Paper 292-25

TAKING ADVANTAGE OF THE SAS® SYSTEM ON THE OS/390 PLATFORM.

David Crow,
SAS Institute Inc., Cary, NC

ABSTRACT

More and more businesses are finding that the Internet provides an ideal way to access both new and legacy applications running on OS/390. This paper suggests some of the ways you can use the SAS® System for OS/390 to provide Web access to information. We will also describe some of the new features of Version 8 regarding performance tuning, use of new operating system features, and use of SAS as a server. See how SAS and OS/390 can work together to provide exciting solutions to problems.

INTRODUCTION

Version 8 of the SAS System for OS/390 contains many new features that can be used not only to build new applications but also to breathe new life into old ones. This paper describes some of these new features and how to take advantage of them. Some of the changes, such as improved multi-volume support and longer character variables help solve problems you have previously encountered when developing SAS applications, but the most dramatic changes have occurred in the area of Web enablement and interoperability with other platforms. This paper looks at improvements in all of these areas, starting with the Web. Nearly every component of Version 8 of the SAS System contains features that help enable Web based applications. Base SAS provides the Output Delivery System (ODS), the FTP access method, and the e-mail access method. SAS/GRAPH® provides Web-ready output in GIF format, and other software products such as webEIS™, webAF™, Enterprise Guide™, and SAS/IntrNet™, are all aimed at providing SAS applications access to the Web.

WEB ENABLEMENT

The Web is here, and everybody knows it's not just the next incremental step in the computer business. When social problems with the Web become a frequent topic in Ann Landers you know the world has changed! For many years, the PC, UNIX, and mainframe systems have appeared to be in a race, and we were looking for one big winner. We have all heard theories as to whether the PC would defeat UNIX, and all kinds of speculation as to how much longer the mainframes could last. But now the Web has arrived, and its methodology encourages a heterogeneous mixture of platforms. OS/390 is ready to participate in this mix. In particular, OS/390 offers operational characteristics, such as security, manageability, scalability, and 24x7 availability that are desperately needed for industrial strength Web applications. But what can you do with mainframe applications that were invented before the Internet and were designed as batch applications, or for display on 3270 terminals? The answer, surprisingly, is quite a lot, and in many cases, a small effort can pay huge dividends in both the appearance and the utility of these mission-critical applications. The applications are already there; all we have to do is hook them up. There are, of course, legitimate concerns that suggest a company should be careful about deploying Web-based applications on OS/390, but the business opportunities are so great that problems once considered insurmountable are now just problems awaiting a solution. As OS/390 sites begin to look at what they must do to Web-enable applications, most will discover they have already solved enough of these problems that they can immediately begin using the Web for a few applications. Then, over time, the once overwhelming collection of insurmountable problems will dwindle away until using the Web becomes the preferred way to access OS/390.

The Web: Parts and Pieces

In its simplest form, a Web-based application produces static reports to be viewed on a desktop using a Web Browser. To do this, the application writes HTML (HyperText Markup Language) text files and a Web Server sends those files to the Web Browser whenever the user asks for them. The communication is done across a TCP/IP network that is either the Internet itself, or a smaller network that may have limited connection to the Internet, or in some cases, no connection at all.

Most OS/390 installations already have all these pieces in place. A TCP/IP network connection provides access to OS/390 from 3270 emulators running on PC desktops, and most business networks have at least one Web Server running somewhere. So for this simplest case, the only problems that remain have to do with producing the HTML text files and providing them to the Web Server.

SAS features for use with the Web

Version 8 of the SAS System provides many tools for dealing with Web reports and the Internet. Here, we will only look at a few of them: The Output Delivery System (ODS), the FTP access method, the Socket access method, and finally the e-mail access method.

