



Creating a Disaster Recovery Plan using Ingres Replicator

INGRES

June 2008

About the Presenter

- **Chip Nickolett has been using Ingres since 1986**
- **He installed and configured the first production productized Replicator installation in 1994**
- **He has over 15 years of proven expertise in Disaster Recovery and Business Continuity Planning**
- **He is currently the Director of Consulting Services in the Americas and can be reached via email at chip.nickolett@ingres.com**

Agenda

- **Defining Disaster Recovery**
- **Ingres Replicator**
- **Replication Planning**
- **Implementing Replicator**
- **Caveats**
- **Failover**
- **Alternatives**
- **Summary**
- **Q&A**

Defining Disaster Recovery

- **Disaster Recovery (DR) is primarily about recovering the infrastructure necessary to operate a business**
- **It assumes that there is some catastrophic event that has led to the need for recovery**
 - The corollary to this is that current production systems will not be available during these recovery efforts
- **It assumes that the event is temporary, and eventually the systems and data will need to be relocated to the production site / systems**

Defining Disaster Recovery (cont.)

- **Common DR / BC Terminology:**
 - RTO: Recovery Time Objective. This is how long a system can be “down” from the point of failure
 - RPO: Recovery Point Objective. This is how much data can be lost due to the failure
 - SDO: Service Delivery Objective. This is the maximum amount of time to have working systems and/or services to provide business functionality, and is usually dependent on multiple systems

Defining Disaster Recovery (cont.)

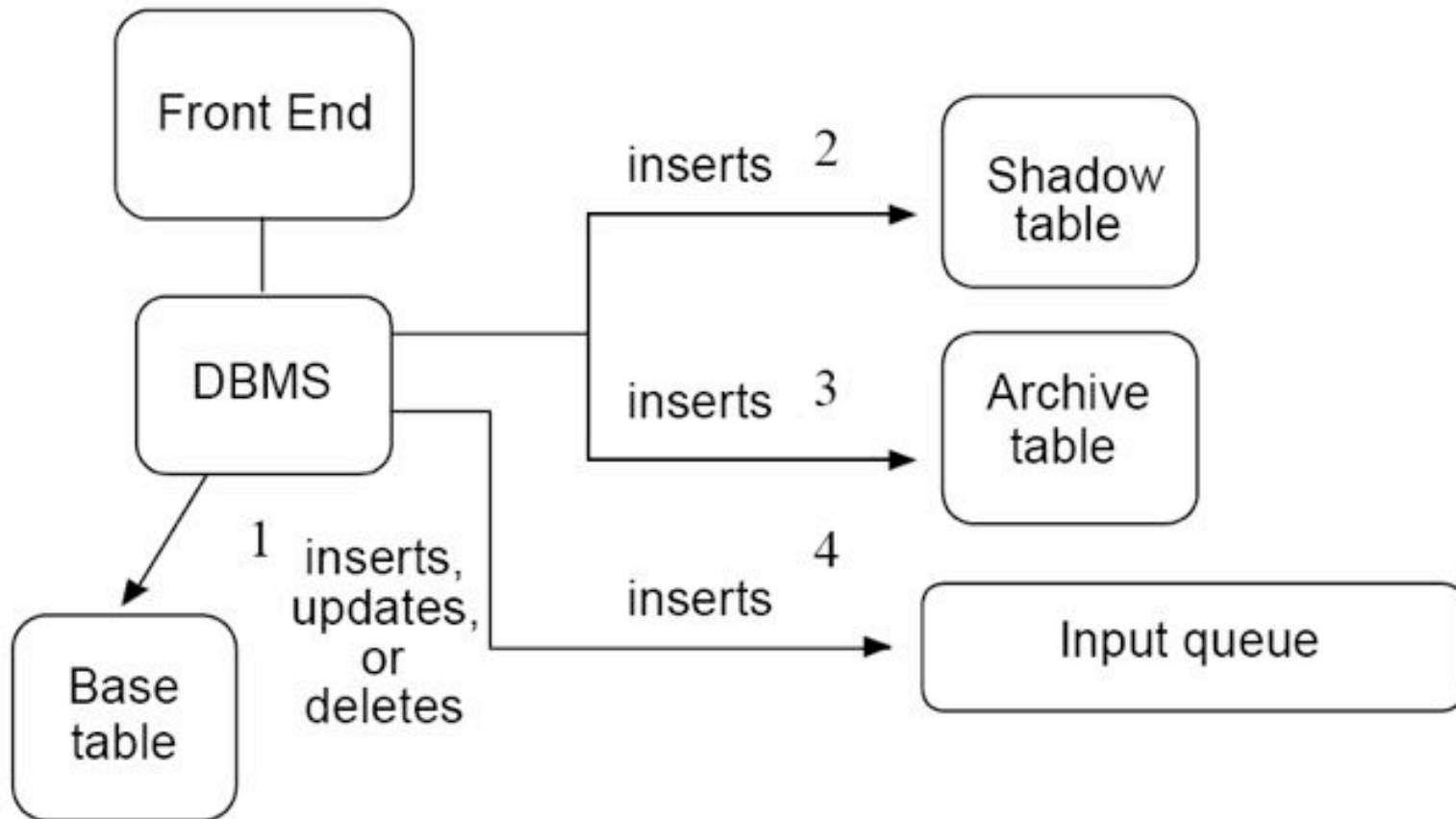
- **Disaster Recovery (DR) is the foundation for Business Continuity (BC) planning**
 - The focus of BC is on business operations, processes and procedures, people and workspaces, etc.
 - BC efforts are more extensive and expensive, and are therefore generally not addressed in a comprehensive manner by most businesses
- **All plans, no matter how good, need to be reviewed, updated, and tested on a regular basis!**

