Tech Info.(vol.2)

Informix-Online, 버전 5.x 에서 Informix-Online Dynamic Server, 버전 7,x 로의 이전

요약

이전(migration)을 무리없이 수행하려면 구체적인 계획이 필요합니다. 이 글에서는 이전시 문제가 될 수 있는 몇 가지 부분에 대하여 설명합니다. 이전 작업에 대한 추천 사항과 자세한 정보를 얻을 수 있는 참고 서적도 함께 소개합니다.

Informix OnLine 초기 5.0x 버전에서 Informix OnLine Dynamic Server 버전 7.x 으로의 이전 작업시 문제가 될 수 있는 것은 다음과 같은 이유 때문입니다.

- Informix OnLine과 Informix OnLine Dynamic Server의 주요 차이점
- Informix OnLine, 버전 5.x의 보완 릴리즈에서의 약간의 변경사항
- OS 버전간의 변경 사항

권장사항

1. 각 시스템에 따른 플랫폼과 제품 버전에 대한 릴리즈 노트를 구해서 검토합니다.
2. 현재 사용중인 버전과 이전할 버전 사이의 버전 릴리즈 노트를 모두 구해서 검토합니다. 필요하면 응용프로그램을 조정합니다.
3. 버전 7.x로 이전하기 전에 먼저 5.x 최종 버전으로 이전합니다. 버전 5.x의 릴리즈의 약간의 변경 사항중에도 버전7.x에 적용되는 것이 많습니다. 2 단계에 걸쳐 이전하면 초기 5.x 버전과 최종 5.x 버전 보강 릴리즈 사이의 변경 사항이 적용됩니다. 버전 5.x에서 버전 7.x로 이전할 때는 OS 갱신이 필요할 수도 있으며, OS에 따라서는 Informix 제품 업그레이드와는 무관하게 이러한 OS 업그레이드 작업이 이루어질 수 있습니다. 여러 가지를 한꺼번에 변경하지 않고 이처럼 단계적으로 밟아 나가는 것이 예기치 못한 문제가 발생할 위험을 줄일 수 있는 방법입니다.
4. Informix OnLine Dynamic Server, 버전 7.x로 업그레이드할 때는 공유메모리가 8MB 추가됩니다. OnLine Dynamic Server에는 적어도 하나의 가상 공유메모리 세그먼트가 필요하고 그 기본 크기가 8MB이기 때문입니다.

확인사항

Informix OnLine, 버전 5.0x 에서 Informix OnLine Dynamic Server, 버전 7.x 로 이전하기 위해서는 Migration Guide 의 설명을 따르십시오. 다음은 사용자가 접할 수 있는 각각의 이전 단계에 초점을 맞춘 확인 사항입니다.

1. OS: 먼저 사용하고자 하는 Informix 제품과 버전이 이전시키고자 하는 시스템의 OS를 지원하는지 확인하고, 시스템에 따른 릴리즈 노트를 검토하여 Informix에서 추천한 OS 사양이 모두 설치되었는지 확인합니다. 그리고 공유메모리 커널 매개변수가 Informix OnLine Administrator's Guide와 여러분의 사용 플랫폼마다 각기 다르게 첨부되는 Informix 제품 버전에 따른 릴리즈 노트의 정보에 맞게 제대로 설정되었는지 확인합니다.
2. 일치성 검사: 데이터 무결성을 보장하려면 Migration Guide의 일치성 검사에 대한 지시를 따릅니다. 일치성 검사는 시스템 이전 전후에 모두 실시합니다. Migration Guide에서 지시하는 tbcheck 명령 집합에 'tbcheck -ce'를 추가로 실행하십시오.
3. 백업: 이전하기 전에 0 레벨 아카이브를 실행해야 합니다. 여러 OnLine (4.10부터 5.04에 이르는) 버전에서 보고된 '이전 BLOB 영역 페이지'와 '기록 파손' 문제의 영향이 미치는 버전에서 이전할 때는 레벨 0 보관을 두 번 연속 실행합니다.(Tech Alerts에서 이 문제를 다룬 항목의 제목은 아래 '참고'를 보십시오.) 이전하기 전 0 레벨 아카이브를 마지막으로 실행하기 전에 먼저 일치성 검사가 완료되었는지 확인하십시오.
4. 리소스(resource): sysmaster 데이터베이스를 생성할 때 사용할 논리 로그가 충분한지 확인합니다. 그리고, 색인을 작성할 리소스가 있는지 확인합니다.