Output Delivery System (ODS)

ODS is a new component of SAS that processes essentially all output from SAS procedures. By default, ODS renders the output in a print format that is basically unchanged from Version 6. However, with a few simple changes, ODS can render output directly into Web-ready HTML, and can even be used to produce hypertext links between related reports. For example, your summary reports can contain links to detail or exception reports so that navigation and drill-down functions can be easily included. In fact, when used with SAS/GRAPH, ODS can provide drill-down links from regions of a graph so you can provide both the high-level overview in graphic form, as well as additional detail in report format. ODS provides the SAS programmer with default format layouts, called TEMPLATES, that simplify startup. ODS also provides the ability to create your own templates for customized layouts with such features as a corporate logo in the background, or color highlighting on individual fields. All of this is done with a new SAS statement, appropriately called ODS, a few new options on the PROC and DATA statements, and a new procedure, PROC TEMPLATE. For more information on the Output Deliver System, see *The Complete Guide to the SAS Output Delivery System, Version 8 (Order #CW57241).*

FTP Access Method

Once a report is in HTML format, the Web Server must have access to it. Unless your Web Server is running on OS/390 (the Domino Go Web Server, or the HTTP server), you will probably need to send the file to another system. You could create the file on OS/390 and add a separate FTP step to the end of your SAS job, but an alternative is to use the FTP access method. With this access method, SAS will FTP the output file to the target system as the file is produced. The FTP access method is invoked with the FILENAME statement. In addition to naming the file to be used, other options on the FILENAME statement specify exactly how the FTP is done:

- HOST – specifies the remote host to use.
- CD – specifies the directory name to use on the remote host.
- USER – specifies the name of the user that is to be used for FTP signon authentication.
- PASSWORD -- The password may either be specified on the FILENAME statement, or in an interactive session, may be specified as “PROMPT”.

See the SAS Language Reference: Dictionary, Version 8 (Order #CW57240) for details on the use of the FTP access method, including examples.

Socket Access Method

The Socket access method allows a SAS session to act as a server and communicate with client computer programs via the TCP/IP access method. The client computer program need not be running SAS. You may never need to use the Socket access method directly, but certain SAS products such as SAS/IntrNet use it. The Socket access method is a part of the base SAS System.

E-mail Access Method

So far, we have described changing traditional SAS programs to produce HTML reports, and using FTP to transfer them to the Web Server. But when this is done from a batch job, the user at the desktop may not know when to look for results. One way to provide this notification is via the e-mail access method. On OS/390, the SAS System produces e-mail output in SMTP format that can be handled by nearly all e-mail packages. This can be used to send notification that the report is ready for viewing, and can include the initial Web URL so that the user can simply click to begin looking at the report. The e-mail access method also supports attachments so the e-mail can contain the full report if desired. To use the e-mail access method, you write the body of the mail to a file. The FILENAME statement is used to identify this as e-mail, and includes options that specify the e-mail address, the subject, and any attachments you wish to include. The e-mail access method was released as an experimental interface in Version 8, and will be a fully supported production interface in 8.1. The e-mail access method is for output only, and cannot be used to retrieve e-mail.

For complete details on using the e-mail access method, including syntax and examples, see the technical note on the tech support web site at:

http://www.sas.com/techsup/download/
technote/ts605/html

Full documentation will be available in the standard SAS Reference material.

Use of HFS files for HTML output

The FTP access method provides the ability to write files directly to the system where the Web Server is running. Of course, if anything goes wrong with that server, or with your network connection, your job will fail. When a job requires significant resources, it may be desirable to produce the report files directly on the OS/390 system, and then transfer them in a separate step.

The SAS System supports writing HTML files to traditional MVS file formats like PDSE and sequential, but when you finally transfer them to the Web Server platform, you have to change the names to use the directory based file system on the server. For simple reports, this is not difficult, but for complex, interrelated reports, it can be confusing and error-prone. An alternative is to use the OS/390 Unix File System called HFS (Hierarchical File System). Although it is called the Unix File System, any job or session can use HFS files. These files are fully supported by the SAS System: anywhere that a SAS program can use a sequential data set, it can use an HFS file. Of course, if you are using an OS/390 Web Server, such as Domino Go Webserver or the HTTP Server in WebSphere, then you should definitely write the files directly to HFS, because the Server will need to read them from HFS. In short, your HTML files are going to end up in a directory based file system eventually, so it's often easiest to write them there in the first place.

