

REPAIR

```
cdr start repair [-c server ] jobname
-c server --connect=server connect to server
```

```
cdr stop repair [-c server ] jobname
-c server --connect=server connect to server
```

```
cdr define repair [-c server ] jobname [-f ] [-e option ] [-b blocksize ] [-r replicate ] [-R replset ]
-S source_server [ target_server | participants ]
-r replname --replicate replname Name of the replicate to repair
-R replset --replset replset Name of the replset set to repair
-b --blocksize Number of rows per repair block
-e option --extratargetrows option Rule for handling target rows.
option: delete | merge | keep
-S server --syncdatasource server Name of the source server for the job
```

```
cdr delete repair [-c server ] jobname -S source_server
-c server --connect=server connect to server
-S server --syncdatasource server Name of the source server for the job
```

```
cdr list repair [ jobname ] [brief|full]
```

MISC

```
Show CDR Errors:
cdr error [-c server] [-fan] [-p first,last] [-s seqno]
cdr finderr <cdrerrornumber>
```

```
Remaster:
cdr remaster [-c server] -M node replicate "select statement"
-c server --connect=server connect to server
-M --master=<node> define master replicate
```

```
Alter:
cdr alter [-c server] -o[-f DB1:OWNER1.TABLE1
[ DB2:OWNER2.TABLE2 .... DBn:OWNERn.TABLEn]
-c server --connect=server connect to server
-o --on set alter mode
-f --off unset alter mode
```

```
Remove server:
cdr remove <object> - (doesn't work in some versions of 11x, use 'cdr delete -f')
```

```
Swap:
cdr swap shadow [-c server] -p name -P ID -s name -S ID
```

```
Repair:
cdr repair ats ats_file [-C]
cdr repair ris ris_file [-C]
```

```
Stats:
cdr stats rqm [-c server] [-TACSR]
cdr stats rcv [-c server]
```

onstat commands

```
onstat -g options
cat [scope | replname ]
Print Enterprise Replication global catalog information
cdr Print Enterprise Replication statistics
dtc Print statistics for the Enterprise Replication delete table cleaner
dss [ UDR | UDRx ]
Print statistics about data sync threads and user-defined data types
grp [ A|E|Ex|G|L|Lx|M|Mz|P|pager|R|S|SI|Sx|T|UDR|UDRx ]
Print statistics about the Enterprise Replication grouper
nif [ all | sites | serverid | sum ]
Print statistics about the Enterprise Replication network interface
que Print statistics for the Enterprise Replication high-level queues
rcv [serverid]
Print statistics about the Enterprise Replication receive manager
rep [replname]
Print events that are in the queue for the schedule manager
rqm [ ACKQ | CNTRLQ | RECVQ | SENDQ | SYNCQ | FULL | BRIEF | VERBOSE ]
Print statistics of the Enterprise Replication low-level queues
sync Print the Enterprise Replication synchronization status
```

cdr view

(available in v10.00.UC9 and higher)

Note: In v10.00.UC9, execute \$INFORMIXDIR/etc/syscdrsmiadd.sql on all nodes (no connections allowed to syscdr or sysmaster). This step not required in newer versions of 11.x

```
cdr view [-c server] [-r interval] object(s) [options]
-c server --connect=server connect to server
-r interval --repeat=repeat interval in seconds
objects: list of objects separated by space
List of supported objects and their suboptions are:
ddr
servers
sendq
nif
apply
rcv
ris
ats
profile
state
atsdir [-R | -C | -v | -d | -q]
risdir [-R | -C | -v | -d | -q]
-R repair
-C check
-v verbose
-d delete option for repair
-q quiet option for repair
```

Key Terminology

Enterprise Replication Server
Informix database server that participates in data replication

Replicate
Defines participants and attributes of how to replicate data

Master Replicate
Method of ensuring replicated tables have consistent attributes on different servers

