There are many presentations on recovering computer systems from disasters and the need for planning BUT what exactly ARE these disasters?
This presentation will give you the opportunity to learn from other people's misfortunes that NOTHING is impossible, and even the most improbable events can (and will) happen to someone
Whilst the presentation doesn't really give any guidelines on how to plan for a disaster, you will certainly leave knowing that you must plan for the worst possible scenario and take nothing for granted

- Objective 1:What makes a disaster so disastrous?
- Objective 2:Why do we need to (keep) testing our disaster recovery plans?
- Objective 3:Are disasters really so unexpected?
- Objective 4:What can we do to avoid our disaster?
- Objective 5:Any more user experiences?

Agenda

- Why do we plan for disasters?
- Why must plans be flexible?
- "Expect the Unexpected"
- Some **MORE** TRUE stories
- Moral

Hopefully the reason WHY we plan is self evident, but the fact that plans must be FLEXIBLE is lost on many people

A rigid plan is GREAT if the disaster you face is the one you planned for

But it won't be

Guaranteed

"Expect the Unexpected" is a favourite management-speak phrase, but it's obviously impossible

You have to expect that your disaster will NOT be something you had planned for, so your recovery plans will have to adapt

Why do we plan for disasters?

- What I really mean is
  "Why do we plan for disaster recovery?"
- Recovering from a complete disaster is NOT simple
- Probably the most complex recovery you will ever have to do
- MUST have a satisfactory backup plan
- MUST have a tried and tested recovery plan
  - Emphasise TRIED AND TESTED!

It's also true that a true disaster recovery is probably the most complex recovery you will ever have to do

You will also be under the highest stress of any recovery scenario you have been in before

The future of your employer is at stake (so is your future employment!)

Your recovery MUST be "second nature" so it must be practised and practised and practised

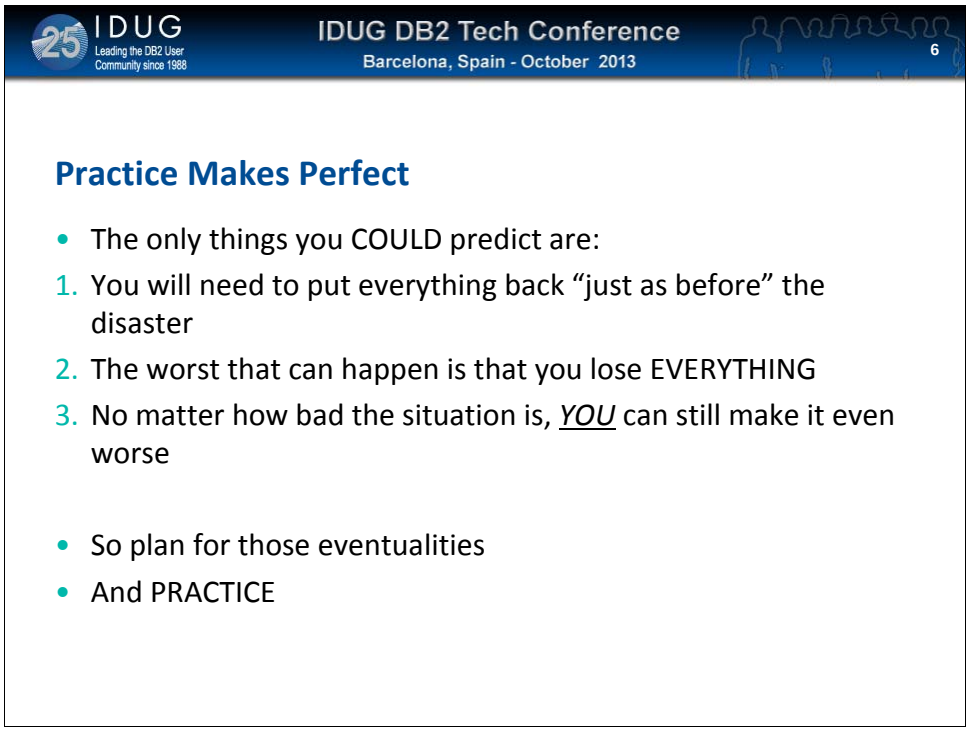
Why must plans be flexible

- Because your disaster will NOT be predictable
- How can you plan for an "unexpected" event?
  - You cannot
- Your recovery must cope with recovering your DB2 subsystem
  - AND all the user data
- Regardless of how they have been lost
  - Or how much has been lost

