API mkdir() (patrz “Dostęp przy użyciu funkcji API” na stronie 74) można użyć do dodania serwerów opartych na technologii Windows znajdujących się poza lokalną podsiecią.

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.

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików QNTC: Komendy, które zawiera sekcja “Dostęp przy użyciu komend języka CL” na stronie 58, działają w systemie plików QNTC z następującymi wyjątkami:

```

ADDLNK
APYJRNCHG
CHGJRNOBJ

```

CHGOWN

CHGAUT

CHGPGP

CHKIN

CHKOUT

DSPAUT

ENDJRN

| 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 sekcji “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

Korzystanie z komendy MKDIR w systemie plików QNTC: Aby dodać katalog serwera do katalogu /QNTC, należy użyć komendy Utworzenie katalogu (Make Directory - MKDIR). Katalog QNTC tworzony jest automatycznie dla wszystkich działających serwerów Windows w domenie serwera iSeries NetServer i lokalnej podsięci. Serwery oparte na technologii Windows 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 podsięcią.

Uwaga: 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.

Używanie funkcji API zintegrowanego systemu plików w systemie plików QNTC: Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QNTC, z wyjątkiem następujących:

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

QjoRetrieveJournalInformation()
QJORJIDI()
QJOSJRNE()
QjoStartJournal()
Qp0lGetPathFromFileID()
readlink()
symlink()
takedescriptor()

| **Włączanie korzystania z protokołu Kerberos w systemie plików QNTC:** QNTC umożliwia w systemie iSeries
| dostęp do serwerów opartych na technologii Windows, które obsługują protokół uwierzytelniania Kerberos V5.
| Zamiast używać hasła zarządzania siecią LAN do uwierzytelniania w każdym serwerze, poprawnie skonfigurowany
| system iSeries będzie mógł uzyskać dostęp do obsługiwanych serwerów CIFS w pojedynczej transakcji logowania.

| Zanim obsługa protokołu Kerberos zostanie włączona w systemie plików QNTC, należy skonfigurować następujące
| usługi:

- | • Usługa uwierzytelniania sieciowego
- | • Odwzorowanie tożsamości dla przedsiębiorstwa (EIM)

| gdy powyższe pozycje zostaną skonfigurowane, można włączyć obsługę protokołu kerberos użytkownikom w systemie
| plików QNTC. Aby umożliwić użytkownikom korzystanie z obsługi protokołu kerberos w systemie plików QNTC,
| należy wykonać następujące czynności.

- | • Profil użytkowników iSeries musi mieć przypisany parametr lokalnego zarządzania hasłami, LCLPDMGT, 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. Jedyne dostępy do systemu będą możliwe poprzez aplikacje obsługujące
| protokół kerberos, takie jak iSeries Navigator.

| Jeśli użytkownik poda wartość ***YES**, hasło będzie obsługiwane przez system i użytkownik zostanie pomyślnie
| uwierzytelniony bez protokołu kerberos.

- | • 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 protokołu kerberos.
 - | – 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)

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

System plików serwera plików OS/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. QFileSvr.400 współdziała z serwerem plików OS/400 w systemie docelowym w celu wykonywania
bieżących operacji na zbiorach.

Więcej informacji na temat systemu plików QFileSvr.400 zawiera sekcja Używanie systemu plików QFileSvr.400 w
interfejsie zintegrowanego systemu plików.

Korzystanie z QFileSvr.400 poprzez interfejs zintegrowanego systemu plików

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

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

Więcej informacji o systemie plików QFileSvr.400 zawierają następujące tematy.

- “Rozróżnianie wielkości znaków w systemie plików serwera plików OS/400”
- “Nazwy ścieżek w systemie plików serwera plików OS/400”
- “Komunikacja w systemie plików serwera plików OS/400”
- “Ochrona i uprawnienia do obiektu w systemie plików serwera plików OS/400” na stronie 52
- “Dowiązania w systemie plików serwera plików OS/400” na stronie 53
- “Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików serwera plików OS/400” na stronie 53
- “Używanie funkcji API zintegrowanego systemu plików w systemie plików serwer plików OS/400” na stronie 53

Rozróżnianie wielkości znaków w systemie plików serwera plików OS/400: System plików QFileSvr.400 zachowuje małe i wielkie litery we wprowadzanej nazwie obiektu dla pierwszego poziomu katalogów, który w zasadzie jest katalogiem “root” (/) systemu docelowego. System QFileSvr.400 nie rozróżnia jednak wielkości liter podczas wyszukiwania nazwy.

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

Nazwy ścieżek w systemie plików serwera plików OS/400:

- Ścieżki dostępu mają następującą postać:

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

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

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

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

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

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

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

- Głębokość struktury hierarchicznej katalogów limitowana jest jedynie przez ograniczenia programu i systemu oraz przez ograniczenia nałożone przez używany system plików.
- Gdy nazwa jest zapisywana, składające się na nią znaki są przekształcane do postaci UCS2 Poziom 1 (patrz “Niezmienność nazw” na stronie 16).

Komunikacja w systemie plików serwera plików OS/400:

- Połączenia TCP z serwerem plików serwera docelowego mogą zostać ustanowione tylko wtedy, gdy jest w nim aktywny podsystem QSERVER.
- Próby połączenia SNA LU 6.2 wykonywane są tylko w przypadku, gdy istnieje nieużywana sesja obsługiwana lokalnie (na przykład sesja ustanowiona specjalnie do użycia przez połączenie LU 6.2). Podczas nawiązywania połączeń LU 6.2 system plików QFileSvr.400 używa trybu BLANK. W systemie docelowym zadanie 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 OS/400.
- Jeśli nie zostało ustanowione połączenie z serwerem docelowym, system plików QFileSvr.400 zakłada, że katalog pierwszego poziomu reprezentuje nazwę hosta TCP/IP. Aby ustanowić połączenie z serwerem docelowym, system plików QFileSvr.400 wykonuje następujące czynności:

1. Zamienia nazwę jednostki zdalnej na adres IP.
2. Łączy się z programem odwzorowującym serwera poprzez port 449 za pomocą otrzymanego adresu IP. Następnie wysyła zapytanie do programu odwzorowującego serwera o usługę "as-file". Odpowiedź na zadane pytanie może być 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 OS/400.
 - Jeśli program odwzorowujący serwera docelowego nie jest aktywny, używany jest domyślny numer portu dla usługi "as-file" (8473).

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

3. Jeśli nazwa jednostki zdalnej nie zostanie zamieniona na adres IP, to zakłada się, że katalog pierwszego poziomu ma nazwę taką, jak SNA LU 6.2. Dlatego też ustanawia się połączenie pomiędzy APPC i serwerem plików OS/400.
- Następnie system plików QFileSvr.400 okresowo (co dwie godziny) sprawdza, czy istnieją jakieś nieużywane połączenia (na przykład z połączeniem nie są powiązane żadne otwarte pliki), dla których nie zanotowano żadnej aktywności w ciągu ostatnich dwóch godzin. Jeśli system znajdzie takie połączenie, to zostanie ono zakończone.
 - System plików QFileSvr.400 nie wykrywa 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, to po krótkim czasie system plików QFileSvr.400 zgłosi błąd przekroczenia czasu oczekiwania (time-out).

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

Ochrona i uprawnienia do obiektu w systemie plików serwera plików OS/400: Jeśli w obu systemach skonfigurowany jest protokół Kerberos i użytkownik jest uwierzytelniony w protokole Kerberos, to protokół Kerberos może być użyty do uwierzytelniania systemu plików znajdującego się na docelowym serwerze iSeries. Jeśli uwierzytelnianie Kerberos nie powiedzie się, to do potwierdzenia dostępu może być użyty identyfikator użytkownika i hasło.

Uwaga: Jeśli ważność biletu uprawniającego do biletu lub biletu serwera wygaśnie po tym jak serwer docelowy potwierdzi dostęp, wygaśnięcie nie nastąpi, dopóki połączenie z serwerem docelowym nie zostanie rozłączone. Więcej informacji o protokole Kerberos zawiera sekcja Usługa uwierzytelniania sieciowego w Centrum informacyjnym iSeries.

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

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

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

Dowiązania w systemie plików serwera plików OS/400: System plików QFileSvr.400 dopuszcza tylko jedno dowiązanie do obiektu. W systemie plików QFileSvr.400 nie można tworzyć ani przechowywać dowiązań symbolicznych. Można jednak uzyskać dostęp do danych w systemie QFileSvr.400, używając dowiązania symbolicznego z systemu plików “root” (/), QOpenSys lub UDFS.

Opis dowiązań zawiera sekcja “Dowiązanie” na stronie 10.

Korzystanie z komend i terminali zintegrowanego systemu plików w systemie plików serwera plików OS/400: Komendy wymienione w sekcji “Dostęp przy użyciu komend języka CL” na stronie 58 działają w systemie plików QFileSvr.400, z wyjątkiem następujących:

```

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

```

Te same ograniczenia odnoszą się do ekranów użytkownika, które opisuje sekcja “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57.

Używanie funkcji API zintegrowanego systemu plików w systemie plików serwer plików OS/400: Funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików QFileSvr.400, z wyjątkiem następujących:

```

chown()
fchown()
| fclear()
| fclear64()
givedescriptor()
link()
QjoEndJournal()

```

QjoRetrieveJournalInformation()
QJORJIDI()
QJOSJRNE
QjoStartJournal
Qp0lGetPathFromFileID()
symlink()
takedescriptor()

Sieciowy system plików (NFS)

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

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

Więcej informacji o systemie plików NFS zawiera temat *Użycie systemu plików NFS w interfejsie zintegrowanego systemu plików*.

Korzystanie z systemów plików NFS poprzez interfejs zintegrowanego systemu plików

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

Więcej informacji o systemie plików NFS zawierają następujące tematy.

- “Charakterystyka sieciowego systemu plików”
- “Odmiany serwerów i klientów w sieciowym systemie plików”
- “Dowiązania w sieciowym systemie plików” na stronie 55
- “Korzystanie z komend zintegrowanego systemu plików w sieciowym systemie plików” na stronie 55
- “Korzystanie z funkcji API zintegrowanego systemu plików w sieciowym systemie plików” na stronie 56

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 oznacza, 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 połączenia klient/serwer, przy każdym z nich NFS funkcjonuje inaczej i inna jest jego charakterystyka:

1. Użytkownik podłącza system plików serwera iSeries w kliencie.
2. Użytkownik podłącza system plików serwera UNIX w kliencie.
3. Użytkownik podłącza system plików z innego serwera do klienta.

W pierwszym przypadku podłączony system plików zachowuje się na kliencie podobnie, jak zachowywałby się na serwerze iSeries. Jednak obie charakterystyki systemu plików NFS i oferowanego systemu plików muszą być uwzględnione. Na przykład, jeśli użytkownik podłącza system plików QDLS z serwera do klienta, będzie miał charakterystyki i ograniczenia systemu plików QDLS. Na przykład, w systemie plików QDLS 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. Aby dowiedzieć się więcej o systemie plików QOpenSys, należy przejrzeć temat "System plików systemów otwartych (QOpenSys)" na stronie 29.

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

Dowiązania w sieciowym systemie plików: W systemie plików NFS może istnieć wiele dowiązań stałych do jednego obiektu. W pełni obsługiwane są dowiązania symboliczne. Można używać dowiązań symbolicznych z systemu plików NFS do obiektów w innych systemach. Możliwość stosowania wielu dowiązań stałych i dowiązań symbolicznych jest całkowicie uzależniona od systemu plików, który został dołączony z NFS.

Opis dowiązań zawiera sekcja "Dowiązanie" na stronie 10.

Korzystanie z komend zintegrowanego systemu plików w sieciowym systemie plików: Wszystkie komendy przedstawione w sekcji "Dostęp przy użyciu komend języka CL" na stronie 58 i ekrany opisane w sekcji "Dostęp przy użyciu pozycji menu i ekranów" na stronie 57 działają również w sieciowym systemie plików (NFS), z wyjątkiem następujących:

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

Istnieją jednak komendy odnoszące się tylko do pewnych podłączonych systemów plików, w tym do NFS. Jednakże używanie tych komend może być niebezpieczne w przypadku procesu wielowątkowego. Poniższa tabela opisuje te komendy. Aby poznać szczegóły dotyczące komend i ekranów związanych bezpośrednio z systemem plików NFS,

należy przejrzeć publikację OS/400 Network File System Support  .

Tabela 6. Komendy CL dotyczące systemu plików NFS

| 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. |


Tabela 6. Komendy CL dotyczące systemu plików NFS (kontynuacja)

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

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

Korzystanie z funkcji API zintegrowanego systemu plików w sieciowym systemie plików: Wszystkie funkcje API wymienione w sekcji “Wykonywanie operacji za pomocą funkcji API” na stronie 91 działają w systemie plików NFS, z wyjątkiem następujących:

- QjoEndJournal()
- 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.

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, służącego do zarządzania i administrowania serwerem z pulpitu systemu Windows.

Istnieje kilka sposobów komunikowania się z zintegrowanym systemem plików:

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.

Dostęp przy użyciu komend języka CL

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

Dostęp przy użyciu funkcji API

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

Dostęp przy użyciu narzędzia iSeries Navigator

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

Dostęp przy użyciu iSeries NetServer

Produkt iSeries NetServer, Windows umożliwia użytkownikom dostęp do ścieżek współużytkowanych katalogów i współużytkowanych kolejek wyjściowych systemu OS/400.

Dostęp przy użyciu programów do przesyłania plików

Programy do przesyłania plików służą do przesyłania plików z kilku różnych systemów plików. Umożliwiają także przesyłanie folderów i dokumentów z systemu plików QDLS.

Dostęp przy użyciu komputera PC

Komputer PC podłączony do systemu iSeries umożliwia wykonywanie operacji na katalogach i obiektach znajdujących się w zintegrowanym systemie plików. Katalogi i obiekty widziane są tak, jakby znajdowały się w komputerze PC.

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

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

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

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

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

Z menu zintegrowanego systemu plików możliwe jest wywoływanie ekranów, na których można wykonywać następujące operacje:

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

Niektóre systemy plików nie obsługują wszystkich powyższych operacji. Aby sprawdzić ograniczenia, które dotyczą wybranego systemu plików, patrz Praca z systemami plików.