Shadow Replicate
Copy of an existing replicate

Participant
Specifies data (database, table and rows) to replicate to which servers

Replicate Set
A set of replicates combined and administered together as one

Template
Mechanism to set up and deploy replication for a group of tables on one or more servers

Global Catalog
The syscdr database on each server that keeps track of ER configuration and state

Key onconfig variables

```
CDR_EVALTHREADS # evaluator threads (per-cpu-vp,additional)
CDR_DSLOCKWAIT # DS lockwait timeout (seconds)
CDR_QUEUEMEM # Maximum amount of memory for any CDR queue (Kbytes)
CDR_NIFCOMPRESS # Link level compression (-1 never, 0 none, 9 max)
CDR_SERIAL # Serial Column Sequence
CDR_DBSPACE # dbspace for syscdr database
CDR_QHDR_DBSPACE # CDR queue dbspace (default same as catalog)
CDR_QDATA_SBSpace # List of CDR queue smart blob spaces
CDR_MAX_DYNAMIC_LOGS # Dynamic log addition disabled by default
CDR_SUPPRESS_ATSRISWARN (see documentation for list of suppression values)
SBSPACETEMP # Temp subspace for grouper paging for evaluating large txns
```



Informix Dynamic Server Enterprise Replication Quick Reference Guide

This Quick Reference Guide was created from the usage listing of the **cdr** utility along with information from the **IBM Informix Dynamic Server Enterprise Replication Guide**. Refer to the documentation for complete information on configuring, using, monitoring and trouble shooting Informix Enterprise Replication.

Quest Information Systems, Inc.
P.O. Box 10495
Burke, VA 22009-0495
www.questinfosys.com
info@questinfosys.com
703-455-6325

CDR Commands

Note: Command options are executed with short or long syntax. Both are notated below.
(I.E. server command options short syntax is '-c server', long syntax is '--connect=server')

SERVER

cdr define server (options) servername

-c server	--connect=server	connect to server
-i min	--idle=min	idle timeout
-s space	--send=space	dbspace where send queue created (obsolete)
-r space	--recv=space	dbspace where recv queue created (obsolete)
-A dir	--ats=dir	directory for Aborted Transaction Spool
-R dir	--ris=dir	directory for Row Information Spool
-I	--init	initialize server
-S server	--sync=server	synchronize catalog (use with -I)
-N	--nonroot	non root server
-L	--leaf	leaf server

cdr delete server servername

cdr modify server [-i min] [-A dir] [-R dir] [-m p | r] [-I on|off] server

-c server	--connect=server	connect to server
-i min	--idle=min	idle timeout
-A dir	--ats=dir	directory for Aborted Transaction Spool
-R dir	--ris=dir	directory for Row Information Spool
-m mode	--mode=mode	set server mode (primary or readonly)

cdr suspend server [-c server] servername

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr resume server [-c server] servername

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr connect server [-c server] servername

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr disconnect server [-c server] servername

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr list server [-c server] servername

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr start [-c server]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr stop [-c server]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

REPLICATE

cdr start repl [-c server] [-S server [-e keep|delete|merge]] replicate [TargetServers]

-c server	--connect=server	connect to server
-S server	--syncdatasource=server	server to be used as a data source for sync
-e	--extratargetrows=[keep delete merge]	handle extra row during sync

TargetServers List of TargetServers to be started. Separate by space.

cdr stop repl [-c server] replicate server(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr define repl [-c server] [-vuiAFIORTm] -C rule(s) [-M master] [-S scope]