'tmpdbspace'에 대한 요구가 있으므로 업그레이드하는 동안 사용할 'tmpdbspace' 공간이 충분한지 확인해야 합니다.

예측 할 수 있는 문제

다음은 Migration Guide 에 설명된 단계를 빠뜨리거나 따르지 않았을 때 일어날 수 있는 문제들입니다.

1. Assertion Failure: 'sysmaster' 데이터베이스가 제대로 만들어지기 전에 OnLine Dynamic Server 버전

7.x을 액세스하면 Assertion Failure가 일어납니다. 버전 7.x로 서버가 최초로 온라인 상태로 되면, sysmaster 데이터베이스가 만들어집니다. sysmaster 데이터베이스 작성이 완료되는 데는 수 분이 걸립니다. 7.x 서버를 처음 액세스하기 전에 메시지 로그를 확인하여 데이터베이스 작성이 완료되었는지 확인하십시오.

2. 성능과 데이터 액세스 문제: Migration Guide의 설명대로, 이전할 데이터베이스 모두에 UPDATE STATISTICS를 실행하지 않으면 성능과 데이터 액세스 문제가 생깁니다.
3. 응용프로그램의 Rowid 비호환: rowid를 사용하는 응용프로그램이 분할된 테이블을 액세스할 때 오류나 예기치 못한 결과가 일어날 수 있습니다. Informix OnLine Dynamic Server 버전 7.x에서 처음 가동하면 rowid 사용이 바뀝니다. 따라서 이전에 사용한 rowid는 일반적으로 7.x의 분할 기능과 호환되지 않습니다. 버전 5.x에서 rowid를 사용하는 응용프로그램은, 7.x에서 테이블 분할을 사용할 경우 rowid 칼럼을 버전 7.x 분할 테이블에 사용자가 명시적으로 지정하지 않으면 호환성 문제에 부딪칩니다. 데이터 테이블을 분할하고 rowid를 사용하는 응용프로그램으로 그 테이블을 액세스하려면 테이블마다 SQL 구문을 사용하여 rowid 칼럼을 추가하십시오.

참고

Informix OnLine Dynamic Server Migration Guide, 버전 7.1

Part number: 000-7625

Informix OnLine Dynamic Server Migration Guide, 버전 7.2

Part number: 000-7887A

UNIX Products Installation Guide, 버전 7.1

Part number: 000-7638B

UNIX Products Installation Guide, 버전 7.2

Part number: 000-7896A

Informix OnLine Dynamic Server Administrator's Guide Vol. 1 와 2, 버전 7.1

Part numbers: 000-7778, 000-7777

Informix OnLine Dynamic Server Administrator's Guide Vol. 1 와 2, 버전 7.2

Part numbers: 000-7885A, 000-7886A

AFDEBUG 설정방법

다음은 assert failure 오류의 원인을 디버깅할 때의 단계입니다. 스택 추적 출력이 오류 진단에 도움이 됩니다.

1. ONCONFIG 매개변수 설정 - DUMPCORE 1 (코어 파일을 만듭니다.)

   - DUMPSHMEM 1(assert failure 오류시의 공유 메모리의 내용을 파일명 shmem.pid.cnt로 덤프 받습니다.)

   - DUMPDIR /tmp (특정 디렉토리로 출력을 보냅니다.)