Więcej informacji o menu i ekranach zintegrowanego systemu plików zawierają następujące sekcje:

- Reguły dotyczące nazw ścieżek dla komend CL i ekranów
- Dostęp przy użyciu komend języka CL

Dostęp przy użyciu komend języka CL

Wszystkie operacje wykonywane przy użyciu menu i ekranów zintegrowanego systemu plików (patrz “Dostęp przy użyciu pozycji menu i ekranów” na stronie 57) 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 sekcje “System plików definiowany przez użytkownika (UDFS)” na stronie 30 i “Sieciowy system plików (NFS)” na stronie 54. Jeżeli 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 pomiędzy katalogiem i obiektem. | |
| ADDMFS | Dodanie dołączonego systemu plików (Add Mounted File System). Dołącza wyeksportowane systemy plików zdalnego serwera do lokalnych katalogów klienta. | MOUNT |
| APYJRNCHG ² | Zastosowanie kronikowanych zmian (Apply Journalled Changes). Korzysta z pozycji kronik do wprowadzenia zmian, które wystąpiły od ostatniego składowania obiektu kronikowanego lub do wprowadzenia zmian do określonego punktu. | |

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

| Komenda | Opis | Alias |
|------------------------|--|-----------|
| 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 ² | 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. | |
| 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. | |

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

| Komenda | Opis | Alias |
|----------------------|--|------------|
| 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 ² | 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). | |
| 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). | |
| RTVDIRINF | Pobranie informacji katalogu. Służy do zbierania atrybutów obiektów w zintegrowanym systemie plików. | |
| SAV | Składowanie (Save). Kopiuje obiekt lub grupę obiektów z systemu do urządzenia składowania. | |
| SNDJRNE ² | Wysłanie pozycji do kroniki (Send Journal Entry). Dodaje do dziennika pozycje kroniki użytkownika opcjonalnie powiązane z kronikowanym obiektem. | |
| STRJRN ² | 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. | |

Tabela 7. Komendy zintegrowanego systemu plików (kontynuacja)

| Komenda | Opis | Alias |
|------------------------|---|-------|
| WRKLNK | Praca z dowiązaniem obiektów (Work with Object Links). Wyświetla listę obiektów w katalogu i zawiera opcje przeprowadzania działań na obiektach. | |
| WRKOBJOWN ¹ | Praca z obiektami według właścicieli (Work with Objects by Owner). Wyświetla listę obiektów należących do profilu użytkownika i zawiera opcje przeprowadzania działań na obiektach. | |
| WRKOBJPGP ¹ | 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. | |

Uwaga:

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

Więcej informacji o komendach CL zintegrowanego systemu plików i ograniczeniach używania tych komend w różnych systemach plików zawierają następujące sekcje:

- Praca z systemami plików
- Reguły dotyczące nazw ścieżek dla komend CL i ekranów
- Praca z danymi wyjściowymi komend RTVDIRINF i PRDIRINF
- Sekcja CL w Centrum informacyjnym iSeries

Reguły dotyczące nazw ścieżek dla komend CL i ekranów

Używając do pracy z obiektem komend lub ekranów zintegrowanego systemu plików, użytkownik identyfikuje obiekt podając ścieżkę do niego. Poniżej przedstawiono zasady, o których należy pamiętać w trakcie podawania ścieżek. Określenie **obiekt** w tych zasadach 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 apostrofach ('). Gdy ścieżki wpisuje się na ekranach, apostrofy nie są obowiązkowe. Jeśli ścieżka zawiera ciągi znaków w cudzysłowie, należy ją ująć w znaki ' '.
- Ścieżkę należy wprowadzać od lewej do prawej strony, poczynając od najwyższego poziomu katalogu i kończąc na obiekcie, na którym ma być wykonana komenda. Nazwy składników ścieżek są rozdzielane ukośnikiem (/) lub ukośnikiem odwrotnym (\), na przykład:
 'Kat1/Kat2/Kat3/ZbiórUż'
 lub
 'Kat1\Kat2\Kat3\ZbiórUż'
- Znaków / lub \ i znaków pustych nie można używać w poszczególnych składnikach ścieżki (ponieważ znaki / i \ są używane jako separatory). Komendy nie zamieniają małych liter na wielkie. Nazwa może zostać zamieniona na pisaną wielkimi literami lub nie, w zależności od tego, czy system plików zawierający obiekt uwzględnia wielkość liter i czy obiekt jest tworzony, czy poszukiwany.
- Długość nazwy obiektu jest ograniczona przez system plików, w którym znajduje się obiekt, i przez maksymalną długość ciągu komendy. Komendy akceptują nazwy obiektów o maksymalnej długości 255 znaków i ścieżki o maksymalnej długości 5000 znaków.

Ograniczenia dotyczące nazw ścieżek w każdym systemie plików opisane zostały w sekcji Praca z systemami plików.

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

```
'/Kat1/Kat2/Kat3/ZbiórUż'
```

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

```
'MójKat/MójZbiór'
```

gdzie MójKat jest podkatalogiem bieżącego katalogu użytkownika.

- Znak tyldy (~) z ukośnikiem (lub ukośnikiem odwrotnym) na początku ścieżki oznacza, że ścieżka rozpoczyna się w katalogu osobistym użytkownika, który wprowadził komendę, na przykład:

```
'~/KatUżytk/ObiektUżytk'
```

- Znak tyldy (~), po którym następuje nazwa użytkownika a następnie ukośnik (lub ukośnik odwrotny), na początku nazwy ścieżki oznacza, że ścieżka rozpoczyna się w katalogu osobistym użytkownika określonego przez podaną nazwę, na przykład:

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

- W niektórych komendach, aby znaleźć nazwy odpowiadające wzorcowi, można w ostatnim składniku ścieżki użyć gwiazdki (*) lub znaku zapytania (?). Znak * oznacza, że system ma szukać nazw zawierających dowolną liczbę znaków w miejscu znaku *. Znak ? oznacza, że system ma szukać nazw zawierających pojedynczy znak w miejscu znaku ?. Poniższy przykład w

