

Willkommen zum Informix Newsletter

Inhaltsverzeichnis

| | |
|---|----|
| Aktuelles..... | 2 |
| TechTipp: Replikation – Übersicht (HDR/RSS/SDS/CDR)..... | 3 |
| TechTipp: Replikation – sqlhosts Beispiel..... | 5 |
| TechTipp: CDR – Setup CDR Server | 5 |
| TechTipp: CDR – Send-Only..... | 6 |
| TechTipp: CDR – Primary - Read-Only | 7 |
| TechTipp: CDR – Update anywhere | 8 |
| TechTipp: CDR - Intervall | 8 |
| TechTipp: CDR – Conditional Replication..... | 9 |
| TechTipp: CDR – Conditional Replication beim UPDATE | 10 |
| TechTipp: CDR – Weitere Einsatzgebiete..... | 10 |
| TechTipp: CDC – InfoSphere Change Data Capture | 10 |
| Über den Tellerrand: Foglight - Der Nebel lichtet sich..... | 11 |
| Nutzung des INFORMIX Newsletters..... | 13 |
| Die Autoren dieser Ausgabe..... | 14 |

Aktuelles

Liebe Leserinnen und Leser,

die Natur startet durch und zeigt uns die schönsten Blüten.

Auch INFORMIX startet im März mit den TechDays im Watson Tower in München und bietet neben vielen Informationen auch wieder die Gelegenheit sich zu begegnen, Erfahrungen auszutauschen und sich weiter zu vernetzen.

Da der Newsletter oft die aktuellen Themen im Informix Team der Abteilung IBM Expert Labs widerspiegelt, ist diese Ausgabe geprägt vom Thema „Replikation“. Informix bietet hierbei eine Vielzahl an Optionen, die man bei Mitbewerbern vergeblich sucht.

Ihr TechTeam



TechTipp: Replikation – Übersicht (HDR/RSS/SDS/CDR)

HDR: High availability Data Replication

RSS: Remote Standalone Secondary (oder Server?)

SDS: Same Disk Secondary

CDR: Continuous Data Replication

Die erste Frage, die sich stellt, ist: „Wofür benötige ich eine Replikation?“

Antworten auf diese Frage können z.B. sein

- Ausfallsicherheit
- Lastverteilung
- Datenverteilung
- Modernisierung
- Migration
- Datenverknüpfung

Die nächste Frage betrifft die Plattform und die Informix Version

- Selbes Betriebssystem und selbe Informix Version
- Anderes Betriebssystem oder andere Informix Version

Für die Ausfallsicherheit bieten sich **HDR** und **RSS** an. Beide bedingen, dass die Primäre Instanz und das Replikat auf dem gleichen Betriebssystem mit identischer Informix Version aufgesetzt werden.

Die Replikation **HDR** ist eng an den Primary Server gekoppelt und kann diesen ggf. auch blockieren, falls die Abarbeitung der Loginformationen am Secondary Server nicht nachkommt. Es kann es nur eine HDR Secondary Server Instanz geben. Meist sind HDR Primary Server und HDR Secondary Server in unterschiedlichen Brandabschnitten im selben Gebäude mit schneller Kommunikation verbunden.

Die Replikation **RSS** kann sowohl „near sync“, als auch mit einem eingestellten Delay arbeiten, bei dem die RSS Instanz immer ein bestimmtes Zeitintervall dem Zustand der Primary Instanz hinterher läuft. Eine RSS Instanz kann den Primary Server nicht blockieren, sondern es vergrößert sich nur kurzzeitig die Latenz zwischen den Datenbeständen. Es können viele RSS Instanzen mit einer Primary Instanz verknüpft werden. Da RSS nicht so eng an den Primären Server angebunden ist, wird diese Option oft für weit entfernte Standorte genutzt.

Sowohl HDR als auch RSS bieten sich als Lösungen zur Ausfallsicherheit an, da diese Secondary Server mittels „Failover“ zum Primary Server werden können. Der Failover kann manuell, oder durch den ConnectionManager veranlasst werden.

Sowohl HDR als auch RSS Server können genutzt werden, um die Last komplexer Abfrage vom Primary Server auf den Secondary Server zu verlagern. (*)

Sowohl HDR als auch RSS Server können als „Updatable“ eingerichtet werden, bei denen der Secondary sowohl Insert, als auch Update und Delete erlaubt. (*)

(*) hierbei ist die Lizenzierung zu beachten.