   - DUMPCNT n (디렉토리에서 허용되는 덤프 수를 지정합니다.)
2. assert failure 오류가 일어날 때 af.xxx 파일이 하나 만들어집니다. 이 파일은 /tmp나 DUMPDIR이 가리키는 곳에 있습니다. af.xxx에는 assert failure 오류 메시지가 들어있습니다. 이 파일에는 유용한 정보가 있으므로 오류 진단을 위해 보존해야 합니다. 플랫폼에 따라 7.12 이후 버전에서는 af.xxx 파일에 스택 추적이 포함됩니다
3. 세션 id가 온라인 로그 shmem.pid.cnt의 assert failure 오류 메시지에 표시되는 경우에는 공유메모리 덤프가 도움이 될 수 있습니다.
4. 이런 문제가 다시 발생하게 되면 다음의 작업을 수행한다면 좋은 기술 지원 정보가 될 것입니다.
   A. $) setenv AFDEBUG 1 (엔진- oninit를 시작하기 전에 실행)

      ** 이 환경 변수는 assert failure 오류가 발생했을 때 OnLine 을 즉시 정지 모드로 변경하지 않고 정체 상태(hang)가 되게 합니다.

   B. assert failure 오류가 발생하면, 온라인 로그를 확인하여 어떤 프로세스가 정체되고 있는지 확인하십시오. assert failure시, onstat -m을 사용하여 메시지 로그를 확인할 수 있습니다.

   C. 일단 온라인 엔진이 정체 상태에 빠지게 되면, 해당 인스턴스에 대해 onstat 명령을 실행할 수 있습니다. 이 방법을 사용하면 장애가 발생한 그 시간에 인스턴스의 공유메모리를 검토할 수 있습니다. onstat의 다음 옵션들을 확인하시는 것이 중요합니다.

onstat -g sts

onstat -g glo

onstat -g stk <thread_id>

onstat -g ses <session_id>

onstat -g seg

onstat -g mem

onstat -a

주의: 온라인 로그 오류 메시지에서 볼 수 있었던 세션 번호가 onstat -g ses 로 확인했을 때 결과로 나오지 않을 수 있습니다. 하지만 thread id 는 나옵니다. session id 를 구하려면 onstat -g ath 를 실행하고 thread id 로 관련된 rstcb id 를 구합니다. 그런 다음 onstat -u 를 실행하여 rstcb id 와 user-thread id 를 같이 참조할 수 있습니다. 다음은 session id 를 추적하는 예입니다. thread id (tid) 19 에 대한 rstcb 값은 a296878 입니다. 이 값을 사용하여 onstat -u 출력에서는 user-thread 칼럼에서 a296878 을 확인할 수 있을 것입니다. *sessid* 칼럼에서 이를 따르면 이 스레드의 session id 가 8 임을 알 수 있습니다. 여기서 얻은 결과는 다음 명령을 사용할 수 있다는 것을 뜻합니다.

onstat -g ses 8

예: /home/tech/user % onstat -g ath

Informix OnLine Version 7.13.UC1 -- On-Line -- Up 01:35:50 - 10488 Kbytes

Threads:

tid	tcb	rstcb	prty	status	vp-class	name
2	a372a98	0	2	sleeping(Forever)	3lio	lio vp 0
3	a372d08	0	2	sleeping(Forever)	4pio	pio vp 0
4	a372fa0	0	2	sleeping(Forever)	5aio	aio vp 0
5	a373238	0	2	sleeping(Forever)	6msc	msc vp 0
6	a373aa0	0	2	sleeping(Forever)	7aio	aio vp 1
7	a373cc8	a296010	4	sleeping(secs: 1)	1cpu	main_loop()

8 a386a50 0 2 running 1cpu sm_poll

9 a39a7b0 0 2 running 8tli tlitcppoll

10 a39acf0 0 2 sleeping(Forever) 1cpu sm_listen

11 a3f8a70 0 2 sleeping(secs: 2) 1cpu sm_discon

12 a3f8eb0 0 3 sleeping(Forever) 1cpu tlitcplst

13 a410788 a296444 2 sleeping(Forever) 1cpu flush_sub(0)

14 a4109f0 0 4 sleeping(Forever) 1cpu kaio

19 a411c20 a296878 2 sleeping(secs: 16) 1cpu btclean

예: /home/tech/user % onstat -u

Informix OnLine Version 7.13.UC1 -- On-Line -- Up 01:40:22 - 10488 Kbytes

User threads

address flags sessid user tty wait tout locks nreads nwrites

a296010 ---P-D 0 Informix - 0 0 0 11 2

a296444 ---P-F 0 Informix - 0 0 0 0 0

a296878 ---P-B 8 Informix - 0 0 0 0 0

a296cac ---P-D 0 Informix - 0 0 0 0 0

4 active, 128 total, 17 maximum concurrent

D. 프로세스 id에 디버거를 사용하려면 다음 명령을 사용하여 수동으로 테이블을 첨부할 수 있습니다.

예 : sdb $InformixDIR/bin/oninit PID

*** 프로세스에 WHERE TRACE 를 실행할 수 있습니다. ***

onstat -g glo 를 실행하여 oninit 프로세스의 PID 를 구할 수 있습니다.