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

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

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

- Nazwy ścieżek nie mogą zaczynać się od pojedynczego znaku gwiazdki (*), aby odróżniały się od wartości specjalnych serwera iSeries. 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'
```

Więcej szczegółów zawiera sekcja: “System plików bibliotek (QSYS.LIB)” na stronie 35.

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

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

Więcej szczegółów zawiera sekcja: “System plików QSYS.LIB niezależnej ASP” na stronie 38.

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

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

Na przykład:

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

lub

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

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 (zobacz “Dostęp przy użyciu funkcji API” na stronie 74). Dodatkowo, wartość szesnastkowa 0 jest niedozwolona w nazwach ścieżek.

Informacje o ograniczeniach obowiązujących dla wybranej komendy zawiera pomoc do komendy lub sekcja Control language (CL) w Centrum informacyjnym iSeries.

Praca z danymi wyjściowymi komend RTVDIRINF i PRVDIRINF

Komenda RTVDIRINF służy do zbierania atrybutów dla obiektów, które zawiera zintegrowany system plików. 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. Więcej informacji, takich jak nazewnictwo zbiorów baz danych lub tabel zawiera sekcja RTVDIRINF.

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

Sekcja “Dostęp do danych RTVDIRINF” na stronie 74 zawiera informacje o dostępie do danych znajdujących się w tych trzech tabelach. Więcej informacji dotyczących używania tych danych zawiera sekcja “Użycie danych RTVDIRINF” na stronie 74.

Sekcja Tabela 8 opisuje pola tabeli, w której przechowywane są atrybuty obiektów. Jeśli dla parametru 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 INFLIB utworzone zostaną pliki QAEZD0001D i QAEZD0001O. Użytkownicy mogą podawać własne prefiksy nazw plików dla bazy danych, o długości do dziewięciu znaków.

Tabela 8. QAEZDxxxxO (Plik zawierający atrybuty obiektów)

| Nazwa pola | Typ pola | Opis pola |
|-------------------------|-------------------|---|
| QEZDIRIDX | INTEGER | Indeks katalogu nadrzędnego. |
| QEZOBJNAM ¹ | VARGRAPHIC (1024) | Nazwa obiektu. ² |
| QEZOBJLEN | INTEGER | Liczba bajtów w polu nazwy obiektu. |
| QEZNMCCSID | INTEGER | Identyfikator CCSID nazwy obiektu. |
| QEZREGION | GRAPHIC (2) | Dwuznakowy ID kraju dla nazwy obiektu. (Decyduje o działaniach definiowanych według lokalizacji, takich jak kolejność zestawiania). |
| QEZLANGID | GRAPHIC (3) | Trzyznakowy ID oznaczający język nazwy obiektu. |
| QEZMODE | INTEGER | Tryb i rodzaj dostępu do pliku. Więcej informacji o trybie zawiera sekcja o funkcji API Otwarcie pliku open(). |
| QEZOBJTYPE ¹ | GRAPHIC (10) | Typ obiektu. |
| QEZCCSID | INTEGER | Identyfikator CCSID danych i atrybutów rozszerzonych obiektu. |
| QEZALCSIZE ¹ | BIGINT | Liczba bajtów przydzielonych dla obiektu. |

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

| Nazwa pola | Typ pola | Opis pola |
|-------------------------|--------------|---|
| 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. |
| QEZCEAS | BIGINT | Liczba krytycznych atrybutów rozszerzonych przypisanych do obiektu. |
| QEZEXTATRS | BIGINT | Całkowita liczba bajtów dla wszystkich danych atrybutów rozszerzonych. |
| QEZCRTTIM | TIMESTAMP | Data i godzina utworzenia obiektu. |
| QEZACCTIM | TIMESTAMP | Data i godzina ostatniego dostępu do danych obiektu. |
| QEZCHGTIMA ¹ | TIMESTAMP | Data i godzina ostatniej zmiany atrybutów obiektu. |
| QEZCHGTIMD | TIMESTAMP | Data i godzina ostatniej zmiany danych obiektu. |
| QEZSTGFREE ¹ | SMALLINT | Informacje o tym, czy dane obiektu zostały odłączone ze zwolnieniem pamięci. Poprawne wartości: 0 - Dane obiektu nie zostały odłączone. 1 - Dane obiektu zostały odłączone. |
| QEZCHKOUT ¹ | 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 nie zawiera danych. |
| QEZLOCAL | SMALLINT | Określa, czy obiekt jest przechowywany lokalnie, czy w systemie zdalnym. Decyzja o miejscu przechowywania obiektu zależy od reguł konkretnego systemu plików. Obiekty w systemach plików, które nie zawierają indykatora lokalnego lub zdalnego, traktowane są jak obiekty zdalne. Poprawne wartości: 1 - Dane obiektu przechowywane są lokalnie. 2 - Dane obiektu przechowywane są w systemie zdalnym. |
| QEZOWN ¹ | 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. |
| QEZUID | INTEGER | Każdy użytkownik w systemie musi mieć unikalny numer identyfikacyjny użytkownika (UID). |
| QEZOWNPGP | GRAPHIC (10) | Nazwa profilu użytkownika będącego grupą podstawową obiektu lub następujące wartości specjalne: *NONE - Obiekt nie posiada skojarzonej z nim grupy podstawowej. *NOUSRPRF - Ta wartość specjalna używana jest przez system plików Network File System do wskazania, że w lokalnym serwerze nie ma profilu użytkownika z ID grupy (GID) odpowiadającym GID obiektu zdalnego. |
| QEZGID | INTEGER | Profile grup identyfikowane są według unikalnych numerów identyfikacyjnych grup (GID). |

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. |
| QEZASP | SMALLINT | Pula pamięci dyskowej, w której obiekt jest przechowywany. |
| QEZJRNSTS ¹ | SMALLINT | Aktualny status kronikowania obiektu. W polu będzie jedna z następujących wartości: 0 (NOT_JOURNALED) - Obiekt nie jest aktualnie kronikowany. 1 (JOURNALED) - Obiekt jest aktualnie kronikowany. |
| QEZJSUBTRE | SMALLINT | Gdy flaga ta zostanie zwrócona, oznacza, że obiekt jest katalogiem z semantyką poddrzewa kronikowania zintegrowanego systemu plików. 0 - Obiekt nie jest kronikowany z semantyką poddrzewa. 1 - Obiekt jest kronikowany z semantyką poddrzewa. Nowe obiekty utworzone w poddrzewie tego katalogu odziedziczają atrybuty kronikowania i opcje katalogu. |
| QEZJOPTENT | SMALLINT | Gdy kronikowanie jest aktywne, kronikowane są pozycje opcjonalne. Lista pozycji opcjonalnych różni się zależnie od typu obiektu. 0 - Obiekt nie jest kronikowany z pozycjami opcjonalnymi. 1 - Obiekt jest kronikowany z pozycjami opcjonalnymi. |
| QEZJAFTERI | SMALLINT | Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu po wprowadzeniu zmiany. 0 - Obiekt nie jest kronikowany z obrazami po zmianach. 1 - Obiekt jest kronikowany z obrazami po zmianach. |
| QEZJBEBORI | SMALLINT | Gdy kronikowanie jest aktywne, kronikowany jest obraz obiektu przed wprowadzeniem zmian. 0 - Obiekt nie jest kronikowany z obrazami przed wprowadzeniem zmian. 0 - Obiekt jest kronikowany z obrazami przed wprowadzeniem zmian. |
| QEZJRNID | GRAPHIC (10) | Pole to przypisuje do kronikowanego obiektu identyfikator, którego można użyć w różnych komendach i funkcjach API związanych z kronikowaniem. |
| QEZJRNNAM | GRAPHIC (10) | Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę aktualnie używanej kroniki. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę kroniki ostatnio używanej dla obiektu. Jeśli obiekt nigdy nie był kronikowany, wszystkie bajty znajdujące się w tym polu otrzymają wartość binarną zero. |
| QEZJRNLIB | GRAPHIC (10) | Jeśli status kronikowania ma wartość JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się aktualnie używana kronika. Jeśli status kronikowania ma wartość NOT_JOURNALED, pole to zawiera nazwę biblioteki, w której znajduje się kronika ostatnio używana dla obiektu. Jeśli obiekt nigdy nie był kronikowany, wszystkie bajty znajdujące się w tym polu otrzymają wartość binarną zero. |
| QEZJRNSTR | TIMESTAMP | Liczba sekund, które upłynęły od epoki odpowiadającej dacie i godzinie uruchomienia kronikowania obiektu. Jeśli obiekt nigdy nie był kronikowany, wszystkie bajty znajdujące się w tym polu otrzymają wartość binarną zero. |

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

| Nazwa pola | Typ pola | Opis pola |
|------------------------|--------------|--|
| QEZAUDT | GRAPHIC (10) | <p>Wartość kontroli przypisana do obiektu. Poprawne wartości:</p> <p>*NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.</p> <p>*USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.</p> <p>*CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.</p> <p>*ALL - Kontrola obiektu jest przeprowadzana dla każdego dostępu do obiektu, uzyskanego przez dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.</p> |
| QEZBLKSIZ | INTEGER | Wielkość bloku obiektu. |
| QEZNLNK | INTEGER | Liczba dowiązań stałych do obiektu. |
| QEZFILEID ¹ | GRAPHIC (16) | ID zbioru dla obiektu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu. |
| QEZFILEIDS | INTEGER | 4-bajtowy ID zbioru. Numer ten jednoznacznie identyfikuje obiekt w systemie plików. Nie może on identyfikować obiektu w całym systemie. |
| QEZGENID | BIGINT | ID generacji przypisany do ID zbioru. |
| QEZFSID | BIGINT | ID systemu plików, w którym znajduje się obiekt. Numer ten jednoznacznie identyfikuje system plików, w którym znajduje się obiekt. |
| QEZRDEV | BIGINT | Jeśli obiekt reprezentuje plik specjalny urządzenia, jest to rzeczywiste urządzenie reprezentowane przez ten obiekt. |
| QEZDOM | GRAPHIC (10) | <p>Domena obiektu. Poprawne wartości:</p> <p>*SYSTEM - Obiekt istnieje w domenie systemowej.</p> <p>*USER - Obiekt istnieje w domenie użytkownika.</p> |
| QEZCRTAUD | GRAPHIC (10) | <p>Wartość kontroli przypisana do obiektu utworzonego w katalogu. Poprawne wartości:</p> <p>*NONE - W trakcie odczytu lub zmiany obiektu nie jest przeprowadzana żadna kontrola bez względu na użytkownika, który uzyskuje dostęp do obiektu.</p> <p>*USRPRF - Kontrola obiektu jest przeprowadzana tylko wtedy, gdy kontrolowany jest bieżący użytkownik. Następuje sprawdzenie użytkownika w celu określenia warunku przeprowadzenia kontroli obiektu. Profil użytkownika może warunkować, czy obiekt kontrolowany jest tylko po wprowadzeniu zmian, czy także po odczycie danych obiektu.</p> <p>*CHANGE - Kontrola obiektu jest przeprowadzana po każdej zmianie wprowadzonej przez dowolnego użytkownika systemu.</p> <p>*ALL - Kontrola obiektu jest przeprowadzana po każdym dostępie do obiektu dowolnego użytkownika systemu. Każdy dostęp do danych definiowany jest jako operacja odczytu lub zmiany.</p> |

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

| Nazwa pola | Typ pola | Opis pola |
|------------|-------------|--|
| 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. Więcej informacji zawiera sekcja “Obsługa skanowania” na stronie 17.</p> <p>Poprawne wartości:</p> <p>'x40' (SCANNING_NO) - Obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania. Uwaga: Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>'x80' (SCANNING_YES) - Obiekt będzie podlegać skanowaniu zgodnie z regułami opisanyymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>'x20' (SCANNING_CHGONLY) - Obiekt będzie podlegać skanowaniu zgodnie z regułami opisanyymi 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 (QSCANFSCCTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES. Uwaga: Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> |
| 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. Więcej informacji zawiera sekcja “Obsługa skanowania” na stronie 17.</p> <p>Poprawne wartości:</p> <p>x'40' - Po utworzeniu w katalogu obiekt nie będzie podlegać skanowaniu według reguł opisanych w programach obsługi wyjścia dotyczących skanowania. Uwaga: Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> <p>x'80' - Po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanyymi w programach obsługi wyjścia dotyczących skanowania w przypadku modyfikacji obiektu lub aktualizacji oprogramowania do skanowania od momentu ostatniego skanowania obiektu.</p> <p>x'20' - Po utworzeniu obiektu w katalogu obiekt będzie podlegać skanowaniu zgodnie z regułami opisanyymi 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 (QSCANFSCCTL). Jeśli wartość ta nie została podana, system podejmie takie działania, jakby podany został atrybut SCANNING_YES. Uwaga: Jeśli dla atrybutu Sterowanie skanowaniem systemów plików (QSCANFSCCTL) nie podano wartości *NOPOSTRST w trakcie odtwarzania obiektu z tym atrybutem, obiekt będzie skanowany przynajmniej raz po odtworzeniu.</p> |

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

| Nazwa pola | Typ pola | Opis pola |
|------------|-------------|---|
| 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> |
| QEZSSIGDF | GRAPHIC (1) | <p>Sygnatury skanowania oznaczają poziom obsługi oprogramowania do skanowania.</p> <p>Gdy obiekt jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z powiązaną sygnaturą skanowania grupy niezależnych pul ASP. Gdy obiekt nie jest w grupie niezależnych pul ASP, sygnatura skanowania obiektu porównywana jest z wartością globalnej sygnatury skanowania. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Porównywane sygnatury nie różnią się.</p> <p>x'01' - Porównywane sygnatury różnią się.</p> |
| QEZSBINARY | GRAPHIC (1) | <p>Określa, czy podczas ostatniego skanowania obiekt podlegał skanowaniu w trybie binarnym. W polu będzie jedna z następujących wartości:</p> <p>x'00' - Obiekt nie podlegał skanowaniu w trybie binarnym.</p> <p>x'01' - Obiekt podlegał skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu w trybie binarnym. Jeśli status skanowania obiektu ma wartość SCAN_FAILURE, skanowanie obiektu w trybie binarnym nie powiodło się.</p> |
| QEZSCCSID1 | INTEGER | <p>Określa, czy obiekt podlegał skanowaniu z wymienionym identyfikatorem CCSID podczas poprzedniego skanowania. Jeśli status skanowania obiektu wskazuje wartość SCAN_SUCCESS, obiekt został pomyślnie poddany skanowaniu z identyfikatorem CCSID. Jeśli status skanowania obiektu wskazuje wartość SCAN_FAILURE, skanowanie obiektu z identyfikatorem CCSID nie powiodło się. Wartość 0 oznacza, że pole to nie obowiązuje.</p> |

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

| Nazwa pola | Typ pola | Opis pola |
|---------------------|-----------|---|
| QEZSCCSID2 | INTEGER | 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. |
| 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 OS/400 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 Qp01SetAttr(). |
| 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 Qp01SetAttr() do wyzerowania licznika dni używania. |
| QEZPRMLNK | SMALLINT | Gdy obiekt ma kilka nazw, pole to otrzyma wartość tylko dla pierwszej znalezionej nazwy. |
| 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. |
| QEZSIG ¹ | SMALLINT | Określa, czy obiekt zawiera podpis cyfrowy OS/400. Poprawne wartości: 0 - Obiekt nie zawiera podpisu cyfrowego OS/400. 1 - Obiekt zawiera podpis cyfrowy OS/400. |
| 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. |
| QEZMLTSIG | SMALLINT | Określa, czy obiekt zawiera więcej niż jeden podpis cyfrowy OS/400. Poprawne wartości: 0 - Obiekt zawiera tylko jeden podpis cyfrowy. 1 - Obiekt zawiera więcej niż jeden podpis cyfrowy. Jeśli w polu QEZSYSSIG znajduje się wartość 1, przynajmniej jeden z podpisów pochodzi ze źródła zaufanego w systemie. |

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

| Nazwa pola | Typ pola | Opis pola |
|------------|----------|---|
| QEZDSTGOPT | SMALLINT | <p>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:</p> <p>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.</p> <p>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.</p> <p>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.</p> |
| QEZMSTGOPT | SMALLINT | <p>Opcja ta służy do określenia sposobu przydzielenia i użycia pamięci głównej w systemie dla wskazanego obiektu. Opcję tę można podać tylko dla plików strumieniowych w systemach plików "root" (/), QOpenSys i użytkownika (UDFS). Poprawne wartości:</p> <p>0 - Pamięć główna zostanie przydzielona normalnie. Przydzielone i użyte zostanie jak najwięcej pamięci głównej. Spowoduje to zminimalizowanie liczby operacji we/wy dla dysku, ponieważ informacje będą przechowywane w pamięci głównej.</p> <p>1 - Pamięć główna będzie przydzielana tak, aby zminimalizować przestrzeń zajmowaną przez obiekt. Przydzielone i użyte zostanie jak najmniej pamięci głównej. Spowoduje to zminimalizowanie zużycia pamięci głównej i zwiększenie liczby operacji we/wy dla dysku, ponieważ mniej informacji będzie przechowywane w pamięci głównej.</p> <p>2 - System dynamicznie określi optymalny przydział pamięci głównej dla obiektu w oparciu o pozostałe działania systemu i rywalizację o pamięć główną. Oznacza to, że gdy rywalizacja o pamięć główną jest niewielka, system przydzieli jak najwięcej pamięci w celu zminimalizowania liczby operacji dyskowych wejścia/wyjścia. Kiedy rywalizacja o pamięć główną jest znaczna, system przydzieli mniej pamięci głównej w celu zminimalizowania rywalizacji. Opcja ta ma znaczenie tylko wtedy, gdy opcja stronicowania puli pamięci ma wartość *CALC. Gdy opcja stronicowania puli pamięci ma wartość *FIXED, system działa tak samo, jak dla opcji STG_NORMAL. Gdy dostęp do obiektu uzyskiwany jest poprzez serwer plików, opcja ta nie ma znaczenia. Jej działanie jest takie samo, jak dla opcji STG_NORMAL.</p> |

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. Więcej informacji zawiera sekcja "Katalogi *TYPE2" na stronie 9. |
| QEZFILTYP2 ¹ | SMALLINT | Format pliku strumieniowego (*STMF). Poprawne wartości: 0 - Plik strumieniowy ma format *TYPE1. 1 - Plik strumieniowy ma format *TYPE2. Więcej informacji zawiera sekcja "Plik strumieniowy" na stronie 14. |
| 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. Więcej informacji zawiera sekcja "Plik strumieniowy" na stronie 14. |
| 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. |
| 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. |
| QEZCASE | SMALLINT | Wskazuje rozróżnianie wielkości znaków systemu plików, w którym znajduje się obiekt. 0 - System plików nie rozróżnia wielkości znaków. 1 - System plików rozróżnia wielkość znaków. |
| QEZOFLOW | SMALLINT | Wskazuje, czy obiekt spowodował przepełnienie puli pamięci, w której się znajduje. Poprawne wartości: 0 - Pula pamięci dyskowej nie jest przepełniona. 0 - Pula pamięci dyskowej jest przepełniona. |
| QEZPCREAD | SMALLINT | Określa, czy obiekt może być zapisany lub usunięty, jeśli jego atrybuty rozszerzone zostały zapisane lub usunięte albo zmieniono jego wielkość. Poprawne wartości: 0 - Obiekt można zmienić. 1 - Obiektu nie można zmienić. |

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

| Nazwa pola | Typ pola | Opis pola |
|-----------------------|-------------|---|
| QEZPCHID ¹ | SMALLINT | Określa, czy obiekt można wyświetlić przy użyciu zwykłego listingu katalogów. 0 - Obiekt nie jest ukryty. 1 - Obiekt jest ukryty. |
| QEZPCSYS | SMALLINT | Określa, czy obiekt jest plikiem systemowym i jest wyłączony z normalnego wyszukiwania katalogów. 0 - Obiekt nie jest plikiem systemowym. 1 - Obiekt jest plikiem systemowym. |
| QEZPCARC | SMALLINT | Określa, czy obiekt zmienił się od ostatniego sprawdzenia obiektu. 0 - Obiekt nie zmienił się. 1 - Obiekt zmienił się. |
| QEZSYSARC | SMALLINT | Określa, czy obiekt zmienił się i powinien być składowany. Gdy czas zmiany obiektu jest aktualizowany, atrybut ten jest włączany, a po składowaniu obiektu atrybut jest wyłączany. 0 - Obiekt nie zmienił się i nie musi być składowany. 1 - Obiekt zmienił się i musi być składowany. |
| QEZJRCVNAM | GRAPHIC(10) | Najstarszy dziennik wymagany do wprowadzenia zmian w dzienniku. Gdy pole informacji o wprowadzeniu zmian zawiera wartość PARTIAL_TRANSACTION, dziennik zawiera dane początku transakcji częściowej. W przeciwnym przypadku dziennik zawiera rozpoczęcie operacji składowania. |
| QEZJRCVLIB | GRAPHIC(10) | Nazwa biblioteki, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. |
| QEZJRCVASP | GRAPHIC(10) | Nazwa puli ASP, która zawiera najstarszy dziennik wymagany do pomyślnego wprowadzenia zmian w dzienniku. Poprawne wartości: *SYSBAS - Dziennik znajduje się w puli ASP systemowej lub użytkownika. Urządzenie ASP - Nazwa urządzenia ASP, które zawiera dziennik. |
| QEZJTRNI | 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 - WRKCMTDFN). 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. |

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

| Nazwa pola | Typ pola | Opis pola |
|--|----------|-----------|
| Uwaga: | | |
| 1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRDIRINF. | | |
| 2. W polu tym przechowywana jest tylko nazwa obiektu. Pozostałą część nazwy ścieżki zawiera Tabela 9, w polu QEZDIRNAM1, jeśli długość nazwy katalogu jest mniejsza niż 1 kB lub polu QEZDIRNAM2, jeśli długości nazw przekraczają 1 kB. | | |

Tabela 9 jest przykładową tabelą zawierającą katalogi przetworzone przez komendę RTVDIRINF.

Tabela 9. QAEZDxxxD (Składowanie atrybutów katalogów)

| Nazwa pola | Typ pola | Opis pola |
|---|-------------------|---|
| QEZDIRIDX | INTEGER | Identyfikator nazwy ścieżki (dotyczy tylko katalogów). |
| QEZDIRNAM1 ¹ | VARGRAPHIC (1024) | Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki nie przekracza 1 kB. |
| QEZDIRNAM2 ¹ | DBCLOB (16M) | Ścieżka katalogu nadrzędnego. Używana tylko wtedy, gdy długość ścieżki przekracza 1 kB. Może pomieścić ścieżki do 16 MB długości. |
| QEZRCCSID | INTEGER | Identyfikator CCSID katalogu. |
| QEZDREGION | GRAPHIC (2) | Identyfikator regionu ścieżki katalogu. |
| QEZLANGID | GRAPHIC (3) | Identyfikator języka ścieżki katalogu. |
| QEZDIRLEN ¹ | INTEGER | Długość nazwy ścieżki katalogu. |
| QEZDIRFID | GRAPHIC (16) | Identyfikator zbioru katalogu. Identyfikator przypisany do obiektu. ID zbioru można użyć w funkcji Qp0lGetPathFromFileID() do pobrania nazwy ścieżki obiektu. |
| QEZDFID | INTEGER | Identyfikator zbioru katalogu. |
| QEZDIRFSID | BIGINT | Systemowy identyfikator zbioru katalogu. |
| QEZDIRGID | BIGINT | Identyfikator pokolenia. |
| Uwaga: | | |
| 1. Pole to jest dołączane do podzestawu pól używanych przez komendę PRDIRINF. | | |

Tabela 10 jest przykładową tabelą tworzoną po wpisaniu komendy RTVDIRINF. Tabela ta służy komendzie PRDIRINF do rozróżnienia, które zbiory baz danych posłużyły do zapisania informacji pobranych przez różne instancje komendy RTVDIRINF.

Tabela 10. QUSRSYS/QAEZDBFILE (Składowanie utworzonych zbiorów)

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

|
| **Dostęp do danych RTVDIRINF:** Dostęp do danych tabeli można uzyskać na kilka sposobów. Poniżej znajdują się sposoby dostępu do danych utworzonych przez komendę RTVDIRINF:

- Używanie komendy 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. Przykładami takich metod są: wbudowany SQL i interfejsy CLI SQL.

| **Użycie danych 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 63 dopuszcza tworzenie zapytań w celu utworzenia raportów statystyk w oparciu o dowolne pozycje tej tabeli. PRTDIRINF nie obejmuje raportów opartych na wszystkich polach tabeli. Użyty zostanie podzestaw pól.
- Tabela 9 na stronie 73 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 63 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/QAEZxxxxO,  
nazwa_biblioteki/QAEZxxxxD WHERE QEZDIRNAM1 = "/MYDIR" AND  
nazwa_biblioteki/QAEZxxxxO.QEZDIRIDX=nazwa_biblioteki/QAEZxxxxD.QEZDIRIDX
```
- Tabela 10 na stronie 73 używana jest głównie przez komendę PRTDIRINF 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. Szczegóły dotyczące używania funkcji API ze zintegrowanym systemem plików zawiera sekcja Wykonywanie operacji z użyciem funkcji API.

Dostęp przy użyciu 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 sekcje opisujące niektóre najczęściej wykonywane zadania systemu plików:

Praca z plikami i folderami

- “Tworzenie folderu” na stronie 107
- “Usuwanie folderu” na stronie 107
- “Odblokowanie pliku” na stronie 107
- “Zablokowanie pliku” na stronie 107
- “Nadawanie uprawnień” na stronie 109
- “Konfigurowanie konwersji plików tekstowych” na stronie 109
- “Wysyłanie pliku lub folderu do innego systemu” na stronie 109
- “Zmiana opcji definicji pakietów” na stronie 110
- “Planowanie daty i godziny wysyłki pliku lub folderu” na stronie 110
- “Włączanie i wyłączanie skanowania obiektów” na stronie 111

Praca z współużytkowanymi zasobami plikowymi

- “Tworzenie współużytkowanego zasobu plikowego” na stronie 110
- “Zmiana współużytkowanego zasobu plikowego” na stronie 110

Praca z systemami plików użytkownika

- “Utworzenie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Podłączanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34
- “Odłączanie systemu plików UDFS w zintegrowanym systemie plików” na stronie 34

Kronikowanie obiektów

- “Uruchomienie kronikowania” na stronie 86
- “Zakończenie kronikowania” na stronie 86

Dostęp przy użyciu iSeries NetServer

iSeries Support for Windows Network Neighborhood (iSeries NetServer) jest funkcją systemu IBM Operating System/400 (OS/400), która umożliwia klientom systemu Windows dostęp do ścieżek współużytkowanych katalogów OS/400 i współużytkowanych kolejek wyjściowych. 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 LINUX z zainstalowanym oprogramowaniem klienckim Samba również mogą uzyskać dostęp do danych i drukarek za pomocą iSeries NetServer. Współużytkowane katalogi iSeries NetServer można podłączać w klientach systemu Linux jako systemy plików Samba (smbfs) podobnie, jak podłącza się systemy plików NFS wyeksportowane z systemu iSeries. Więcej informacji zawiera sekcja iSeries NetServer w Centrum informacyjnym iSeries.

Współużytkowany zasób plikowy iSeries NetServer jest ścieżką do katalogu, którą iSeries NetServer udostępnia klientom PC w sieci iSeries. Współużytkowany zasób plikowy może składać się z dowolnych katalogów zintegrowanego systemu plików w iSeries. Zanim możliwe będzie korzystanie z funkcji współużytkowania plików z użyciem systemu iSeries NetServer, należy utworzyć współużytkowany zasób plikowy iSeries NetServer i jeśli to konieczne, zmienić współużytkowany zasób plikowy iSeries NetServer używając narzędzia iSeries Navigator.

Aby uzyskać dostęp do współużytkowanych zasobów plikowych zintegrowanego systemu plików za pomocą 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**.

- Wybierz dla współużytkowanego zasobu literę oznaczającą wolny napęd (na przykład napęd I:).
- Podaj nazwę współużytkowanego zasobu plikowego 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.

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

Znacznie większą wydajność udostępniania plików dla klienta PC w porównaniu z innymi systemami plików iSeries zapewnia bazowy system plików "root" (/). Użytkownik może przenosić pliki z bazowego systemu plików "root" (/). Więcej informacji zawiera sekcja "Przenoszenie plików lub folderów do innego systemu plików" na stronie 108.