Ingres Replicator

- **Currently included FREE with the Ingres distribution**
- **Documentation available at**
<http://www.ingres.com/downloads/ingres-documentation.php>
- **Can be configured using Visual DBA (VDBA) or on the command line with Replication Manager (repmgr)**
- **Is “rules based” and can therefore be tailored with regard to what is replicated (horizontally and vertically)**
- **Database procedures used to capture data changes**

Ingres Replicator (cont.)

The following figure illustrates the tasks of the Change Recorder:



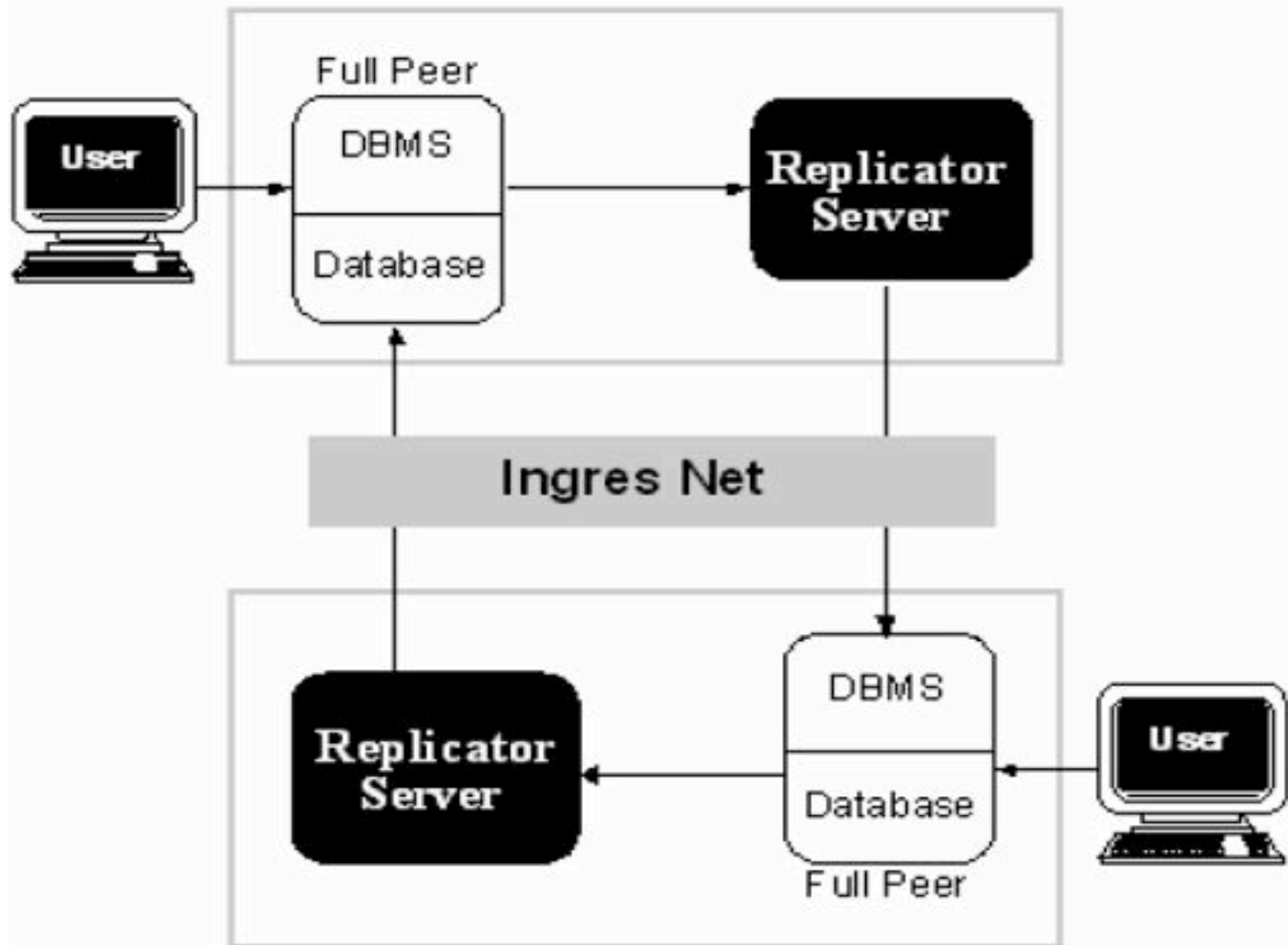
Ingres Replicator (cont.)

