Getting Started With Enterprise Replication

With Tom Beebe
tom@advancedatatools.com

Webcast on Dec 14th 2017

Advanced DataTools

Tom Beebe

Tom is a Senior Database Consultant and has been with Advanced DataTools for over 10 years. He has been working with Informix since college with a long time fondness for open source languages. Tom is the lead consultant for Networking, Unix System Administration and Web Development needs. Currently, he is the Project Manager and lead developer on a variety of Web Development projects.

Contact Info:

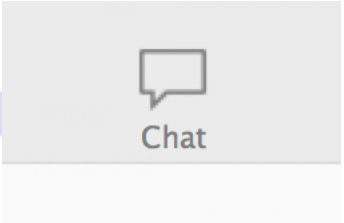
tom@advancedatatools.com

www.advancedatatools.com

703-256-0267 x 106

Webcast Guidelines

- The Webcast is being recorded. The Webcast replay and slides may be available in a few days.
- Please Mute your Line. Background sounds will distract everyone.
- Use the Chat Button in the upper right to ask questions.



About This Webcast

- Intended as an introduction to ER
- Wanted a fairly simple step by step guide to getting up and running
- Explanation of the different Informix replication techniques and when you would want one versus the others
- Avoiding more advanced or complex topics on ER in this particular webcast

Agenda

- What is ER
- Differences between ER and HDR
- Setting up the Informix environment
- Defining the servers
- Defining simple replicates
- Multiple targets from a master server
- Checking the replication status
- Repairing replicates
- Questions

What Is Enterprise Replication

- Log based
- Asynchronous data replication
- Can be Primary-Target or update anywhere
- Efficient and Flexible
- Allows for data repair and synchronization
- Runs on multiple platforms

ER Vs HDR

HDR	Enterprise Replication
Complete instance replication	Replicates tables or even just rows
Does not replication BYTE/TEXT	Replicates BYTE/TEXT
(Default) synchronous replication	Asynchronous replication
Intended for DR	For HA, can be used for DR
All systems must be identical	Can be a mixed environment
Simple to set up	Can be complex to set up
Schema changes automatically replicated	Schema changes must be made individually
Network performance can impact primary	Replication status will not impact the primary server
No schema requirements	Requires PK, ERKeys or Unique Index

Reasons To Use ER

- Only want some of your data on the target server
- Setting up an update-anywhere environment
- Need replication but using BYTE or TEXT columns
- Mixed environment
- Unstable network between source and target servers
- Combining multiple targets into a single target server
- Feeding data from a source server into multiple instances
- Spreading your data and load across multiple instances

ER Replication Types

- Update anywhere
  - All systems are replicated r/w
  - An update at one point will (eventually) filter to all other nodes
- Primary-Target
  - All replicates are read only other than the master
  - Updates will filter down but not back up
  - Tables may not be in sync if there are changes made to target tables directly