The HFS file system has been available in all releases of OS/390, but a fairly large number of traditional MVS shops have never needed it, so you may find you are one of the first users in your installation to ask about using HFS files. Some key points to remember:

- HFS files are contained in libraries, similar in concept to a PDS, or a SAS data library. Each entry in the directory of an HFS library is either a file or another directory. The complete name of an HFS file traces the directory hierarchy from the system root all the way down to the file name, so HFS files have names like /u/user1/reports/file2. This name is called the PATH, and each entry between slashes identifies a directory.
- HFS libraries must be created on a volume managed by OS/390 System Managed Storage (SMS)
- Usually, an HFS library is allocated for a single user, and all files and directories created by that user must fit in that one HFS library. Other users may also use or create files in this library if they have been granted appropriate authority.
- The HFS library must be assigned a “mount point”. This is the first few directory levels in the PATH that OS/390 uses to figure out which HFS library to use. For example, all HFS files whose path begins /u/user1/ will be in one HFS library, while all that begin /u/user2/ would be in another library. The mount point can be assigned explicitly using the MOUNT operator command, or implicitly via the AUTOMOUNT facility.
- For RACF users, the USER definition requires an OMVS segment. The OMVS segment contains the default home directory path, and other information about the user's access to HFS files.
- HFS files are not restricted to OS/390 Unix applications. They can be used as an ordinary MVS sequential data set. To reference an HFS file in a batch job, use the PATH parameter instead of the DSNAME parameter on the DD statement.
- IBM has excellent documentation on the HFS file system. Refer *to OS/390 V2R8.0 Unix System Service Planning* (SC28-1890-08), and Redbook *OS/390 Version 2 Release 6 Unix System Services Implementation and Customization (*SG24-5178-00).

Web-enabling an application often means that you will change several PROC statements and

DATA Steps to direct output to files instead of printed reports. So instead of a single print file, you may now end up creating a dozen or more HTML files. To simplify coding, and to collect related files together, you can create a fileref to be used as a common directory path. This allows use of a simple file name, and if the file does not exist, it will be created. This is similar to writing multiple members into a PDSE, but it is more flexible, because you can mix file types, and the file name can be more descriptive.

Example: Assume that you have a TSO userid named “myuser” that has been authorized to create HFS files, and you want ODS to produce several files in a directory called public_html/dailyjob. First, you use a filename statement like the following to create a fileref that you can use in later statements:

```
        FILENAME odsout
        '/u/myuser/public_html/dailyjob';
```

To direct ODS to create a report to the directory named by this fileref, you might code:

```
        ODS html path=odsout
        body='report_index.html';
        PROC PRINT DATA=INDEX;
        ODS html path=odsout
        body='report_detail.html';
        PROC PRINT DATA=DETAIL;
```

The SAS System would then create two separate HFS files:

```
        /u/myuser/public_html/dailyjob/
        report_index.html
        /u/myuser/public_html/dailyjob/
        report_detail.html
```

These filenames don’t look like traditional MVS DSNAMES, but they are absolutely standard for OS/390, and are fairly typical of the names used in Web applications. As mentioned above, ordinary sequential files or PDSE members can be used instead of HFS files, but don’t try to use a traditional PDS. ODS needs to write multiple files at once, and this does not work with a PDS.

As you develop complex Web reports with many different types of files, the power and flexibility of the HFS file system will become more and more apparent. The SAS System also supports the use of HFS files for SAS data files; and this will be described in the section on interoperation with other platforms.

The SAS/IntrNet product

So far, we have discussed static Web publishing capabilities that are part of the base SAS System in which the user at the Web Browser does not have an opportunity to affect how or when the program runs. The SAS/IntrNet product adds these elements to provide Dynamic Web processing. In some cases, especially when retrieving information from online databases, you may wish to have a query run on demand, and you may even need to provide some input parameters for the query. One way to do this with SAS/IntrNet is to use is the CGI-broker. A CGI-Broker is invoked when the Web Server recognizes that the Web Browser has requested a special kind of URL. This URL names the CGI-Broker, and includes any additional values for specific variables that may be needed. The Web Server calls the CGI-Broker, which returns a single HTML page to be displayed at the Browser. The broker provided by SAS/IntrNet operates by establishing a connection to a SAS Session via the Socket access method. Each time the Broker receives a request, it writes the data to this TCP/IP socket to tell the SAS Application Server what program to run, along with other parameters that it will need. This program, using ODS, produces a Web page and returns it to the Broker, which in turn passes it back to the Web Server for display at the Browser. If the report needs more than one HTML page, or if it needs multiple files to construct a single page, then the other pages will be written to a temporary directory that is referenced by the page that is returned to the Broker. In this way, the report that is produced by a single query can be arbitrarily complex. Note that the SAS/IntrNet CGI-Broker must run on the same platform as the Web Server. If that is IBM’s HTTP server, then the CGI-Broker runs on OS/390. If it is a Windows/NT server, it runs on NT. But whether the Server runs on OS/390 or on a different platform, the broker can still connect to the SAS Application Server running on OS/390 by way of the Socket access method.