SDS ist im eigentlichen Sinn keine Replikation, da hier gemeinsame Platten für die Datenhaltung genutzt werden. Hierbei werden die Ressourcen wie CPU und Hauptspeicher weiterer Rechner zur Verarbeitung genutzt, um den Primären Server zu entlasten. Jede dieser Instanzen nutzt eigene Temporäre Bereiche und nutzt für die Verarbeitung den eigenen Cache im Hauptspeicher. Für das Lesen und Schreiben der Daten wird koordiniert auf die gemeinsamen Daten der DBSpaces auf Platte zugegriffen. Auch eine SDS Instanz kann im Notfall den Primary Server ersetzen, jedoch nicht, falls es ein Problem mit dem Speichersystem gab.

Sowohl HDR, als auch RSS und SDS übernehmen Strukturänderungen wie das Hinzufügen eines neuen DBSpaces, eines Chunks, sowie Änderungen an den Datenbanken und Tabellen automatisch.

Die **Enterprise Replikation**, auch als **CDR** bekannt, bietet die Möglichkeit zwischen unterschiedlichen Versionen von INFORMIX zu replizieren, die zudem auf unterschiedlichen Plattformen installiert sein können.

So kann z.B. eine Informix Version 11.70 auf Windows mit Informix 12.10 auf AIX und Informix 14.10 auf Linux Daten austauschen.

Die Replikation basiert nicht wie bei HDR und RSS auf einer Kopie der Primary Instanz, sondern repliziert Daten logisch zwischen Tabellen mit gleicher Struktur.

Die Namen der Datenbanken, Tabellen und Spalten von Ziel und Quelle dürfen sich unterscheiden.

Die Replikation kann gerichtet oder bidirektional eingerichtet werden.

Es besteht die Möglichkeit die replizierten Daten sowohl im Umfang (Spaltenauswahl), als auch durch eine Where-Bedingung einzuschränken.

Die Replikation wird auf Basis von Tabellen definiert, die je Tabelle gestartet, gestoppt oder suspendiert werden kann.

Da die CDR keine Kopie einer Primären Instanz darstellt, ist eine automatische Übernahme nicht vorgesehen.

Werden jedoch alle Datenbanken und Tabellen repliziert, und entsteht somit eine Kopie, kann mittels Gruppe in der Datei sqlhosts ein Redirect von Clients eingerichtet werden, sollte einer der Server nicht erreichbar sein.

Jeder der CDR Server ist eine eigenständige Instanz und damit uneingeschränkt nutzbar.

Das Hinzufügen von DBSpaces oder Chunks wird nicht an die anderen Instanzen übertragen.

Änderungen an Datenbanken oder Tabellen werden nicht automatisch übernommen und müssen manuell angepasst werden.

TechTipp: Replikation – sqlhosts Beispiel

Die Basis für jegliche Replikation ist neben den Transaktionslogs die Verbindung zwischen den Instanzen. Diese wird über die Datei „sqlhosts“ definiert, in der je Zeile eine Verbindungsdefinition enthalten ist.

Das folgende Beispiel enthält für die Instanzen in München die Einträge zum Primary Server, zur HDR Instanz, RSS Instanz und SDS Instanz, sowie die Server in Frankfurt und Hamburg.

Die mit **ROT** gekennzeichneten Einträge sind für die Enterprise Replikation notwendig. Hierbei ist zu sehen, dass z.B. im Cluster München alle Instanzen dieses Standortes der CDR bekannt sind. Sollte eine Übernahme der Primären Rolle durch HDR, RSS oder SDS erfolgen, so würde die Replikation mit den Instanzen der anderen Standorte problemlos weiter funktionieren.

```

mucl_rep   group   -      -      i=1
mucl        onsoctcp  mucl   9001   g=mucl_rep # Primary Munich
muc2        onsoctcp  muc2   9002   g=mucl_rep # HDR MUC
muc3        onsoctcp  muc3   9003   g=mucl_rep # RSS MUC
muc4        onsoctcp  mucl   9004   g=mucl_rep # SDS MUC

fra5_rep   group   -      -      i=5
fra5        onsoctcp  fra1   9005   g=fra5_rep # Primary Frankfurt

ham6_rep   group   -      -      i=6
ham6        onsoctcp  ham1   9006   g=ham6_rep # Primary Hamburg
ham7        onsoctcp  ham2   9007   g=man6_rep # RSS for HH

```

TechTipp: CDR – Setup CDR Server

Die Basis für die Replikation sind die Replikations Server, die mit den folgenden Befehlen definiert werden:

```

cdr define server -c mucl_rep mucl_rep -I
cdr define server -c fra5_rep fra5_rep -I -S mucl_rep
cdr define server -c ham6_rep ham6_rep -I -S mucl_rep

```

Der zuerst definierte Server ist die Basis, auf die sich die weiteren Replikations Server synchronisieren.