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

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

Dostęp przy użyciu programów przesyłania plików

Klient protokołu File Transfer Protocol (FTP) umożliwia przesłanie plików znajdujących się w serwerem iSeries, łącznie z plikami z systemu plików "root" (/), QOpenSys, QSYS.LIB, Independent ASP QSYS.LIB, QOPT i QFileSvr.400. Pozwala także na przesyłanie folderów i dokumentów w systemie plików usługi biblioteki dokumentów (QDLS). Klienta FTP można uruchomić interaktywnie w trybie nienadzorowanego zadania wsadowego, w którym komendy są czytane z pliku, a odpowiedzi na nie zapisywane do pliku. Zapewnia on także inne funkcje umożliwiające przetwarzanie plików na serwerze, na którym działa.

Obsługę protokołu FTP można używać przy przesyłaniu plików do i z jednego z poniższych systemów plików:

- system plików "root" (/),
- system plików systemów otwartych (QOpenSys),
- system plików bibliotek (QSYS.LIB),
- system plików QSYS.LIB niezależnej puli ASP,
- system plików obsługujący obiekty biblioteki dokumentów (QDLS),
- system plików nośników optycznych (QOPT),
- sieciowy system plików (Network File System - NFS),
- system plików NetWare (QNetWare),
- system plików 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.

Informacje o protokole FTP zawierają następujące sekcje kategorii **Sieci** w Centrum informacyjnym iSeries:

- 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 tak, 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 z użyciem interfejsu systemu Windows mogą być automatycznie poddane konwersji z kodu EBCDIC na 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. Od systemu w wersji OS/400 V4R4, iSeries NetServer można także skonfigurować tak, aby przeprowadzał konwersję plików.

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 Windows w celu uzyskania dostępu do systemu iSeries NetServer, działającego w serwerem iSeries.

Przesyłanie plików za pomocą protokołu FTP

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

Praca z plikami za pomocą programu iSeries Navigator

iSeries Access Family zawiera iSeries Navigator, który łączy się z serwerem iSeries i sprawia, że zintegrowany system plików jest dostępny dla komputera PC. iSeries Navigator jest graficznym interfejsem użytkownika służącym do zarządzania i administrowania serwerem iSeries z pulpitu systemu Windows.

Praca z plikami za pomocą iSeries NetServer

iSeries NetServer jest częścią systemu OS/400, która umożliwia współpracę klientów współużytkownika plików i drukarek wbudowanych w system Windows z serwerem.

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

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 V5R3 system rozpocznie automatyczną konwersję katalogów do formatu *TYPE2 dla wszystkich systemów plików, które jeszcze nie zostały poddane tej konwersji. Nie powinna ona poważnie wpłynąć na działanie systemu.

- “Przegląd konwersji formatu *TYPE1 do *TYPE2”
- “Wskazówki: konwersje”

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 sekcji “Obsługa skanowania” na stronie 17.

Bezpośrednio po zainstalowaniu systemu w wersji V5R3 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.

Więcej informacji o sprawdzaniu, czy funkcja konwersji została pomyślnie zakończona, zawiera sekcja “Określanie statusu konwersji” na stronie 79.

Systemy plików, które można poddać automatycznej konwersji, to "root" (/), QOpenSys i system plików użytkownika dla pul ASP od 1 do 32. Więcej informacji o systemach plików użytkownika w niezależnych ASP zawiera temat “Wskazówki: niezależne ASP” na stronie 81.

Uwaga: Można uniknąć automatycznej konwersji do formatu katalogów *TYPE2, jeśli konwersja ta zostanie przeprowadzona ręcznie przed zainstalowaniem systemu w wersji V5R3. Patrz sekcja “Używanie katalogów *TYPE2 w systemach OS/400 w wersji V5R1 lub V5R2” na stronie 10.

Wskazówki: konwersje

Bezpośrednio po zainstalowaniu systemu w wersji V5R3 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.

Podczas przeprowadzania konwersji, należy wziąć pod uwagę następujące zagadnienia:

- “Określanie statusu konwersji” na stronie 79
- “Tworzenie profili użytkowników” na stronie 79
- “Zmiana nazw obiektów” na stronie 79
- “Uwagi na temat profilu użytkownika” na stronie 80
- “Wymagania pamięci dyskowej” na stronie 80
- “Wskazówki: dowiązanie symboliczne” na stronie 80
- “Wskazówki: niezależne ASP” na stronie 81
- “Wskazówki: składowanie i odtwarzanie” na stronie 81
- “Odzyskanie pamięci (Reclaim Storage - RCLSTG)” na stronie 81
- “Skanowanie zintegrowanego systemu plików” na stronie 81

Określanie statusu konwersji

Bezpośrednio po zainstalowaniu systemu w wersji V5R3 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 sekcja CVTDIR.

Ponieważ zadanie QFILESYS1 przetwarza konwersję, można wyświetlić protokół zadania QFILESYS1 i przejrzeć ewentualne komunikaty o błędach w procesie konwersji. Dodatkowo, do tego 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.

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.

Zmiana nazw obiektów

Katalogi *TYPE2 wymagają, aby nazwy dowiązań były poprawnymi nazwami UTF-16. Katalogi *TYPE1 mają nazwy UCS2 Poziom 1. Dlatego podczas konwersji katalogów mogą pojawiać się nieprawidłowe lub zduplikowane nazwy. 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 na temat znaków zawiera sekcja "Znaki złożone" lub "Znaki odpowiedniki" na stronie 80

Więcej informacji o UTF-16 można znaleźć na stronie głównej Unicode (<http://www.unicode.org> ).

Znaki złożone: Niektóre znaki mogą składać się z więcej niż jednego znaku Unicode. Na przykład są to znaki akcentowane lub umlaut. Te znaki muszą być zmienione lub znormalizowane do wspólnego formatu zanim zostaną umieszczone w katalogu, aby wszystkie obiekty miały unikalne nazwy. Normalizacja znaków złożonych jest procesem, w którym znak jest przekształcany do nowego i przewidywalnego formatu. Format wybrany dla katalogu *TYPE2 to złożona postać kanoniczna. Jeśli w katalogu *TYPE1 znajdują się dwa obiekty zawierające te same znaki złożone, są one normalizowane do tej samej nazwy. Powoduje to kolizję, nawet jeśli jeden obiekt zawiera połączone złożone znaki, a drugi zawiera rozłączone złożone znaki. Dlatego jeden z nich musi mieć zmienioną nazwę zanim zostanie dowiązany w katalogu *TYPE2.

Znaki odpowiedniki: Niektóre znaki nie mają prawidłowych odpowiedników w systemie Unicode. Znaki te mają specjalne wartości, które składają się z dwóch znaków Unicode, w dwóch określonych przedziałach, z których pierwszy znak znajduje się w pierwszym zakresie (na przykład 0xD800-0xD8FF), a drugi w drugim zakresie (na przykład 0xDC00-0xDCFF). Nazywa się to parą odpowiedników. Jeśli jeden ze znaków Unicode zostanie usunięty lub są one podane w złej kolejności (tylko jeden znak), nazwa jest nieprawidłowa. Nazwy tego typu są dozwolone w katalogach *TYPE1, ale nie są dozwolone w katalogach *TYPE2. Aby konwersja mogła być kontynuowana, jeśli nazwa zawiera jeden z nieprawidłowych znaków, jest zmieniana zanim obiekt zostanie dowiązany w katalogu *TYPE2.

Uwagi na temat profilu użytkownika

Podczas konwersji system próbuje zapewnić, aby ten sam profil użytkownika, który jest właścicielem dowolnego katalogu *TYPE1 był również właścicielem odpowiedniego katalogu *TYPE2. Ponieważ katalogi *TYPE1 i *TYPE2 przez pewien czas istnieją jednocześnie, ma to wpływ na wielkość pamięci należąca do profilu użytkownika i liczbę pozycji w profilu użytkownika.

Więcej informacji o profilach użytkowników zawierają następujące tematy.

- “Zmiana maksymalnej ilości pamięci dla profilu użytkownika”
- “Zmiana właściciela katalogu”

| **Zmiana maksymalnej ilości pamięci dla profilu użytkownika:** Podczas konwersji katalogu, pewna liczba
| katalogów istnieje w obu formatach jednocześnie i ich właścicielem jest ten sam profil użytkownika. Jeśli 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 opisany w sekcji Tworzenie profili użytkowników. Do protokołu zadania QFILESYS1 i kolejki
| komunikatów QSYSOPR wysyłany jest komunikat CPIA08B, a konwersja jest kontynuowana.

Wymagania pamięci dyskowej

| Wymagania pamięci dyskowej należy rozważyć, gdy w systemie rozpocznie się konwersja katalogów w systemie
| plików do formatu *TYPE2. Istnieje kilka problemów odnoszących się do wymaganej przestrzeni dyskowej:

- końcowa wielkość katalogów po konwersji do formatu *TYPE2,
- dodatkowa przestrzeń dyskowa wymagana, gdy funkcja konwersji jest uruchomiona.

W wielu przypadkach, końcowa wielkość katalogu *TYPE2 jest mniejsza niż katalogu *TYPE1. Zazwyczaj katalogi *TYPE2 zawierające mniej niż 350 obiektów wymagają mniej pamięci dyskowej niż katalogi *TYPE1 zawierające tę samą liczbę obiektów. Katalogi *TYPE2 zawierające więcej niż 350 obiektów są o 10 procent większe (średnio) niż katalogi *TYPE1.

| Gdy funkcja konwersji jest uruchomiona, wymagana jest dodatkowa przestrzeń dyskowa. Funkcja konwersji wymaga,
| aby obie wersje katalogów, w formacie *TYPE1 i *TYPE2 istniały jednocześnie.

| **Uwaga:** Zanim rozpocznie się instalacja wersji V5R3, należy rozważyć uruchomienie komendy V5R2 (CVTDIR) z
| opcją *ESTIMATE, która dostarczy dokładnych danych szacunkowych ilości wolnej pamięci dyskowej
| wymaganej do przeprowadzenia operacji konwersji.

Wskazówki: dowiązanie symboliczne

Dowiązania symboliczne to obiekty w zintegrowanym systemie plików, które zawierają 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. Szczegóły dotyczące zmiany nazw obiektów można znaleźć w sekcji Obiekty o zmienionej nazwie.

