Informix Basic Administration

Part 2: Backup & Recovery

Barrie Shaw

Virtual-DBA Informix/PostgreSQL Lead DBA &
Delivery Manager
XTIVIA Inc.

A Publication of

XTIVIA

TABLE OF CONTENTS

Introduction

Barrie Shaw is a Senior IBM Informix Certified System Administrator with over twenty-five years of experience using Informix software, including database administration, systems administration, application development and support. She leads the XTIVIA Informix Virtual-DBA team for managed services and consulting. Barrie was a member of IBM's Gold Consultant group for 10 years. She enjoys interacting with her clients, teaching them to use Informix and wowing them with her vi skills.

The following eBook covers the backup and recovery of an Informix instance. This includes the Informix tools for archive and restore, performing a table level restore, and backing up the logical logs. Best practices for monitoring and managing the logical logs and archives are also covered.

Part 2

Backup & Recovery

1

Archiving the Instance

1.1 Archiving the Instance

- Informix archives must be performed while the instance is online

- There are three levels: 0, 1 and 2
  - Level 0 is a full archive
  - Level 1 will archive all changes since the last level 0
  - Level 2 will archive all changes since the last level 1

- Informix provides two utilities to use for backups
  - ontape
  - onbar

- Both utilities allow for the backup and restore of storage spaces and logical logs
- ontape
  - Very easy to set up
  - Archives can be written to tape, file, directory or STDIO
  - Does not use a storage manager
  - Can be used to change the logging mode of a database

onbar

- Requires a storage manager (PSM is provided in lieu of 3rd party)

- Requires that critical files be backed up and available for restore

- Allows you to:
  - Select specific storage spaces
  - Restore to a specific point in time
  - Perform separate physical and logical restores
  - Back up and restore different storage spaces in parallel
  - Use multiple tape drives concurrently for backups and restores
  - Perform imported restores
  - Perform external backups and restores

Important things to note:

- onbar and ontape are NOT compatible – You must restore with the same utility that you used to archive

- Archive and logical log backups likewise must be backed up using the same utility

- An archive cannot be restored to a different version of Informix than the one on which it was taken. This includes major version numbers and point releases.

- The OS of the server where the restore is performed must be the same or similar to the one where the archive was taken

1.2 Backing Up with ontape

- Be sure to save a copy of the onconfig file for use during a restore

- Be sure that the tape device or directory for the archive exists

- The TAPEDEV and LTAPEDEV directories or files should be owned by informix:informix

- Permissions on the backup directory must be RWX for the informix user and group

- If using a file for TAPEDEV and LTAPEDEV, permissions should be 660

- Informix will manage file names if you use a directory
- ontape syntax

  – Archive

    - ontape –s   Archive will prompt for level number
    - ontape –s –L 0   Can specify level number

- If not using a directory, ontape will prompt for you to mount a "tape". If the file exists, simply hit return.

```
>ontape -s -L 0

Please mount tape 1 on
/opt/IBM/informix_rss/backups/archives/arch.0 and press
Return to continue ...
```

1.3 Backing Up with onbar

- onbar works in conjunction with a storage manager

- You will need to back up the ixbar and oncfg files associated with the archive for use in a restore

- You should save a copy of the onconfig file

- The archive and logical log devices are configured via the storage manager

onbar syntax

> onbar –b -w

- -w will do a whole system backup, including all storage spaces and logical logs based on a single checkpoint

> onbar –b

- Will back up all storage spaces and used logical logs

Additional options:

- The default is level 0; use "–L" option to specify a level number
- Use the "-cf" option to specify whether or not to back up the critical files with the archive. Valid values are "yes", "no", and "only"
- See the *Informix Backup and Restore Guide* for complete syntax and options

2

Logical Log Backups

2.1 Logical Log Backups

- Transactions for a logged database are written to the logical log files

- In order to be able to restore to the point of failure, logical log files must be backed up and saved