Mit der Initialisierung wird die Datenbank syscdr erstellt, und die Threads für die Replikation werden gestartet.

Die Verbindung der Replikations Server kann mittels `cdr list server` überprüft werden:

```

cdr list server
SERVER          ID STATE   STATUS      QUEUE   CONNECTION   CHANGED
-----
muc1_rep        1 Active   Local       0
fra5_rep        5 Active   Connected   0       Feb 23 07:45:50
ham6_rep        6 Active   Connected   0       Feb 23 07:45:56

```

Mit der Verbindung der CDR Server wird noch nichts repliziert. Für die Replikation müssen Replicates je Tabelle definiert werden.

TechTipp: CDR – Send-Only

Die häufigste Anwendung der Replikation ist die „gerichtete Replikation“, bei der eine Seite als Quelle, die andere Seite als Ziel definiert wird.

Um sicherzustellen, dass keine Daten in die falsche Richtung repliziert werden, besteht die Option zu einem Replikat einen Server als „Send-Only“ zu definieren.

Im folgenden Beispiel ist der Server `muc1_rep` nur Quelle und die beiden anderen Server nur Ziel:

```

cdr define repl -c muc1_rep
-i -A -C ignore -R -S tran -f y -n n artikel_repl
"S demo@muc1_rep:informix.artikel" "select * from artikel"
"R demo@fra5_rep:informix.artikel" "select * from artikel"
"R beispiel@ham6_rep:informix.artikel" "select * from artikel"

```

Durch die Angabe `S=Source` wird definiert, dass `muc1_rep` in diesem Replikat nur Sender ist, die anderen beiden Instanzen sind ausschliesslich Empfänger.

`cdr list repl:`

```

DEFINED REPLICATES ATTRIBUTES
-----
REPLICATE:      artikel_repl
STATE:          Active ON:muc1_rep
CONFLICT:       Ignore
FREQUENCY:      immediate
QUEUE SIZE:     0
PARTICIPANT:    demo:informix.artikel
OPTIONS:        transaction,ris,ats,fullrow
REPLID:         65561 / 0x10019
REPLMODE:       SEND-ONLY ON:muc1_rep
APPLY-AS:       INFORMIX ON:muc1_rep
REPLTYPE:      Master

```

TechTipp: CDR – Primary - Read-Only

Das nächste Beispiel zeigt zwei Primary Knoten in der Replikation, die sowohl Sender, als auch Empfänger sind, sowie einen Knoten, der Read-Only ist:

Im folgenden Beispiel ist der Server ham6_rep nur Ziel. Die beiden anderen Server sind Quelle und Ziel:

```
cdr define repl -c mucl_rep
-i -A -C ignore -R -S tran -f y -n n artikel_repl
"P demo@mucl_rep:informix.artikel" "select * from artikel"
"P demo@fra5_rep:informix.artikel" "select * from artikel"
"R beispiel@ham6_rep:informix.artikel" "select * from artikel"
```

Durch die Angabe R=Receiver wird definiert, dass ham6_rep in diesem Replikat nur Empfänger ist und keine Daten senden kann.

cdr list repl:

```
DEFINED REPLICATES ATTRIBUTES
-----
REPLICATE:      artikel_repl
STATE:          Active ON:mucl_rep
CONFLICT:       Ignore
FREQUENCY:      immediate
QUEUE SIZE:     0
PARTICIPANT:    demo:informix.artikel
OPTIONS:        transaction,ris,ats,fullrow
REPLID:         65561 / 0x10019
REPLMODE:       PRIMARY ON:mucl_rep
APPLY-AS:       INFORMIX ON:mucl_rep
REPLTYPE:      Master
```