E. 다음은 명령 순서의 예입니다.

F. example% setenv AFDEBUG 1

G. example% oninit

H. <처리중입니다>

I. example% onstat -m

J. 07:20:18 checkpoint completed: duration 13 seconds

K. 07:22:20 bsession.c, line 861, thread 2854, procid 4975,

L. (destroy_session) mem pools not freed.

M. 07:22:22 invoke_debugger:execlp error 2

N. 07:22:22 execlp(,,,4975,0);

O. 07:22:22 oninit pid 4975 can be attached to manually

P.

Q. example% onstat -g glo (cpu VP 의 PID 구함)

R. example% sdb $InformixDIR/bin/oninit PID

디버거 사용: (sdb, xdb, dbx, gdb)

*** WHERE TRACE 실행***

S. onconfig 매개변수 설정에 문제가 있으면 명령줄에서 설정해도 됩니다. 매개변수를 명령줄에서 설정하면 onconfig에 설정된 값을 대신하게 됩니다.

T. 모든 작업이 끝나면 엔진을 중지시키십시오. 먼저 onmode -ky를 시도해 보고, 엔진이 해제되지 않으면 관련 공유 메모리 세그먼트를 삭제하고 onstat -g glo 명령으로 찾은 oninit 프로세스의 PID를 이용하여 kill -9 PID 명령으로 그 프로세스를 삭제합니다.

U. 이제 "oninit"를 실행하여 엔진을 다시 온라인 모드로 가동하십시오.

데이터베이스의 사용자와 그 권한을 리스트하는 보고서

Informix 데이터베이스 권한으로 작업할 보안 제품을 테스트하던 중 다음 ace 보고서를 개발하였는데 다른 사람들에게도 도움이 될 것 같습니다. 가능한 상황은 모두 다루었음을 알려드립니다.(현재 칼럼 레벨의 권한이나 참조 권한은 출력되지 않습니다.)

데이터베이스에 접근할 수 있는 사용자 모두와 각 테이블에 대한 그들 각각의 권한 정보가 필요할 경우가 있습니다.