We've already said that the plan will have to be flexible

To cater for ANY recovery of DB2

You can make one assumption for sure – if you plan for a recovery following the loss of EVERYTHING (data, hardware, people etc.) then your real disaster is unlikely to be even worse



I know I am repeating myself here, but it does bear repeating

PRACTICE

Some potential disasters

- Hurricane Sandy and developers
- Overenthusiastic housekeeping
- Pay attention to maintenance
- Load Replace …. What?
- Controller failure
- Fat fingers

- Some aspects of these stories have been changed
  - To protect the embarassed ☺
- And we meet "Murphy"
  - And his law

So we will now look at some not-so-typical incidents

All of which could have led (and in some cases did lead) to a full disaster recovery

Many of them have something in common which we will touch on later

Although they are all based on real events, I have tried to anonymise them as much as possible

And I need to introduce some of you to Mr Murphy

Murphy's Law

- Thanks to Wikipedia:
- *"If anything can go wrong, it will"*
- *"If there's more than one possible outcome of a job or task, and one of those outcomes will result in disaster or an undesirable consequence, then sometime somebody will do it that way"*
- *"Whatever can go wrong will go wrong, and at the worst possible time, in the worst possible way"*

Murphy's Law

Lots of you will know – if you don't, here are some definitions

Basically Murphy says "Not only will something go wrong, it will do so in the most surprising way or to cause the most damage"

Some of you will have had the feeling, whilst driving, that "If I ever have an accident, I will hit a policeman, a judge or someone really important" – THAT's Murphy

1) Hurricane Sandy and developers

- Here is a rarity – a REAL disaster

- Company on US East Coast well protected against disasters
- All the right backups and all the right plans
- Computers are in a high floor, away from danger
- So are the generators

- But where is the fuel stored?
  - Yes, you guessed it

Hurricane Sandy and developers

- So DR had to be invoked
- No problem, it's all tested and work was moved to one of a number of standby sites
- Business continues
- But what about development?

Hurricane Sandy and developers

- There never was a disaster plan for non-production systems

- Rebuilding all these (including QA) from scratch took WEEKS
  - Meanwhile, lots of developers with no work to do
- Where was the documentation on all these system?
- Yes, under water

Hurricane Sandy and developers

- Question:
- Does your disaster plan encompass non-production systems

- If not, is that a conscious decision?
  - How many developers do you have
  - You have to get their systems back somehow

2) Overenthusiastic housekeeping

- Regular backups of DB2 catalog are being taken
- Regular cleanup of BSDS ensures logs are not kept longer than necessary
- BSDS cleanup and backup frequency don't match

- Congratulations
  You now have a hidden bomb waiting to go off

Overenthusiastic housekeeping

- One day, discovery of a broken page in the catalog
- No problem, just run a recovery – we have regular backups
- AFTER overwriting the table space, recover utility discovers missing log datasets
  - And abends

So now the pages are NOT marked in error, but neither have they been recovered

Of course, none of this is immediately obvious, so normal processing can continue

Until the damage becomes obvious (broken pages, missing data, index/table inconsistencies etc.)

A regular RECOVER utility was run to attempt to recover these broken pages – but the RECOVER utility expected these types or errors to be corrected by the –START command

Overenthusiastic housekeeping

- So now what??
- No backup was taken before recovery was started, so no "Plan B" is possible
- Could do RECOVER TOCOPY instead
  - And gloss over missing data
- But this was the catalog remember!