Führt man die selbe Abfrage auf dem Server ham6_rep aus, so zeigt sich:

```
REPLICATE:      artikel_repl
STATE:          Active ON:ham6_rep
CONFLICT:       Ignore
FREQUENCY:      immediate
QUEUE SIZE:     0
PARTICIPANT:    beispiel:informix.artikel
OPTIONS:        transaction,ris,ats,fullrow
REPLID:         65561 / 0x10019
REPLMODE:       READ-ONLY ON:mah6_rep
APPLY-AS:       INFORMIX ON:ham6_rep
REPLTYPE:      Master
```

TechTipp: CDR – Update anywhere

Oft sollen Daten zwischen allen Servern synchron gehalten werden, so dass jede Änderung an einem Server sofort die Daten auf allen anderen Servern anpasst. Um dies einzurichten wird der Replikations Modus einfach weggelassen. Damit sind alle beteiligten Server PRIMARY und können sowohl Daten senden, als auch empfangen:

```
cdr define repl -c mucl_rep
-i -A -C ignore -R -S tran -f y -n n artikel_repl
"demo@mucl_rep:informix.artikel" "select * from artikel"
"demo@fra5_rep:informix.artikel" "select * from artikel"
"beispiel@ham6_rep:informix.artikel" "select * from artikel"
```

Spätestens an dieser Stelle sollte man sich Gedanken machen, was bei Konflikten passiert. Als Optionen für die Konfliktresolution (-C) gibt es:

- ignore
- always
- timestamp
- deletewins

Details hierzu würden den Umfang des Newsletters sprengen.

Beim Scope (-S) ist die Sache einfacher.

Hier gibt es nur die Auswahl aus „row“ und „transaction“. Sobald Foreign Keys mit im Spiel sind, macht nur noch „transaction“ Sinn.

TechTipp: CDR - Intervall

Die Replikation kann sofort und kontinuierlich, oder in Intervallen ablaufen. Dazu wird bei der Definition der Replikate das Intervall angegeben als

- -i # Sofort (immediate)
- -e xxx # Alle x Minuten (every)
- -a xx:xx:xx # Zu bestimmten Zeiten (at)

Beispiele:

```
-e 42 # Alle 42 Minuten
-e 4:23 # Alle 4 Stunden und 23 Minuten
-a 19:30 # Um 19:30
-a Mon.05:13 # Jeden Monat um 05:13
-a 1.02:55 # Am Ersten Tag des Monats um 02:55
-a L.23:55 # Am letzten Tag des Monats um 23:55
```

TechTipp: CDR – Conditional Replication

Die Replikation des gesamten Inhalts von Tabellen ist nicht immer gewünscht. Die CDR bietet die Möglichkeit, sowohl die Auswahl der Spalten zu reduzieren, die übertragen werden sollen, als auch die Auswahl der Datensätze mittels Where-Bedingung zu kontrollieren.

```
cdr define repl -c mucl_rep
-i -A -C ignore -R -S tran -f y -n n order_FRA_repl
"P demo@mucl_rep:informix.order"
"select o_id, c_nr, ship_location, o_date
  from order
  where ship_location = 'FRA'"
"R demo@fra5_rep:informix.order"
"select o_id, c_nr, ship_location, o_date
  from order
  where ship_location = 'FRA'"
```

```
cdr define repl -c mucl_rep
-i -A -C ignore -R -S tran -f y -n n order_HAM_repl
"P demo@mucl_rep:informix.order"
"select o_id, c_nr, ship_location, o_date
  from order
  where ship_location in ('BRE','HAM')"
"R demo@ham6_rep:informix.bestellung"
"select b_id, c_nr, lieferort, bestelldatum
  from bestellung
  where lieferort in ('BRE','HAM')"
```

In diesem Beispiel werden nur vier Felder der Bestellungen von München an die Standorte Frankfurt und Hamburg übertragen. Alle anderen Felder sind nur in der Instanz in München verfügbar.

Nur die Bestellungen mit Lieferort „**BRE**men“ oder „**HAM**burg“ werden an die Instanz in Hamburg repliziert. Von den Bestellungen mit Lieferort Frankfurt sieht Hamburg nichts. Die Bestellungen mit Lieferort „**FR**ankfurt“ werden nur nach Frankfurt repliziert. Lieferungen in andere Gebiete werden nicht repliziert und bleiben bei der Erfassung in München.

