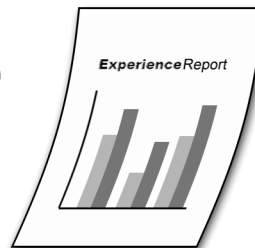


iSeries



Backing up the integrated file system

Experience Report



iSeries



Backing up the integrated file system

Contents

Introduction	1	Improve the performance of your backups	4
Understand the data in your integrated file system	1	Use online backups	6
Integrated file system backup scenarios	3	Back up less data	6
Setting realistic expectations for backup and recovery	4	References and resources	8
Tips and techniques for improved backup and recovery performance	4	Disclaimer	8

Introduction

The integrated file system was introduced in OS/400^(R) V3R1M0. Every iSeries^(TM) server has some data stored in the integrated file system. To be able to recover your server, it is critical that the data be included in your backup. Use of the integrated file system has increased dramatically. Many of the new applications on the iSeries server store their data in the integrated file system.

Some of the applications and data stored in the integrated file system include:

- Lotus^(R) applications such as Domino^(TM), Quickplace^(TM), and Sametime^(R)
- Content Manager OnDemand for iSeries
- Content Manager for iSeries
- WebSphere^(R) Application Server
- HTML files
- JPEG files
- Image files
- Java^(TM) programs
- Help text for some applications

For many iSeries server customers the amount of data stored in the integrated file system is not a concern. The data can be backed up within the customer's window of time available to perform a backup. However, some customers may have additional considerations when backing up their integrated file system data.

This experience report will:

- Identify several scenarios that cause additional considerations for integrated file system backup
- Help you identify the amount and structure of integrated file system data on your system
- Help you set realistic expectations for integrated file system backup on your system
- Provide tips and techniques that you can consider to help deal with integrated file system backups
- Provide references to other useful resources

Understand the data in your integrated file system

The first step to address additional considerations for integrated file system backup is to understand the integrated file system concepts. You can start by reading about the integrated file system in the iSeries^(TM) Information Center.

Once you understand the concepts, you need to understand your data. It is critical to know both the amount and structure of your integrated file system data. The information in this section will help you to understand your data.

iSeries Navigator

iSeries Navigator is the graphical user interface for managing and administering your systems from your Windows^(R) desktop. iSeries Navigator makes operation and administration of your system easier and more productive. You can use the iSeries Navigator to browse the directories and files stored in the integrated file system on your system. You can also display attributes about the data such as the size of the objects. You can read more about how to access the integrated file system with iSeries Navigator in the iSeries Information Center.

WRKLNK command

The Work with Object Links (WRKLNK) command shows a list of names of specified objects in directories and options to work with the objects. You can learn more about the WRKLNK command in the Programming topic of the iSeries Information Center.

QSRSRV program

V5R1 PTF SI05156 and V5R2 PTF SI05155 are available to gather metrics data about your integrated file system data. Depending on the amount of integrated file system data on your system the program may take a long time to complete. It is best to run this program when your system is least busy. You can load the PTF and then run the following command to obtain data about all of your integrated file system data.

```
call qsrsv parm("METRICS" '/')
```

To get the most accurate backup performance data, it would be best to have the system in the same state you would have it when backing up the system (user-defined file systems (UDFS) unmounted, Windows network servers varied on/off, etc). However this may not be possible, so there is another optional parameter which tells the tool not to process QNTC, QNETWARE, or QLANSRV directories. If you don't normally save those directories because the network servers are varied off, then specify "EPFS" as the third parameter:

```
call qsrsv parm("METRICS" '/' "EPFS")
```

You can also run the command against a specific directory. For example:

```
call qsrsv parm("METRICS" '/mydir/mysubdir')
```

The command will produce a spool file with the name of QSRSRV. The listing is in English only. The listing will include a line for each directory on your system. Each entry includes the number of links or objects (LINK COUNT), the number of multiple linked objects (DFRD LINKS), the total size in kilobytes of all the objects in the directory (SIZE IN K BYTES), and the name of the directory (NAME).

