

# **Recital: Using C-ISAM Files**

---

## **Using C-ISAM Files**

Recital Corporation,  
100 Cummings Center, Suite 318J  
Beverly, MA 01915

Recital may have patents and/or patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.  
COPYRIGHT ©1988-2007 Recital Corporation. All rights reserved. All Recital products are trademarks or registered trademarks of Recital Corporation, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.

Last Updated September, 2007

## INDEX

---

<b>INTRODUCTION</b>	<b>1</b>
<b>CREATING THE RECITAL TABLE</b>	<b>2</b>
<b>CREATING THE BRIDGE FILE</b>	<b>2</b>
<b>USING THE BRIDGE</b>	<b>4</b>
<b>DATA TYPES</b>	<b>5</b>
<b>C-ISAM ERROR MESSAGES</b>	<b>6</b>
<b>JAVA/JDBC Example</b>	<b>7</b>

## **INTRODUCTION**

---

Recital Terminal Developer and the Recital Database and Mirage Application Servers support access to Informix compliant C-ISAM files.

Data access is achieved through a C-ISAM Bridge. This requires the creation of a Bridge file and an empty Recital table that has the same structure as the C-ISAM file.

## Creating the Recital Table

---

Create a Recital table with the same structure as the C-ISAM file. The fields/columns in the structure file must exactly match the data type and length of those in the C-ISAM file. The Recital table will have one byte more in total record length due to the Recital record deletion marker.

To create the table, use the SQL CREATE TABLE command or the Recital Terminal Developer CREATE worksurface. The table should be given a '.str' file extension (rather than the default '.dbf') to signify that this is a structure file only.

Please see the end of this document for information on matching Informix and Recital data types

## Creating the Bridge File

---

In Recital Terminal Developer, the Bridge File can be created using the CREATE BRIDGE worksurface. For Server clients, the Bridge File can be created with the SQL CREATE BRIDGE command or 4GL CREATE BRIDGE FROM <ini> commands.

### Maximums Widths

The following maximum widths apply to the bridge elements:

Element	Maximum Width in Characters	Description
Type	10	Bridge type: CISAM
External	80	External file name
Metadata	80	Recital 'structure' table name
Alias	10	Alias name

### CREATE BRIDGE (SQL)

The CREATE BRIDGE SQL command defines and creates the bridge in one step:

e.g.

```
exec sql
CREATE BRIDGE cisamdemo.dbf
TYPE "CISAM"
EXTERNAL "cisamdemo.dat"
METADATA "cisamdemo.str"
ALIAS "cisamdemo";
```

//or

```
exec sql
CREATE BRIDGE cisamdemo.dbf
AS "TYPE=CISAM;EXTERNAL=cisamdemo.dat;METADATA=cisamdemo.str;ALIAS=cisamdemo";
```

## **CREATE BRIDGE FROM <ini>**

Firstly, an 'ini' file should be created on the server in the data directory where the external data file is held. The ini file has the following contents:

```
[bridge]
bridgetype=<bridgetype>
externalname=<name of the external data file>
databasename=<name of the Recital structure table>
alias=<the name to use to access your file>
```

e.g. cisamdemo.ini

```
[bridge]
bridgetype=CISAM
externalname=cisam.dat
databasename=cisamstru.str
alias=cisamdemo
```

Then the CREATE BRIDGE command should be issued:

```
create bridge cisamdemo.dbf from cisamdemo
```

## Using the Bridge

---

The Bridge can now be used. To access the C-ISAM file, use the 'alias' specified in the Bridge definition.

e.g.

Select \* from cisamdemo

e.g.

- use cisamdemo
- edit

## Data Types

---

<b>Informix</b>	<b>Recital</b>
Byte	Numeric
Char	Character
Character	Character
Date	Date
Datetime	Character
Decimal	Numeric
Double Precision	Float
Float	Real
16 Bit Integer	Short
Integer	Numeric
Interval	Character
32 Bit Long	Integer
Money	Numeric
Numeric	Numeric
Real	Numeric
Smallfloat	Numeric
Smallint	Numeric
Text	Unsupported
Varchar	Character

## **C-ISAM RDD Error Messages**

---

The following errors relate to the use of the Recital CISAM Replaceable Database Driver (RDD). They can be received as an 'errno <expN>' on Recital error messages:

<b>ERRNO()</b>	<b>Error Description</b>
100	Duplicate record
101	File not open
102	Invalid argument
103	Invalid key description
104	Out of file descriptors
105	Invalid ISAM file format
106	Exclusive lock required
107	Record claimed by another user
108	Key already exists
109	Primary key may not be used
110	Beginning or end of file reached
111	No match was found
112	There is no "current" established
113	Entire file locked by another user
114	File name too long
115	Cannot create lock file
116	Memory allocation request failed
117	Bad custom collating
118	Duplicate primary key allowed
119	Invalid transaction identifier
120	Exclusively locked in a transaction
121	Internal error in journaling
122	Object not locked

## Example Java Program for JDBC C-ISAM Access

---

```
/*#=====
*# Copyright (C) 2006 Recital Corporation Inc.
*# As an unpublished licensed proprietary work.
*# All rights reserved worldwide.
*#=====
* MODULE : cisam_test.java
* PURPOSE : Recital Universal JDBC Driver CISAM file access
* AUTHOR : Recital Corporation
* DATE : Aug-2006
*=====*/

//-----
/-- Standard imports required by Recital JDBC Driver
//-----
import java.sql.*;
import java.io.*;
import java.net.URL;
import java.math.BigDecimal;
import Recital.sql.*;

public class cisam_test {

    public static void main(String argv[]) {
        int i;
        ResultSet rs;
        ResultSetMetaData rsmd;

        try {
            //-----
            //-- Load the Client Driver for the
            //-- Recital Database Server
            //-----
            new RecitalDriver();
            //-----
            //-- Build the connection URL:
            //-----
            String url = "jdbc:Recital: " +
                "SERVERNAME=servername; " +
                "DATABASE=jdbc_test; " +
                "USERNAME=username; " +
                "PASSWORD=password";
            //-----
            //-- Connect
            //-----
            Connection con = DriverManager.getConnection(url);
            //-----
            //-- Create a statement on the connection
            //-----
            Statement stmt = con.createStatement();
            //-----
            //-- Query info from the driver
```

```

//-----
DatabaseMetaData dbmd = con.getMetaData();
System.out.println(dbmd.getDriverVersion() + "\n" +
                    dbmd.getDriverName());
//-----
//-- Create table to define the metadata of the CISAM table on the server
//-----
System.out.println("Create meta data file cisamdemo.str. ");
stmt.execute("create table cisamdemo.str " +
             " (dd char(4), " +
             "confirm char(6), " +
             "procdat char(6), " +
             "control char(5), " +
             "dollars num(13,2), " +
             "dealer char(5), " +
             "territory char(2), " +
             "worep char(12), " +
             "currtan char(3), " +
             "traddate char(6), " +
             "city char(10), " +
             "account char(11), " +
             "pretran char(2), " +
             "afsrep char(14), " +
             "repkey char(9), " +
             "branch char(3), " +
             "wodealer char(5), " +
             "bankcode char(2), " +
             "commrate num(6,4), " +
             "newrep char(1), " +
             "settle char(1), " +
             "postdate char(6) ");

//-----
//-- Create Bridge file
//-----
System.out.println("Create Bridge File cisamdemo.dbf. ");
stmt.execute("create bridge cisamdemo.dbf " +
             "type 'CISAM'+
             "external 'cisamdemo' + // Name of the CISAM data file.
             "metadata 'cisamdemo.str'+ // Name of the Metadata file.
             "alias 'cisamdemo'");

//-----
//-- Query for data
//-----
System.out.println("Select rows from cisamdemo. ");
rs = stmt.executeQuery("SELECT * from cisamdemo");
rsmd = rs.getMetaData();
int nr_cols = rsmd.getColumnCount();
while (rs.next()) {
    for (i = 1; i <= nr_cols; i++) {
        System.out.println("rs Column[" + i + "]: " +
                           rsmd.getColumnName(i) + "<" +
                           rsmd.getColumnTypeName(i) + "> = " +
                           rs.getString(i));
    }
}

```

```

        System.out.println("Got results: ");
    }
    //-----
    //-- Release the statement
    //-----
    stmt.close();
    //-----
    //-- Disconnect from the server
    //-----
    con.close();
} catch (Exception e) {
    System.out.flush();
    System.err.flush();
    DriverManager.println("Recital JDBC driver exception: " + e.getMessage());
    e.printStackTrace();
}
try {
    System.out.println("Press any key to continue...");
    System.in.read();
} catch (IOException ie) {
    ;
}
}
}

```