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)
```

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.

Odzyskanie pamięci (Reclaim Storage - RCLSTG)

Gdy system przetwarza systemy plików "root" (/), QOpenSys i UDFS w puli ASP użytkownika do obsługi formatu katalogów *TYPE2, nie można uruchomić komendy RCLSTG w żadnym katalogu zintegrowanego systemu plików, również w katalogach niezależnej ASP. Można jednak użyć parametru OMIT(*DIR), aby pominąć katalogi zintegrowanego systemu plików i umożliwić odzyskanie obiektów nie związanych z katalogami. Aby sprawdzić, kiedy systemy plików zostaną w pełni przetworzone, aby możliwe było uruchomienie operacji RCLSTG dla katalogów zintegrowanego systemu plików, należy przejrzeć temat "Określanie statusu konwersji" na stronie 79.

Skanowanie zintegrowanego systemu plików

Obiekty w systemie plików "root" (/), QOpenSys i UDFS ASP użytkownika nie będą skanowane z użyciem punktów wyjścia związanych ze skanowaniem zintegrowanego systemu plików do czasu zakończenia konwersji systemów do formatu katalogów *TYPE2. Atrybuty związane ze skanowaniem można przypisać do obiektów w katalogach *TYPE1 i *TYPE2 w celu określenia, czy obiekty mają być skanowane, czy nie, nawet jeśli system plików nie został w pełni przetworzony. Więcej informacji o obsłudze skanowania zintegrowanego systemu plików zawiera temat "Obsługa skanowania" na stronie 17.

Gdy system przetwarza obiekty z formatu *TYPE1 do *TYPE2, uwzględni wartość systemową sterowania skanowaniem 'Skanowanie przy następnym dostępie po odtworzeniu obiektu' (Scan on next access after object has been restored) tak, jakby obiekty były odtwarzane. Na przykład, jeśli wartość 'Skanowanie przy następnym dostępie po odtworzeniu obiektu' została podana w trakcie konwersji, wówczas obiekt z atrybutem 'obiekt nie będzie skanowany' w formacie katalogu *TYPE1 zostanie poddany skanowaniu przynajmniej raz po zakończeniu pełnej konwersji. Więcej informacji o wartościach systemowych związanych ze skanowaniem zawiera sekcja "Pokrewne wartości systemowe" na stronie 18.

Aby sprawdzić, kiedy systemy plików zostaną w pełni przetworzone, aby możliwe było uruchomienie operacji skanowania zintegrowanego systemu plików, należy przejrzeć temat "Określanie statusu konwersji" na stronie 79.

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

Więcej informacji o kronikowaniu obiektów zawierają następujące tematy.

- "Przegląd kronikowania" na stronie 82
- "Uruchomienie kronikowania" na stronie 86
- "Zmiana kronikowania" na stronie 86

- “Zakończenie kronikowania” na stronie 86

Przegląd kronikowania

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

- “Zarządzanie kronikami”
- “Jakie obiekty należy uwzględnić w kronikowaniu”
- “Kronikowane obiekty zintegrowanego systemu plików”
- “Kronikowane operacje” na stronie 84
- “Specjalne zagadnienia dotyczące pozycji kronik” na stronie 84
- “Zagadnienia dotyczące używania wielu napędów dysków i kronikowania” na stronie 85

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

Zarządzanie kronikami

Podstawowym celem zarządzania kronikami jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Zarządzanie kronikami może również służyć do:

- zapisów kontrolnych dla działań dotyczących obiektów w systemie;
- rejestrowania innych niż kronikowane działań dotyczących obiektów;
- szybszego odtwarzania z nośników, na których przeprowadza się składowanie podczas użycia;
- asysty 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. Więcej uwag dotyczących kronikowania obiektów zawiera sekcja “Jakie obiekty należy uwzględnić w kronikowaniu”. W zintegrowanym systemie plików można kronikować pliki strumieniowe, katalogi i dowiązania symboliczne. Obsługiwane są tylko obiekty następujących systemów plików: bazowego (root, "/"), QOpenSys i UDFS.

Jakie obiekty należy uwzględnić w kronikowaniu

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

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

Kronikowane obiekty zintegrowanego systemu plików

Obsługa kronikowania w systemie OS/400 umożliwia kronikowanie niektórych typów obiektów zintegrowanego systemu plików. Obsługiwane typy obiektów to pliki strumieniowe, katalogi i dowiązania symboliczne. Jedynymi systemami plików obsługującymi kronikowanie obiektów tych typów są bazowy system plików (/), QOpenSys i UDFS. Obiekty zintegrowanego systemu plików mogą być kronikowane zarówno poprzez tradycyjny interfejs systemowy

(komendy CL lub funkcje API), jak i z wykorzystaniem programu iSeries Navigator. Operacje Uruchomienie kronikowania i Zakończenie kronikowania można wykonać w aplikacji iSeries Navigator, podobnie jak można wyświetlić informacje o statusie kronikowania.

Uwaga: Odzworowane w pamięci pliki strumieniowe oraz pliki strumieniowe wykorzystywane przez Integrated xSeries Server for iSeries (IXS) jako przestrzeń dyskowa napędu wirtualnego nie mogą być kronikowane. Katalogi, które mogą zawierać blokowe obiekty plików specjalnych, nie mogą być kronikowane. Przykładami są: /dev/QASP01, /dev/QASP22 i /dev/IASPNAM.

Poniższa lista przedstawia podsumowanie dotyczące obsługi kronikowania w zintegrowanym systemie plików:

- Do wykonywania operacji kronikowania na obsługiwanych typach obiektów można używać komend ogólnych i funkcji API. Interfejsy te akceptują identyfikację obiektów poprzez nazwę ścieżki, identyfikator zbioru lub obie te metody jednocześnie.
- Niektóre komendy 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.
 - Maksymalny limit zapisu obsługiwany przez różne interfejsy programów piszących to 2 gigabajty-1. Maksymalna wielkość pozycji kroniki, jeśli podano wartość RCVSIZOPT (*MAXOPT2 lub *MAXOPT3) wynosi 4 000 000 000. W przeciwnym razie maksymalna wielkość pozycji kroniki wynosi 15 761 440 bajtów. Jeśli kronikowany jest plik strumieniowy, a programy piszące przekraczają wielkość 15 761 440 bajtów, należy użyć obsługi *MAXOPT2 lub *MAXOPT3, aby uniknąć wystąpienia błędów.

Bardziej szczegółowe informacje dotyczące kronikowania obiektów, które zawiera zintegrowany system plików, można znaleźć w temacie Zarządzanie kronikowaniem.

Szczegóły dotyczące zawartości i formatów danych właściwych dla pozycji kronik zintegrowanego systemu plików, a także informacje o różnych układach pozycji kronik zawiera plik w języku C o nazwie qp0ljrnl.h, dostarczany w podzbiorze QSYSINC/H (QP0LJRNL).

Pełną listę wszystkich pozycji kronikowania utworzonych dla obiektów zintegrowanego systemu plików i definicji wyżej wspomnianych układów można znaleźć w sekcji Wyszukiwarka informacji pozycji kronikowania w temacie Zarządzanie kronikowaniem.

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

Bardziej szczegółowe informacje dotyczące kronikowania obiektów zintegrowanego systemu plików można znaleźć w temacie Zarządzanie kronikowaniem. Pełną listę wszystkich pozycji kronikowania utworzonych dla obiektów zintegrowanego systemu plików można znaleźć w sekcji Wyszukiwarka informacji pozycji kronikowania w temacie Zarządzanie kronikowaniem.

Specjalne zagadnienia dotyczące pozycji kronik

Wiele kronikowanych operacji w systemie zintegrowany system plików korzysta z kontroli transakcji w celu uformowania pojedynczej transakcji z wielu funkcji wykonywanych podczas działania. Tych kronikowanych operacji nie można uznać za kompletne, dopóki w cyklu kontroli transakcji nie pojawi się pozycja kroniki Zatwierdzenie (Commit) (Kod kroniki C, Typ CM). Operacje kronikowane zawierające pozycję kroniki Wycofanie (Rollback) (Kod kroniki C, Typ RB) w cyklu kontroli transakcji to operacje, które się nie powiodły, a pozycje kronik dla tych operacji nie powinny być powtarzane ani replikowane.

Kronikowane pozycje zintegrowanego systemu plików (Kod kroniki B) używające w ten sposób kontroli transakcji obejmują:

- AA — Zmianę wartości kontroli (Change Audit Value),
- B0 — Rozpoczęcie tworzenia (Begin Create),
- B1 — Utworzenie podsumowania (Create Summary),
- B2 — Dodanie dowiązania (Add link),
- B3 — Zmianę nazwy/Przesunięcie (Rename/Move),
- B4 — Usunięcie dowiązania (Katalog nadrzędny) (Unlink (Parent Directory)),
- B5 — Usunięcie dowiązania (Dowiązanie) (Unlink (Link)),
- FA — Zmianę atrybutu (Attribute Change),
- JT — Uruchomienie kronikowania (Start Journal) (tylko gdy kronikowanie się rozpoczyna, ponieważ operacja została wykonana w katalogu z włączonym atrybutem kronikowania),
- OA — Zmianę uprawnień (Authority Change),
- OG — Zmianę grupy głównej obiektu (Object Primary Group Change),
- OO — Zmianę właściciela obiektu (Object Owner Change).

Kilka pozycji kronik dla zintegrowanego systemu plików ma specyficzne pola wskazujące, czy dana pozycja jest pozycją podsumowania. Operacje wysyłające pozycje kronik zawierające podsumowania, przesyłają dwie pozycje do kroniki. Pierwsza z nich zawiera podzbiór danych właściwych dla danej pozycji. Druga pozycja zawiera pełny zbiór danych właściwych dla pozycji i jest oznaczona jako pozycja podsumowania. Programy, które powielają obiekt lub ponownie wykonują operację, będą głównie zainteresowane pozycjami podsumowań.

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

Niektóre operacje kronikowane muszą wysyłać pozycje kronik, które są przeciwne do tych operacji. Na przykład cykl kontroli transakcji zawierający pozycje kroniki B4 (Usunięcie dowiązania - Unlink) może również zawierać pozycję B2 (Dodanie dowiązania - Add Link). Jednak taki przypadek wystąpi tylko podczas operacji, w wyniku których pojawi się pozycja kroniki Wycofanie (Rollback) (C — RB).

Może się tak zdarzyć z dwóch powodów:

1. Wystąpiły problemy podczas operacji i pozycja ta była potrzebna wewnętrznie w procedurze czyszczącej błąd ścieżki,
2. Operacja została przerwana przez wyłączenie systemu i podczas procedury IPL wykonano odtwarzanie konieczne do wysłania pozycji w celu wycofania przerwanej operacji.

| **Zagadnienia dotyczące używania wielu napędów dysków i kronikowania**

| Informacje o dowiązaniach stałych, ich definicje i sposób działania, zawiera temat "Dowiązanie stałe" na stronie 11.

| 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.

Uruchomienie kronikowania

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

- | Aby uruchomić kronikowanie obiektu przy użyciu iSeries Navigator:
 - | 1. Wyświetl system w aplikacji **iSeries Navigator**.
 - | 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.

Bardziej szczegółowe informacje dotyczące kronikowania obiektów zintegrowanego systemu plików zawiera temat Zarządzanie kronikowaniem w Centrum informacyjnym iSeries.

Zmiana kronikowania

- | Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o uruchamianiu kronikowania obiektu zawiera temat Uruchamianie kronikowania. Gdy kronikowanie zostanie uruchomione dla obiektu, można zmienić atrybuty kroniki dla obiektu bez konieczności zatrzymywania i ponownego uruchamiania kronikowania. Komenda Zmiana kronikowanego obiektu (Change Journal Object - CHGJRNOBJ) umożliwia zmianę kronikowanych obiektów.

Zakończenie kronikowania

Podstawowym celem kronikowania jest umożliwienie odtworzenia zmian obiektu dokonanych od czasu ostatniego składowania tego obiektu. Więcej informacji o uruchamianiu kronikowania obiektu zawiera temat Uruchamianie kronikowania. Gdy kronikowanie obiektu zostanie uruchomione, istnieje możliwość przerwania kronikowania tego obiektu.

- | Aby zatrzymać kronikowanie obiektu przy użyciu iSeries Navigator:
 - | 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.

Bardziej szczegółowe informacje dotyczące kronikowania obiektów zintegrowanego systemu plików zawiera temat Zarządzanie kronikowaniem w Centrum informacyjnym iSeries.

Obsługa programowania

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

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

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

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

- Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych
- Kopiowanie danych pomiędzy plikami strumieniowymi a zbiorami składowania
- Wykonywanie operacji za pomocą funkcji API
- Obsługa gniazd
- Nazewnictwo i obsługa międzynarodowa
- Konwersja danych
- Przykład: funkcje C zintegrowanego systemu plików

Kopiowanie danych pomiędzy plikami strumieniowymi i zbiorami baz danych

Użytkownicy znający zasady pracy ze zbiorami baz danych za pomocą narzędzi do przetwarzania rekordów, takich jak język opisu struktur danych (DDS), mogą stwierdzić istnienie zasadniczych różnic w pracy z plikami strumieniowymi. Wynikają one z innej struktury (lub też braku struktury) plików strumieniowych w porównaniu ze zbiorami baz danych. Dostęp do pliku strumieniowego uzyskuje się podając przesunięcie w bajtach (byte offset) i długość. Zwykle dostęp do bazy danych uzyskuje się określając, które pola mają zostać użyte i ile rekordów ma zostać przetworzonych.

Ponieważ format i charakterystyka pliku opartego na rekordach definiowane są z góry, system operacyjny zna układ pliku i jest w stanie pomóc uniknąć działań niewłaściwych dla danego formatu i charakterystyki pliku. W przypadku plików strumieniowych system operacyjny ma niewielką lub wręcz żadną wiedzę na temat formatu pliku. Program użytkowy musi wiedzieć, jak wygląda plik i jak poprawnie wykonywać na nim działania. Pliki strumieniowe oferują niezwykle elastyczne środowisko programistyczne, ale dzieje się to kosztem rezygnacji z pomocy systemu operacyjnego. Pliki strumieniowe bardziej nadają się do użycia w niektórych przypadkach programowania; w innych przypadkach lepiej jest użyć plików opartych na rekordach.

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

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

Kopiowanie danych za pomocą komend CL

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

- CPYTOSTMF i CPYFRMSTMF,
- CPYTOIMPF i CPYFRMIMPF.

Komendy CPYTOSTMF i CPYFRMSTMF

Do kopiowania danych pomiędzy plikami strumieniowymi a podzbiorami baz danych można używać komendy Kopiowanie z pliku strumieniowego (Copy from Stream File- CPYFRMSTMF) i Kopiowanie do pliku strumieniowego (Copy to Stream File - CPYTOSTMF). Plik strumieniowy można utworzyć z podzbioru bazy danych za pomocą komendy CPYTOSTMF. Można też utworzyć podzbiór bazy danych z pliku strumieniowego, używając komendy CPYFRMSTMF. Jeśli zbiór lub podzbiór, do którego są kopiowane dane, nie istnieje, to zostanie on utworzony.

Pojawiają się jednak pewne ograniczenia. Zbiór bazy danych musi być albo zbiorem fizycznym opisywanym przez program i zawierającym tylko jedno pole, albo źródłowym zbiorem fizycznym zawierającym tylko jedno pole tekstowe. Omawiane komendy obejmują wiele opcji konwersji i zmiany formatu kopiowanych danych.

Komendy CPYTOSTMF i CPYFRMSTMF mogą również zostać użyte do kopiowania danych między plikiem strumieniowym a zbiorem składowania.

Komendy CPYTOIMPF i CPYFRMIMPF

Do kopiowania danych między plikami strumieniowymi a podzbiórami baz danych można również użyć komend Kopiowanie do zbioru importu (Copy to Import File - CPYTOIMPF) i Kopiowanie ze zbioru importu (Copy from Import File - CPYFRMIMPF). Komendy CPYTOSTMF i CPYFRMSTMF uniemożliwiają przenoszenie danych ze złożonych, zewnętrznie opisanych (za pomocą DDS) zbiorów baz danych. Określenie *plik importu (import file)* odnosi się do pliku typu strumieniowego; termin ten zazwyczaj oznacza plik utworzony w celu kopiowania danych pomiędzy heterogenicznymi bazami danych.

Podczas kopiowania z pliku strumieniowego (lub pliku importu) komenda CPYFRMIMPF pozwala użytkownikowi określić plik definicji pola (field definition file - FDF), który opisuje dane zawarte w tym pliku strumieniowym. Użytkownik może też określić, czy plik strumieniowy jest ograniczony i jakie znaki zostaną użyte do oznaczenia granic łańcuchów, pól i rekordów. Dostępne są też opcje przekształcania specjalnych typów danych, takich jak data i godzina.

Konwersja danych jest w tych komendach dostępna, jeśli docelowy plik strumieniowy lub podzbiór bazy danych już istnieje. Jeśli ten plik nie istnieje, można wykorzystać przedstawioną poniżej dwustopniową metodę uzyskiwania przekształconych danych.

1. Użyj komend CPYTOIMPF i CPYFRMIMPF do skopiowania danych z pliku zewnętrznie opisanego do źródłowego zbioru fizycznego.
2. Komend CPYTOSTMF i CPYFRMSTMF (które przeprowadzają pełną konwersję danych bez względu na to, gdzie znajduje się plik docelowy) można użyć do skopiowania danych między fizycznym plikiem źródłowym, a plikiem strumieniowym.

A oto przykład:

```
CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML('X'07') STRDLML('') DATFMT(*USA) TIMFMT(*USA)
```

Parametr DTAFMT określa, czy strumień wejściowy (plik importu) jest ograniczony; można też wybrać parametr DTAFMT(*FIXED), który wymaga podania pliku definicji pola. Parametry FLDDLML, RCDDLML i STRDLML określają znaki, które działają jako ograniczniki lub separatory pól, rekordów i łańcuchów.

Parametry DATFMT i TIMFMT określają format wszelkich informacji dotyczących daty i godziny, które są kopiowane do pliku importu.

Komendy te są użyteczne, ponieważ mogą zostać umieszczone w programie i działają wyłącznie na serwerze. Jednak ich interfejsy są złożone.

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

Kopiowanie danych za pomocą funkcji API

Aby skopiować podzbiory zbioru bazy danych do pliku strumieniowego w aplikacji, należy użyć funkcji zintegrowanego systemu plików open(), read() i write() w celu otwarcia podzbioru, odczytania danych i zapisania ich w nim lub w innym pliku. W sekcji Funkcje API zintegrowanego systemu plików w Centrum informacyjnym iSeries można znaleźć więcej informacji.

Kopiowanie danych za pomocą funkcji przesyłania

Zaletami aplikacji przesyłania danych iSeries Access 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 zasobów komputera PC i serwera iSeries, a także korzystania z komunikacji pomiędzy nimi.

Jeśli na komputerze PC i na serwerze jest zainstalowany produkt iSeries Access Family, można go wykorzystać do przesyłania danych między plikami strumieniowymi i zbiorami baz danych. Można również przysłać dane do nowego zbioru bazy danych na podstawie istniejącego zbioru bazy danych, do zewnętrznie opisanego zbioru bazy danych lub do nowej definicji zbioru bazy danych i do nowego zbioru.

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

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

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

1. Połącz się z serwerem.
2. Przypisz dysk sieciowy do odpowiedniej ścieżki w systemie plików iSeries.
3. W oknie iSeries Access for Windows wybierz ikonę **Transfer danych z serwera iSeries**.
4. Wybierz serwer, z którego chcesz przysłać dane.
5. Używając baz danych biblioteki iSeries i nazwy pliku, wybierz nazwę pliku, z którego będziesz kopiował, a także napęd sieciowy dla wynikowego pliku strumieniowego. Możesz też wybrać przycisk **Szczegóły** w sekcji **PC** w celu wybrania formatu pliku PC dla pliku strumieniowego. Przesyłanie danych obsługuje najczęściej spotykane typy plików PC, takie jak: tekstowy ASCII, BIFF3, CSV, DIF, tekstowy ograniczony znakami tabulacji oraz WK4.
6. Kliknij przycisk **Przesyłanie danych z iSeries**, aby uruchomić przesyłanie plików.

Tego typu przesyłanie danych można też przeprowadzić uruchamiając aplikację do przesyłania danych w zadaniu wsadowym. W tym celu należy wykonać czynności opisane powyżej, ale z wybraniem opcji menu **Plik**, aby zapisać żądanie przesyłania. Aplikacja Transfer danych do serwera iSeries tworzy plik .DTT lub .TFR. Natomiast aplikacja Transfer danych z serwera iSeries tworzy plik .DTF lub .TTO. W katalogu iSeries Access 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.

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

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

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

Tego typu przesyłanie danych można też przeprowadzić uruchamiając aplikację do przesyłania danych w zadaniu wsadowym. W tym celu należy wykonać czynności opisane powyżej, ale z wybraniem opcji menu **Plik**, aby zapisać żądanie przesyłania. Aplikacja Transfer danych do serwera iSeries tworzy plik .DTT lub .TFR. Natomiast aplikacja Transfer danych z serwera iSeries tworzy plik .DTF lub .TTO. W katalogu IBM e(logo)server 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.

Przesyłanie danych do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru: Aby przesłać dane do nowo utworzonej definicji zbioru bazy danych i do nowego zbioru, wykonaj następujące czynności:

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

Zostanie uruchomiony kreator, który pomoże utworzyć nowy zbiór bazy danych iSeries z istniejącego pliku PC. Konieczne będzie podanie nazwy pliku PC, z którego zostanie utworzony zbiór iSeries, nazwy tworzonego zbioru iSeries oraz kilku innych potrzebnych szczegółów. To narzędzie analizuje składnię podanego pliku strumieniowego, w celu określenia liczby, typu i wielkości pól wymaganych w wynikowym zbiorze bazy danych. Narzędzia tego można następnie użyć do utworzenia definicji zbioru bazy danych na serwerze.

Tworzenie pliku opisu formatu: Jeśli dane są przesyłane do istniejącej definicji zbioru bazy danych, aplikacja Transfer danych do iSeries wymaga użycia powiązanego pliku opisu formatu (FDF). Plik FDF opisuje format pliku strumieniowego. Jest on tworzony przez aplikację Transfer danych z serwera iSeries po przesłaniu danych ze zbioru bazy danych do pliku strumieniowego.

Aby utworzyć plik .FDF, wykonaj następujące czynności:

1. Utwórz opisany zewnętrznie zbiór bazy danych o formacie odpowiadającym źródłowemu plikowi strumieniowemu (liczba pól, typy danych).
2. Utwórz jeden tymczasowy rekord w zbiorze bazy danych.
3. Funkcja Przesłanie danych z serwera iSeries służy do tworzenia pliku strumieniowego i przypisanego do niego pliku .FDF ze zbioru bazy danych.
4. Teraz można użyć funkcji Przesyłanie danych do serwera iSeries. Określ plik .FDF oraz źródłowy plik strumieniowy, który ma być przesłany.

Kopiowanie danych pomiędzy plikami strumieniowymi a zbiorami składowania

Zbiór składowania jest używany, wraz z komendami składowania i odtwarzania, do zachowywania danych, które w przeciwnym wypadku byłyby zapisywane na taśmie lub na dyskietce. Zbioru tego można również używać jak zbioru bazy danych do odczytywania i zapisywania rekordów zawierających informacje o składowaniu/odtworzeniu. Zbioru składowania można ponadto używać do wysyłania obiektów do innych użytkowników w sieci SNADS.

Do kopiowania zbioru składowania do pliku strumieniowego i odwrotnie służy komenda CPY. Jednak przy kopiowaniu pliku strumieniowego z powrotem do zbioru składowania, dane muszą mieć poprawny format danych składowania (muszą to być dane pochodzące ze zbioru składowania i skopiowane do pliku strumieniowego).

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.

Wykonywanie operacji za pomocą funkcji API

Wiele interfejsów API, które wykonują operacje na obiektach w zintegrowanym systemem 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 OS/400,
- 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 12 na stronie 96.

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

Tabela 11. Zintegrowany system plików API

| Funkcja | Opis |
|---------------------|--|
| access() | Określenie dostępności pliku |
| accessx() | Określenie dostępności pliku dla klasy użytkowników |
| chdir() | Zmiana bieżącego katalogu |
| chmod() | Zmiana uprawnień do pliku |
| chown() | Zmiana właściciela i grupy dla pliku |
| close() | Zamknięcie deskryptora pliku |
| closedir() | Zamknięcie katalogu |
| creat() | Utworzenie nowego lub wyczyszczenie istniejącego pliku |
| creat64() | Utworzenie nowego lub wyczyszczenie istniejącego pliku (również dla dużych plików) |
| DosSetFileLocks() | Blokowanie i anulowanie blokady zakresu bajtów w pliku |
| DosSetFileLocks64() | Blokowanie i anulowanie blokady zakresu bajtów w pliku (obsługa wielkich plików) |
| DosSetRelMaxFH() | Zmiana maksymalnej liczby deskryptorów plików |
| dup() | Powielenie deskryptora otwartego pliku |
| dup2() | Powielenie deskryptora otwartego pliku do nowego deskryptora |
| faccessx() | Określenie dostępności pliku dla klasy użytkowników przez deskryptor |
| fchdir() | Zmiana bieżącego katalogu przez deskryptor |
| fchmod() | Zmiana uprawnień do pliku przez deskryptor |
| fchown() | Zmiana właściciela i grupy dla pliku przez deskryptor |
| fclear() | Usunięcie zawartości pliku |
| fclear64() | Usunięcie zawartości pliku (obsługa wielkich plików) |
| fcntl() | Wykonanie kontroli pliku |
| fpathconf() | Pobranie konfigurowalnych zmiennych z nazwami ścieżek przez deskryptor |

Tabela 11. Zintegrowany system plików API (kontynuacja)

| Funkcja | Opis |
|------------------|---|
| 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 |
| getwpmam() | 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 |
| pathconf() | Pobranie konfigurowalnych zmiennych z nazwami ścieżek |
| pipe() | Utworzenie międzyprocesowego kanału z gniazdami |
| pread() | Wczytanie z deskryptora z przesunięciem |

Tabela 11. Zintegrowany system plików API (kontynuacja)

| Funkcja | Opis |
|----------------------------------|---|
| pread64() | Wczytanie z deskryptora z przesunięciem (również dla dużych plików) |
| pwrite() | Zapis do deskryptora z przesunięciem |
| pwrite64() | Zapis do deskryptora z przesunięciem (również dla dużych plików) |
| QjoEndJournal() | Zakończenie kronikowania |
| QjoRetrieveJournal Information() | Odtworzenie informacji kroniki |
| QJORJIDI() | Odtworzenie informacji o identyfikatorze kroniki |
| QJOSJRNE() | Wysłanie pozycji do kroniki |
| QjoStartJournal() | Uruchomienie kronikowania |
| QlgAccess() | Określenie dostępności pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgAccessx() | kreślenie dostępności pliku dla klasy użytkowników (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgChdir() | Zmiana bieżącego katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgChmod() | Zmiana autoryzacji pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgChown() | Zmiana właściciela i grupy dla pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgCreat() | Utworzenie nowego lub przepisanie istniejącego pliku (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgCreat64() | Utworzenie nowego lub przepisanie istniejącego pliku (również dla dużych plików i z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgCvtPathToQSYSObjName() | Tłumaczenie nazwy ścieżki zintegrowanego systemu plików na nazwę obiektu QSYS (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetAttr() | Pobranie atrybutów systemowych dla obiektu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetcwd() | Pobranie nazwy ścieżki dla bieżącego katalogu (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetPathFromFileID() | Pobranie nazwy ścieżki obiektu z jego identyfikatora (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetpwnam() | Pobranie informacji o użytkowniku dla danej nazwy użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetpwnam_r() | Pobranie informacji o użytkowniku dla danej nazwy użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetpwuid() | Pobranie informacji o użytkowniku dla danego identyfikatora użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |
| QlgGetpwuid_r() | Pobranie informacji o użytkowniku dla danego identyfikatora użytkownika (z wykorzystaniem nazwy ścieżki obsługującej narodowe wersje językowe) |

Tabela 11. Zintegrowany system plików API (kontynuacja)

| Funkcja | Opis |
|---------------------|--|
| 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) |
| 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) |

Tabela 11. Zintegrowany system plików API (kontynuacja)

| Funkcja | Opis |
|---------------------------|--|
| 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() | Zeskładowanie danych obiektów i zwolnienie ich pamięci |
| Qp0lSetAttr() | Ustawienie atrybutów systemowych dla obiektu |
| Qp0lUnlink() | Odłączenie pliku z nazwą ścieżki dostępną dla NLS |
| qsyssetgid() | Ustawienie obowiązującego ID grupy |
| qsyssetuid() | Ustawienie obowiązującego ID użytkownika |
| qsyssetgid() | Ustawienie ID grupy |
| qsyssetregid() | Ustawienie rzeczywistego i obowiązującego ID grupy |
| qsyssetreuid() | Ustawienie rzeczywistego i obowiązującego ID użytkownika |
| qsyssetuid() | Ustawienie ID użytkownika |
| QZNFRTVE() | Odzyskanie informacji dotyczących eksportu NFS |
| read() | Wczytanie z pliku |
| readdir() | Wczytanie pozycji katalogu |

Tabela 11. Zintegrowany system plików API (kontynuacja)

| Funkcja | Opis |
|------------------|---|
| 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 OS/400. Ograniczenia używania tych funkcji dla wybranych systemów plików opisano w temacie Praca z systemami plików. Przykładowy program używający funkcji C zintegrowanego systemu plików zawiera temat “Przykład: funkcje C zintegrowanego systemu plików” na stronie 101.

Tabela 12. 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. |

Więcej informacji o funkcjach API zintegrowanego systemu plików zawierają następujące sekcje:

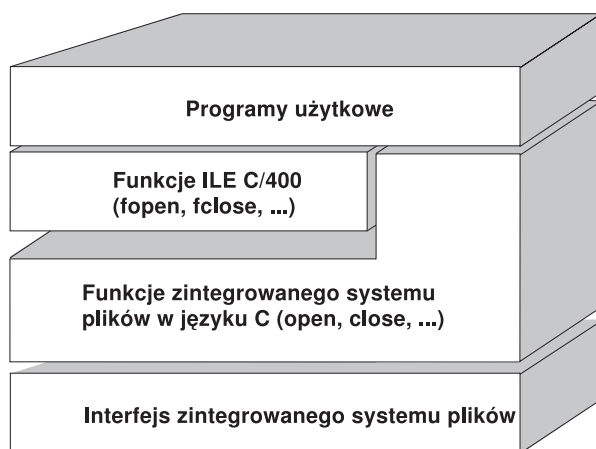
- Funkcje ILE C/400
- Obsługa dużych plików przez funkcje API

- Reguły dotyczące ścieżek dla funkcji API
- Deskryptor pliku
- Ochrona
- Kopiowanie danych za pomocą funkcji API
- Temat Aplikacyjne interfejsy programistyczne (API) w Centrum informacyjnym iSeries.

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 obsługi strumieniowych danych wejścia/wyjścia zintegrowanego systemu plików.

Więcej informacji na temat używania funkcji C ILE C/400 z danymi strumieniowymi wejścia/wyjścia zintegrowanego

systemu plików zawiera publikacja WebSphere Development Studio: ILE C/C++ Programmers Guide  .

Szczegółowe informacje o każdej funkcji C ILE C/400 zawiera publikacja 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. W zintegrowanym systemie plików, w systemach plików "root" (/), QOpenSys i użytkownika, obsługiwane są pliki strumieniowe wielkości maksymalnie około jednego terabajta.

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. Szczegóły dotyczące każdej 64-bitowej funkcji API zawiera temat Zintegrowany system plików API w Centrum informacyjnym iSeries.

Poniższe etykiety umożliwiają aplikacjom użycie obsługi dużych plików:

1. Jeśli w czasie kompilacji zostanie zdefiniowana etykieta makro `_LARGE_FILE_API`, aplikacje mają dostęp do 64-bitowych funkcji API i struktur danych. Na przykład aplikacja, która zamierza używać funkcji API `stat64()` i struktury `stat64`, będzie wymagała zdefiniowania podczas kompilacji etykiety `_LARGE_FILE_API`.
2. Jeśli etykieta makro `_LARGE_FILES` zostanie zdefiniowana przez aplikację podczas kompilacji, istniejące funkcje API i struktury danych są odwzorowane na swoje wersje 64-bitowe. Na przykład, jeśli aplikacja definiuje podczas kompilacji 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.

Reguły dotyczące ścieżek dla funkcji API

Gdy zintegrowany system plików lub funkcja API ILE C/400 używane są do wykonania działań na obiekcie, 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
```

Odwrotny ukośnik (`\`) nie jest rozpoznawany jako separator. Jest on traktowany jak zwykły znak w nazwie.

- Nazwy obiektów w ramach katalogu nie mogą się powtarzać.
- Maksymalna długość każdego składnika ścieżki i maksymalna długość łańcucha określającego ścieżkę mogą się zmieniać w zależności od systemu plików. Sekcja “Porównanie systemów plików” na stronie 23 opisuje ograniczenia występujące w każdym systemie plików.
- Znak `/` na początku ścieżki oznacza, że zaczyna się ona w katalogu głównym (“root”,`/`), na przykład:

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

- Jeśli ścieżka nie zaczyna się od znaku `/`, przyjmuje się, że zaczyna się ona w bieżącym katalogu, na przykład:

```
MójKat/MójPlik
```

gdzie `MójKat` jest podkatalogiem bieżącego katalogu.

- Aby uniknąć pomyłek związanych z wartościami specjalnymi serwera iSeries, nazwy ścieżek nie mogą rozpoczynać się od pojedynczego znaku gwiazdki (`*`). Aby podać nazwę ścieżki, która rozpoczyna się od dowolnych cyfr lub znaków, należy użyć dwóch gwiazdek (`*`), na przykład:

```
'**.plik'
```

Należy wiedzieć, że dotyczy to tylko względnych nazw ścieżek, w których przed gwiazdką (`*`) nie występują inne znaki.

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

```
/QSYS.LIB/PAYROLL.LIB/PAY.FILE
```

Więcej szczegółów zawiera sekcja: “System plików bibliotek (QSYS.LIB)” na stronie 35.

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

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

Więcej szczegółów zawiera sekcja: “System plików QSYS.LIB niezależnej ASP” na stronie 38.

- W nazwach ścieżek nie wolno używać dwukropka (`:`). Ma on specjalne znaczenie w systemie.
- W odróżnieniu od nazw ścieżek w komendach zintegrowanego systemu plików (patrz “Reguły dotyczące nazw ścieżek dla komend CL i ekranów” na stronie 61), gwiazdka (`*`), znak zapytania (`?`), apostrof (`'`) oraz cudzysłów (`"`) i

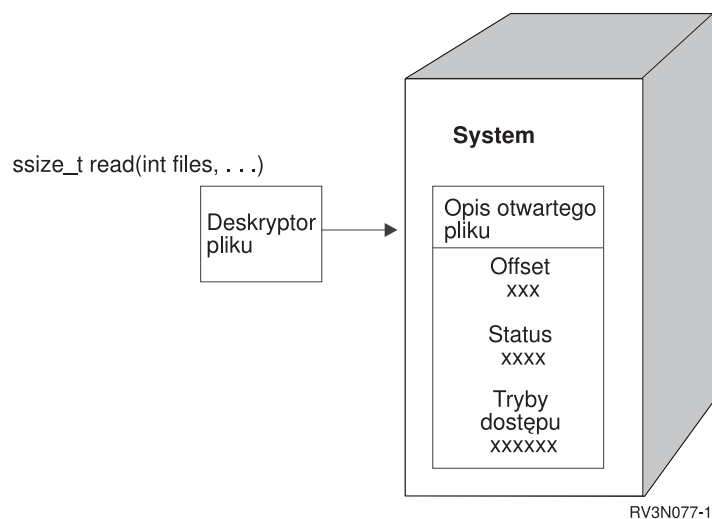
tylda (~) nie mają specjalnego znaczenia. Traktowane są one jak zwykłe znaki w nazwie. Jedynymi funkcjami API, których zasada ta nie dotyczy, są funkcje QjoEndJournal i QjoStartJournal.

- 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.

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**. Jest on dodatnią liczbą całkowitą, która musi być unikalna dla każdego zadania. Zadanie używa deskryptora pliku do identyfikacji otwartego pliku podczas przeprowadzania na nim operacji. Deskryptor pliku reprezentowany jest w funkcjach w języku C działających na zintegrowanym systemie plików przez zmienną *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 w zintegrowanym systemie 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 pomiędzy zdaniami. opis tych funkcji zawiera temat Programowanie z użyciem gniazd lub Funkcje API gniazd w Centrum informacyjnym iSeries.

Ochrona

Gdy używane są funkcje API zintegrowanego systemu plików, można ograniczyć dostęp do obiektów tak, jak przy użyciu interfejsów zarządzania danymi. Trzeba jednak mieć świadomość, że przejmowanie uprawnień nie jest obsługiwane. Funkcja API zintegrowanego systemu plików korzysta z uprawnień pobieranych z profilu użytkownika, który jest właścicielem zadania.

Każdy system plików może mieć własne wymagania dotyczące uprawnień. Zadania serwera NFS są jedynym odstępstwem od tej reguły. Żądania skierowane do serwera NFS są obsługiwane za pomocą profilu użytkownika, którego numer identyfikacyjny (UID) serwer NFS otrzymał w momencie zgłoszenia żądania.

Uprawnienia 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 `chmod()`. Do sterowania ustawieniami bitów praw dostępu do plików tworzonych przez dane zadanie można również wykorzystywać funkcję `umask()`.

Szczegóły ochrony danych i uprawnień opisane zostały w publikacji *Ochrona* .