- **Replicator has the concept of a “Consistent Distributed Data Set,” or CDDS**
 - Defines what data is replicated
 - Defines where that data resides (“nodes”)
 - Defines the relationships between the various sets of data or nodes
 - Full Peer (source node or peer-to-peer nodes)
 - Protected Readonly (checks for “collisions”)
 - Unprotected Readonly (fast but no safeguards)

Ingres Replicator (cont.)

- **Replicator uses “Data Propagation Paths” to define how data will flow between the nodes**
 - Define the path via originator node, current node, and target node
 - Many ways to configure
 - Peer-to-peer
 - Hub-and-Spoke
 - Cascading
 - Central to Backup

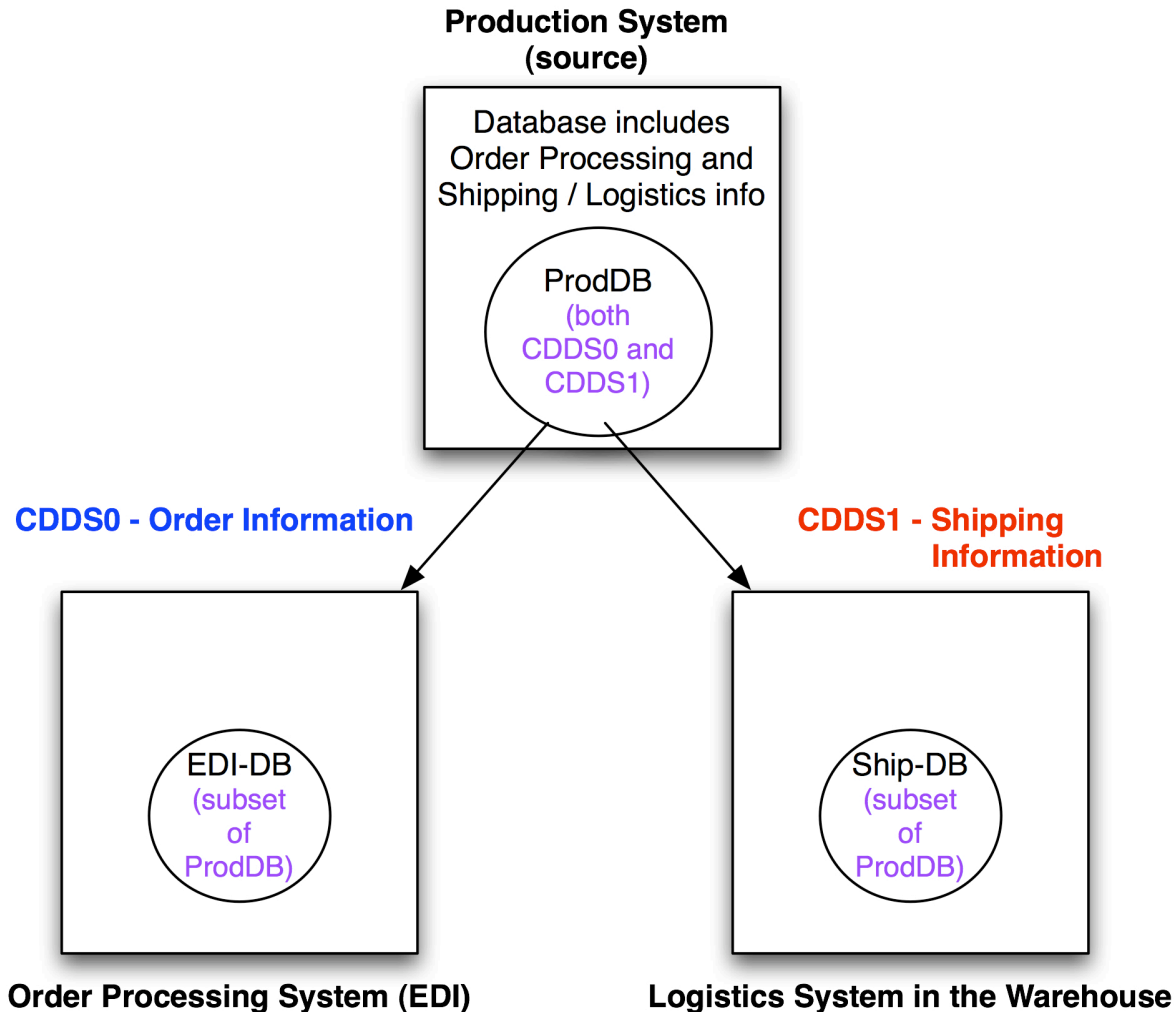
Ingres Replicator (cont.)