[-n y/n] [-t] [-a time] [-e intvl] [-f y/n] [-D y/n]	repl participant	
-M	--master=<node>	define master replicate
-t	--empty	Empty mastered replicate
-n	--name=y/n	mastered replicate name verification
-v	--verify	verify the existing replicates using master definition
-u	--autocreate	automatically create tables if they do not exists
-a time	--at=time	replicate at specified time
-c server	--connect=server	connect to server
-e intvl	--every=intvl	replicate every intvl minutes
-i	--immediate	continuous replication (default)
-A	--ats	aborted transaction spooling
-C rule[,rule]	--conflict=rule[,rule]	conflict resolution rule(s)
-F	--floatcanon	transfer floating point in canonical form (deprecated)
-I	--floatieee	transfer floating point in IEEE form (recommended)
-O	--optimize	don't call procedure unless different server
-R	--ris	row information spooling
-T	--firetrigger	fire triggers when replicating
-S scope	--scope=scope	scope of conflict resolution (row or trans)
-f y/n	--fullrow y/n	Enable/Disable sending of full row for updates
-m <primary repl>	--mirrors <primary repl>	Define a shadow replicate
-D y/n	--ignoredel y/n	do not process any deletes on targets

REPLICATE (cont)

cdr delete repl [-c server] replicate(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr modify repl [-AORT] -C rule(s) [-i] [-a time] [-e min] [-f y/n] replicate

-a time	--at=time	replicate at specified time
-c server	--connect=server	connect to server
-e intvl	--every=intvl	replicate every intvl minutes
-i	--immediate	continuous replication (default)
-A (y/n)	--ats	aborted transaction spooling
-C rule[,rule]	--conflict=rule[,rule]	conflict resolution rule(s)
-O	--optimize	don't call procedure unless different server
-R (y/n)	--ris	row information spooling
-T (y/n)	--firetrigger	fire triggers when replicating
-S scope	--scope=scope	scope of conflict resolution (row or trans)
-f y/n	--fullrow y/n	Enable/Disable sending of full row for updates
-D y/n	--ignoredel y/n	do not process any deletes on targets
-n	--name=n	mastered replicate name verification

cdr change repl [-c server] [-a | -d] [-v | -u] replicate participant

-c server	--connect=server	connect to server
-a	--add	add participant
-d	--del	remove participant
-v	--verify	verify the existing replicates using master definition
-u	--autocreate	automatically create tables if they do not exists

cdr suspend repl [-c server] replicate(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr resume repl [-c server] replicate(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr list repl [-c server] [brief/full] [replicate(s)]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr check repl [-c server] options -m master -r repl target list

-a	--all	include all nodes as targets
-r	--repl=<replicate name>	
-m	--master=<master node>	
-v	--verbose	
-e	--extratargetrows=<delete merge keep>	
-R	--repair	Repair rows in error
-T	--firetrigger=<off on follow>	set trigger firing behavior for sync target servers
off		Do not fire triggers at the sync target servers while applying sync data (default)
on		Always fire triggers at the sync target servers even if the replicate does not have the --firetrigger option enabled.
follow		Fire triggers at the sync target servers only if the replicate has the --firetrigger option enabled.

cdr sync repl [-c server] options -m master -r repl target list

-a	--all	include all nodes as targets
-r	--repl=<replicate name>	
-m	--master=<master node>	
-e	--extratargetrows=<delete merge keep>	
-T	--firetrigger=<off on follow>	set trigger firing behavior for sync target servers
off		Do not fire triggers at the sync target servers while applying sync data (default)
on		Always fire triggers at the sync target servers even if the replicate does not have the --firetrigger option enabled.
follow		Fire triggers at the sync target servers only if the replicate has the --firetrigger option enabled.