TechTipp: CDR – Conditional Replication beim UPDATE

Die Replikation im vorhergehenden Beispiel war so definiert, dass Bestellungen mit Zielort Bremen nur nach Hamburg repliziert wurden.

Was passiert, wenn nun eine solche Bestellung in München geändert wird, so dass sich der Lieferort auf Frankfurt ändert ?

Die Replikation macht aus diesem UPDATE in München einen DELETE in Hamburg und einen INSERT in Frankfurt, so dass die Bestellung mit dem Commit des Updates nur noch in Frankfurt zu sehen ist.

Damit stellt die Enterprise Replikation nicht nur ein einfaches Verteilen von Daten dar, sondern kann auch eine gewisse Logik der Datenverteilung gewährleisten.

Viele weitere Optionen der Enterprise Replikation konnten hier im Newsletter nicht im Detail dargestellt werden, wie z.B. Templates, Master Replicates, und Vieles mehr.

TechTipp: CDR – Weitere Einsatzgebiete

Da die CDR Replikation auf allen unterstützten Betriebssystemen und Versionen von Informix einsetzbar ist, wird diese häufig genutzt um eine Informix Datenbank zu migrieren.

Hier könnte z.B. die Quelle eine INFORMIX 12.10 auf HP-UX sein, die auf eine aktuelle Version 14.10 auf Linux repliziert wird, um so eine Downtime bei der Migration zu minimieren. Wird eine bidirektionale Replikation eingerichtet, so lässt sich die Migration Abteilung für Abteilung realisieren.

Die CDR Replikation lässt es zu, dass von einer 7-Bit, bzw. 8-Bit Datenbank in eine Datenbank mit UTF8 repliziert wird. Damit lässt sich die Modernisierung auf internationale Märkte mit sehr geringer Downtime realisieren. In diesem Fall ist jedoch keine Replikation in die Gegenrichtung möglich, so dass bis zum Umstieg auf das neuen System nur lesende Anwendungen sinnvoll sind.

TechTipp: CDC – InfoSphere Change Data Capture

Wem die INFORMIX Welt der Replikation nicht genug ist, der kann mittels IBM DataReplication (früher bekannt unter Change Data Capture CDC) auch zwischen der INFORMIX Welt und anderen Relationalen Datenbanken, oder nach Kafka replizieren.

In einem heterogenen Umfeld mit INFORMIX, DB2, POSTGRES, ... und KAFKA sind die Optionen der Replikation damit nahezu unbegrenzt.

Über den Tellerrand: Foglight - Der Nebel lichtet sich

Jedes Unternehmen nutzt Tools, um den Zustand ihrer kritischen IT zu überwachen. Überwacht werden dabei meist viele Komponenten und deren Zusammenspiel, wie z.B. Datenbanken, Betriebssysteme, Storage Systeme, Webanbindung und Vieles mehr. Da viele der Komponenten von unterschiedlichen Herstellern stammen, gilt es ein Monitoring System zu finden, das alle Bereiche abdeckt und eine komplette Übersicht bietet.

In diesem Zusammenhang hat uns ein langjähriger Informix Kunde gefragt, ob wir die Anbindung von "Foglight" unterstützen können. Foglight vom Hersteller Quest, das Verfügbarkeiten und Performance von Datenbanken und Systemen monitort, ist bei diesem Unternehmen für viele Bereiche im Einsatz, unterstützt jedoch (derzeit) noch nicht offiziell INFORMIX.