테이블명에 대한 info 권한은 공용 권한이거나 테이블 소유권이므로 테이블 접근 권한을 가지고 있는 사람이 누구인지 보여주는 것은 아닙니다. 모든 사용자와 각 테이블에 대한 그들의 권한을 보여주는 ace 보고서를 작성해 보았습니다. 테이블이나 사용자가 많으면 물론 보고서가 길어질 수 있으며, Informix 4.0와 5.0, SE와 Online을 기준으로 하였습니다.

```
{################################################################}
:
: Module: @(#)privileg.ace
: Author:  Lester B. Knutsen
:
: 설명:             데이터베이스의 모든 사용자와 각 테이블에 대한 그들의 권한을
:       보여주는 Informix Ace 보고서입니다. 프로그램에서 먼저
:       sysusers, systables, systabauth 에서 모든 사용자, 소유자, 권한
:       부여자, 권한을 부여받은 사용자를 검색하여 임시 테이블을 만듭니다.
:       그런 다음 보고서에서 데이터베이스의 각 테이블에 대한 공용 권한에 관
:       한 임시 테이블을 만듭니다. 마지막으로 모든 사용자와 모든 테이블을 조
:       인하여 권한을 보여줍니다. 각 테이블에 대한 권한 Select, Update,
:       Columns, Insert, Delete, Index 와 Alter 등입니다. 각 권한에 대
:       해 다음 사항으로 표시합니다.
:
:       Y - 사용자가 이 권한을 부여받았습니다.
:       N - 사용자가 이 권한을 부여받지 않았습니다.
:       O - 사용자가 테이블 소유자이고 이 권한이 있습니다.
:       P - 공용이므로 사용자가 이 권한이 있습니다.
:
:       사용자에게 다른 사람이 부여한 또 다른 권한이 있다면 두 권한을 모두
:       표시합니다.
:
:       이 프로그램은 다음 명령으로 컴파일하십시오.
:               saceprep privilege
```

```
:
:           보고서의 데이터베이스 섹션은 Informix 의 데모 데이터베이스인
:           stores 데이터베이스를 명시합니다. 만일 변경하고자 한다면 변경 후
:           재컴파일하십시오.
:
:           이 보고서는 다음의 명령으로 실행할 수 있습니다.
:                       sacego -d database privilege
:
:                       위에서 database 는 데이터베이스 이름입니다.
:
}
database stores end
define
            variable last_tabid integer
            variable last_grantor           char(8)
            variable last_tabauth           char(8)
end
output
            left margin 0
            right margin 80
            TOP margin 3
            bottom margin 3
            page length 66
            report to "privilege.out"
{           report to pipe "more"}
end
{#############################################################}
{ 테이블 권한이 있는 모든 사용자, 테이블 소유자 등을 선택합니다. }
select      unique username
```

```
from     sysusers
union
select     unique owner
from     systables
where     systables.tabid > 99   { 시스템 테이블은 건너뜁니다.}
union
select     unique grantor
from     systabauth
union
select     unique grantee
from     systabauth
into temp users1;
{ 위 임시 테이블에서 유일한 사용자 목록을 선택합니다. }
select     unique users1.username,
            sysusers.usertype
from      users1, outer sysusers
where     users1.username = sysusers.username
and       users1.username != "Informix"
into temp users;
{#################################################################}
{ 공용 권한을 검색합니다.}
select     systabauth.tabid,
            systabauth.tabauth     pub_tabauth,
            systabauth.grantor     pub_grantor,
            sysusers.usertype      pub_type
from      systabauth, outer sysusers
where     systabauth.grantee = "public"
and       systabauth.grantee = username
into temp pub;
```

```
{#################################################################}
{ 사용자를 데이터베이스의 모든 테이블, 테이블 권한, 공용 권한을 나타낸
  users, pub 테이블과 조인합니다.}
select    users.username,
          users.usertype,
          systables.tabname,
          systables.owner,
          systables.tabid,
          systabauth.grantor,
          systabauth.grantee,
          systabauth.tabauth,
          pub.pub_tabauth,
          pub.pub_grantor,
          pub.pub_type
from      users, systables, outer systabauth, outer pub
where     systables.tabid = systabauth.tabid
and       users.username = systabauth.grantee
and       systables.tabid = pub.tabid
and       systables.tabid > 99{ 역시 시스템 카타로그 테이블은 제외합니다 }
and       tabname not in
          ("sysmenus", "sysmenuitems", "syscolatt", "syscolval",
            "dtgroup", "dttabauth", "dtuser" )
order by username, tabname, grantor
end
{#################################################################}
format
page header
print     column 1, "Date: ", today using "MM/DD/YY",
          column 26, "DataTools - User Privileges",
```

```
            column 70, "Page: ", pageno using "####"
print        column 1, "---------------------------------------",
                       "---------------------------------------"
skip 1 line
{#############################################################}
page trailer
skip 1 line
print        column 1, "---------------------------------------",
                               "---------------------------------------"
print
"Y-부여된 권한    N-권한 없음   O-테이블 소유자      P-공용 권한"
{#############################################################}
before group of username
need 7 lines
let last_tabid = 0
let last_grantor = "               "
let last_tabauth = "               "
print        column 1, "User login: ", username,
             column 22, "DB Access : ";
if ( usertype = "C" ) then print "Connect"
else if ( usertype = "R" ) then print "Resource"
else if ( usertype = "D" ) then print "DBA"
else if ( pub_type = "C" ) then print "Connect (Public)"
else if ( pub_type = "R" ) then print "Resource (Public) "
else if ( pub_type = "D" ) then print "DBA (Public) "
else print "None"
skip 1 line
print        column 8, "Table",
             column 22, "Grantor",
```

```
            column 32, "Select",
            column 39, "Update",
            column 46, "Column",
            column 53, "Insert",
            column 60, "Delete",
            column 67, "Index",
            column 74, "Alter"