REPLICATESET

cdr start replicaset [-c server] [-S server [-e keep|delete|merge]]

RepSetName [TargetServers]		
-c server	--connect=server	connect to server
-S server	--syncdatasource=server	server to be used as a data source for sync
-e	--extratargetrows=[keep delete merge]	handle extra row during sync

TargetServers List of TargetServers to be started. Separate servers in the list by space.

cdr stop replicaset [-c server] RepSetName

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr define replicaset [-c server] [-i] [-e hh:mm] [-a day.hh:mm] [-x] RepSetName repl

-a time	--at=time	replicate at specified time
-c server	--connect=server	connect to server
-e intvl	--every=intvl	replicate every intvl minutes
-i	--immediate	continuous replication (default)
-X	--exclusive	Exclusive Replset.

cdr delete replicaset [-c server] RepSetName

-c server	--connect=server	connect to server
-----------	------------------	-------------------

REPLICATESET (cont)

cdr modify replicaset [-a time] [-c server] [-e min] [-i] RepSetName

-a time	--at=time	replicate at specified time
-c server	--connect=server	connect to server
-e intvl	--every=intvl	replicate every intvl minutes
-i	--immediate	continuous replication (default)

cdr change replicaset [-c server] [-a | -d] groupname replicate(s)

-c server	--connect=server	connect to server
-a	--add	add participant
-d	--del	remove participant

cdr suspend replicaset [-c server] RepSetName(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr resume replicaset [-c server] RepSetName(s)

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr list replicaset [-c server] [RepSetName]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr check replset [-c server] options -m master -s replset target list

-a	--all	include all nodes as targets
-s setname	--replset=<replset name>	
-m	--master=<master node>	
-v	--verbose	
-e	--extratargetrows=<delete merge keep>	
-R	--repair	Repair rows in error
-T	--firetrigger=<off on follow>	set trigger firing behavior for sync target servers
off		Do not fire triggers at the sync target servers while applying sync data (default)
on		Always fire triggers at the sync target servers even if the replicate does not have the --firetrigger option enabled.
follow		Fire triggers at the sync target servers only if the replicate has the --firetrigger option enabled.

cdr sync replset [-c server] options -m master -s replset target list

-a	--all	include all nodes as targets
-s setname	--replset=<replset name>	
-m	--master=<master node>	
-e	--extratargetrows=<delete merge keep>	
-T	--firetrigger=<off on follow>	set trigger firing behavior for sync target servers
off		Do not fire triggers at the sync target servers while applying sync data (default)
on		Always fire triggers at the sync target servers even if the replicate does not have the --firetrigger option enabled.
follow		Fire triggers at the sync target servers only if the replicate has the --firetrigger option enabled.

TEMPLATE

cdr define template TemplateName -C rule(s) -M server [-S scope]

[-c server] [-xaFIORTX] [-D y/n] -d database [-f file][Table(s)]		
-C rule[,rule]	--conflict=rule[,rule]	conflict resolution rule(s)
-S scope	--scope=scope	scope of conflict resolution (row or trans)
-M	--master=<nnode>	define master replicate
-c server	--connect=server	connect to server
-A	--ats	aborted transaction spooling
-F	--floatcanon	transfer floating point in canonical form (deprecated)
-I	--floatieee	transfer floating point in IEEE form (recommended)
-O	--optimize	don't call procedure unless different server
-R	--ris	row information spooling
-T	--firetrigger	fire triggers when replicating
-D y/n	--ignoredel y/n	do not process any deletes on targets
-X	--exclusive	Exclusive template.
-d	--database=<name>	database to be used to obtain the table info
-a	--all	include all tables in template
-f	--file=<name>	file to obtain the list of table participants

cdr delete template [TemplateName(s)] [-c server]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr list template [-c server] [brief/full] [TemplateName(s)]

-c server	--connect=server	connect to server
-----------	------------------	-------------------

cdr realize template TemplateName [-c server] [-D dbspace] [-ouvt]

[-S server] [-e delete keep merge] "db@serv" ...		
-c server	--connect=server	connect to server
-D	--dbspace=<spacename>	dbspace to be used to create tables.
-u	--autocreate	automatically create tables if they do not exists
-v	--verify	verify the existing replicates using template definition
-t	--target	realize template as a receive only server
-S server	--syncdatasource=server	server to be used as a data source for sync
-e	--extratargetrows=[keep delete merge]	handle extra row during sync
-o	--applyasowner	realize template as the owner of the table