For example:

LINK COUNT	DFRD LINKS	SIZE IN DIRECTORY K BYTES	NAME
35	0	112832	/QIBM/ProdData/OS400/Java400/ext

Summary statistics are listed at the bottom of the listing. For example:

```
TOTALS FOR /
65516      5 10920873
*STMF *DIR *SYMLNK *FIFO *SOCKET *BLKSF *CHRSE *DDIR *DSTMF *SOMOBJ *OOPool OTHER
48089 12135 5173      0      12      0     103      1      0      0      0      2
Total mounted UDFSs      0; Total Unmounted UDFSs      0
Objects per Directory - Max: 10000; Average: 5
Max Directory Depth: 130; Max Directory Width: 10000
```

RTVDSKINF command

Another alternative available to help you understand the data in your integrated file system if you select to not use the QSRSRV program is the Retrieve Disk Information (RTVDSKINF) command. The RTVDSKINF command is used to collect disk space information. The collected information is stored in the library QUSRSYS. The file name depends on the auxiliary storage pool (ASP) device for which disk space information is retrieved. If the information was retrieved from the system and basic ASPs, the collected information will be stored in file QAEZDISK. If the information was retrieved from an independent ASP device, the collected information will be stored in file QAEZDnnnnn, where 'nnnnn' is the ASP number of the independent ASP. The information will be stored in a database file member named QCURRENT.

Each time this command is run, existing information in QCURRENT is overwritten. To save existing information in QCURRENT, rename file QAEZDISK or QAEZDnnnnn, or copy the member to another file.

To get the most accurate results, this command should be run at a time when there is very little system activity. This command can take a long time to complete. Also, the command has limitations for integrated file system data. It will report on integrated file system objects but it doesn't show the names of the objects. However, you can use it to get statistics on the number and size of the integrated file system data on your system.

For example: You can query the file produced by RTVDSKINF to get the count and total size for all of the object types that are saved as part of the integrated file system:

```
SELECT DIOBTP, COUNT(*), SUM(DIOBSZ) FROM QUSRSYS.QAEZDISK
WHERE DIOBTP IN ('DIR', 'STMF', 'SYMLNK', 'SOCKET', 'BLKSF', 'CHRSF', 'FIFO')
GROUP BY DIOBTP ORDER BY SUM(DIOBSZ) DESC
```

Integrated file system backup scenarios

The iSeries^(TM) performance and development groups in Rochester, Minnesota have investigated and measured integrated file system backups. They have identified several scenarios that often result in additional considerations for integrated file system backup. These are the scenarios identified:

Tens of thousands or more directories

Measurements show that many directories negatively decrease backup performance. In general, the more directories you need to back up, the more your performance decreases. Additional considerations may be required when the number of directories gets to be tens of thousands or more.

The Content Manager OnDemand for iSeries is an example of an application environment that often has many directories.

Tens of thousands or more files in directories

Measurements show that many files in a single directory decrease backup performance. Additional considerations may be required when the number of files in a single directory gets to be tens of thousands or more.

The Content Manager for iSeries is an example of an application environment that often has many files stored in each directory.

Applications that must remain active

An increasing number of customers have applications that use the integrated file system. Often these applications must remain active during the backup. For example, your users may expect to be able to access their mail while the backup is running.

The Lotus^(R) Domino^(TM) and Lotus Quickplace^(TM) products are examples of products that use the integrated file system and can remain active during a backup.

Multiple links to objects

Measurements show that having many multiple hard links to objects can dramatically decrease backup performance. Performance will decrease as the total number of hard links in each file system increases. You can read more about saving objects with multiple hard links in the iSeries Information Center.

Saving all of your integrated file system objects in a single tape file

Many iSeries customers back up all of their integrated file system data into a single tape file. For example, if you are using option 21 on the SAVE menu to back up your entire system it saves all of your integrated file system data into a single tape file.

The following SAV command is typically used to save all of the integrated file system data:

```
SAV DEV('/QSYS.LIB/media-device-name.DEVD')OBJ('/*')
('/QSYS.LIB' *OMIT)('/QDLS' *OMIT) UPDHST(*YES)
```

This approach makes for a simple and effective backup of all your integrated file system data. It is also a simple method to recover your entire system. However, since all of the data is stored in one large tape file, this can cause performance problems during a recovery. For example, if you need to recover a subset of your integrated file system, such as a single directory or object, the system will need to read through the entire tape file to find the data you want to recover. The tape file may contain many objects and may span multiple tape volumes. Also, if you need to recover all of your integrated file system data, you will be limited to a single job that serially uses a tape device to restore all of your data. If you use this approach, you cannot benefit from multiple concurrent jobs that each use a different tape device to restore different parts of your integrated file system data.

Setting realistic expectations for backup and recovery

Once you understand the structure and amount of integrated file system data on your system, you need to set realistic expectations for your backup. It is important to realize that there are many factors that affect the performance of your backup and recovery.

The iSeries^(TM) performance group has published a chapter on Save/Restore in the iSeries Performance Capabilities Reference. Chapter 15 explains all of the factors that affect Save/Restore performance. It also provides measurement data for a variety of backup and recovery scenarios. You can use this information to estimate and set realistic expectations for the backup and recovery of your integrated file system data.

