PANAGON™

TECHNICAL NOTICE

INSTALLING CS TO SHARE A SQL SERVER DATABASE ENGINE WITH IS

Document 9812130-002

For updates to any Content Services (CS) documentation, choose the Documentation link on the FileNET Worldwide Support web site http://www.css.filenet.com/, and navigate to the current CS release for your platform. If you do not have a customer support services (CSS) Web Account, click the New User button and follow the online instructions.

Overview

Many sites will install user interfaces (such as IDM Desktop) on client PCs so that users can access both Image Services (IS) and Content Services (CS) library systems. Some sites may want a CS library system to use the database engine already being used by an existing IS library system. This technical notice describes what to do *before* you set up the initial components of a CS library system on the same server and database engine used by an existing IS system.

The procedure described in this technical notice requires you to:

- Back up your existing stored data (for IS and other applications using any associated *site-controlled* SQL Server engine).

  WARNING The multistep procedure in this document is high-risk. Failure to perform prior backups may result in loss of data or system integrity.
- Use the Microsoft SQL BCP utility to export the IS database.
- Rebuild the master database with the network configurations, character set, and sort order specified by the CS software requirements.

  CAUTION If other applications besides IS are using a particular *site-controlled* SQL Server engine, be sure they can accommodate the settings indicated in the following topic, "Requirements and Restrictions" (e.g., for sort order).
- Retrieve that exported data and import it into the rebuilt database.

Requirements and Restrictions

There are several major issues to consider when collocating IS and CS on the same SQL Server database engine:

- Running CS and IS databases on the same SQL Server platform currently requires a **Windows 2000 operating system and a Microsoft SQL Server 2000 database** environment. See the "Supported Configurations under SQL Server" topic for details.

- CS library systems run in a case-*insensitive* database environment (i.e., ***Dictionary order, case-insensitive, accent-insensitive* sort order***)*. These library systems consider uppercase and lowercase characters in property values (known as *indexes* to IS users) to be identical. On the other hand, IS, by default, installs into a database configured to be case-*sensitive* for indexes.

  Therefore, you must analyze your current working environment and decide whether you can adapt your currently stored data (for IS or any other applications that are sharing the SQL Server engine) to a case-insensitive model. If not, you will have to install CS on a dedicated SQL database engine and retain your existing database engine(s) for the existing IS systems.

- In English and Latin-1 language environments, new CS library systems run best on the following database code page: **1252/ISO Character Set** (also known as 8859-1).
  If you have Image Services already installed, you will likely have to rebuild the SQL Server Master database to reset it to this code page (and to the sort order indicated in the previous bullet).

- In IS (using fn_edit) you must, at a minimum, accommodate CS requirements for the following settings (CS typically uses the SQL Server defaults, as indicated here):

  - **User Connections (20)**
  - **Open Objects (500)**
  - **Recovery Interval (5)**

- You can set up IS to run with a *FileNET-controlled* or *site-controlled* SQL Server engine. We recommend, however, that you set up **IS in site-controlled mode** so you have independent control over the database instances for IS and CS. If you run in FileNET-controlled mode, whenever you shut down or restart Image Services, the same action will occur to the CS library systems; this may be disruptive to your users, their client applications, and general CS processes.

Supported Configurations under SQL Server

You can install CS to share a server and SQL Server database engine with an existing IS system only under the following configuration:

Panagon IDM Server-Based Services

- Image Services 3.6.0
- Content Services 5.2.0

Operating Systems:

- Microsoft Windows 2000 Server with Service Pack 2

  or
- Microsoft Windows 2000 Advanced Server with Service 2

Database:

- Microsoft SQL Server 2000 Enterprise Edition with Service Pack 1

Procedure

➥ To install CS to share a database engine with an existing IS system:

1. Back up the IS database, as well as any databases used by other applications that are sharing the *site-controlled* SQL Server engine.