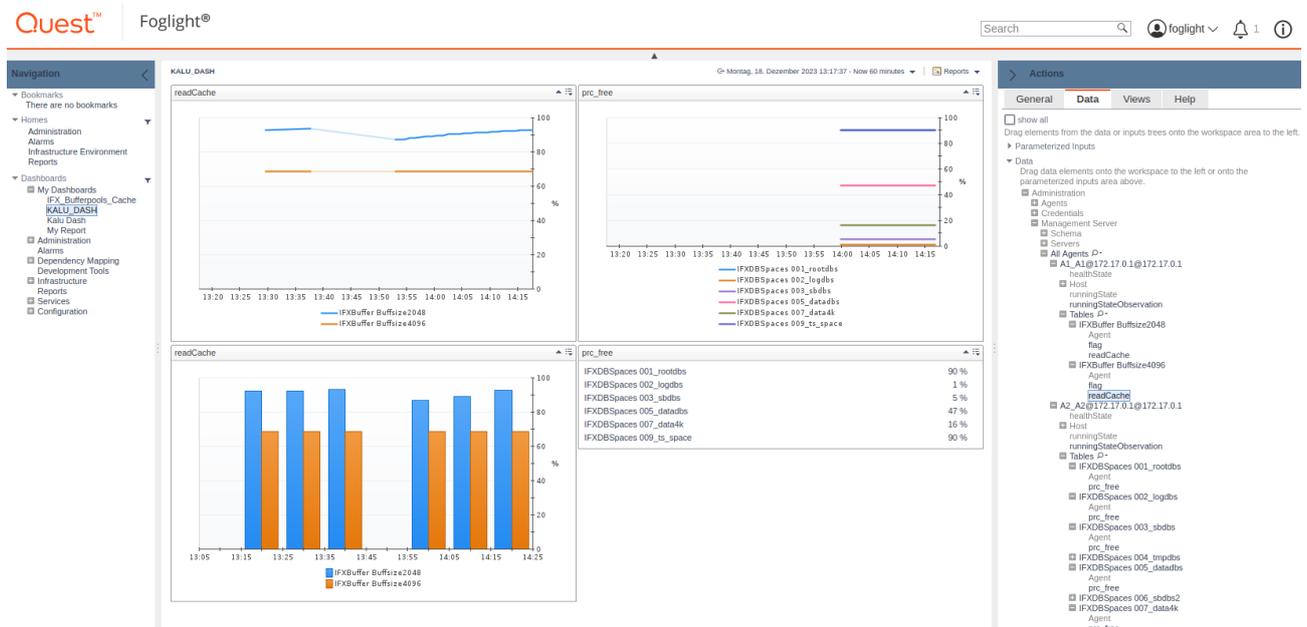
Dank der freundlichen und sehr kompetenten Zusammenarbeit mit der Firma Quest, konnten wir gemeinsam ein "Informix Cartridge" erstellen, das die wichtigen Komponenten von Informix überwacht, und in die Gesamtübersicht des Monitorings integriert wird.

Das Informix Cartridge ist für beliebig viele Informix Instanzen konzipiert und flexibel auf die eigenen Bedürfnisse anpassbar und erweiterbar.

Mit variablen Limits sind Warnungen und Alarme individuell einstellbar, ebenso wie eine Anpassung der Darstellung, um ein einheitliches Bild im Monitoring zu bieten, das denen der anderen Komponenten entspricht.

Hier einige Eindrücke vom ersten Entwurf des Informix Cartridge:

1. Die Übersicht (Auf der rechten Seite ein Menu, über das die Metriken ausgesucht und beliebig angeordnet werden können)



2. Alarmübersicht für alle Instanzen und Alarm Details:

InformixSummary Freitag, 8. März 2024 07:53:17 - Now 60 minutes Reports

| Agent Name | Informix Servername | Fatal Alarms | Critical Alarms | Warning Alarms | Mode | Host |
|---------------------------------------|---------------------|--------------|-----------------|----------------|--------|-----------|
| Informix_IFX_kalu@127.0.1.1 | kalu | | 99 | | Online | 127.0.1.1 |
| Informix_informix@127.0.1.1@127.0.1.1 | kalu | | 76 | | Online | 127.0.1.1 |
| Informix_IFX_test1@127.0.1.1 | test1 | | 34 | | n/a | 127.0.1.1 |

Alarm X

Diagnostic time range Donnerstag, 15. Februar 2024 13:02 - 17:02 4 hours

Informix_informix@127.0.1.1@127.0.1.1: Die Anzahl Pages im Transaktionslog für User 41_informix liegt bei 14192,00

Summary History/Notes Source

Alarm Summary

| Object | Details |
|------------|---|
| IFXTXPages | Triggered at: 15.02.24 16:03 Informix_informix@127.0.1.1@127.0.1.1: Die Anzahl Pages im Transaktionslog für User 41_informix liegt bei 14192,00 1: Die Anzahl Pages im Transaktionslog für User 41_informix liegt bei 14192,00. |

Recent History

No history metrics information available for this alarm.