Obsługa gniazd

Jeśli aplikacja korzysta z systemu plików “root” (/), QOpenSys lub UDFS, można wykorzystać obsługę **gniazd lokalnych** (local sockets) zintegrowanego systemu plików. Obiekt typu gniazdo lokalne (typ obiektu `*SOCKET`) umożliwia nawiązanie komunikacji między dwoma zadaniami działającymi w tym samym systemie.

Jedno z zadań ustanawia punkt połączeniowy tworząc gniazdo lokalne za pomocą funkcji `bind()` w języku C. Drugie zadanie podaje nazwę gniazda lokalnego w funkcji `connect()`, `sendto()` lub `sendmsg()`. Funkcje te i zagadnienia gniazd opisano ogólnie w temacie Programowanie z użyciem gniazd w Centrum informacyjnym iSeries.

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 pomiędzy zadaniami każde z nich używa funkcji `close()` w celu zamknięcia połączenia przez gniazdo. Lokalny obiekt gniazda pozostaje w systemie dopóki nie zostanie usunięty za pomocą funkcji `unlink()` lub komendy Usuwanie dowiązania (Remove Link - RMVLNK).

Gniazda lokalnego nie można zeszkładować.

Nazewnictwo i obsługa różnych języków

Obsługa systemów plików “root” (/) i QOpenSys zapewnia niezmiennosc znaków w nazwach obiektów przy przechodzeniu pomiędzy schematami kodowania używanymi w różnych językach narodowych i w różnych urządzeniach. Podczas przekazywania nazwy obiektu do systemu, każdy jej znak jest poddawany konwersji do postaci 16-bitowej, w której wszystkie znaki mają standardowo kodowaną reprezentację (patrz “Niezmienność nazw” na stronie 16). W chwili użycia nazwy następuje jej konwersja do postaci kodowanej właściwej dla używanej strony kodowej.

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

Ponieważ znaki pozostają niezmienione we wszystkich stronach kodowych, wykonując działania nie powinno się zakładać, że dany znak zostanie zamieniony na inny określony znak, gdy użyta zostanie konkretna strona kodowa. Na przykład nie powinno się przyjmować, że symbol liczby zostanie zamieniony na znak funta szterlinga, nawet jeśli mają one tę samą reprezentację kodową w różnych stronach kodowych.

Należy pamiętać, że atrybuty rozszerzone obiektu poddawane są konwersji w ten sam sposób, co nazwa obiektu; stosują się więc do nich te same uwagi.

Więcej informacji o stronach kodowych zawiera sekcja Globalizacja w Centrum informacyjnym iSeries.

Konwersja danych

W trakcie dostępu do plików poprzez zintegrowany system plików zawarte w tych plikach dane mogą zostać przekształcone lub nie, w zależności od trybu otwarcia żadanego w momencie otwierania tego pliku.

Otwarty plik może znajdować się w jednym z dwóch trybów otwarcia:

Binarnym (Binary)

Dane są odczytywane z pliku i zapisywane do niego bez konwersji. Za operowanie danymi odpowiada aplikacja.

Tekstowym (Text)

Dane są odczytywane z pliku i zapisywane do niego przy założeniu, że mają one postać tekstową. Podczas odczytu danych z pliku dokonywana jest ich konwersja z identyfikatora kodowanego zestawu znaków (CCSID) pliku na identyfikator CCSID aplikacji, zadania lub systemu odbierającego dane. Przy zapisie danych do pliku są one poddawane konwersji z identyfikatora CCSID aplikacji, zadania lub systemu na identyfikator CCSID pliku. Dla rzeczywistych plików strumieniowych wszystkie znaki formatowania wiersza (takie jak znak powrotu karetki, znak tabulacji i znak końca pliku) są po prostu poddawane konwersji z jednego identyfikatora CCSID na inny.

Podczas odczytu z plików o strukturze opartej na rekordach, które są traktowane jako pliki strumieniowe, na końcu danych z każdego rekordu dołączane są znaki końca wiersza (znak końca wiersza i znak nowego wiersza). Przy zapisie do pliku o strukturze opartej na rekordach:

- znaki końca wiersza są usuwane,
- znaki tabulacji zastępowane są odpowiednią liczbą spacji do następnej pozycji tabulacji,
- wiersze uzupełniane są spacjami (w przypadku podzbiorów fizycznych zbiorów źródłowych) lub znakami pustymi (w przypadku podzbiorów fizycznych zbiorów danych) do końca rekordu.

W żądaniu otwarcia można wpisać jedną z opcji:

Binarny, wymuszony (Binary, Forced)

Dane są przetwarzane jako binarne, niezależnie od faktycznej zawartości pliku. Za poprawne operowanie danymi odpowiada aplikacja.

Tekstowy, wymuszony (Text, Forced)

Zakłada się, że dane są tekstem. Dane są poddawane konwersji z identyfikatora CCSID pliku na identyfikator CCSID aplikacji.

Domyślna opcja *Binarne, wymuszone* użyta jest dla funkcji zintegrowanego systemu plików `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.

```
#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;
}

```

I **Uwaga:** Temat “Informacje dotyczące kodu” na stronie 1 zawiera istotne informacje prawne.

Praca z plikami i folderami z wykorzystaniem produktu iSeries Navigator

Dla plików i folderów można wykonać następujące zadania:

- “Odblokowanie pliku” na stronie 107
- “Zablokowanie pliku” na stronie 107
- “Tworzenie folderu” na stronie 107
- “Usuwanie folderu” na stronie 107
- “Przenoszenie plików lub folderów do innego systemu plików” na stronie 108
- “Nadawanie uprawnień” na stronie 109
- “Konfigurowanie konwersji plików tekstowych” na stronie 109
- “Wysyłanie pliku lub folderu do innego systemu” na stronie 109
- “Zmiana opcji definicji pakietów” na stronie 110
- “Planowanie daty i godziny wysyłki pliku lub folderu” na stronie 110
- “Tworzenie współużytkowanego zasobu plikowego” na stronie 110

- “Zmiana współużytkowanego zasobu plikowego” na stronie 110
- “Włączanie i wyłączanie skanowania obiektów” na stronie 111

Odblokowanie pliku

Aby odblokować plik, wykonaj następujące czynności:

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

Jeśli korzystasz z techniki przeciągnij i upuść, możesz podłączyć system plików 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.

Zablokowanie pliku

Aby zablokować plik, wykonaj następujące czynności:

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

Tworzenie folderu

Aby utworzyć folder, wykonaj następujące czynności:

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

Należy także rozważyć, czy obiekty utworzone w tym folderze mają być skanowane, czy nie. Więcej informacji na ten temat zawiera sekcja “Włączanie i wyłączanie skanowania obiektów” na stronie 111.

Gdy w serwerze iSeries tworzony jest folder, należy zdecydować, czy nowy folder (lub obiekty) ma być chroniony przez kronikowanie. Więcej informacji można znaleźć w sekcji Zarządzanie kronikami.

Tematy pokrewne:

- Uruchomienie kronikowania
- Zakończenie kronikowania

Usuwanie folderu

Aby usunąć folder, wykonaj następujące czynności:

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

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 zintegrowanego systemu plików i ich charakterystykami. Więcej informacji zawiera temat Praca z systemami plików.

Należy także rozważyć następujące kwestie:

- Czy używa się aplikacji wykorzystujących zalety systemu plików, w którym obecnie znajdują się obiekty?

Niektóre systemy plików obsługują interfejsy, których nie obsługuje zintegrowany system plików. Aplikacje używające tych interfejsów mogą nie mieć dostępu do obiektów przeniesionych do innego systemu. Na przykład systemy plików QDLS i QOPT obsługują hierarchiczny system plików (HFS). Obsługują również funkcje API i komendy, które służą do pracy z dokumentami i folderami. Nie można używać tych interfejsów w przypadku obiektów znajdujących się w innych systemach plików.

- 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. Porównanie różnych charakterystyk każdego systemu plików zawiera temat "Porównanie systemów plików" na stronie 23.

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.

Nadawanie 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 nadać uprawnienia do pliku lub folderu, wykonaj następujące czynności:

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

Konfigurowanie konwersji plików tekstowych

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

Aby skonfigurować konwersję plików tekstowych, wykonaj następujące czynności:

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

Wysyłanie pliku lub folderu do innego systemu

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

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

Informacje te można jednak zmienić, co zostało opisane w sekcjach “Zmiana opcji definicji pakietów” na stronie 110 i “Planowanie daty i godziny wysyłki pliku lub folderu” na stronie 110.

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

Zmiana opcji definicji pakietów

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

Aby zmienić opcje dla definicji pakietu, wykonaj następujące czynności:

1. Wykonaj kroki opisane w sekcji “Wysyłanie pliku lub folderu do innego systemu” na stronie 109.
2. Kliknij zakładkę **Opcje**. Domyślnie włączone są opcje dołączania podfolderów podczas pakowania i wysyłania plików oraz zastępowania istniejących plików plikami wysyłanymi.
3. Zmień te opcje zgodnie z wymaganiami.
4. Aby ustawić opcje składowania i odtwarzania, kliknij przycisk **Zaawansowane**.
5. Kliknij **OK**, aby zapisać opcje zaawansowane.
6. Kliknij **OK**, aby wysłać plik lub kliknij **Harmonogram**, aby ustawić godzinę wysyłki pliku.

Tematy pokrewne:

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

Planowanie daty i godziny wysyłki pliku lub folderu

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

1. Wykonaj kroki opisane w sekcji “Wysyłanie pliku lub folderu do innego systemu” na stronie 109.
2. Kliknij **Harmonogram**.
3. Określ porę wysłania pliku lub folderu.

Tworzenie współużytkowanego zasobu plikowego

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

Aby utworzyć współużytkowany zasób plikowy, wykonaj następujące czynności:

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

Zmiana współużytkowanego zasobu plikowego

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

Aby zmienić współużytkowany zasób plikowy, wykonaj następujące czynności:

1. W oknie **iSeries Navigator** rozwiń system.

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

Włączanie i wyłączanie skanowania obiektów

Aby włączyć lub wyłączyć skanowanie obiektów:

1. W oknie **iSeries Navigator** rozwiń system.
2. Rozwiń gałąź **Systemy plików**.
3. Rozwiń gałąź **Zintegrowany system plików**.
4. Rozwiń folder lub wybrany plik.
5. Kliknij prawym klawiszem myszy folder lub plik, a następnie wybierz opcję **Właściwości**
6. Kliknij zakładkę **Ochrona**.
7. Wybierz opcję w polu **Skanowanie obiektów**.

Więcej informacji o opcjach znajduje się poniżej. 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 zaznaczona 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, ten atrybut "tylko w przypadku zmiany obiektów" nie będzie używany, a obiekt będzie podlegał skanowaniu po modyfikacji i gdy oprogramowanie do skanowania wskaże aktualizację.

Uwaga: W zakładce tej dla plików można określić status skanowania obiektu.

Uwaga: Jeśli opcja 'Skanuj przy kolejnym dostępie po przywróceniu obiektu' jest zaznaczona w wartościach systemowych, obiekt będzie podlegał skanowaniu co najmniej raz po przywróceniu.

Informacje pokrewne dotyczące zintegrowanego systemu plików

Informacje pokrewne dotyczące zintegrowanego systemu plików można znaleźć w następujących tematach:

- "Wywołanie TI-RPC (Transport Independent Remote Procedure Call)" na stronie 112
- "Odniesienia" na stronie 116
- "Raporty doświadczenia" na stronie 117

Wywołanie TI-RPC (Transport Independent Remote Procedure Call)

Opracowane przez firmę Sun Microsystems wywołanie RPC (Remote Procedure Call - zdalne wywołanie procedury) w prosty sposób oddziela aplikacje klienta od mechanizmu serwera. Obejmuje ono standardową reprezentację danych o nazwie XDR (eXternal Data Representation), która wielu typom maszyn udostępnia przesyłanie danych. Najnowszą wersją RPC jest TI-RPC (Transport Independent RPC). Jest to metoda odizolowania od protokołu, który jest używany w warstwie sieci, co umożliwia płynne przejścia pomiędzy protokołami. 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.

Więcej informacji o wywołaniach Transport Independent Remote Procedure Calls zawierają następujące tematy:

- “Wybór sieci”
- “Konwersja nazwy na adres”
- “Zewnętrzna reprezentacja danych (XDR)” na stronie 113
- “Uwierzytelnienie” na stronie 114
- “TI-RPC (Transport Independent RPC)” na stronie 114

Wybór sieci

Poniższe funkcje API zapewniają środki do wyboru transportu aplikacji.

Te funkcje API wymagają, aby w systemie istniał plik *STMF /etc/netconfig. Jeśli w katalogu /etc nie ma pliku netconfig, użytkownik musi skopiować go z katalogu /QIBM/ProdData/OS400/RPC. Plik netconfig zawsze jest w katalogu /QIBM/ProdData/OS400/RPC.

| Funkcja API | Opis |
|-------------------|--|
| endnetconfig() | Zwalnia wskaźnik do rekordów przechowywanych w pliku netconfig. |
| freenetconfignt() | Zwalnia strukturę netconfig, która została zwrócona na skutek wywołania do funkcji getnetconfignt(). |
| 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(). |

Konwersja nazwy na adres

Poniższe funkcje API umożliwiają aplikacji otrzymywanie adresu usługi lub podanego hosta w sposób niezależny od transportu.

| Funkcja API | Opis |
|--------------------|--|
| netdir_free() | Zwalnia struktury, które zostały przydzielone przez funkcje API konwersji nazwy na adres. |
| netdir_getbyaddr() | Odwzorowuje adresy na nazwy hostów i nazwy usług. |
| netdir_getbyname() | Odwzorowuje podane w parametrze usługi, nazwę hosta oraz nazwę usługi na zbiór adresów, które są zgodne z transportem zidentyfikowanym w strukturze netconfig. |
| netdir_options() | Stanowi interfejs dla funkcji transportu, takich jak adres rozgłośni i zarezerwowany port TCP i UDP. |

| Funkcja API | Opis |
|--------------------|--|
| netdir_sperror() | Zwraca komunikaty opisowe, określające dlaczego jedna z funkcji API konwersji nazwy na adres nie powiodła się. |
| taddr2uaddr() | Dokonuje konwersji adresu specyficznego dla danego transportu (lokalnego) na adres niezależny od danego transportu (uniwersalny). |
| uaddr2taddr() | Dokonuje konwersji adresu niezależnego od danego transportu (uniwersalnego) na adres specyficzny dla danego transportu (lokalny) (struktura netbuf). |

Zewnętrzna reprezentacja danych (XDR)