- Logical log backups must be performed using the same utility as the archives

- The ALARMPROGRAM can be configured to back up logs automatically as they fill

- If logical logs are not backed up and all of the logs become full, the instance will hang

- Monitor logical log status using "onstat –l"
  ontape syntax for logical log backups

  - ontape -a   Archive the full logs; will prompt to mount tape and ask if you want to back up the current log

  - ontape -a -d Archive logs to a directory without prompts

  - ontape -c   Continuous log archive; logs back up as they fill. Requires an open session where the command is run. You can use ctrl-C to stop backing up logs.

  onbar syntax for logical log backups

  - onbar -b -l   Archive the full logs
  - onbar -b -l -c   Archive the current log

  If you don't need to save the logical log files, LTAPEDEV can be set to /dev/null so they will be marked as backed up as they fill.

  You will need to perform an archive and restart the instance for this to take effect.

2.2 The Alarmprogram

- ALARMPROGRAM
  - Onconfig parameter that points to a script to be run when Informix triggers an event
  - One event that will trigger the program is a full logical log file
  - Informix provides three scripts in $INFORMIXDIR/etc:
    - alarmprogram.sh - need to set BACKUPLOGS to Y
    - log_full.sh
    - no_log.sh – exits without backing up logs; use only if logs are not being saved
- For alarmprogram.sh and log_full.sh the default setting for BACKUP_CMD is "onbar –b –l"

- When using ontape, change BACKUP_CMD in the script to:
  - "ontape –a –d" when the onconfig LTAPEDEV parameter is set to a directory
  - "ontape –a" when the onconfig LTAPEDEV parameter is set to a file name
  - When backing up to a file, use the following syntax in the script:

    $BACKUP_CMD 2>&1 > $LOGOUT << !

    n
    !

3

Performing a Restore

3.1 Performing a Restore

Restore concepts

- In order to perform a restore to the point of failure, you will need both the archive and logical log backups

- If using level 0 and level 1 archives, you need only the latest level 0 and the latest level 1 archives along with the logical log backups

3.2 Preparing for a Full Restore

Make sure the instance is offline

For ontape:

- Put the file to be restored in the proper location
    - If you need a specific file, put the file name in TAPEDEV
    - If you need the latest archive in the directory, simply use the directory name
- Be sure that logical log backup files are available on the server
- If restoring to a new system, make sure the necessary chunk files are in place and you have put a copy of the onconfig file in $INFORMIXDIR/etc

For onbar:

- The restore file or tape needs to be available to the storage manager
- Be sure that logical log backup files are available to the storage manager
- Put the ixbar and oncfg files associated with the archive in $INFORMIXDIR/etc, or include the "-cf" option to include critical files on the restore
- If not using the latest archive, update ixbar accordingly
- If restoring to a new system, make sure the necessary chunk files are in place and you have put a copy of the onconfig file in $INFORMIXDIR/etc

- ontape restore syntax
  - ontape –r
    - Will prompt for mounted level 0 tape
    - Will display dbspace and chunk list and ask if you want to continue
    - Will ask if you want to back up the logs
    - Once the level 0 is restored, will prompt for level 1
    - Once the level 1 is restored, will prompt for level 2
    - When finished with archives, will prompt to ask if you have logical log files to restore
    - If you answer y when you should have said n, you will need to start from the beginning

3.4 Performing a Restore with onbar

- onbar restore examples

  onbar –r

  - Perform a restore. If the server is off-line, a full cold restore will be performed. If the server is on-line, only the dbspaces that are off-line will be restored

  onbar –r -w

  - Perform a whole system restore. You may only use this if you did a whole system archive

  onbar –r –t *timestamp*

  - Point-in-time restore

  See the ***Informix Backup and Restore Guide*** for complete syntax and options

- If your source server and target server have different chunk names, you will need to use the "rename" option on the restore command

- To rename chunks during a restore, create a file containing the source and target chunk names, including the full path and offset of the chunk