In addition to the CGI-Broker, SAS/IntrNet provides htmSQL access from the Web Browser to a SAS/SHARE® server, and JAVA interfaces that allow JAVA applications to access a SAS/SHARE Server or a SAS Session via SAS/CONNECT®.

Combining SAS/IntrNet Web Applications with Batch processing

In one sense, the function of the SAS application program is limited only by your imagination, but for practical purposes, it must be able to run in a few seconds. If it takes too long the user at the Web Browser will become convinced that "www" stands not for World Wide Web, but really means Wait, Wait, Wait. But some of the most interesting queries take more than a few seconds. One possible solution to this problem is to have the SAS Application Server submit a batch job rather than trying to do all the work directly in the server. This batch job can be customized according to the user request, and it can use the e-mail access method to notify the user when it completes. Thus the user can continue with other work without having to worry about when the job finishes.

Getting Started with Web Enablement

Like most things, developing Web applications requires learning a few tricks, and then gradually adding more and more complexity. The key is to start small, and build from there. And, like most things, you will encounter a few problems. For instance, all the common workstations use ASCII, and OS/390 produces data in EBCDIC. Converting between these encodings is not difficult, but there are a couple of rough spots. Use ODS to do the conversion with TRANTAB=ASCII. Pay particular attention to the end-of-line character, and to the special characters like the dollar sign and the curly braces. If you start small, it is much easier to tell what is going wrong, and you won't waste nearly as much time.   Also, pay special attention to the examples in the SAS Reference material, and look on the tech support Web site.

Web enablement: Summary

SAS applications that were written with no thought of the Web can be easily modified to produce Web-ready HTML output using ODS. These HTML files can be sent to the Web Server system via the FTP access method, or they can be written to the OS/390 file system using either traditional MVS data sets or HFS files. The HFS file system offers many advantages, and the SAS System fully supports HFS files as sequential data sets. The SAS/IntrNet product takes this one step further, and allows a user to make a dynamic request for a SAS program to be run, and to customize the request. The requested program can be run immediately and the results sent directly back to the user, or the request can be deferred and run in a batch job, in which case the user can be notified via e-mail.

PLATFORM INTEROPERABILITY

Another aspect of the heterogeneous nature of today's computing systems is the need for programs to communicate between different platforms. Constantly transferring data back and forth between platforms is not only time consuming, it is error-prone. We need ways to access data directly. With improved access to data on other platforms, you have more freedom to decide where the data should be stored, and where it should be processed.

Version 8 of the SAS System both enhances old ways of operating with other platforms, and also provides new ways of operating with those platforms.

SAS/CONNECT: Spawner.

The spawner simplifies the creation of SAS sessions on the OS/390 server. It supports TCP/IP socket inheritance which simplifies its use when a TCP/IP firewall is involved. The use the spawner on OS/390 requires:

- OS/390 Version 2 Release 6 (or later)
- APF-authorization
- Authorization to use some special Unix System Services.

The spawner runs as an ordinary MVS started task, and although it is not a Unix program, it operates by using Unix System Services to create new copies of SAS that can be dedicated to single user. For more information about the spawner, see *Installation Instructions and System Manager's Guide for the Version 8 (TS M0) of the SAS System under OS/390*.

SAS/CONNECT: Asynchronous remote submits.

SAS/CONNECT now provides the ability for a remote system to submit a request and then continue processing, perhaps submitting other requests to be run in parallel. This facility provides the remote system with the ability to wait for completion of some or all of the currently submitted requests and then to resume processing.

Data Encryption using SAS/SECURE™. SAS/SECURE allows users of SAS/CONNECT and SAS/SHARE to have access to the RSA BSAFE Crypto-C encryption tools. These tools provide a choice of encryption among RSA RC2, RSA RC4, DES, and triple-DES. Due to export restrictions, the stronger forms of encryption are not available outside of North America at this time. The hardware encryption facilities available on S/390 machines are not exploited by this support. For more information on data encryption, refer to Appendix "Encryption Services" in *SAS/CONNECT User's Guide, Version 8* (Order #CW57187).