Tips and techniques for improved backup and recovery performance

At this point you should understand the structure of your integrated file system data and the scenarios that may require additional considerations. This section lists some tips and techniques that you can consider.

There are three general ways to improve your integrated file system backups:

- Improve the performance of your backups
- Use online backups
- Back up less data

Improve the performance of your backups

The following topics are things that you can consider to improve the performance of your backups.

Convert to *TYPE2 directories

For V5R2, the integrated file system has added function and support for *TYPE2 directories in the “root” (/), QOpenSys, and user-defined file systems (UDFS). The *TYPE2 directory format is an enhancement of the original *TYPE1 directory format. *TYPE2 directories have a different internal structure and different implementation than *TYPE1 directories. There are also PTFs available that allow you to convert to *TYPE2 directories if you are running V5R1.

Performance measurements done by the iSeries^(TM) server performance group have shown performance improvements in backup times by converting to *TYPE2 directories. The largest improvements were seen in scenarios that involved many directories. Chapter 15 in the iSeries Performance Capabilities Reference includes the measurements that were done with several different integrated file system scenarios.

For more information and procedures to convert to *TYPE2 directories, see *TYPE2 directories in the Integrated file system topic of the iSeries Information Center.

Load and apply PTFs to improve performance

The following PTFs improve the integrated file system backup performance on a system with many directories. The biggest benefit will be for systems that have many directories with a small number of files in each directory. You may see minor improvements in other integrated file system backups.

- SI05856 (V5R1)
- SI05599 (V5R2)

Use concurrent backup operations

OS/400^(R) allows you to run multiple concurrent backups. Many iSeries server customers have significantly reduced their backup windows by using multiple concurrent backups. To implement this approach you will need to determine some way to subset and group your integrated file system data. Then you need separate SAV commands to concurrently save each of the subsets. You will need to consider the potential resource contention that can occur on the hardware resources being used. For example, performing concurrent backups on groups of data that are stored on the same set of disk units may cause contention on those disk units.

You may decide to use multiple tape drives or a tape library system with multiple drives to run multiple concurrent SAV commands.

For more information about concurrent backups, see save to multiple devices in the Back up your server topic of the iSeries Information Center.

Save to save files (SAVF) then save the SAVFs to tape with SAVSAVFDTA

Some customers have found that they can reduce their backup window by first backing up their data to a save file (SAVF) rather than saving directly to tape. In V5R2 of OS/400, significant performance improvements were made to backups to save files. Of course if you back up to a save file, you need to have adequate disk space available for the save file. Chapter 15 of the iSeries Performance Capabilities Reference can help you evaluate this approach on your system. You also will need to back up your save files to tape by using the Save Save File Data (SAVSAVFDTA) command. However, the SAVSAVFDTA command does not need to be completed during your backup window.

For more information, see the SAVSAVFDTA command in the Programming topic of the iSeries Information Center.

Reduce or eliminate auditing during backup or recovery operations

Measurements show that performing security auditing during save or restore operations (*SAVRST auditing) can decrease performance. Auditing provides valuable information about the actions being performed on your system and who is performing those actions. However, you need to balance the value of that information against the time you have available to perform a backup or recovery. This is especially true if you need to recover all or many objects in the integrated file system.

For more information about security auditing, see chapter 9 of Security Reference.



Reduce number of objects scanned during the backup

Specifying the SCAN parameter during the backup may have significant performance impacts if scanning is enabled on the system. Scanning objects may be a valuable part of your system security, but you need to consider the amount of time scanning will add to your backup window.

For more information about scanning, see the QsrSave API and the System API Reference.


Use online backups

The following topics are things that you can consider to use online backups.

Use BRMS online backup of Lotus^(R) servers support

Backup, Recovery and Media Services (BRMS) supports online backups of Lotus server databases (such as Domino^(TM) and Quickplace^(TM)). An online backup is a backup that you do while your Lotus server databases are in use; there are no save-while-active synchronization points. You can direct your online backups to a tape device, media library, save file, or a Tivoli^(R) Storage Manager server. BRMS can also create control groups that make it easy to use concurrent backups. Performing an online backup does not improve the performance of your backup. However, since your applications remain active, the duration of the backup is less important.

For more information about BRMS online backups, see Backup Recovery and Media Services. 

If you decide to use the BRMS online backup support, you can tune the performance of the backup to your data. For more information, see performance tuning  on the BRMS web page.

Use save-while-active

The SAV command provides the SAVACT, SAVACTMSGQ, and SAVACTOPT parameters to support saving objects while active.

For more information, see save-while-active in the Back up your server topic of the iSeries Information Center.