Replication Planning

- 1. Identify the Nodes that will be participating in the Replication Environment**
- 2. Identify the consistent data sets (CDDS) relationships between those nodes**
 - It is possible to have multiple CDDS relationships within a single physical database (see next slide)
 - Determine what tables apply, and whether it is the full table or just a subset of the table

Replication Planning (cont.)



Replication Planning (cont.)

- 3. Define the Data Propagation Paths for each CDDS being configured**
- 4. Determine how often replication will occur**

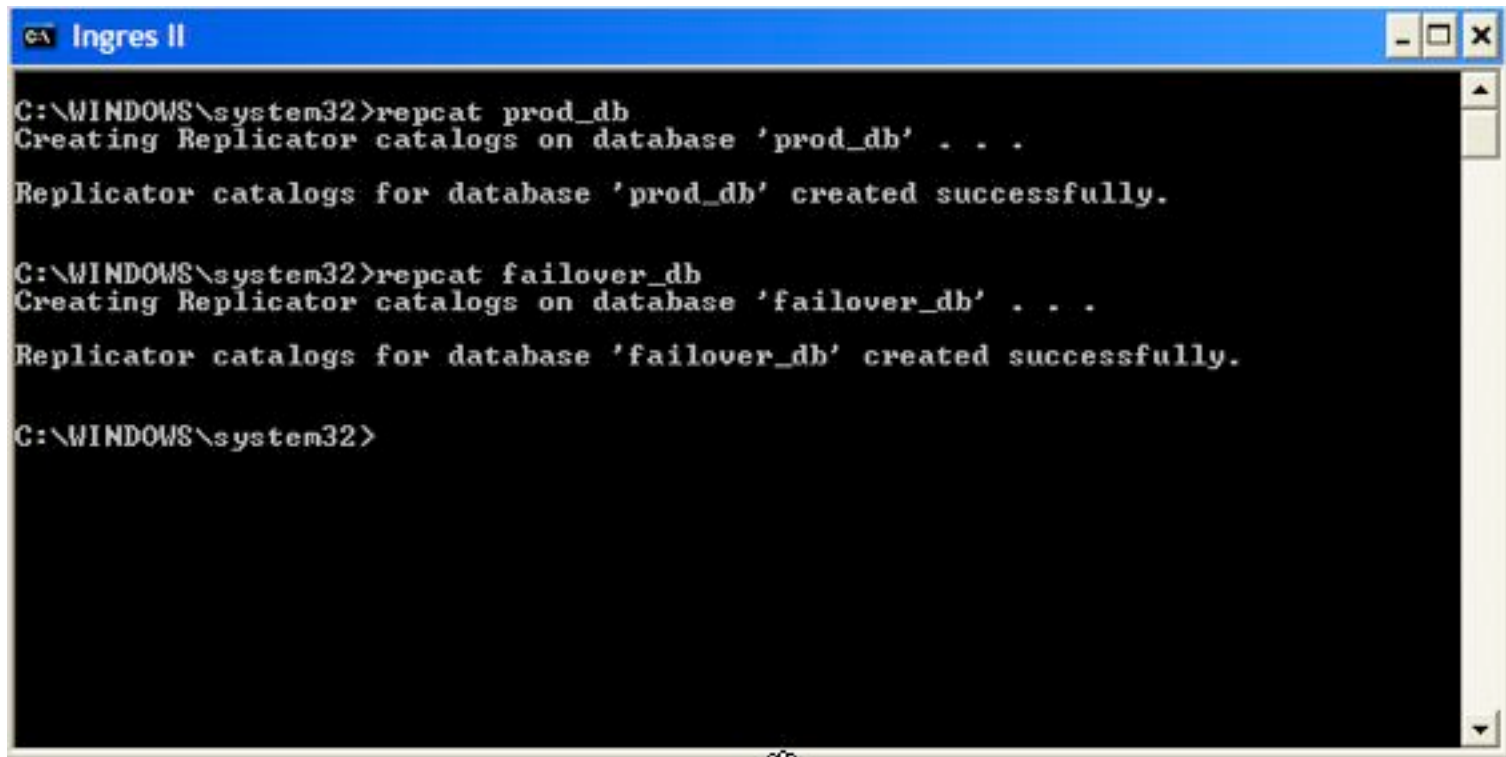
Please note that the worksheets provided in the Replicator User Guide (pages 69, 74, and 75) are very useful. It is highly recommended that they be utilized (if nothing more than a template) for these planning efforts.