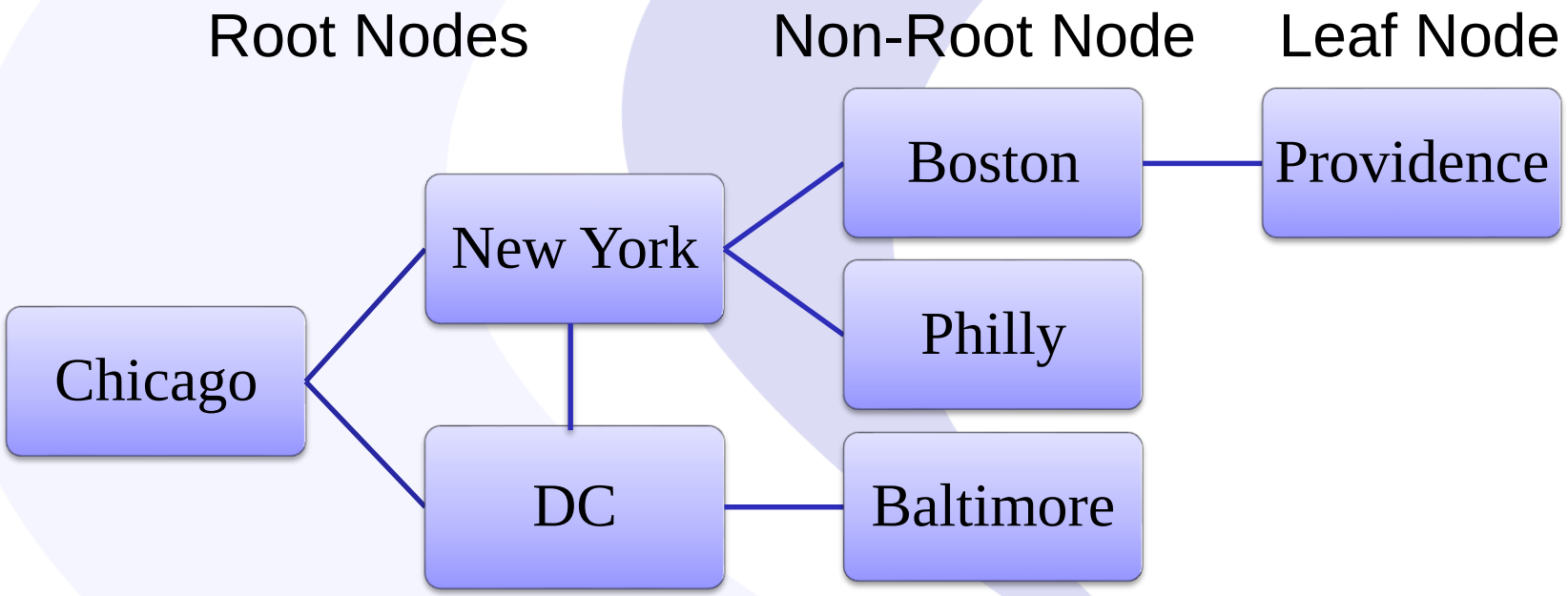
ER Terminology

- Domain
  - An entire cluster of ER Nodes
- Node
  - A particular server
- Replicate
  - A table or set of columns that is being replicated
- Replicate Set
  - Group of replicates
- ATS – Aborted Transaction Spooling
  - Informix on the full failed transaction
- RIS – Row Information Spooling
  - The row information for failed rows

ER Node Types

- Root Nodes
  - Connect to all attached servers
  - Will expect to replicate with all other root nodes
- Non-root Node
  - Will connect to root nodes and other leaf nodes
- Leaf Node
  - Only connects to local non-root or root node
  - Will not connect to other servers
  - Does not contain the full catalog of servers

Node Types


Root Nodes
Non-Root Node
Leaf Node
Chicago
New York
DC
Boston
Philly
Baltimore
Providence

Conflict Resolution

- How the replicates handle cases when two rows collide
- When defining the replicate:
  - Always – always apply the new record regardless
  - Ignore – Do not resolve conflicts
  - Timestamp – Row with the most recent update
  - Deletewins – Delete requests win, otherwise use timestamp
  - Stored Procedure – Use a SPL to determine which row takes precedence
  - Timestamp with SPL – Use a SPL only if the timestamps tie

Setup Steps For Informix

- Set up a sbspace to hold the send and receive queues
- Create dbspace for transaction records
- Define dbspace for syscdr database
- Verify connection between all servers that will replicate
- Verify there are sufficient logs on all systems
- Set up sqlhosts
- Set Informix to use tcp as DBSERVERNAME
- Make ATS and RIS directories

Needed ONCONFIG Changes

- CDR_DBSPACE – where syscdr will be created, rootdbs is default
- CDR_QHDR_DBSPACE – where record headers will be stored
  - 110 bytes per record
  - Estimate records possibly created in 72 hours for sizing
- CDR_QDATA_SBSPACE – Where spooled transactions will be housed
  - Estimate 72 hours of transactions for storage
  - Can have more then one defined, comma separated

Trusted Connections

- Hosts.equiv, .rhosts,
- REMOTE_SERVER_CFG – version 12
  - Can add entries using sql
- Onpassword if it is an untrusted network

