Lessons in Logon Load Balancing

Janet Hutchison

With over four years of SAP experience, Janet joined PricewaterhouseCoopers in 1998 as a Principal Consultant. Janet develops and instructs training classes in advanced ERP Technical Infrastructure topics, and is responsible for courses such as SAP Basis Administration, SAP System Performance, Oracle Database Administration, and Backup and Recovery.

(complete bio appears on page 24)

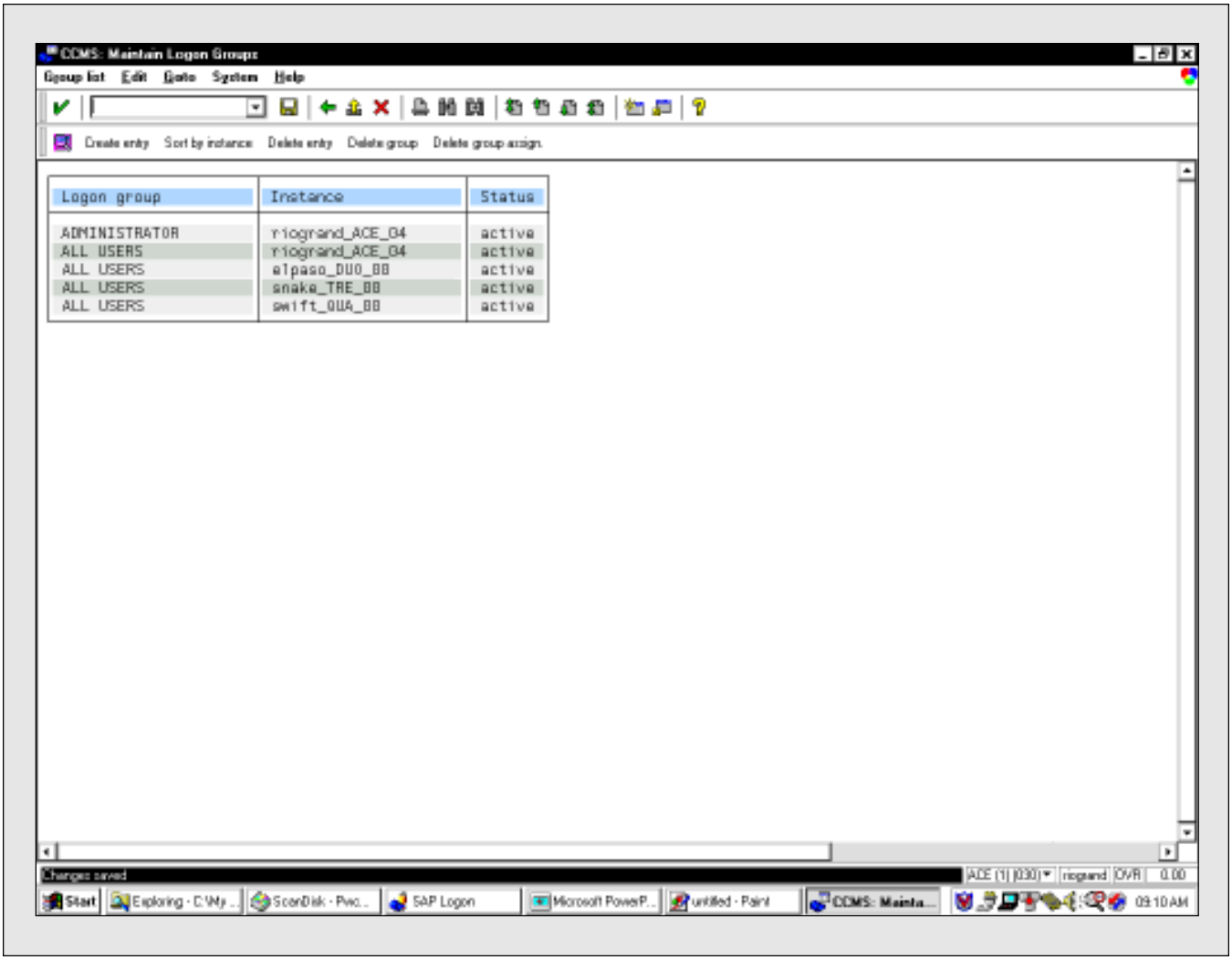
In large SAP R/3 environments that require multiple application instances, you obviously want to make sure that the system workload gets distributed among those processing resources. Otherwise, you defeat the purpose of having multiple application instances.

One simple, albeit not terribly efficient, way to accomplish workload distribution is to assign different users, or groups of users, to a specific set of application instances, then allow users to specify the application instance to which they want to connect. Of course, if that application instance is unavailable, users who specify it will not be able to connect to the system. And if the application instance becomes unavailable in the midst of some user processing, user connections will be lost, and users will not be able to reconnect until the application instance once again becomes available.