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

| Funkcja API | Opis |
|--------------------|---|
| xdr_array() | Filtr, który dokonuje konwersji pomiędzy tablicami o zmiennej długości i odpowiadającymi im zewnętrznymi reprezentacjami. Funkcja ta jest wywoływana do kodowania i dekodowania każdego elementu tablicy. |
| xdr_bool() | Filtr, który dokonuje konwersji pomiędzy zmiennymi typu Boolean (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 pomiędzy tablicami bajtów i ich zewnętrznymi reprezentacjami. Funkcja ta operuje na tablicach, których elementy mają rozmiar 1 bajtu z wbudowanym zewnętrznym opisem każdego elementu. Długość ciągu bajtów jest jawnie podana jako liczba całkowita bez znaku (unsigned integer). Ciąg bajtów nie jest zakończony znakiem null. Zewnętrzna i wewnętrzna reprezentacja bajtów jest taka sama. |
| xdr_char() | Filtr, który dokonuje konwersji pomiędzy znakami języka C i ich zewnętrznymi reprezentacjami. |
| xdr_double() | Filtr, który dokonuje konwersji pomiędzy liczbami o podwójnej precyzji języka C i ich zewnętrznymi reprezentacjami. |
| xdr_double_char() | Filtr, który dokonuje konwersji pomiędzy dwubajtowymi znakami języka C i ich zewnętrznymi reprezentacjami. |
| xdr_enum() | Filtr, który dokonuje konwersji pomiędzy liczbami typu wyliczeniowego (enum) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_free() | Rekurencyjnie zwalnia obiekt wskazany przez przekazany wskaźnik. |
| xdr_float() | Filtr, który dokonuje konwersji pomiędzy liczbami zmiennoprzecinkowymi (znormalizowane liczby zmiennoprzecinkowe pojedynczej precyzji) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_int() | Filtr, który dokonuje konwersji pomiędzy liczbami całkowitymi języka C i ich zewnętrznymi reprezentacjami. |
| xdr_long() | Filtr, który dokonuje konwersji pomiędzy długimi liczbami całkowitymi (long integer) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_netobj() | Filtr, który dokonuje konwersji pomiędzy nieprzezroczystymi (opaque) danymi o zmiennej długości i ich zewnętrznymi reprezentacjami. |
| xdr_opaque() | Filtr, który dokonuje konwersji pomiędzy nieprzezroczystymi danymi o stałej długości i ich zewnętrznymi reprezentacjami. |
| xdr_pointer() | Zapewnia uchwycenie (chasing) wskaźnika wewnątrz struktur i szereguje wskaźniki null (dokonuje ich konwersji do postaci XDR). Może reprezentować rekurencyjne struktury danych, takie jak drzewa binarne lub listy. |

| Funkcja API | Opis |
|--------------------|--|
| xdr_reference() | Filtr, który zapewnia uchwycenie wskaźnika wewnątrz struktur. Umożliwia on szeregowanie, deszeregowanie i zwalnianie wskaźników wewnątrz jednej struktury, do której istnieją odniesienia z innej struktury. Funkcja xdr_reference() nie nadaje specjalnego znaczenia wskaźnikowi null podczas szeregowania i przesłanie adresu wskaźnika null może spowodować błąd pamięci. Dlatego programista musi opisywać dane za pomocą unii o dwóch składowych. Jedna składowa jest używana, gdy wskaźnik jest poprawny, druga, gdy wskazuje na wartość null. |
| xdr_short() | Filtr, który dokonuje konwersji pomiędzy krótkimi liczbami całkowitymi (short integer) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_string() | Filtr, który dokonuje konwersji pomiędzy ciągami znaków języka C i ich zewnętrznymi reprezentacjami. |
| xdr_u_char() | Filtr, który dokonuje konwersji pomiędzy znakami języka C (unsigned char) i ich zewnętrznymi reprezentacjami. |
| xdr_u_int() | Filtr, który dokonuje konwersji pomiędzy liczbami całkowitymi bez znaku (unsigned integer) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_u_long() | Filtr, który dokonuje konwersji pomiędzy długimi liczbami całkowitymi bez znaku (unsigned long integer) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_u_short() | Filtr, który dokonuje konwersji pomiędzy krótkimi liczbami całkowitymi bez znaku (unsigned short integer) języka C i ich zewnętrznymi reprezentacjami. |
| xdr_union() | Filtr, który dokonuje konwersji pomiędzy uniami języka C i ich zewnętrznymi reprezentacjami. |
| xdr_vector() | Filtr, który dokonuje konwersji pomiędzy tablicami o stałej długości i odpowiadającymi im zewnętrznymi reprezentacjami. |
| xdr_void() | Bez parametrów. Jest przesyłana do innych funkcji RPC, które wymagają parametrów, ale nie przesyłają danych. |
| xdr_wrapstring() | Funkcja, która wywołuje funkcję API xdr_string(xdr, sp, maxuint), gdzie maxuint jest maksymalną wartością liczby całkowitej bez znaku. Funkcja xdr_wrapstring() jest użyteczna, ponieważ pakiet RPC przesyła maksymalnie dwie funkcje XDR jako parametry, a funkcja xdr_string() wymaga trzech. |

Uwierzytelnienie

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

| Funkcja API | Opis |
|--------------------|--|
| 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. |

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.

Więcej informacji o wywołaniach Transport Independent Remote RPC (TI-RPC) zawierają następujące tematy:

- “Uproszczone funkcje API aplikacji TI-RPC” na stronie 115
- “Funkcje API poziomu najwyższego aplikacji TI-RPC” na stronie 115
- “Funkcje API poziomu średniego aplikacji TI-RPC” na stronie 115
- “Funkcje API poziomu eksperckiego aplikacji TI-RPC” na stronie 115

- “Inne funkcje API aplikacji TI-RPC” na stronie 116

Uproszczone funkcje API aplikacji TI-RPC:

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

| Funkcja API | Opis |
|-------------|---|
| rpc_call() | Wywołuje zdalną procedurę w podanym systemie. |
| rpc_reg() | Rejestruje procedurę z pakietem usług RPC. |

Funkcje API poziomu najwyższego aplikacji TI-RPC: Poniższe funkcje API umożliwiają aplikacji podanie typu transportu.

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

Funkcje API poziomu średniego aplikacji TI-RPC:

Poniższe funkcje API są podobne do funkcji API najwyższego poziomu, ale aplikacje użytkownika wybierają informacje specyficzne dla transportu za pomocą funkcji API służących do wyboru sieci:

| Funkcja API | Opis |
|------------------|------------------------|
| clnt_tp_create() | Tworzy uchwyt klienta. |
| svc_tp_create() | Tworzy uchwyt serwera. |

Funkcje API poziomu eksperckiego aplikacji TI-RPC: Poniższe funkcje API umożliwiają aplikacji wybór typu transportu. Oferują także zwiększony zakres sterowania szczegółami uchwytów CLIENT i SVCXPRT. Te funkcje API są podobne do funkcji API pośredniego poziomu z dodatkowym sterowaniem zapewnionym poprzez używanie funkcji API służących do konwersji nazwy na adres.

Używanie funkcji API służących do konwersji nazwy na adres zapewnia dodatkowe sterowanie.





| 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(). |




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. |

Odniesienia

Niniejsza bibliografia zawiera bardziej szczegółowe omówienie informacji o serwerze iSeries zawartych w tym podręczniku.

- Temat Control language w kategorii **Programowanie** w Centrum informacyjnym iSeries zawiera opis języka CL serwera iSeries i jego komend. Opis każdej komendy obejmuje diagram składniowy, parametry, wartości domyślne, słowa kluczowe i przykład.
- Temat Globalizacja w Centrum informacyjnym serwera iSeries wyjaśnia zagadnienia obsługi języków narodowych (NLS), takie jak zestawy znaków i strony kodowe oraz zawiera informacje niezbędne do oszacowania, zaplanowania i wykorzystania obsługi NLS i możliwości obsługi wielu języków w serwerze iSeries.
- Temat API w kategorii **Programowanie** w Centrum informacyjnym serwera iSeries zawiera opis każdej funkcji API OS/400, w tym również funkcje API obsługiwane przez zintegrowany system plików, jak i punkty wyjścia i programy obsługi wyjścia zintegrowanego systemu plików.
- Temat Zarządzanie kronikami w kategorii **Zarządzanie systemami** w Centrum informacyjnym iSeries zawiera informacje o konfigurowaniu, zarządzaniu i rozwiązywaniu problemów związanych z ochroną ścieżek dostępu zarządzanych przez system (SMAPP) oraz lokalnymi i zdalnymi kronikami w serwerze iSeries.
- Temat Kontrola transakcji w kategorii **Baza danych** w Centrum informacyjnym iSeries wyjaśnia, jak definiować i przetwarzać grupy zmian zasobów, takich jak zbiory baz danych lub pliki zintegrowanego systemu plików jako logiczne jednostki pracy.
- OS/400 Network File System Support  W podręczniku tym opisano system plików NFS z punktu widzenia serii powszechnie używanych aplikacji. Zawarto w niej informacje na temat eksportu, dołączania, blokowania plików i na temat ochrony. Można się z niej dowiedzieć, jak używać systemu NFS do zbudowania i utrzymywania bezpiecznego sieciowego systemu plików.
- Optical Support  Publikacja ta służy jako przewodnik i podręcznik do obsługi nośników optycznych IBM w systemie OS/400. Zawarte w niej informacje powinny pomóc użytkownikom zrozumieć koncepcję serwera danych biblioteki optycznej, zaplanować administrowanie biblioteką optyczną i serwerem danych biblioteki optycznej oraz rozwiązywać dotyczące ich problemy.
- WebSphere Development Studio: ILE C/C++ Programmers Guide  Publikacja ta zawiera informacje niezbędne do projektowania, edycji, uzupełniania, uruchamiania i poprawiania programów ILE C/400 w serwerze iSeries.
- WebSphere Development Studio: C/C++ Language Reference  Publikacja ta zawiera informacje o strukturze programów ILE C/400 i zawiera szczegóły funkcji bibliotek i plików dołączanych (nagłówków).

- Ochrona  Publikacja ta zawiera szczegółowe informacje techniczne o bezpieczeństwie OS/400, 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) w 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 w serwerze IBM iSeries.

| **Raporty doświadczenia**

- | Raporty te są pisane przez programistów IBM dokumentując ich bezpośrednie doświadczenia w realnym wprowadzaniu scenariuszy i rozwiązań. Służą do przekazywania doświadczeń z konkretnymi implementacjami rozwiązań iSeries wraz z instrukcjami krok po kroku i wskazówkami.
- | Z plikami i systemami plików związane są następujące raporty:
- | Składowanie zintegrowanego systemu plików

Dodatek. Uwagi

Niniejsze informacje zostały opracowane dla produktów i usług oferowanych w Stanach Zjednoczonych.

IBM może nie udostępniać produktów, usług lub innych opcji omawianych w niniejszej dokumentacji w innych krajach. Aby uzyskać informacje dotyczące aktualnie dostępnych produktów w danym kraju należy skontaktować się z lokalnym przedstawicielem IBM. Odniesienia do produktów, programów lub usług IBM nie oznaczają ani nie sugerują, że można użyć tylko produktów, programów, i usług firmy IBM. Wszelkie produkty, programy lub usługi funkcjonalnie równorzędne, które nie naruszają żadnych własności intelektualnych firmy IBM mogą być zamiennie stosowane. Jednakże odpowiedzialność za weryfikację funkcjonalności wszelkich produktów, programów i usług innych dostawców spoczywa na użytkowniku.

IBM może być właścicielem patentów lub prowadzić procedury patentowe dotyczące zagadnień pisanych w niniejszej dokumentacji. Udostępnienie tej dokumentacji nie oznacza udzielenia żadnej licencji na te patenty. Zapytania o licencje można wysyłać na adres:

- | IBM Director of Licensing
- | IBM Corporation
- | 500 Columbus Avenue
- | Thornwood, NY 10594-1785
- | U.S.A.

Zapytania licencyjne dotyczące informacji dwubajtowych (DBCS) należy kierować do Departamentu Własności Intelektualnych IBM we własnym kraju lub wysyłać pisemnie na adres:

- | IBM World Trade Asia Corporation
- | Licensing
- | 2-31 Roppongi 3-chome, Minato-ku
- | Tokyo 106, Japan

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 może ulepszać i/lub zmieniać produkty i/lub programy opisane w tej publikacji dowolnie, bez uprzedniego 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 zamieszczone w tych serwisach nie stanowią części materiałów dotyczących tego produktu firmy IBM i korzystając z nich użytkownik robi to na własne ryzyko.

- | IBM może użyć lub rozpowszechnić wszelkich dostarczonych przez klientów informacji w dowolny sposób, bez przyjmowania żadnych zobowiązań wobec klientów.

Licencje do programu dla osób, które szukają informacji dotyczących licencjonowania w celu: (i) wymiany informacji pomiędzy niezależnie utworzonymi programami (włączając niniejszy program) i (ii) obustronnego wykorzystania wymienionych informacji, powinny nawiązać kontakt z następującą jednostką:

- | IBM Corporation

| Software Interoperability Coordinator, Department 49XA
| 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 niniejszych informacjach i wszystkich materiałach licencjonowanych dla niego dostępnych jest udostępniany przez firmę IBM na warunkach umowy IBM Customer Agreement, IBM International Program License Agreement, lub dowolnego odpowiednika umowy między stronami.

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 prowadziła testów tych produktów i nie może potwierdzić dokładności wydajności, zgodności ani innych danych związanych z produktami tych dostawców. 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.

Wszystkie podane ceny są jedynie sugerowanymi cenami firmy IBM i mogą się zmienić bez uprzedniego powiadomienia. Ceny sprzedawców mogą od nich odbiegać.

Informacje te służą tylko celom planowania. Mogą ulec zmianie zanim opisany produkt będzie dostępny.

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. Te programy przykładowe można kopiować, zmieniać i rozpowszechniać w dowolnej formie bezpłatnie, do celów projektowych, użytkowych, marketingowych lub dystrybucyjnych zgodnie z interfejsem programistycznym dla platformy operacyjnej, dla której przykładowe programy zostały napisane. Przykłady te nie zostały gruntownie przetestowane. Dlatego firma IBM nie gwarantuje ani nie sugeruje niezawodności, przydatności ani funkcjonalności tych programów. Te programy przykładowe można kopiować, zmieniać i rozpowszechniać w dowolnej formie bezpłatnie, do celów projektowych, użytkowych, marketingowych lub dystrybucyjnych zgodnie z interfejsem programistycznym dla platformy operacyjnej, dla której przykładowe programy zostały napisane.

Każda kopia lub część tych przykładowych programów lub ich pochodna musi zawierać następujące uwagi dotyczące praw autorskich:

© IBM Corp., 2004. Części kodu pochodzą z przykładowych programów IBM Corp. © Copyright IBM Corp. 2004. All rights reserved.

Jeśli informacje te są przeglądane elektronicznie, fotografie i kolorowe ilustracje mogą być niewidoczne.

Informacje o interfejsie programistycznym

- | Niniejsze informacje dokumentują interfejsy programistyczne, które umożliwiają klientom tworzenie programów w celu uzyskania usług Zintegrowanego systemu plików.

Znaki handlowe

Następujące nazwy są znakami towarowymi firmy International Business Machines Corporation w Stanach Zjednoczonych i/lub w innych krajach:

Application System/400
AS/400
e (logo)
IBM
iSeries
Operating System/400
OS/400
400
C/400
DB2
Integrated Language Environment
OfficeVision
OS/2
WebSphere
xSeries

- | Lotus, Freelance i WordPro są znakami handlowymi International Business Machines Corporation i Lotus Development Corporation w Stanach Zjednoczonych i/lub w innych krajach.

Microsoft, Windows, Windows NT i logo Windows to znaki handlowe Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX to zastrzeżony znak handlowy The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Nazwy innych firm, produktów i usług mogą być znakami towarowymi lub znakami usług osób trzecich.

Warunki dotyczące pobierania i drukowania publikacji

Firma IBM udziela zgody na używanie publikacji wybranych do pobrania z zastrzeżeniem następujących warunków i wyrażeniem na nie zgody.

Użytek osobisty: Publikacje można powielać do własnych celów niekomercyjnych, jeśli zachowane zostaną wszystkie uwagi dotyczące praw własności. Nie wolno rozpowszechniać ani upubliczniać ani tworzyć prac w oparciu o całe lub części niniejszych publikacji bez pisemnej zgody firmy IBM.

Użytek komercyjny: Publikacje można powielać, rozpowszechniać i upubliczniać do celów komercyjnych tylko wewnątrz przedsiębiorstwa, jeśli zachowane zostaną wszystkie uwagi dotyczące praw własności. Nie wolno tworzyć prac w oparciu o całe lub części niniejszych publikacji ani upubliczniać bądź rozpowszechniać publikacji lub prac poza przedsiębiorstwem bez pisemnej zgody firmy IBM.

Z wyjątkiem osobnej zgody, nie przysługują żadne inne zgody, licencje ani prawa do publikacji, wyrażone ani domniemane, co dotyczy także informacji, oprogramowania, danych i pozostałej własności intelektualnej zawartej w niniejszej publikacji.

IBM zastrzega sobie prawo cofnięcia wszelkich zagwarantowanych tu zezwoleń bez konieczności powiadamiania, jeśli użycie publikacji zostanie uznane przez firmę IBM za szkodzące jej interesom lub niepoprawnie stosowane.

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 GWARANCJI NA ZAWARTOŚĆ NINIEJSZYCH PUBLIKACJI. PUBLIKACJE UDOSTĘPNIANE SĄ W STANIE, W JAKIM SIĘ ZNAJDUJĄ. NIE UDZIELA JAKICHKOLWIEK GWARANCJI (W TYM RĘKOJMI), A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ ORAZ PRZYDATNOŚCI DO OKREŚLONEGO CELU.

Wszystkie materiały objęte są prawami autorskimi IBM Corporation.

Pobranie lub drukowanie publikacje z tego serwisu WWW oznacza zgodę z powyższymi warunkami.

IBM