print       column 1,  "------------------",
            column 22, "--------",
            column 32, "------",
            column 39, "------",
            column 46, "------",
            column 53, "------",
            column 60, "------",
            column 67, "------",
            column 74, "------"
{#################################################################}
after group of username
print       column 1, "--------------------------------------",
                      "----------------------------------------"
skip 1 line
{#################################################################}
on every row
{           중복 행은, 여러 권한 부여자가 공용 권한을 부여한 경우 pub 테이블과
다른 여러 테이블을 조인했을 때 만들어집니다. 따라서 중복 행을 필터로
골라내야 합니다.}
if ( last_tabid = tabid and last_grantor = grantor and last_tabauth = tabauth )
            then
            begin
```

```
                { 중복 행-아무 작업도 하지 않습니다. }
                let last_tabid = tabid
                let last_grantor = grantor
                let last_tabauth = tabauth
        end
else if ( last_tabid = tabid and username = owner )
        then
        begin
                { 중복 행-아무 작업도 하지 않습니다. }
                let last_tabid = tabid
                let last_grantor = grantor
                let last_tabauth = tabauth
        end
else
        {#########################################################}
begin
        let last_tabid = tabid
        let last_grantor = grantor
        let last_tabauth = tabauth
        print   column 1,   tabname;
        {매 행마다 가능한 권한은 4 가지입니다.
        1. 테이블 소유자이며 테이블에 대해 모든 권한이 있습니다.
        2. 이 사용자는 여러 사용자로부터 권한을 부여 받았습니다.
        3. 권한을 부여받지 않았거나 권한이 취소되었지만 공용으로 권한을
                부여받았습니다. 모든 사용자에게 공용 권한이 있습니다.
        4. 권한이 없고 공용 권한도 없습니다.}
        {#########################################################}
        {사용자가 테이블 소유자인지 확인합니다.}
        if ( username = owner ) then
```

```
begin
        print       column 22, username,
                    column 34, "0",
                    column 41, "0",
                    column 48, "0",
                    column 55, "0",
                    column 62, "0",
                    column 69, "0",
                    column 76, "0"
end
{#######################################################}
{ 사용자에게 권한이 있는지(systabauth.tabauth is not null)
        확인하고 사용자 권한을 리스트하거나
  공용 권한이 없으면(systabauth.tabauth is null
        where grantee = public)
        "N"을 모두 인쇄합니다.
}
else if ( tabauth is not null or pub_tabauth is
null )then begin
        print       column 22, grantor;
        if (( tabauth[1] = "s" ) or ( tabauth[1]   = "S"   ))
        then
                    print column 34, "Y"
        else        print column 34, "N";
        if (( tabauth[2] = "u" ) or ( tabauth[2]   = "U"   ))
        then
                    print column 41, "Y";
        else        print column 41, "N";
        if (( tabauth[3] = "*" ) or ( tabauth[3]   = "*"   ))
```

```
        then
                print column 48, "Y";
                else        print column 48,   "N";
        if (( tabauth[4] = "i" ) or ( tabauth[4]   = "I"   ))
        then
                print column 55, "Y";
        else        print column 55, "N";
        if (( tabauth[5] = "d" ) or ( tabauth[5]   = "D"   ))
        then
                print column 62, "Y";
        else        print column 62, "N";
        if (( tabauth[6] = "x" ) or ( tabauth[6]   = "X"   ))
        then
                print column 69, "Y";
        else        print column 69, "N";
        if (( tabauth[7] = "a" ) or ( tabauth[7]   = "A"   ))
        then
                print column 76, "Y"
        else        print column 76, "N"
end
else
{#######################################################}
{사용자는 권한이 없고 공용 권한이 있으면 공용 권한을 리스트합니다.}
begin
        print       column 22, pub_grantor;
        if (( pub_tabauth[1] = "s" ) or ( pub_tabauth[1]
        = "S"   )) then
                print column 34, "P";
        else        print column 34, "N";
```

```
                if (( pub_tabauth[2] = "u" ) or ( pub_tabauth[2]
                = "U"   )) then
                        print column 41, "P";
                else    print column 41, "N";
                if (( pub_tabauth[3] = "*" ) or ( pub_tabauth[3]
                = "*"   )) then
                        print column 48, "P";
                else    print column 48, "N";
                if (( pub_tabauth[4] = "i" ) or ( pub_tabauth[4]
                = "I"   )) then
                        print column 55, "P";
                else    print column 55, "N";
                if (( pub_tabauth[5] = "d" ) or ( pub_tabauth[5]
                = "D"   )) then
                        print column 62, "P";
                else    print column 62, "N";
                if (( pub_tabauth[6] = "x" ) or ( pub_tabauth[6]
                = "X"   )) then
                        print column 69, "P";
                else    print column 69, "N";
                if (( pub_tabauth[7] = "a" ) or ( pub_tabauth[7]
                = "A"   )) then
                        print column 76, "P"
                else    print column 76, "N"
        end
end
end
{############################################################}
```