   WARNING The multistep procedure that follows is high-risk. Failure to perform database backups may result in loss of data or system integrity.
2. Write down the record count for each IS database table, and also note a few of the actual records so that you can check

them at the end of the following procedure prior to installing CS. To access the IS database tables, do the following:

a. From the Start menu, choose Programs > Microsoft SQL Server 2000 > Enterprise Manager > SQL Server Group.

b. In the Console Root window, expand the SQL server group; then expand the database server containing Image Services (e.g., LOCAL or the appropriate system name).

c. Expand Databases, expand the IS database (for example, *fnsys*), and click Tables.

3. Analyze *all* indexes (property values) in every table in the entire IS database (not just the index database) and modify them if necessary to ensure unique names in a case-insensitive environment. This will also affect any other applications that might be sharing the SQL Server engine.

   You can use the SQL Server Query Analyzer to get just the names of classes and indexes, and that data can be easily exported and sorted so that you can determine duplicates. You can also use bcp, ddexim, and SEC_tool's export to get such data in a usable text form that you can easily work with (e.g., you could export the results to a product such as Microsoft Excel and then sort by name value).

   Here are some simple queries you can run on each IS table to return a list of associated values. For the key index of each table, use the column name.

   ```
   Select f_docclassname from f_sw.document_class
   order by f_docclassname

   Select f_indexname from f_sw.user_index order by
   f_indexname
   ```

   CAUTION This step is critical. If you do not resolve duplicate indexes (for example, you cannot allow the same value to be repeated, one in uppercase and one in lowercase), the following may occur:

   - associated IS tables may be dropped altogether
   - your database will not be imported as expected

- your database will not be recoverable in the rebuilt database
- associated applications, such as COLD, may be adversely affected, and you may not be immediately aware of it

Therefore, be sure to check for errors, as documented later in the procedure, before you proceed with the CS installation.

4. Turn off the automatic ANSI to OEM option in the SQL Client Configuration Utility.
   a. From the Start menu, choose Programs > Microsoft SQL Server 2000.
   b. Choose the Client Network Utility.
   c. Click the DB Library Options tab.
   d. Uncheck the Automatic ANSI to OEM option under DB Library Configuration.
5. Export the database using BCP, as follows:
   a. Copy the names of the IS tables into an **export.bat** file as follows:
      i. From the Start menu, choose Programs > Microsoft SQL Server 2000 > Enterprise Manager > SQL Server Group.
      ii. In the Console Root window, expand the SQL server group and expand the database server containing Image Services (e.g., LOCAL or the appropriate system name).
      iii. Expand Databases, expand the IS database (for example, *fnsys*), and click Tables.
      iv. Go to Tools > Query Analyzer.
      v. Log in to the appropriate IS server.
      vi. Select the IS database from the drop-down list.
      vii. Enter the following query statement (on one line):

```
select uid, name from sysusers where name
like 'f_%'
```

viii. Verify that the users *f_xxx* are displayed and make note of the uids (user IDs).

ix. Enter the following query statement (on one line):

```
select name, type, uid from sysobjects where
type = 'U' and uid in (n1, n2, n3...nX) order
by name
```

where *n1* through *nX* are the uids obtained for *f_xxx* users in the query you ran in step vii.

x. Verify the returned information, then edit the query statement in step ix and re-run it as:

```
select name from sysobjects where type = 'U'
and uid in (n1, n2, n3...nX) order by name
```

xi. Right-click in the Results window and choose "Select all". Choose File > Save As to save the list of IS table names returned in the query to an **export.bat** file in a newly created directory folder (e.g., **c:\exp**).

b. Open the **export.bat** file in a text editor.

c. In place of each IS database table now listed, create the following line in the file:

```
bcp IS_database_name.IS_user_name.table_name out
table_name.data -n -e table_name.err -o
table_name.out -Usa -P sa_password
```

For example:

```
bcp fnsys.f_sw.doctaba out doctaba.data -n -e
doctaba.err -o doctaba.out -Usa -P
```

d. Save and close the **export.bat** file.

e. Open a DOS window, change to the export directory you just created, and type:

```
export
```

to run **export.bat**. For each specified table, the batch operation creates in your export directory the following three files, into which it exports associated data in native format:

***table name*.data**

***table name*.err**

***table name*.out**

f. Check the ***table name*.err** and ***table name*.out** files for any logged errors.

