

External Tables – What & Why

Mike Walker

Webcast on November 15, 2018

Advanced DataTools

Mike Walker



IBM CHAMPION 

Mike Walker has been using Informix databases for over 20 years, as a developer and as a database administrator. Mike is an IBM Champion.

Mike runs the Remote DBA Support for Advanced DataTools Corporation.

mike@advanceddatatools.com

www.advanceddatatools.com

Office: 703-256-0267

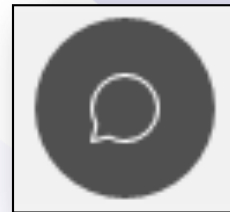
Advanced DataTools

Webcast Guidelines

The Webcast is being recorded. The Webcast replay and slides will be available in a few days.

Please Mute your Line. Background sounds will distract everyone.

Use the Chat Button – in the toolbar at the bottom of the screen



Informix External Tables

Interface between SQL and filesystem files
Simple way to read/write data to/from files

Fast & Easy to Use

Use for data unloads and loads
Use for table reorgs and migrations
Use for ETL

Read a Flat File using SQL

File containing comma delimited data

datafile.txt

```
Mike,101,CO,  
Tom,102,VA,  
Lester,100,VA,  
Art,201,NJ,  
Jack,202,MA,
```

**Want to read
this data file
using Informix
SQL**

Read a Flat File using SQL

Define an external table, referencing the file:

```
create external table employees (  
    name char(20),  
    idnum smallint,  
    state char(2)  
)  
using  
    (datafiles (  
        "DISK:/home/informix/external_table  
s/datafile.txt"),  
        format 'delimited',  
        delimiter ',',  
    )  
);
```



Use the full path –
will not be the
current directory!

Read a Flat File using SQL

Can now read from the *file* using simple SQL:

```
select * from employees;
```

name	idnum	state
Mike	101	CO
Tom	102	VA
Lester	100	VA
Art	201	NJ
Jack	202	MA

Read a Flat File using SQL

Use the External Table in SQL Select statements as a regular table, including joins to other tables:

```
select e.name, s.sname
from employees e,  External Table
    state s
where e.state = s.code
    and s.sname != "Colorado"
order by 1;
```

Query External Tables - Limitations

Cannot use an external table:

- In a subquery
- In an Outer Join – as the OUTER table

Further limitations with BLOB/CLOB columns

26214: Cannot perform this operation on an external table

Read a File (fixed format) using SQL

Use external tables to read **fixed format** files

datafile.txt4

Mike	101CO
Tom	102VA
Lester	100VA
Art	201NJ
Jack	202MA

Read a File (fixed format) using SQL

The **external** definition describes the **format of the field in the file**. The source file format does **not** need to match the table

```
create external table employees (  
    name char(20) external char(10),  
    idnum smallint external char(3),  
    state char(2) external char(2),  
    dummy char(1) external char(1)  
)  
using  
(datafiles  
("DISK:/home/informix/external_tables/datafile.txt4"),  
format 'fixed'  
);
```

Need a placeholder for the newline

Read a Flat File using SQL

The ability to *read from text files* makes external tables very useful for loading data warehouses and ETL

```
truncate table empnames;  
  
insert into empnames  
select name  
from employees_ext;
```


Limitations

Cannot change the data in the existing file via an external table

No DELETE, UPDATE, INSERT INTO ... VALUES

Cannot perform the following operations on external tables:

TRUNCATE, UPDATE STATISTICS, START VIOLATION, STOP VIOLATION, ALTER TABLE, CREATE TRIGGER, CREATE INDEX, CONNECT BY, LBAC operations

Error 26191

Limitations

External Tables are not logged – not replicated

No Commit Interval

- Beware when loading large tables from an external table to avoid long transactions or excessive locks – consider using RAW

Write to a Flat File

While you *cannot* modify the existing data via an external table, you *can* write to a new file with:

INSERT INTO <external table> ...SELECT...