SAS/CONNECT Cross Environment Data Access (CEDA)

CEDA allows a SAS System running on one platform to read a SAS Data File from another platform without requiring it to be explicitly converted in format. SAS can gain access to the foreign platform's files either directly via network access, or indirectly using FTP. For example, suppose you have run SAS on a UNIX server to create a SAS Data file, and would like to have an OS/390 batch job read that file. In Version 6 you would have to convert the file to a transport format, transfer the file to OS/390, load it into a SAS data library, and finally run the job to use it. In Version 8, with CEDA, you can simply FTP the file to the OS/390 HFS system, and read the file directly with OS/390. When reading a file that was created on another system, the SAS System handles conversion of text data from EBCDIC to ASCII, and it converts numeric data from the other host's floating point representation to the IBM hexadecimal float. Of course, this data conversion takes time, and CEDA only provides read access, so you may still prefer to transport the file to OS/390 for some applications. CEDA also provides other platforms with the ability to directly access SAS data files stored in OS/390 HFS, but not SAS data files stored in traditional MVS SAS data libraries.

HFS provides the ability to network mount a remote file system on OS/390, and HFS files can be mounted on the file systems of other platforms. With this kind of network connectivity enabled, Version 8 of the SAS System can read the file directly from the remote system, and CEDA will still handle all the data conversions.

Integration Technologies: Integrated Object Methods (IOM)

IOM provides a different way to access data on network connected computers, and does not require that you run SAS on the client workstation. IOM allows an application that runs on a PC workstation to access mainframe SAS data using network communication to a SAS server running on OS/390. IOM provides access to SAS data objects such as SAS data sets, as well as providing the ability to submit programs to the SAS object server and retrieve the LOG, LIST, and OUTPUT files. Enterprise Guide is an example of the power of the IOM concept, but you can write your own applications using a language such as Visual Basic. For example, a Visual Basic program running on a PC can use IOM objects to produce a listing from an OS/390 Catalog.

New features for traditional SAS Applications

So far, this paper has focused on Web enablement and interoperation with other platforms. Version 8 of the SAS System also includes new features for traditional applications. This section groups them into the following categories

- Support for 3270 Display Terminals
- Access to Data
- Performance issues
- Help and other Documentation

Support for 3270 Display terminals

Most customers now run 3270 emulators on PC workstations instead of using real 3270, and these emulators provide extended graphic capabilities that were rarely found in real 3270 terminals. Version 8 has significantly enhanced SAS support for 3270 graphics to enable customers to take advantage of these emulators, and Version 8 makes extensive use of these new graphic elements. One example of this new 3270 function is the Explorer mode. Explorer mode is a point and click interface into SAS that will be especially useful to SAS users who are familiar with that interface on PC or Unix platforms.

The user interface of some SAS products that were developed for 6.10, 6.11, and 6.12 use graphic elements that cannot be supported on 3270 devices. In most cases, these applications were never available for SAS on OS/390, but there is one notable exception: SAS/AF® BUILD FRAME. The SAS/AF product redesigned the user interface for this application to fit more naturally into a workstation environment, and it requires graphic elements that are not supported in OS/390. As a result, BUILD FRAME is not supported in Version 8 of the SAS system for OS/390. For users of this application who would like to move to Version 8, there are two options:

- Maintain a copy of Version 6 to use for BUILD FRAME operations, and use the CPORT/CIMPORT functions to move those new frame definitions from Version 6 to Version 8. This has the drawbacks that you cannot use any of the new graphic elements that are supported in Version 8 and you must keep an old copy of Version 6 SAS around.
- Obtain a copy of Version 8 of the SAS system to run on a PC or a UNIX workstation. Run the BUILD FRAME application on that system, and use the CPORT/CIMPORT functions to move the new frame definitions to OS/390. This allows the use of the new graphic elements, but does not prevent you from trying to use elements that are not supported for the 3270, and, finally, the exact presentation on the 3270 will be slightly different from that of the PC or UNIX workstation.