Things Affected

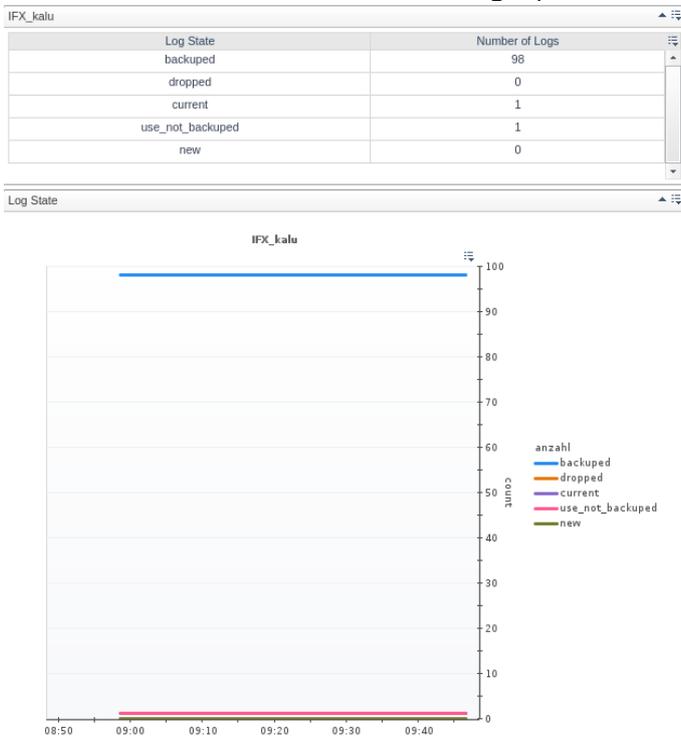
IFX_Log_User affects:

- Informix_informix@127.0.1.1@127.0.1.1

Troubleshooting

Sorry, there is no troubleshooting available for this alarm.

3. Der Zustand der Transaktionslogs (Gesichert, Genutzt, Neu, ...)



Die Darstellungen bei Foglight können individuell angepasst werden. Es stehen immer die Optionen

- Liste
- LineGraph
- BarGraph
- StackedGraph
- PieGraph
- Status

zur Verfügung.

Die Anordnung kann über Drag&Drop verändert werden. Im Alarmmenu können für jeden definierten Alarm unterschiedliche Stufen wie "Info", "Warnung", "Critical" und "Fatal" gewählt werden. Eine Alarmierung über Mail oder andere Alarmwege ist möglich.

Derzeit ist das Cartridge (noch) nicht allgemein verfügbar. Wir würden uns aber freuen, wenn INFORMIX bei Quest in die Liste der im Standard unterstützten Datenbanken aufgenommen würde. Für Fragen stehen wir (Markus Schröder / Gerd Kaluzinski) gerne zur Verfügung (siehe Autoren am Ende des Newsletters).

Anbei ein Link mit einer Übersicht zum Monitoring von Datenbanken mit Foglight:
<https://www.quest.com/de-de/products/foglight-for-cross-platform-databases/>

Nutzung des INFORMIX Newsletters

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit.

Die IUG hat sich dankenswerterweise dazu bereit erklärt, den INFORMIX Newsletter auf ihren Web Seiten zu veröffentlichen.

Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht, falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Rückmeldungen hierzu sind herzlich Willkommen !

Die gefundenen Tippfehler dürfen zudem behalten und nach Belieben weiterverwendet werden.

Eine Weiterverbreitung in eigenem Namen (mit Nennung der Quelle) oder eine Bereitstellung auf der eigenen HomePage ist ausdrücklich erlaubt. Alle hier veröffentlichten Scripts stehen uneingeschränkt zur weiteren Verwendung zur Verfügung.

Die Autoren dieser Ausgabe

Andreas Legner **INFORMIX Advanced Support**
HCL Software

Martin Fuerderer **Database Development**
HCL Software

Gerd Kaluzinski **IBM Expert Lab DACH Consultant**
Data & AI Software
gerd.kaluzinski@de.ibm.com

+49-175-228-1983

Unterstützung bei der Integration von Foglight durch:

Markus Schröder **Principal Presales Consultant**
Markus.Schroeder@quest.com

Dank auch an die vielen Helfer im Hintergrund.

Nicht zu vergessen der Dank an die Informix User Group, ohne die es den INFORMIX Newsletters heute nicht mehr geben würde, und die dankenswerter Weise die Verteilung übernimmt.

Foto Nachweis:
Redaktionsgarten

(Gerd Kaluzinski)