다음은 위 보고서를 OnLine 5.1 버전에서 stores5 데이터베이스를 가지고

실행한 결과입니다.

Date: 03/25/97 DataTools – User Privileges Page: 1

User login: DB Access : Resource (Public)

Table	Grantor	Select	Update	Column	Insert	Delete	Index	Alter	
call_type	Informix	P	P	N	P	P	P	N	
catalog	Informix	P	P	N	P	P	P	N	
cust_calls	Informix	P	P	N	P	P	P	N	
customer	Informix	P	P	N	P	P	P	N	
custview	Informix	P	P	N	P	P	N	N	
items	Informix	P	P	N	P	P	P	N	
manufact	Informix	P	P	N	P	P	P	N	
orders	Informix	P	P	N	P	P	P	N	
someorders			N	N	N	N	N	N	N
state	Informix	P	P	N	P	P	P	N	
stock	Informix	P	P	N	P	P	P	N	

User login: public DB Access : Resource

Table	Grantor	Select	Update	Column	Insert	Delete	Index	Alter	
call_type	Informix	Y	Y	N	Y	Y	Y	N	
catalog	Informix	Y	Y	N	Y	Y	Y	N	
cust_calls	Informix	Y	Y	N	Y	Y	Y	N	
customer	Informix	Y	Y	N	Y	Y	Y	N	
custview	Informix	Y	Y	N	Y	Y	N	N	

items	Informix	Y	Y	N	Y	Y	Y	N	
manufact	Informix	Y	Y	N	Y	Y	Y	N	
orders	Informix	Y	Y	N	Y	Y	Y	N	
someorders			N	N	N	N	N	N	N
state	Informix	Y	Y	N	Y	Y	Y	N	
stock	Informix	Y	Y	N	Y	Y	Y	N	

Y-권한 부여됨 N-권한 없음 O-테이블 소유자 P-공용 권한

Illustra와 Universal Server 의 차이점

제품이름

새 제품 이름은 Informix Universal Server 입니다. Universal Data Server 라고는 부르지 않으므로 "UDS"로 사용하지 마십시오.

"DataBlade" 라는 용어는 등록 상표이며 형용사로만 쓰이므로 "DataBlade" 대신 "DataBlade Module"로 사용하십시오.

아키텍처

Informix Universal Server 는 Dynamic Scalable Architecture(DSA)에 바탕을 두고 있습니다. Extension Virtual Processor (EVP)라는 가상 프로세서가 Universal Server 에 추가되는데, CPU VP 와 달리, EVP 는 직접 파일 시스템에 접근하므로 개발자가 명시적으로 프로세서를 만들지 않아도 됩니다. CPU VP 에서 실행되는 사용자 정의 함수는 스레드-보호적(thread-safe)이어야 하며 운영 체제 호출이나 표준 POSIX 라이브러리 루틴과 같은, 비보호적인(unsafe) 호출을 해서는 안됩니다. 사용자 정의 함수에 대해서는 LIBMI 를 사용해야 하며, 빈 메모리를 할당하기 위해서는 mi_alloc()와 mi_free()를 사용해야 합니다. 그리고 전역

변수나 정적 변수를 사용하면 안됩니다.

Universal Server 에서 지원되는 ILLustra 기능

Universal Server에는 Online Dynamic Server에서는 볼 수 없던 완전히 새로운 기능이 몇 가지 있습니다. 이 새로운 기능은 다음과 같습니다.