File is recreated – not appended to

Write to a Flat File

```
create external table tables_ext (  
    tabid int,  
    tabname char(30)  
)  
using  
(datafiles  
("DISK:/home/informix/external_tables/tablist.txt")  
);
```

```
insert into tables_ext  
select tabid, tabname  
from systables  
where tabid < 10;
```

Write to a Flat File

tablist.txt

```
1| systables |  
2| syscolumns |  
3| sysindices |  
4| systabauth |  
5| syscolauth |  
6| sysviews |  
7| sysusers |  
8| sysdepend |  
9| syssynonyms |
```

**Pipe delimited is
the default**

**DBDELIMITER
does not apply**

Write to a Flat File

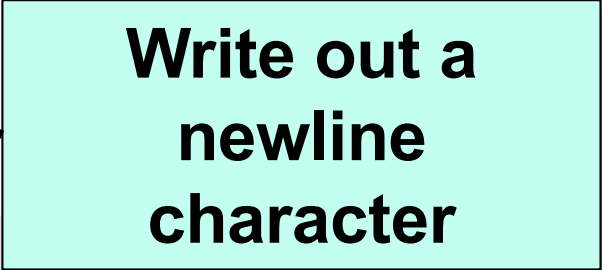
Writing **fixed format** files...

Output a newline character in the select statement if you want each record on its own line

```
execute procedure ifx_allow_newline('t');
```

```
insert into tables_ext  
select tabid, tabname, "
```

```
from systables  
where tabid < 10;
```



**Write out a
newline
character**

Write to a Flat File

Can use a complex SELECT statement to INSERT into the External Table

- Pull data from multiple tables
- Use CASE statements
- Populate derived columns
- Format the output...and subsequently the file format

Write to a Flat File

The ability to *write to text files* makes external tables very useful for extracting data to feed other systems and applications

Read from/Write to a File

DEMO

Use of External Tables

A good use of external tables is to use ***both*** the **write** (*unload*) and **read** (*load*) abilities to:

- Reorg tables
- Move a table to a dbspace with different page size
- Speed up table ALTERs that can't be done in-place
- Make copies of tables
- Copy data between databases
- Migrate large tables

External Tables for Unload/Load

Unload from Source

1 - Create external table matching the source *table*

```
create external table mytable_ext (  
  id integer,  
  name char(20))  
using  
  (datafiles ("DISK:/tmp/mytable.unl")) ;
```

2 - Insert into external table – select rows from source table

```
insert into mytable_ext  
select * from mytable;
```

External Tables for Unload/Load

Load into Target

1 - Create external table matching the source *file*

```
create external table mytable_ext (  
  id integer,  
  name char(20))  
using  
  (datafiles ("DISK:/tmp/mytable.unl")) ;
```

2 - Insert into table – select rows from external table

```
truncate table mytable_copy;  
  
insert into mytable_copy  
select * from mytable_ext;
```

Syntax

Column Definition

```

|--+--SAMEAS--template-----+-----+
| .-,-----|.
| v          ||
'---column--data_type--| Other Optional Clauses |--'

```

Other Optional Clauses

```
|-----+-----|
'-EXTERNAL--CHAR--(--size--)-----+-'
                                     +-NULL--'--null_string--'+
                                     '-NOT NULL-----'
```

DATAFILES Clause

```
|--DATAFILES----->
.
.
.
V
|
>--(---'--+DISK+---:--+fixed_path-----+---'
+--)--|
      '-PIPE-'      '-formatted_path-' |      .-;------. |
      |              |              |      V              | |
      |              |              |      '-;---+BLOBDIR+---:--+fixed_path+--'
      |              |              |              '-CLOBDIR-'
```

Syntax

Table Options

```

. - , ----- .
V          (1)  .-DELIMITED-.      |
|-----+--FORMAT-----+--INFORMIX--+--|-----+
|          '-FIXED-----'          |
+--DEFAULT-----+
|          (1) (2) |
+-| Loading mode options |-----+
'-+-DBDATE--'date_format'-----+-'
  +--DBMONEY--'currency'-----+
  +--DELIMITER--'field_delimiter'--+
  +--RECORDEND--'record_delimiter'--+
  +--MAXERRORS--num_errors-----+
  +--REJECTFILE--'filename'-----+
  |   (1)          .-ON-----.      |
  +-----+--ESCAPE--+--+-----+
  |          '-OFF-'          |
  '-+-NUMROWS+--num_rows-----+'
      '-SIZE-----'