- The format is:

  source-path source-offset target-path target-offset

  Example:

  /opt/IBM/informix_prd/db_spaces/rootdbs 0
  /opt/IBM/informix_rss/db_spaces/rootdbs 0

  /opt/IBM/informix_prd/db_spaces/data01 0
  /opt/IBM/informix_rss/db_spaces/data01 0

  /opt/IBM/informix_prd/db_spaces/temp01 0
  /opt/IBM/informix_rss/db_spaces/temp01 0

- Before restoring, ROOTPATH in the target onconfig file needs to be updated to contain the full path of the *source* server's root dbspace

- Informix will automatically change ROOTPATH back to the path of the *target* root dbspace during the restore

  Command syntax:

  ontape –r –rename –f *renamefile*

  onbar –r –rename –f *renamefile*

3.6 Full-System Restore

A full system restore is needed when a critical dbspace is damaged and needs to be restored. This could be due to a disk failure or corrupted data.

The following list shows critical dbspaces:

• The root dbspace

• The dbspace that contains the physical log

• A dbspace that contains logical-log files

You must start a full-system restore with a cold (off-line) restore.

3.7 Warm, or Partial, Restore

You can do a warm, or partial, restore when damage has occurred to a non-critical dbspace, blobspace, or sbspace.

- A "warm" restore can be performed with the instance online

- If you do not need to restore a critical space, you can restore only those storage spaces that contain a damaged chunk or chunks

- When a disk failure occurs in one chunk of a storage space that spans multiple chunks, all active transactions for that storage space must terminate before it can be restored

- The restore operation can be started before the database server finishes the transactions, but the restore will be delayed until the database server verifies that all transactions that were active at the time of the failure have completed.

4

Monitoring Archives

4.1 Monitoring Archives

- Archives can be monitored with:
  - onstat –g arc

  - Reviewing the online log
    - Messages will be written to the online log at the start and completion or abort of an archive

onstat –g arc

```
IBM Informix Dynamic Server Version 12.10.FC4XK -- On-Line -- Up 27
days 14:00:47 -- 17860352 Kbytes

num    DBSpace            Q Size Q Len  Buffer partnum    size scanner

Dbspaces - Archive Status

name               number level date            log       log-position

rootdbs            1      0     10/15/2016.03:00 3930      0xdf3018

logspace           2      0     10/15/2016.03:00 3930      0xdf3018

swxdbs             6      0     10/15/2016.03:00 3930      0xdf3018

swxlr              7      0     10/15/2016.03:00 3930      0xdf3018

swxlrh             8      0     10/15/2016.03:00 3930      0xdf3018

plogdbs            9      0     10/15/2016.03:00 3930      0xdf3018
```

5

Table Level Restore

5.1 Table Level Restore Using archecker

- Informix provides a program called archecker that will allow you to restore a single table or a subset of a table to a point in time or to the last archive

- Archecker is located in $INFORMIXDIR/bin

- It is capable of performing a physical or logical restore:
  - A physical restore would be performed if your logical log files haven't been backed up or you just want to restore the table to the time that the level 0 archive was performed.
  - A logical restore can restore the database to any point in time that is encompassed by the logical logs. This is useful to restore data that was accidentally deleted or to recover a table that was dropped.

- It should be noted that you cannot use a shared memory connection when performing a table-level restore.
- To perform a table level restore, you'll need the following items:
  - A level 0 archive, from either onbar or ontape
  - An exact copy of the schema of the table to be restored
  - An archecker command file

- The archecker command file consists of 5 sections.
  - Database
  - Set
  - Create table
  - Insert into
  - Restore

5.1 Table Level Restore (continued)

- Database section
  - Specifies the database in which the restore is being performed.

    database stores_demo;

- SET section
  - Controls the different features in the table-level unload library
  - Two options are available: commit and workspace
  - Commit sets the number of records to insert before committing during a physical restore. The default is 1000.
  - Workspace sets the dbspaces to use for the working storage space. The default is the root dbspace.
  - You cannot use temporary dbspaces for the working storage space.