True, you can provide users with the ability to switch to an alternate application instance in the event of a failure, but this requires that users understand the concept of application instances and be able to ascertain which application instances are available. It also requires that you provide for multiple application entries in each user's SAPLOGON menu.

Once this is done, there is no guarantee which instance any given user will select for any particular logon session! That is, as soon as more than one application instance is made available to the user, all control is lost as to which application instance the end user selects as the primary, and which instance is used only when the primary application instance is unavailable. I suspect that it is implementations like this that cause Basis administrators to complain that some application instances

Figure 1 Main Screen of Transaction SMLG



are overworked and performing poorly while others remain unused.

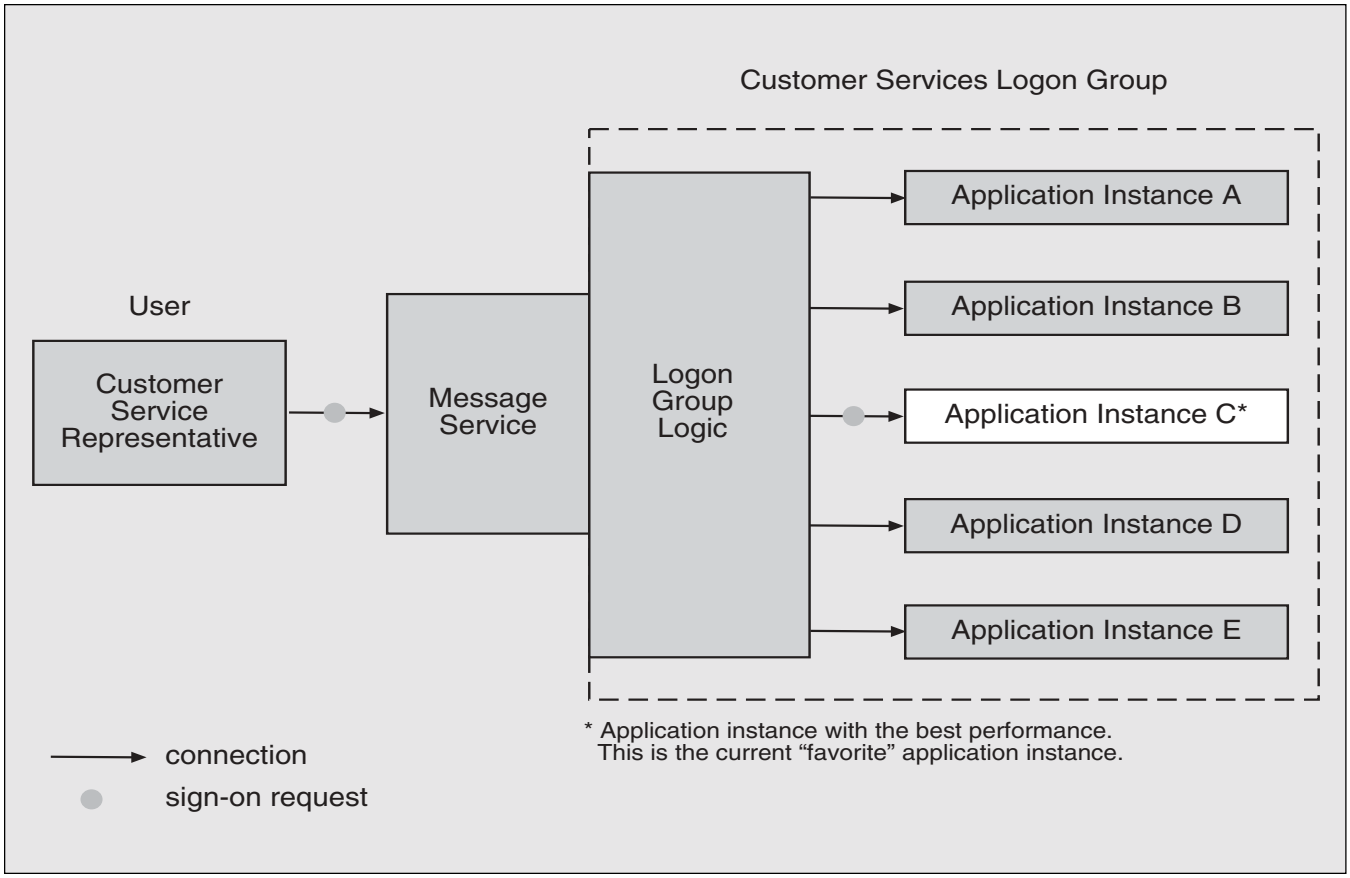
There is a better way! You can achieve intelligent, automated distribution of workload across multiple application instances, with minimal impact to the end users, through logon groups.