One exciting new feature for 3270 users is the syntax-sensitive program editor. This editor, available in Release 8.1 of the SAS System, provides a quick analysis of the syntax of a SAS program, and displays the different elements of the program in different colors. This editor helps the programmer detect mistakes such as missing quotes and semi-colons that can be very difficult to find. To enable this editor when using SAS in TSO:

- Specify the COLORPGM option at invocation of the SAS System.
- Issue the command SYNCOLOR ON from the command line after SAS comes up.

Long term, we do not believe that the 3270 is a viable environment for high-end interactive use. However, we also know that many of our customers have invested heavily in applications that depend on 3270 support in SAS, and we will continue to maintain that support. At the same time, we have no plans to significantly enhance the 3270 display functions beyond what is available in Version 8. Interoperation with desktop workstations using products such as Enterprise Guide™ and WebAF™ will provide the high-end interactive displays in the future, and we will look to enhance offerings in these areas.

ACCESS TO DATA

Version 8 of the SAS System provides significant enhancement to the ways you can access data. There is room in this paper to describe only a few, so many features of interest to OS/390 users are not mentioned here.

Multi-volume SAS Data Libraries

Version 8 significantly enhances the support for multi-volume SAS data libraries:

- The rules for allocating multi-volume libraries are less restrictive. Version 8 supports allocation of the library on a single volume so it is eligible to extend to additional volumes dynamically.
- Use of the Guaranteed Space attribute is no longer required.
- The Version 6 style of pre-allocating the space continues to work in Version 8.
- Multi-volume libraries can be migrated and recalled using DFhsm.
- The SAS System tolerates the fixup actions taken by some 3rd party vendor products that intercept X37 abends and provide additional space in ways other than the standard OS/390 end of volume support.

DB2 Views

In Version 6, most uses of DB2 data base required use of a DB2 View. In Version 8, the View is still supported, but it is no longer required. For example, to access DB2 from SAS, you might use a LIBNAME statement such as:

```
LIBNAME dblib db2 authid=myuser
in='database inventory';
```

Then in the DATA step, simply say:

```
SET dblib.tablename;
```

The DATA step uses the same variable names as the DB2 columns.

In Memory Files

Version 8.1 of the SAS system includes improved support for maintaining an entire SAS data file in memory. This can be a significant performance improvement when the same file is used many, many times in the same SAS session, such as a SHARE server. To specify a file as an In Memory File, use the

SASFILE statement. Options on this statement provide two ways of using the In Memory File:

- Allocate as many file buffers as needed to pre-stage the entire file into memory.
- Do not pre-stage the file, but once memory has been allocated for a file buffer, maintain that buffer image in memory for the duration of the SAS session.

Before using this feature on a large file, you should consult your OS/390 systems staff to be sure the system has enough real storage to support this usage, and that the OS/390 resource scheduler will allocate enough real storage to your SAS session. Otherwise, you may find that the SAS job actually takes longer to run due to a dramatic increase in operating system paging.

Interoperation between Version 6 and Version 8

A key part of the design of Version 8 is to allow interoperation with previous releases of SAS. In order to support mixed case variable names, long character strings, ODS templates, and other new features, there have been several significant changes in the format of the SAS Catalog, and the SAS Data Libraries. All of these changes were introduced in Version 7, and for the most part, there are no problems allowing Version 7 and Version 8 to share the same SAS Data Libraries and Catalogs. The real issues have to do with sharing SAS data between Version 6 and Version 7 or Version 8. Steve Beatrous and James Holman have written an excellent paper on this topic for SUGI 24, and this paper is available at the SAS Institute Web site at:

http://ftp.sas.com/techsup/
download/v7papers/V6andV7.pdf

Some of the key points mentioned in this paper:

- In general, Version 6 does not know anything about Version 8 files, and cannot deal with them.
- Version 8 can read, but cannot write Version 5 SAS data libraries.
- Version 8 can neither read nor write Version 6.06 data libraries.
- Version 8 can read and write Version 6 data libraries.
- Version 8 can read, but cannot write Version 6 catalogs and SQL views.
- The library concatenation feature (new) can simplify the process of migrating from Version 6 to the new format for Data Library and Catalog.
- In order to simplify the migration from Version 6 to Version 8, use the VALIDVARNAME=V6 option to prevent Version 8 from using mixed case variable names.
- SAS/SHARE, when using TCP/IP, can connect Version 6 clients to Version 8 servers, and Version 8 clients can connect to Version 6 Servers. However, when using the XMS access method, the client and server level must match exactly. You should not even attempt to mix different maintenance levels of client and server when using XMS, much less attempt to cross release or version boundaries. This restriction essentially requires you to change all of your clients to a new release at the same time you upgrade the server, or to temporarily switch from using XMS to the TCP/IP access method until all clients and servers are running the same level.