```

Loading mode options

```

|-----+--EXPRESS+-----|
      '-DELUXE--'

```

SAMEAS Clause

Define an external table simply by using an existing table as a template

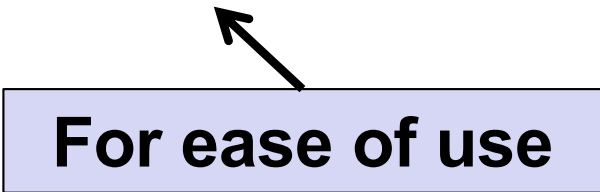
```
create external table <ext tab name>  
sameas <tablename>  
using (  
    datafiles ("DISK:<filename>")  
);
```

Creates the external table with the same columns as the existing table <tablename>

Multiple Files

Spread the IO over multiple disks by specifying multiple files

```
create external table if not exists  
<ext tablename>  
sameas <tablename>  
using (  
    datafiles (  
        "DISK:<filename1>",  
        "DISK:<filename2>",  
        "DISK:<filename3>")  
    ) ;
```



For ease of use

Multiple Files

Name multiple files automatically with
`%r(<start>..end>)`

...

DATAFILES

`("DISK:/tmp/items_%r(1..3).unl")`

```
-rw-rw-rw- 1 informix informix 115715 Oct  3 09:35 items_1.unl
-rw-rw-rw- 1 informix informix 139023 Oct  3 09:35 items_2.unl
-rw-rw-rw- 1 informix informix  69506 Oct  3 09:35 items_3.unl
```

Use Internal Format

When working on the same Informix version, O/S and hardware, use the internal “INFORMIX” format for ***much*** faster unloads/loads

```
create external table <ext tablename>
sameas <tablename>
using (
    datafiles ("DISK:<filename>"),
    format "informix"
);
```

Use Internal Format

FORMAT "INFORMIX"

- Can be used to migrate tables to dbspaces of different page sizes
- Can not effectively be used when changing data types – stick to FORMAT “DELIMITED”
- INFORMIX formatted files are compatible with HPL’s “No-Conversion” files (in limited testing!)

Deluxe vs Express Mode

The mode is only applies to **loading** through external tables

Express (faster):

Can only be used on non logged tables
Target table must have no indexes

**Express will be
used when
possible**

```
create external table <ext tablename>  
sameas <tablename>  
using (  
    datafiles ("DISK:<filename>"),  
    express) ;
```

Might be ignored

Deluxe (slower):

Evaluates indexes and unique constraints during load
Data can be accessed during the load
Cannot specify a commit interval – long tx possible & many locks!

Deluxe vs Express Mode

If you define an external table as **EXPRESS**, but the conditions are not met (e.g. the table being loaded is a *logged* table), then the load will run as **DELUXE**

No error will be shown by the command, but the Informix log will show a message similar to:

```
17:25:52   Switching load on target  
table informix.stock to DELUXE
```

Reject File

Records that fail to load are recorded in a reject file

```
create external table mytable_ext (  
  id integer,  
  name char(20))  
using  
  (datafiles ("DISK:/tmp/mytable.unl"),  
   rejectfile "/tmp/mytable.rej",  
   maxerrors 5);
```

mytable.rej:

```
/tmp/mytable.unl~2~CONVERT_ERR~id~X|Tom|~
```

Read/Write to Named Pipes

Instead of writing to and reading from files, can eliminate the need for the files by using named pipes

Use pipes to send output to another program

```
create external table mytable_ext  
(id integer,  
  name char(20))  
using  
(datafiles ("PIPE:/tmp/pipe.1"))
```

Read/Write to Named Pipes

```
mkfifo pipe.1
```

```
cat pipe.1 | gzip > file.unl.gz
```

```
insert into mytable_ext  
select * from mytable;
```

```
-rw-r--r-- 1 informix informix 2295 Oct 3 09:44 file.unl.gz
```


INTO EXTERNAL

Shortcut to unload data to an external table without having to define it first

External table can then be used to reload from the file

```
select *  
from mytable  
into external mytable_ext  
using  
(datafiles ("DISK:/tmp/mytable.unl")  
);
```