Back up less data

The following topics are things that you can consider to back up less data.

Use the CHGPERIOD parameter to only save changed objects

The SAV command provides a CHGPERIOD parameter that can be used to find and save only objects that have changed. In some cases, this can be an effective way to reduce the amount of data you need to back up. However, the system still needs to look at each object to determine which objects have changed. If you have many files it may still take a long time to determine which objects have changed.

For more information, see saving changed objects in directories in the Back up your server topic of the iSeries Information Center.

Structure your directories to easily back up new files, omit data, or subset your data

It may be beneficial to consider your backup strategy when you structure and name your directories. You may be able to group and name your files in some way that will make it easier to include or omit groups of directories or objects from your backups. You might want to group the directories such that you can back up all of the directories and files for an application, a user, or specified time period.

For example, if you are creating many files each day or each week, it might be useful to create a directory to contain the new files. Consider implementing a naming convention for the directories such that you can back up only the directory that contains the new objects or omit older directories.

Example: Create a directory structure that uses the year, month, and week to store new objects.

```
/2003 All files created in 2003
 /01 All files created in January of 2003
  /01 All files created in first week of January of 2003
  /02 All files created in second week of January of 2003
  /03 All files created in third week of January of 2003
  /04 All files created in fourth week of January of 2003
 /02 All files created in February of 2003
```

Omit objects from the backup

OBJ Parameter

The SAV command provides the OBJ parameter that specifies the objects to be included and omitted from the backup. The OBJ parameter lets you specify a list of 300 values to be included or omitted from the SAV command. The values can be either specific directories or objects or generic values that provide wild card support for the objects to be included or omitted.

Here are some examples of reasons why you might want to omit a directory or object from your backup:

- The directory or object is temporary and is not required if you need to recover your system.
- The directory or object is already backed up and has not changed since the last full backup.
- You are trying to subset your integrated file system data so you can run multiple concurrent SAV commands.

Pattern Parameter

The SAV command provides the PATTERN parameter which lets you specify a list of 300 values which are used to subset the save by either including or omitting objects which qualify for the save based on the OBJ parameter. The values can be either specific object names or generic values that provide wild card support for the objects to be included or omitted.

Here are some examples of reasons why you might want to include or omit objects which qualify for the save based on the OBJ parameter:

- You want to save an entire directory tree, but omit objects of a specific type or name.
- You want to save all objects of a specific type without specifying which directories they may reside in.

Note: While less data may be saved, the amount of time to save the data may be increased. If patterns are specified on the PATTERN parameter, any object which qualifies for the save is compared to the list of objects on the PATTERN parameter.

For more information about the SAV command parameters, see SAV in the Programming topic of the iSeries Information Center.

Journal changes and save journal receivers

In OS/400 V5R1M0, journal support was added to the integrated file system. You can journal changes to directories, stream files, and symbolic links. If you implement journaling on your integrated file system data, you may need to change your save strategy. Your new strategy should be to back up the objects less frequently and instead back up the journal receivers that contain the changes you've made to the objects. This could reduce the amount of data you need to back up. However, you will need to understand and consider the implications to your recovery procedures.

For more information, see journal support in the Integrated file system topic of the iSeries Information Center.

Implement Hierarchical Storage Management (HSM)

If you have historical integrated file system data that is infrequently needed you might benefit from implementing Hierarchical Storage Management. Hierarchical Storage Management (HSM) automatically and transparently manages customer data across a storage hierarchy. The storage hierarchy can consist of high performance disk, compressed disk, and tape libraries.

When and how often data is accessed on your server depends on the type of data. A set of data that is currently being used may be accessed many times a day (hot data), or it may have become historical data which is accessed less frequently (cold data).

Through the Backup, Recovery and Media Services (BRMS) user-defined policies, HSM can migrate or archive and dynamically retrieve infrequently used data or historical data up or down a hierarchy of storage devices.




For more information, see Hierarchical Storage Management. 

References and resources





iSeries^(TM) Information Center

- Integrated file system
- Back up your server
- Backup, Recovery and Media Services

Manuals

- Backup and Recovery  (about 600 pages)
- Backup Recovery and Media Services for iSeries  (about 300 pages)
- Security Reference  (about 700 pages)

Web sites

- Backup Recovery and Media Services (BRMS) for iSeries 
- Performance Management Resource Library for iSeries 
- Hierarchical Storage Management (HSM) 
- iSeries Storage 

Disclaimer

Information is provided "AS IS" without warranty of any kind. Mention or reference to non-IBM products is for informational purposes only and does not constitute an endorsement of such products by IBM.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.



Printed in USA