Version 8 Performance

Since Version 6, a lot of new code has been added to the SAS. Some features, such as mixed case variable names and long character strings, require the execution of additional code, even when the job doesn't use those features. At the same time, a very large effort has gone into reducing the impact of these changes. In general, when moving work from Version 6 to Version 8 without making any changes to take advantage of new features in Version 8, you should expect to find:

- CPU consumption for some jobs will increase slightly, and will decrease slightly for others. For an entire data center, you will probably see an overall increase of a few percent. One known case that causes a significant increase in CPU consumption in Version 8, when compared to Version 6, is the production of a very large number of very small reports with the use of BY variables. This is an unfortunate result of a fundamental change in how the SAS System handles this situation. But most often, when we have found instances of dramatic increases in CPU consumption between Version 6 and Version 8, the developers have been able to find and fix the problem. We know that our customers are our best source of this kind of information, so if you believe you have a CPU performance problem in Version 8, you may wish to report this as a problem to tech support.

- Version 8 requires significantly more virtual storage than Version 6. Program modules and dynamic storage requirements have both approximately doubled, and the bundles that are loaded into the OS/390 Link Pack area have more than doubled because they include many new modules. All of this increase comes in above-the-line storage. Below the line storage in Version 8 is essentially unchanged from Version 6, although there have been some slight decreases.
- I/O consumption is approximately the same as in Version 8, although individual jobs may show a slight increase or decrease.

MEMSIZE=0, MEMLEAVE OPTIONS

Due to the significant increase in Virtual Storage consumption in Version 8, and based on customer reported problems in Version 6, we have added the MEMLEAVE option, and have changed MEMSIZE processing so that SAS will not attempt to use more storage than is available. During initialization of the SAS system, the available storage in the region is examined, and the amount of storage that SAS will use is set based on the MEMSIZE and MEMLEAVE options. If a specific, non-zero MEMSIZE has been specified, SAS will use that storage, and will insure that at least MEMLEAVE storage is available for use by other tasks in the region. If the region is not large enough to satisfy both the MEMSIZE and the MEMLEAVE request, then the MEMSIZE value is reduced, and a warning is issued. If MEMSIZE=0 has been specified, MEMSIZE is calculated to be the largest possible MEMSIZE value that can still honor the MEMLEAVE requirement. By setting the MEMLEAVE option to the storage required by tasks other than SAS, and setting MEMSIZE=0, any increase in the MVS REGION parameter results in additional storage for SAS.

Documentation and Help

Help from the 3270 is much improved over Version 6, and includes the ability to use HTML style links to move from one part of Help to another. It also has navigation commands to allow you to move back and forth between pages that you have already visited. Version 8 also provides online documentation with online search facilities that can be viewed from a Web Browser. If you need hardcopy documentation, you can still obtain it from SAS Institute, or you may obtain a CD with all the documentation in PDF format and print those parts of the documentation that you need in hardcopy.

SUMMARY

OS/390 has had many exciting changes in recent years, and The SAS System for OS/390 continues to be enhanced to take advantage of these opportunities. Version 8 has provided dramatic new facilities for developing Web based applications, there are new opportunities for operation with other platforms, and many improvements in traditional MVS facilities.
As OS/390 continues to evolve, look for SAS to continue to enhance its offering to take advantage of new facilities.

REFERENCES

The Complete Guide to the SAS Output Delivery System, Version 8 (Order #CW57241*)*

SAS Language Reference: Dictionary, Version 8 (Order #CW57240)

OS/390 V2R8.0 Unix System Service Planning (SC28-1890-08)

OS/390 Version 2 Release 6 Unix System Services Implementation and Customization (SG24-5178-00)

Beatrous, Steve and Holman, James (1999), “Version 6 and Version 7: A Peaceful Co-Existence”, Proceedings of the Twenty-fourth Annual SAS Users Group International Conference

AUTHOR

Dave Crow
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
(919) 677-8000
dave.crow@sas.com