6. Create a SQL script to create *wqm* and *doctaba* tables.

   a. From the Start menu, choose Programs > Microsoft SQL Server 2000 > Enterprise Manager.

   b. In the Console Root window, expand the SQL server group, and then expand the database server containing IS.

   c. Expand Databases, right-click the IS database (for example, *fnsys*), and in the context menu choose the All Tasks > Generate SQL Script command. The Generate SQL Scripts dialog box opens.

   d. In the General tab, under Objects to Script deselect Script All Objects. This moves all tables into the bottom left list box labelled *Objects on fnsys* (or whatever your IS database is called).

   e. In the General tab, under Objects to Script select the All Tables check box.

   f. Modify the lower right list box so it contains *only* the following tables:

      - *doctaba*
      - *wqm*-prefixed tables

      If any of these tables are in the left list box, select them and click the Add button to move them to the right list box. To remove other tables from the right list box, select them and click the Remove button.

   g. In the Formatting tab, make sure only the *Generate the CREATE object_name command* check box is selected for each object.

   h. In the Options tab, under Table Scripting Options select the Script Indexes check box.

   i. In the General Tab, click the Preview button in the upper right corner to view the SQL script prior to running it.

j. Click OK. A SQL script will be generated to create all tables except **doctaba** and **wqm** tables.

k. When prompted, save the SQL script as ***drive*:\mssql\binn\fntables**.

l. Click OK once the scripting is complete.

7. In the Services control panel, stop the services for Image Services and then MS SQL Server.

8. Change the supported Network Libraries Configuration, as follows:

   a. From the Start menu, choose Programs > Microsoft SQL Server 2000 > Server Network Utility.

      The SQL Server Network Utility dialog box opens.

   b. Click Add.

      The Add Network Library Configuration dialog box opens.

   c. Under Network Libraries, select Named Pipes, leave the default Pipe Name (**\\.\pipe\sql\query**), and click OK.

      The SQL Server Network Utility dialog box re-opens.

   d. Click Add.

      The Add Network Library Configuration dialog box re-opens.

   e. Under Network Libraries, select TCP/IP, leave the default Port Number (1433), and click OK.

   f. Click Apply and then OK to close the SQL Server Network Utility dialog box.

CAUTION Before running the Microsoft command-line utility **rebuildm.exe** to rebuild the SQL Server Master database, see article Q273572 (bug 236130) in the Microsoft Knowledge Base at http://support.microsoft.com. This article describes how **rebuildm.exe** fails when the source directory containing data files is on a CD. The article also provides a workaround for this bug, as follows:

   a. Insert the *Microsoft SQL Server 2000* CD.

   b. Copy the entire contents of the CD to a network location, to serve as the source directory for the data files.

c. Remove the Read-only attribute on the source directory, its subdirectories, and files.

9. Rebuild the SQL Server Master database, as follows:

   a. From the Start menu, choose Run.

   b. In the Open field, browse to find the directory containing the Rebuild Master command, type the following, and click OK:

      `rebuildm.exe`

      The Rebuild Master dialog box opens.

   c. Click Settings.

      The Character Set/Sort Order/Unicode Collation dialog box opens.

   d. In the Character Set field select *1252/ISO Character Set (default)*, in the Sort Order field select *Dictionary order, case-insensitive, accent-insensitive*, and click OK. These settings will accommodate the requirements of Content Services.