So what happens now

Performing a disaster recovery will wind us back to a prior point in time, but these are key financial applications – that is not an option

The only solution is to try and undo the damage "on the fly" – of course, the systems must be down while we do this

And we'd better shout for help too – we need all the help we can get our hands on now

Overenthusiastic housekeeping

- As it happens, there really isn't another option
- Luckily this wasn't a production DB2

- As we saw before though, "non production" does not always equal "not important"

- PLEASE ensure that your backups are actually usable before using them!

3) Pay attention to maintenance

- DBA decides a bufferpool is too small
- Arbitrarily decides to make it bigger
- Much MUCH bigger
- Bigger, in fact, than this DB2 member can support

- Member abends with a very specific reason code outlining the problem

Pay attention to maintenance

- As it happens, this was a known problem
- With a fix
- It was on the Sysprogs desk
  - Instead of being applied to DB2

- Anyway, the member was restarted again
- And failed again
- With another very specific reason code

Pay attention to maintenance

- This was also a known problem
- With a fix
- It was also on the Sysprogs desk 
  - Instead of being applied to DB2
- So, the only(?) way forward was a group restart

Pay attention to maintenance

- Production was down for a day
- It took over a week to iron out all the data inconsistencies
- And they lived with all the inevitable -911s

4) Load Replace …. What?

- DBA is practising clearing out all the data in a single partition
- He's using LOAD REPLACE with a dummy input
  - Always the easiest and fastest method

- And the hard way, he learns the difference between
- `LOAD DATA … REPLACE … INTO TABLE … PART n` and
- `LOAD DATA INTO TABLE … PART n … REPLACE`

Load Replace …. What?

- So now the whole table is empty, not just one partition
- At which point he discovers he actually ran the job in production……..
  - Seriously

- Never mind, this is production, lots of backups available
- So he recovered to the most recent image copy …..

Load Replace …. What?

- And lost all of todays updates in the process
- Following a severity 1 call to their tools vendor, the updates were rescued from the DB2 log
- Shortly after, the company decided they had one DBA too many

5) Controller failure

- Hardware problem in DASD controller causes data corruption in DB2 tables
- First symptoms are GRECP/LPL

- All data is mirrored, but surely the problem can't be that widespread
- So let's just do a local recovery to a point prior to the problem surfacing

Controller failure

- Some of the recoveries worked 
- Some of them failed due to missing maintenance 
- Some failed because of missing log data due to the controller problems 

- Luckily, the recovery utility they used detected these problems and refused to recover the data
- So they used an alternate recovery utility which ended cc=0
  - But still didn't recover the data 

Controller failure

- So the recovery plan ended up being:

- *"Every time DB2 finds a broken page, we'll use REPAIR to fix it"*
- Even though just about every repair is unique

Controller failure

- Oh, and that recovery (2-3 slides back) was to the the WRONG PIT
  - Forgot local timezone corrections

- Production access was re-enabled following that recovery
- So following the recovery to the correct PIT
  - Updates occurring between recoveries needed to be applied from the log

Controller failure

- Would switching to the mirror have been a better option?
- Who knows
- But it's hard to see how it would have been a worse choice
- And isn't that what mirrored DASD is for anyway?

6) Fat fingers

- DBA wants to tidy up production
- Amongst the many table spaces, are quite a few obsolete ones

- Easy to work through a list in SPUFI
  - DROP
  - DROP
  - DROP

- They are all small and not being used, so the drops run pretty quickly

Fat fingers

- So why is this one taking longer than the others?
- Yikes! – Mistyped the name and it's an IN USE table space that's being dropped

Fat fingers

- Luckily it is a read-only table space during the day
  - Only updated over night
- So relatively easy to recreate and repopulate
  - And the rebinds were easy too
- A VERY lucky escape
  - And a VERY relieved DBA

Fat fingers

- Would RESTRICT ON DROP have helped?
  - Designed to protect against accidental drops
- Maybe, but our DBA (being totally sure of what he is doing)
  - Would have also coded the requisite ALTER … DROP RESTRICT ON DROP
- Circumventing the protection
- In the good old days, this might have caught the typing error
- But now we do a lot of “cut and paste”ing

ARE YOU WORRIED OR RELIEVED?