SQLHosts

- All ER connections need to be part of a group

```
er_grp          group   -       -       i=10
system_tcp      onsoctcp        systemname      port    g=er_grp
```

- Make sure DBSERVERNAME is the tcp connection and is the same as INFORMIXSERVER
- All servers must have each other listed in their sqlhosts files

Sample SQLHosts

```
g_chicago       group       -          -       i=1
chicago_tcp     onsoctcp    chiserver  sqlexec g=g_chicago

g_ny            group       -          -       i=2
ny_tcp          onsoctcp    nyserver   sqlexec g=g_ny

g_dc            group       -          -       i=3
dc_tcp          onsoctcp    dcserver   sqlexec g=g_dc
```

Set Up Trusted Hosts

- Edit ~informix/.rhosts
- Add <hostname> Informix
- Use dbaccess to verify that the servers can talk both ways
- If you get the error:
  - 951: Incorrect password or user informix@dc is not known on the database server.
  - Add the listed hostname

Setting Up The Data

- Make sure the table has a primary key
  - (v11 can use with ERKey)
  - (v12 can use –key to pass a unique index)
- Use ADD CRCOLS if using timestamp CR
- If you expect to use a good deal of data checking ADD REPLCHECK columns
- ALTER TABLE <table> ADD CRCOLS;
- ALTER TABLE <table> ADD ERKEY;

Define First Node

```
cdr define server \
        -A $INFORMIXDIR/ats \
        -R $INFORMIXDIR/ris \
        -I g_chicago
```

- -A sets the ats directory
- -R sets the ris directory
- -I tells it to initialize the ER system
- The last parameter is the server being initialized

Define Second Node

```
cdr define server \
        -A $INFORMIXDIR/ats \
        -R $INFORMIXDIR/ris \
        -I g_ny \
        -S g_chicago
```

- -A sets the ats directory
- -R sets the ris directory
- -I tells it to initialize the ER system
- -S Tells it to synchronize the catalog with the defined server

CDR List Server

```
informix@chicago:~/scripts$ cdr list server
SERVER                ID STATE     STATUS       QUEUE   CONNECTION CHANGED
------------------------------------------------------------------------
g_chicago             1 Active     Local            0
g_dc                  3 Active     Connected        0       Dec 14 16:16:12
g_ny                  2 Active     Connected        0       Dec 14 16:17:01
```

Define First Replicate

```
cdr define repl -c g_chicago -C always repl_customer \
"stores_demo@g_chicago:informix.customer" "select * from customer" \
"stores_demo@g_ny:informix.customer" "select * from customer"
```

- cdr define repl – tells cdr to define a replicate
- -c - tells it to connect to server g_chicago to run the command
- -C – this is the conflict resolution rule
- The first line is the first replicate source followed by the sql for it
- Second line is the second replicate followed by the sql
- Can have named columns instead of using ‘*’
- Can use different database or table names

Output

Interpreting this replicate as a master replicate.

Verification of stores_demo@g_chicago:informix.customer started

Verification of stores_demo@g_chicago:informix.customer is successful

Verification of stores_demo@g_ny:informix.customer started

Verification of stores_demo@g_ny:informix.customer is successful

cdr list repl

```
informix@chicago:~/scripts$ cdr list repl repl_customer

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          repl_customer
STATE:              Inactive ON:g_chicago
CONFLICT:           Always Apply
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.customer
OPTIONS:            transaction,fullrow
REPLID:             65542 / 0x10006
REPLMODE:           PRIMARY  ON:g_chicago
APPLY-AS:           INFORMIX ON:g_chicago
REPLTYPE:           Master
```

Remote cdr list repl

```
informix@chicago:~/scripts$ cdr list repl -c g_ny repl_customer

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          repl_customer
STATE:              Inactive ON:g_ny
CONFLICT:           Always Apply
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.customer
OPTIONS:            transaction,fullrow
REPLID:             65542 / 0x10006
REPLMODE:           PRIMARY  ON:g_ny
APPLY-AS:           INFORMIX ON:g_ny
REPLTYPE:           Master
```