    Examples:

    SET COMMIT to 2000;

    SET WORKSPACE to dbspace1;

Create table section

- Defines the table that is being restored and the table to which you want to restore it

- The definition of the table being restored must be identical to the table as it existed in the database when the archive was performed, including:

  – the dbspace(s) where the table is located

  – any fragmentation schemes, etc.

- You do not need to include the index and constraint definitions

- When defining the restore table you have the following options:

  – a full restore

  – restoring a subset of the columns

  – restoring specific rows within the table.

- The table can be placed in any valid dbspace on the system

- If the table doesn't exist then it will be created; if it does exist, then it will be appended to.

Insert section

- The insert section defines what data you would like to restore from the source table

- The syntax is a standard "insert into.. select * from" command

- Limit the data you want restored, such as the columns or a subset of the rows, by using a standard where clause

      insert into customer_restore

      select * from customer;

Restore section

- The restore section defines the type of restore that you want to perform, logical or physical
  - To perform a physical restore you would use the entry:

    restore to current with no log;

  - For a logical restore you would use:

    restore to "2016-10-17 14:00:00";

Archecker command file example:

```
DATABASE stores_demo;
 create table "informix".customer
      (customer_num serial not null ,
      fname char(15),
      lname char(15),
      company char(20),
      phone char(18)
      ) extent size 16 next size 16 lock mode page;
 create table "informix".customer_restore
      (customer_num serial not null ,
      fname char(15),
      lname char(15),
      company char(20),
      phone char(18)
      ) in datadbs extent size 16 next size 16 lock mode page;
insert into customer_restore
select * from customer;
restore to current with no log;
```

5.1 Table Level Restore (continued)

- Once you have created the command file, run the archecker command specifying the file name.

     archecker -tvds -f customer_restore.cmd

- After the archecker process has completed, the new table, customer_restore, will contain the data that was defined in the insert section.

Conclusion

We hope you found the Informix Basic Administration - Part 2 material useful. It was compiled by our Informix team to assist clients who desired to learn how to do things for themselves. Our customers wanted to know how to handle the basics like installation and configuration and basic maintenance tasks, like how to do backups and restores. Our goal is to provide the basics you need to keep a generic Informix installation running and recoverable. If you want to know the more complicated stuff, like Enterprise Replication (ER), High-availability Data Replication (HDR), Connection Manager, Remote Standalone Secondaries (RSS), Shared Disk Secondaries (SDS), or TimeSeries functionality, please give us a call.

Informix is a robust database system that is in use all over the globe. We've been known to get calls from clients who found it running on a server in their environment, very often hidden in a closet, doing something mission critical. Usually, no version is too old for us to help them, as our team has been doing this for a very long time.

XTIVIA has been providing Informix services for more than 20 years. Most of our team used to work directly for Informix and all have been specializing in Informix for many years. That is the same team you will work with when:

you sign up for our Virtual-DBA managed services

you get an Informix Health check

you need an experienced Informix DBA to help you out

you need someone to talk to about Informix licensing or maintenance

Let us help you. Give us a call today, our DBAs are nice, you'll see.

This eBook is brought to you by

XTIVIA

your trusted partner for Informix technology.

You can look to XTIVIA for all of your Informix support needs, including:

[Virtual-DBA Database Monitoring and Remote Administration](#)

[Health Checks and Performance Tuning](#)

[Informix Replication and High Availability](#)

[Informix DBA Consulting Services](#)

[Emergency Problem Resolution and Troubleshooting](#)

[Informix Software Licensing and Maintenance](#)

Thanks for reading

Informix Basic Administration Part 2

Need more guidance with Informix services?
Call 1-888-685-3101 ext. 2
Or visit virtual-dba.com for more info