Implementing Replicator

- **Checkpoint your databases (ckpdb) before starting**
- **Use either Visual DBA or Replication Manager to perform the required operations**
- **Have your planning worksheets ready**
- **Track your activity and progress**
 - Detailed records will help with troubleshooting in the event of problems
 - This provides an opportunity to gather useful metrics for future reference

Implementing Replicator (cont.)

- Install the Replicator objects using “repcat”



```
C:\WINDOWS\system32>repcat prod_db
Creating Replicator catalogs on database 'prod_db' . . .
Replicator catalogs for database 'prod_db' created successfully.

C:\WINDOWS\system32>repcat failover_db
Creating Replicator catalogs on database 'failover_db' . . .
Replicator catalogs for database 'failover_db' created successfully.

C:\WINDOWS\system32>
```


Implementing Replicator (cont.)

- **Enter the configuration information using “repmgr” or “vdba”**
- **This will usually take two to four hours to fully configure and test a simple configuration**
 - I recommend that creating simple test databases and configuring replication there to get an understanding of the tool without disrupting production installations
 - Note: Production configuration should be performed during a quiet time due to generated schema objects

Implementing Replicator (cont.)

- **Ingres/Net virtual nodes (VNODES) are required for Replicator**
 - I recommend using a system hostname (or “floating hostname”) instead of an IP address (or “floating IP address”) since it is much easier to reconfigure at a system level during a DR event. This minimizes changes to the Ingres and/or applications
- **Traffic will always use Ingres/Net unless the hostname and the VNODE name are the same**

Implementing Replicator (cont.)

■ Troubleshooting

- Define the environment variable DD_RSERVERS to point to the \$II_SYSTEM/ingres/rep directory
- Make sure that there is a “runrepl.opt” configuration file in each server directory
 - Default configuration can be created in repmgr
<dbname> | Monitor | Server | EditConfig | Default
 - repmgr | Resolver | CDDS (generates a report) is a quick and simple configuration sanity check

Caveats

- **Ingres Replicator change logging becomes part of your transaction**
 - Well designed transactions that are concise work best
 - The Ingres Transaction Log file might need to be increased in size to accommodate additional logging
 - Locks per Transaction might need to be increased
- **Tables being replicated do require unique keys**
 - Not always possible or practical (e.g., full row required for uniqueness) in all database schemas

Failover

- **How failover occurs is largely dependent on other aspects of the operating environment and the type / severity of the failure**
 - If the database server system fails then failover can often be quickly performed by “floating” the hostname to the target (replica) system
 - More severe problems (such as a fire in the computer room) may require supporting system recovery before service is resumed

Failover (cont.)

- **Replication can help improve fault tolerance**
 - It is far more likely to have a system outage as opposed to a catastrophic event
 - Replication can minimize service outages in a minimally disruptive manner
 - Database connections will need to be restarted
 - In-progress transactions will be rolled-back on the source system and will need to be redone

Alternatives

- **The Ingres Replicator is a good product, but might not be ideal for all environments**
- **Third-party alternatives that use log-based replication for higher performance**
 - Golden Gate's Transactional Data Management
<http://www.goldengate.com/solutions/highavail.html>
 - PSB's High Volume Replicator (HVR)
<http://www.hvr.biz>

Summary

- **Disaster Recovery planning efforts generally involve multiple systems. The Ingres system will likely be just one piece of that puzzle.**
- **Ingres Replicator can provide the means to update a remote DR system, and then receive changes later as the production systems restored**
- **Even with an automated replication system, testing failover and “failback” operations is still critical**
- **Additional Ingres DR information available at <http://www.ingres.com/downloads/insights.php>**



Questions?

INGRES