Starting the Replicate

```
informix@chicago:~/scripts$ cdr start repl repl_customer
informix@chicago:~/scripts$ cdr list repl repl_customer

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          repl_customer
STATE:              Active ON:g_chicago
CONFLICT:           Always Apply
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.customer
OPTIONS:            transaction,fullrow
REPLID:             65542 / 0x10006
REPLMODE:           PRIMARY  ON:g_chicago
APPLY-AS:           INFORMIX ON:g_chicago
REPLTYPE:           Master
```

Multiple Targets

```
cdr define repl -c g_chicago -C ignore repl_stock \
"stores_demo@g_chicago:informix.stock" "select * from stock" \
"stores_demo@g_ny:informix.stock" "select * from stock" \
"stores_demo@g_dc:informix.stock" "select * from stock"

cdr start repl repl_stock
```

Output from cdr list repl

```
informix@chicago:~/scripts$ cdr list repl -c g_dc

CURRENTLY DEFINED REPLICATES
------------------------------
REPLICATE:          repl_stock
STATE:              Active ON:g_dc
CONFLICT:           Ignore
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.stock
OPTIONS:            transaction,fullrow
REPLID:             65541 / 0x10005
REPLMODE:           PRIMARY  ON:g_dc
APPLY-AS:           INFORMIX ON:g_dc
REPLTYPE:           Master
```

Sendq

```
informix@chicago:~/scripts$ cdr view sendq
RQM SENDQ
Server          Trans.    Trans.    Trans.      Data     Memory       ACKS
                in que    in mem    spooled  in queue     in use    pending
--------------------------------------------------------------------------
g_chicago            1         1          0       113        113          0
g_dc                 0         0          0         0          0          0
g_ny                 0         0          0         0          0          0
```

Primary-Target Example

```
cdr define repl -c g_chicago -C ignore repl_stock \
"P stores_demo@g_chicago:informix.stock" "select * from stock"
\
"R stores_demo@g_ny:informix.stock" "select * from stock" \
"R stores_demo@g_dc:informix.stock" "select * from stock"

cdr start repl repl_stock
```

Chicago - Primary

```
informix@chicago:~/scripts$ cdr list repl repl_stock

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          repl_stock
STATE:              Active ON:g_chicago
CONFLICT:           Ignore
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.stock
OPTIONS:            transaction,fullrow
REPLID:             65543 / 0x10007
REPLMODE:           PRIMARY  ON:g_chicago
APPLY-AS:           INFORMIX ON:g_chicago
REPLTYPE:           Master
```

NY - Target

```
informix@chicago:~/scripts$ cdr list repl -c g_ny repl_stock

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          repl_stock
STATE:              Active ON:g_ny
CONFLICT:           Ignore
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores_demo:informix.stock
OPTIONS:            transaction,fullrow
REPLID:             65543 / 0x10007
REPLMODE:           READ-ONLY ON:g_ny
APPLY-AS:           INFORMIX ON:g_ny
REPLTYPE:           Master
```

Replicate Status Check

cdr check replicate --master=g_chicago --repl=repl_stock g_dc g_ny

- --master is the primary server to check from
- --repl is the replicate name
- The trailing options are any servers to check
- Large tables can take a while

```
Dec 14 2017 16:57:31 ------   Table scan for repl_stock start  --------

Node                    Rows     Extra   Missing  Mismatch Processed
--------------- --------- --------- --------- --------- ---------
g_chicago                 74         0         0         0         0
g_dc                      74         0         0         0         0
g_ny                      74         0         0         0         0
```

Repairing Replicates

```
cdr check replicate --master=g_chicago \
--repl=repl_stock --repair --all
```