- 사용자 정의 자료형.

  이제 고정된 내장 자료형 지원에만 한정되지 않습니다. CHAR, VARCHAR, DATE, DATETIME, MONEY 외의 자료형을 사용할 수 있습니다.

- C로 작성된 사용자 정의 함수.

  이 함수는 CREATE FUNCTION문으로 데이터베이스에 추가되고, 컴파일된 C 이진 코드가 공유 메모리에 로드되어 데이터베이스 일부가 됩니다.

물론, 내장 프로시저도 계속 지원되고, 사용자 정의 함수 역시 다음과 같은 단순 SQL로 호출할 수 있습니다.

```
SELECT myFunction(column1), column2 FROM myTable WHERE myFunction2('input',column2)
```

- 사용자 정의 함수의 정보를 관리함으로써 옵티마이저에게 최적의 경로를 가질 수 있도록 "Hint" 제공
- 사용자 정의 색인을 비롯한 사용자 정의 접근 방법.

  Illustra는 현재 전형적인 btree 외에도 rtree (spatial 데이터 검색), dtree (전체 텍스트 검색) 등 몇 가지 색인형을 지원합니다.

- SELECT문에서 구동하는 트리거

UNIVERSAL SERVER에서 지원되지 않는 ILLUSTRA 기능

다음의 Illustra 기능은 Informix Universal Server에서 지원되지 않습니다.

- 18자를 넘어서는 테이블, 함수, 칼럼 이름

- 대소문자를 구별하는 이름.

  Universal Server에서는 대소문자를 구별하지 않습니다.

- 룰(Rule).

  Universal Server에는 트리거가 있습니다. Universal Server는 SELECT 문에서 구동하는 트리거를 지원합니다. 그러나 Universal Server는 다중 SELECT 리거(또는 다중 INSERT, UPDATE, DELETE 트리거)를 지원하지 않습니다. DO INSTEAD(예를 들어, 트리거를 실행하다가 권한이 없어서 실행을 하지 못하게 되면, 그 대신에 다른 행동을 취할 수 있도록 하는 기능)를 지원하지 않습니다.

- Alerter (특정 이벤트가 발생했을 얼러터(alerter)때 그것을 확인할 필요성이 있는 클라이언트에게 해당 이벤트가 발생했음을 알려주는 기능) 은 지원되지 않습니다.

- Time travel과 Versioning 기능 (갱신되거나 삭제된 이전 데이터를 그대로 테이블에 가지고 있게 함으로써, 필요시 시간 조건을 주어서 변경되기 이전의 데이터도 검색할 수 있는 기능)은 지원되지 않습니다.

- 엘란 라이센스 관리자(Elan license manager)

- SQL 사용자 정의 함수를 작성하는 Illustra 구문.

  대신 Informix 내장 프로시저 언어(SPL)를 사용하십시오. SPL에는 오류 처리 루프와 조건문이 포함되어 있으므로 SQL보다 강력합니다.

- 집합(setof)을 반환하는 SQL 사용자 정의 함수는 SPL에서 WITH RESUME문을 사용하여 코드를 다시 작성할 수 있습니다.

- 다중 상속과 재그 행(multi inheritance and jagged row).

  이후 버전에서는 지원될 것입니다.

- 배열

  Universal Server에서는 LIST라는 새 자료형이 배열 형식의 데이터를 지원하지만 구문이 다릅니다.

- 다음 자료형은 지원되지 않습니다:

  ABSTIME, CID, EXTERNAL_FILE, OID, RELTIME, TIME, TIME_TZ, TIMESTAMP, TIMESTAMP_TZ.

UNIVERSAL SERVER에서 지원되지 않는 Informix ONLINE 기능

Universal Server에서는 다음 내용을 포함한 모든 Informix Online Dynamic Server 기능을 지원할 것입니다.

- 동적 확장 구조, 다중 스레드, 병렬 데이터베이스 엔진
- 병렬 데이터 질의(PDQ)
- 원시 자료 저장 영역(Raw data storage)
- 분산 데이터베이스와 2단계 완료(2-phase commit)
- 테이블 분할화(Fragmentation of Tables)