

Using SAS/CONNECT to Distribute Data to a LAN File Server from an MVS SAS Job

Harmon L. Jolley

*Healthsource Provident Plan Administrators, Inc.
Chattanooga, TN*

Abstract

Most large organizations have a wealth of data stored in mainframe data bases. The evolution of local area networks and file servers has brought about the need to distribute mainframe data, yet also to keep the data current and secure. This paper will show how that SAS/CONNECT® and APPC® can be used to accomplish this.

The Application

We have promoted the use of the Query Window of SAS/ASSIST® as a way for data analysts to do ad hoc queries of mainframe DB2® data. This is done in a client/server fashion by using SAS/ACCESS® DB2/2® and IBM's DDCS® gateway to DB2. The current version of the Query Window provides an option to reduce network traffic that is caused by having to download data from the DB2 system catalog such as table and column names. It provides the ability to use "mirror" tables: local SAS datasets which contain information from the DB2 catalog tables, and which is quickly available to the Query Window's list boxes.

Our testing showed that usage of mirror tables accomplished our goal of improving response time. However, we needed to ensure that the DB2 mirror tables were kept current with the mainframe source. Through using SAS/CONNECT in an MVS job which reads the DB2 catalog and sends data to the IBM LAN Server®, the data is kept current.

Some Assembly Required...

In order to send data from a mainframe SAS program to an IBM LAN server via APPC, the following software products must be present:

Mainframe

- Base SAS for MVS
- SAS/CONNECT for MVS

IBM LAN Server

- Base SAS for OS/2®
- SAS/CONNECT for OS/2

Mainframe and LAN Server

- APPC
- Communication Manager/2®

Some of these products may already be in place if the site is using APPC for client/server development. While our application uses APPC as the communications protocol, similar applications should be able to be developed using other protocols supported by the SAS System.

With the software installed, the following set-up is needed to provide the connection from a mainframe job to the LAN server:

1. Two transaction definitions are added to the Network Descriptor File (NDF) which is present in the Communications Manager/2 directory.
2. A configuration file is defined for use in the SAS for OS2 session. This defines the communications protocol to be used.
3. A mainframe SAS program is written to sign on to the LAN server, then distribute data. Note that things are somewhat reversed from a

common way to use SAS/CONNECT, which is to download data from the mainframe after signing on to it from a SAS session on the workstation. In this application, it is the LAN server which is the remote session, so the data is transferred via PROC UPLOAD. Other SAS program code could precede this, and the data can either be a SAS dataset or flat file.

If you have any questions or comments, please contact:

Harmon Jolley
Healthsource Provident Plan Administrators, Inc.
2 Fountain Square
Chattanooga, TN 37402
(615) 755-8282
jolleyh@hlthsrc.com

An example of the code for each of these three items is shown at the end of this paper.

Conclusions

We are very pleased with the results of our application to distribute mainframe data to a server. Other applications may be based on this in the near future which will also distribute mainframe data to servers.

References

With any client/server development, it is important to have highly skilled and supportive people in communications and systems programming. I wish to thank three such people who have helped greatly with client/server development, including this application:

John Phleger

Mike Tribble

Gerry Williamson

SAS, SAS/ACCESS, and SAS/ASSIST are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

APPC, Communications Manager/2, DB2, DB2/2, DDCS, LAN Server, and OS/2 are registered trademarks of IBM.

The following SAS manuals were used as references for the material presented in this paper:

Installation Instructions: The SAS System under OS/2, release 6.10

SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition

Exhibit 1: Network Descriptor File installed on LAN server to provide connection from SAS mainframe session

Note that this file has a .NDF extension, and is present in the subdirectory containing Communications Manager/2.

```
DEFINE_TP TP_NAME(SASTP62)
          DESCRIPTION(Internal SAS Service TP)
          PIP_ALLOWED(NO)
          FILESPEC(SASroot directory\SASTP62.EXE)
          CONVERSATION_TYPE(EITHER)
          CONV_SECURITY_RQD(YES)
          SYNC_LEVEL(EITHER)
          TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
          PROGRAM_TYPE(BACKGROUND)
          INCOMING_ALLOCATE_QUEUE_DEPTH(255)
          INCOMING_ALLOCATE_TIMEOUT(120)
          RECEIVE_ALLOCATE_TIMEOUT(60);
```

```
DEFINE_TP TP_NAME(SASRMT)
          DESCRIPTION(Remote SAS Initiation TP)
          PIP_ALLOWED(NO)
          FILESPEC(SASroot directory\SAS.EXE)
          PARM_STRING(-CONFIG C:\SAS\CONFIG.DMR)
          CONVERSATION_TYPE(EITHER)
          CONV_SECURITY_RQD(YES)
          SYNC_LEVEL(EITHER)
          TP_OPERATION(NONQUEUED_AM_STARTED)
          PROGRAM_TYPE(PRESENTATION_MANAGER)
          RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

Exhibit 2: Configuration Files installed on LAN server and MVS to provide connection from SAS mainframe session

The following code is in the file referenced in PARM_STRING of exhibit 1.