- Similar syntax but adding –repair fixes mistakes
- --all will check all replicates
- Treats the master as the primary system, it will remove and update rows on all of the other replicates to match the master, so make sure to use the correct master
- Will use logical logs

cdr check flags

- --background – Runs the command in the background
- --name – Sets a name for the check job, needed to look up the status
- --extratargetrows – allows you to define how to handle extra rows found on the target
- --firetrigger – How to handle triggers on the target systems
- --skipLOB – Do not compare large object data
- --timestamp – Instead of master you can base the updates on what row has the most recent timestamp
- --where – Use a where clause to only look at specific parts of the data

cdr sync replicate

- Used to copy missing data to target replicates
- Meant to add missing rows
- Bypasses logical logs
- Good to use if populating a table for the first time

Other Notes

- OAT Does support ER
- Most ER commands can be run through task/admin commands in sysadmin
- ER can be set up using SSL connections
- Extensive monitoring commands (onstat –g cdr, onstat –g rqm, etc)
- cdr error will show any replication errors
- Templates of replicates can be created to easily add new systems to the network

More Notes

- Replicatesets can be created for groups of replicates which allows for managing of those replicates with a single command
- Replicates and ER can be both paused and stopped and started as needed
- ER Grid was introduced to allow for easier management of systems
- If running versions prior to 11 note that while it can replicate with version 11+ systems there can be unexpected issues
- Can set ER up to only replicate on a schedule

Informix Version Notes

- 7.22 – ER Introduced
- 9.3 – UDT and parallel apply added
- 9.40 – Large transaction, HDR and Encryption support
- 10 – Templates, sync/check
- 11.10 – Truncate, onconfig change support, extensive bug fixes
- 11.70 – Flexible grid, coning ER with ifxclone, erkey
- 12.10 – Timeseries, sharding support, removed primary key requirement

Links

- Setting up Informix ER For the First Time
  - http://www-01.ibm.com/support/docview.wss?uid=swg21153338
- Creating a diagram for Informix ER
  - https://www.ibm.com/developerworks/data/tutorials/dm-1204informixerdiagram/index.html
- Nagaraju Inturi's ER Presentation To The WAIUG
  - http://waiug.org/wp-content/uploads/2017/08/Nagaraju_Inturi_ER_WAIUG_Aug_2017.pdf
- ONCONFIG configuration parameters
  - https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.adref.doc/ids_adr_1072.htm#ids_adr_1072__er

Questions?

Send follow-up questions to
tom@advancedatatools.com

Advanced DataTools

Next Webcasts - 2018

- **Stay tuned, the 2018 webcast schedule will be posted soon**

Please register for each webcast here at:
http://advancedatatools.com/Informix/NextWebcast.html

Informix Training 2018

- **Advanced Informix Performance Tuning**
  - February 5-8, 2018
- **Informix for Database Administrators**
  - April 23-26, 2018

- All courses can be taken online on the web from your desk or at our training center in Virginia.
- We guarantee to *NEVER* cancel a course and will teach a course as long as one student is registered!
- Please register early as the last two courses have filled up and we have not been able to accommodate everyone.

http://advancedatatools.com/Training/InformixTraining.html

New Training Servers

Each Student in class will have a server running Informix with:

- 8 Cores
- 16GB RAM
- 1 SSD Disk
- Additional HDDs

Informix Support and Training from the Informix Champions!

Advanced DataTools is an Advanced Level IBM Informix Data Management Partner, and has been an authorized Informix partner since 1993. We have a long-term relationship with IBM, we have priority access to high-level support staff, technical information, and Beta programs. Our team has been working with Informix since its inception, and includes 8 Senior Informix Database Consultants, 4 IBM Champions, 2 IIUG Director's Award winners, and an IBM Gold Consultant. We have Informix specialists Lester Knutsen and Art Kagel available to support your Informix performance tuning and monitoring requirements!

- ***Informix Remote DBA Support Monitoring***
- ***Informix Performance Tuning***
- ***Informix Training***
- ***Informix Consulting***
- ***Informix Development***

Free Informix Performance Tuning Webcast replays at:
http://advancedatatools.com/Informix/Webcasts.html
Email: info@advancedatatools.com
Web: http://www.advancedatatools.com

Thank You

Thomas Beebe

Advanced DataTools Corporation

tom@advancedatatools.com

For more information:

http://www.advancedatatools.com