      The Rebuild Master dialog box re-opens.

   e. In the *Source Directory containing Data Files* field, type the following and click Rebuild:

      `CD drive:\x86\Data`

   For further details on running **rebuildm.exe**, see the following topics in the SQL Server *Books Online*:

   - "Changing the Character Set, Sort Order, and Unicode Collation"
   - "Rebuild Master Utility"

10. Reboot the server once the rebuild is complete.
11. Re-create the FileNET IS database (*fnsys*, for example).
12. Initialize the IS database by running *fn_util initrdb* using the command line in DOS.
13. Delete (drop) the *doctaba* table, as follows:

    a. From the Start menu, choose Programs > Microsoft SQL Server 2000 > Enterprise Manager.

b. In the Console Root window, expand the SQL server group, and then expand the database server containing IS (e.g., LOCAL or the appropriate system name).

c. Expand Databases, then expand the database for the IS system (e.g., *fnsys*) and click Tables.

d. In the details pane, right-click the *doctaba* table and then in the context menu click Delete. The Drop Objects dialog box opens.

e. Click Drop All. This drops the *doctaba* table from your IS database.

14. Truncate all tables with data, as follows:

a. From the Console Root\Microsoft SQL Servers\SQL Server Group window in SQL Enterprise Manager, choose SQL Query Analyzer from the Tools menu. The Query window opens.

b. Right-click each table in the IS database and choose Properties to find out (and note for the next step) whether it contains data.

c. For each table in the IS database that contains data, run the following command in the query window:

```
truncate table user name.table name
```

NOTE For the *user name* value, see the Owner column in the List of tables for the database under Console Root.

For example:

```
truncate table f_sw.docataba
```

15. Run the SQL script to create *wqm* and *doctaba* tables, as follows:

a. Click the New Query toolbar button (CTRL+N) to start a new query in the Query window.

b. In the SQL Server Query Analyzer dialog box, choose File > Open.

c. Select ***drive*:\mssql\binn\fntables.sql** and click the Open button.

d. In the SQL Server Query Analyzer dialog box, choose Query > Execute. The SQL script you generated prior to rebuild will now create the *doctaba* and *wqm* tables in the rebuilt database.

16. Close the SQL Query Analyzer tool.
17. Verify that the automatic ANSI to OEM option is off, as indicated in step 4 on page 6.
18. Import data using BCP, as follows:
    a. Copy **export.bat** to a file named **import.bat**. Be sure to place this file in the same directory where the ***.data** files are located.
    b. Modify each line in the **import.bat** file to the following format:

```
bcp IS_database_name.IS_user name.table_ name
in table_name.data -n -E
-e table_name_in.err -o table_name_in.out
-Usa -P sa_password
```

    For example:

```
bcp fnsys.f_sw.doctaba in doctaba.data -n -E
-e doctaba_in.err -o doctaba_in.out -Usa -P
```

    where: *fnsys* is the IS database name, *f_sw* is the IS user name, and *doctaba* is a table name.
    c. Run **import.bat** using the command line in a DOS window. This will import the data in each IS table back into the master database using the ***.data** files.
    d. Check the ***.err** and ***.out** logs for errors.
19. (IS *FileNET-controlled* systems only) Set the memory and user connections for MS SQL Server to accommodate the CS library system you intend to install, as follows.
    a. From the Start menu on your IS server, choose Run.
    b. In the Open field, browse to find the IS System Configuration Editor (**fn_edit**) and click OK.
    c. Click OK in the Open Configuration Database dialog box.
    d. Select the Relational Databases tab in the FileNET Image Services System Configuration Editor dialog box.

e. Select the MS SQL tab.

NOTE For details on what CS requires concerning the following three substeps, refer to “Preparing a SQL Server Database” in the “Prerequisites to Installing Content Services” topic of the Content Services *Installation Guide*.

f. Locate the Memory parameter for the particular database server and modify to the desired value.

g. Locate the User Connections parameter for the particular database server and modify to the desired value (minimum of 20 for CS).

h. Locate the Recovery Interval parameter for the particular database server and modify to the desired value (minimum of 5 for CS).

i. From the File menu choose Exit.

j. Click Yes when prompted to save your changes.

20. Restart the IS library system.
21. Check that the record counts and information you wrote down in step 2 on page 4 still match the IS database tables.
22. You may now create the database for, and install, a CS library system on the SQL Server engine. See the CS server documentation for details.