```
/* SAS APPC config file to allow OS/2 as a remote partner */
-NOLOGO /* required to prevent remote signon hang */
-DMR
-COMAMID APPC
-REMOTE LU-name of server
-NOTERMINAL
-NO$SYNTAXCHECK
-SET APPC_LU62MODE RDBLOW

/* The following file contains any start-up procedures needed in the
remote SAS session */
-AUTOEXEC c:\sas\autoexec.dmr
```

The following code is in the file referenced as the CONFIG file in the MVS SAS program in Exhibit 3.

```
* APPC OPTIONS MF TO PC - ADD TO MF CONFIG FILE
LUPOOL=ALL
LUPREFIX=PLAS40
LUFIRST=1
LULAST=50
LU62MODE=RDBLOW
```

Exhibit 3: SAS MVS Job which reads the DB2 catalog, then sends a subset of it to a LAN server

```
//Job name JOB (accounting),H.JOLLEY,CLASS=run class,MSGCLASS=message class

/**
/** QUERY WINDOW DB2 "MIRROR TABLE" UPDATE.
/** THE MIRROR TABLES ARE PC COPIES OF DB2 CATALOG, AND ARE USED TO
/** IMPROVE QUERY WINDOW PERFORMANCE BY AVOIDING DOWNLOADING FROM
/** DB2 CATALOG.
//SAS01 EXEC SAS,OPTIONS='MACRO,DQUOTE',
// CONFIG='config file'
//DB2DEFS DD DSN=SAS library,DISP=OLD
//APPCID DD DSN=SAS library,DISP=OLD
//SYSIN DD *
OPTIONS ERRORABEND;

* OBTAIN SUBSET OF AVAILABLE TABLES AND COLUMNS FROM DB2 CATALOG;

PROC SQL;
CONNECT TO DB2 (SSID=DB2 subsystem ID);

CREATE TABLE DB2DEFS.DB2TABS AS
SELECT *
FROM CONNECTION TO DB2
(
SELECT NAME, CREATOR, DBNAME, COLCOUNT, REMARKS
FROM DB2.SYSTABLES
WHERE (DBNAME LIKE 'G%' OR DBNAME LIKE 'S%') AND
CREATOR NOT LIKE '$G%'
);
* THE FOLLOWING DISPLAYS SQL RETURN CODE AND MESSAGE IN LOG WINDOW ;
%PUT &SQLXRC &SQLXMSG;

CREATE TABLE DB2DEFS.DB2COLS AS
SELECT NAME, TBNAME, TBCREATO, REMARKS
FROM CONNECTION TO DB2
(
SELECT A.NAME, A.TBNAME, A.TBCREATOR, A.REMARKS
FROM DB2.SYSCOLUMNS A, DB2.SYSTABLES B
WHERE A.TBNAME = B.NAME AND
A.TBCREATOR = B.CREATOR AND
(B.DBNAME LIKE 'G%' OR DBNAME LIKE 'S%') AND
A.TBCREATOR NOT LIKE '$G%'
);
* THE FOLLOWING DISPLAYS SQL RETURN CODE AND MESSAGE IN LOG WINDOW ;
%PUT &SQLXRC &SQLXMSG;
DISCONNECT FROM DB2;

* UPDATE DB2DEFS.DB2TABS AND DB2COLS USING SAS/CONNECT.
DB2DEFS IS A LIBRARY STORED ON THE SERVER. ;
%INCLUDE APPCID;
SIGNON;
RSUBMIT;
LIBNAME DB2DEFS 'SAS library path';
PROC UPLOAD INLIB=DB2DEFS OUTLIB=DB2DEFS;
RUN;
ENDRSUBMIT;
```

Note that the dataset referenced by APPCID contains a single line of SAS program code as follows. The ID and password must be defined on the server to Communications Manager/2. SAS/CONNECT signs on using the ID and password. For security reasons, I recommend placing this SAS code in a dataset that is secured so that only the job and the security administrator can see the ID and password.

```
OPTIONS COMAMID=APPC REMOTE=LU-name of serverAPPCSEC='userid.password';
```