External Tables

DEMO

Performance Comparisons

Timings performed against a table (not fragmented) with 14 million rows, 440 MB

Linux, IDS 12.10.FC5

All loads performed into a RAW table with no indexes

Performance Comparisons

Unload

	No PDQ (seconds)	PDQPRIORITY 50 (seconds)
SQL Unload	26	22
External Table	28	11
External Table – 3 files	27	11
External Table – informix format	3	4

Performance Comparisons

Load

	No PDQ (seconds)	PDQPRIORITY 50 (seconds)
SQL Load	149	167
DBLOAD	145	150
External Table – Deluxe	156	110
External Table - Express	36	20
External Table – 3 files	33	16
External Table – informix format	23	17

Performance Comparisons

Timings performed against a table (fragmented) with 16.5 million rows, 1.8 GB

AIX 7.1, IDS 12.10.FC8

All loads performed into a RAW table with no indexes

Unload

	No PDQ (seconds)	PDQPRIORITY 100 (seconds)
onunload	400+	n/a
SQL Unload	136	131
External Table	58	54
External Table – 3 files	58	54
Ext. Table, informix format	12	12
Ext. Table, informix format, 3 files	12	12
HPL Delimited	103	103
HPL Delimited – 3 files	39	39
HPL no-conversion	22	22
HPL no-conversion – 3 files	20	20

Load

	No PDQ (seconds)	PDQPRIORITY 100 (seconds)
dbload	430	415
SQL Load	318	300
External Table – Express	162	181
External Table – 3 files	176	164
Ext. Table, Express, informix	150	126
Ext. Table, Express, informix, 3 files	139	127
External Table – Deluxe	203	216
External Table – 3 files	202	211
Ext. Table, Deluxe, informix	163	195
Ext. Table, Deluxe, informix, 3 files	200	185

Load

	No PDQ (seconds)	PDQPRIORITY 100 (seconds)
HPL, Delimited, Express	52	52
HPL, Delimited, Express – 3 files	24	24
HPL, No-conversion, Express	18	19
HPL, No-conversion, Express – 3 files	19	18
HPL, Delimited, Deluxe	108	108
HPL, Delimited, Deluxe – 3 files	100	102

Summary

- External tables are simple to use and flexible
- Good for ETL and migrations
- They are faster than using SQL and DBLOAD, esp. when using the INFORMIX format, and PDQ
- They are much easier to use than the High Performance Loader
 - Unloads *may* be faster than HPL
 - HPL *may* be faster for loads
- Because of bugs in some versions, do not test in Production!

Summary

YMMV
Very subjective!!!

Simplicity

SQL > Ext. Tables > dbload > HPL

Flexibility

HPL > Ext. Tables > dbload > SQL

Speed (Unload)

Ext. Tables* > HPL* > SQL > onunload

Speed (Load)

HPL* > Ext. Tables* > SQL > dbload

* Depends on PDQ, fragmentation, mode



Next Webcast

Informix Best Practices

**dostats and Informix
Update Statistics**

December 6, 2018 2 PM EST

Please register for each webcast at:

<http://advanceddatatools.com/Informix/NextWebcast.html>

Replays available at:

<http://advanceddatatools.com/Informix/Webcasts.html>

Advanced DataTools



Informix Training 2019

Advanced Informix Performance Tuning

- March 11-14, 2019

Informix for Database Administrators

- April 22-25, 2019
- September 16-19, 2019

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

<http://advanceddatatools.com/Training/InformixTraining.html>

Eight New Training Servers



Each Student in class will have a server running Informix 12.10 with:

- 8 CPU Cores
- 16 GB RAM
- 1 SSD Disk
- 1- 4 disks

Questions?



Send follow-up questions to
info@advanceddatatools.com

Advanced DataTools



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://advanceddatatools.com/Informix/Webcasts.html>

Email: info@advanceddatatools.com

Web: <http://www.advanceddatatools.com>



Advanced DataTools



Thank You

For more information:

info@advanceddatatools.com

<http://www.advanceddatatools.com>

Advanced DataTools