Considerations

- Many of the worst disasters are a combination of things going wrong
  - Sometimes three or more
- Each problem can be a disaster in its own right
  - Or could even be fairly trivial
- But the combined effects can be catastrophic
- Planning specifically for a "combination disaster" is impossible
  - How many different disaster possibilities are there?

One common thread you may have noticed is that big disasters are not generally caused by a big problem or a big incident

They are caused by a coincidence of two (or more) seemingly innocuous problems that combine to be catastrophic

Or just plain stupidity/carelessness

This is what makes the planning a bit of a challenge – it is possible to imagine a whole list of things that could go wrong, but the possible combinations of these is pretty close to infinite

Considerations

- This is why a disaster plan should NOT make assumptions about the type of disaster

- Also, PLEASE consider making "**Take a backup before you start**" part of your recovery plan

Hopefully, you can now see why a rigid disaster plan is not the safety net that it appears to be

You need to be flexible in order to cope with anything that a disaster can throw at you

Also, I'd strongly recommend making "take a backup before you start" part of your disaster plan

Considerations

- Seems an odd thing to do – to secure data that you know (or suspect) has been compromised
  - But as we saw with some of our stories
  - There is ALWAYS opportunity to make things even worse
  - At least a backup allows you to get back to where you started
  - AND gives you some "thinking time" while the backup is running
  - There's nothing worse than jumping to the WRONG decision just to be seen to be doing *something*

Taking a backup of something that you are about to over write as part of the recovery seems an odd thing to do, BUT it does have a number of useful advantages

1) No matter how bad things seem to be they can always get even worse – at least a backup means you can get back to where you started if you need to
2) Whilst the backup is running, you have more time to consider your recovery plan without jumping into the first solution that comes to mind

As we saw earlier, there is almost nothing worse than having a disaster and then making it even worse by doing the wrong sort of recovery

A word of warning

- Whilst we are talking about disasters and disasters in the making
- Consider this

- In DB2 V8, we have Simple, Segmented and Partitioned table spaces
- In DB2 9 we have Segmented, Partitioned and Universal

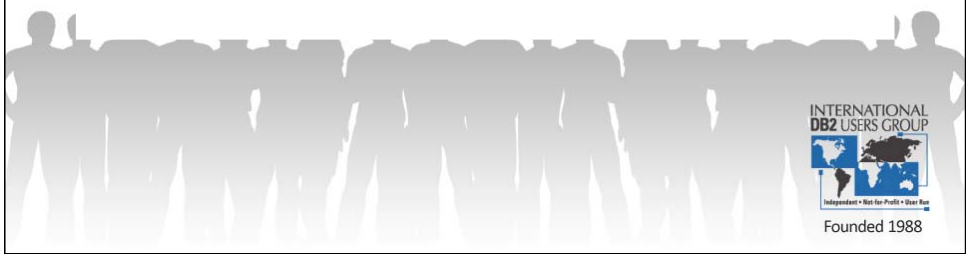
A word of warning

- IF you
  - Accidentally drop a SIMPLE table space
- You CANNOT now recreate it as a simple table space again
- So even if you know where your image copies are
  - You can't recover the data with RECOVER or with DSN1COPY
  - Unload (but how?)/Reload is the only way

And Finally

- I hope none of YOU appear in any future versions of this presentation
- And I hope all your Disaster Recoveries are TESTS
  - And not For Real

Any Questions?



INTERNATIONAL DB2 USERS GROUP
Independent • Not-for-Profit • User Run
Founded 1988

Phil Grainger
BMC Software
phil_grainger@bmc.com

F08

It could happen to YOU
Scary disaster stories, lessons learned, NOTHING is impossible
Part 2

INTERNATIONAL DB2 USERS GROUP
Independent • Not-for-Profit • User Run
Founded 1988