In this article, I will explain how logon groups work and how to use them to establish an intelligent logon load balancing strategy that can improve system performance, elevate end-user satisfaction, and help attain High Availability objectives.

Logon Groups — An Overview

A logon group is a logical construct that groups application instances together within an SAP R/3 system. It always contains one or more application instances. **Figure 1** shows you the main screen of transaction SMLG, where you can see that in this system, two logon groups have been created: ADMINISTRATOR and ALL USERS. (Logon group names are case-sensitive.) The ADMINISTRATOR logon group has only one application instance entry: riogrand_ACE_04. The ALL USERS logon group has four application instance entries:

Figure 2 Logon Groups Provide Access Paths to Application Instances



riogrand_ACE_04, elpaso_DUO_00, snake_TRE_00, and swift_QUA_00. You define a logon group *within* an SAP R/3 system; logon groups do not cross SAP R/3 systems.

In this article, I will explain how logon groups work and how to use them to establish an intelligent logon load balancing strategy that can improve system performance, elevate end-user satisfaction, and help attain High Availability objectives.

Logon groups basically provide access paths to application instances. You can think of a logon group as a traffic cop. When the system receives a logon request, the logon group logic identifies the application instance with the best performance, and directs the logon request to that application instance, as shown in **Figure 2**.

Logon group logic and functionality are delivered by the message service. The data that uniquely identifies the message service includes the TCP/IP address of the message server host, the host name, and the SAP SID two-digit number of the instance where the message service runs.

To pass along this information you turn to the SAPLOGON, SAPLGPAD, or Session Manager

Figure 3 The SAP Logon Screen

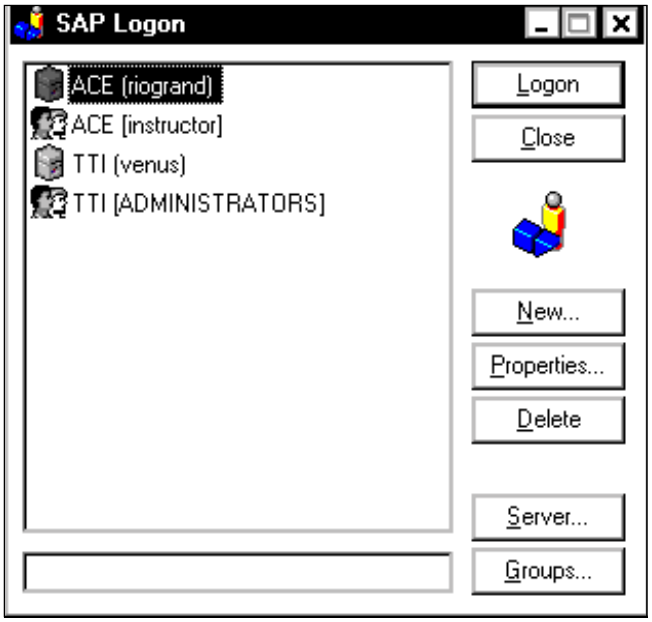
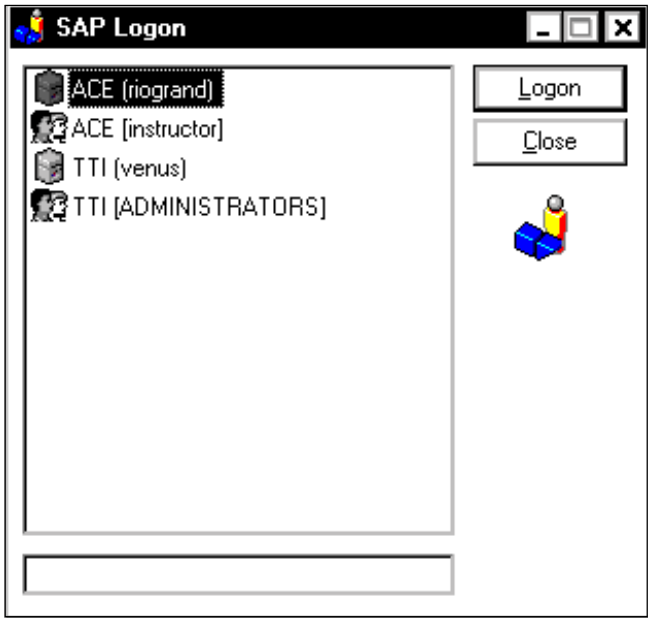


Figure 4 SAPLGPAD.EXE (Release 4.5)

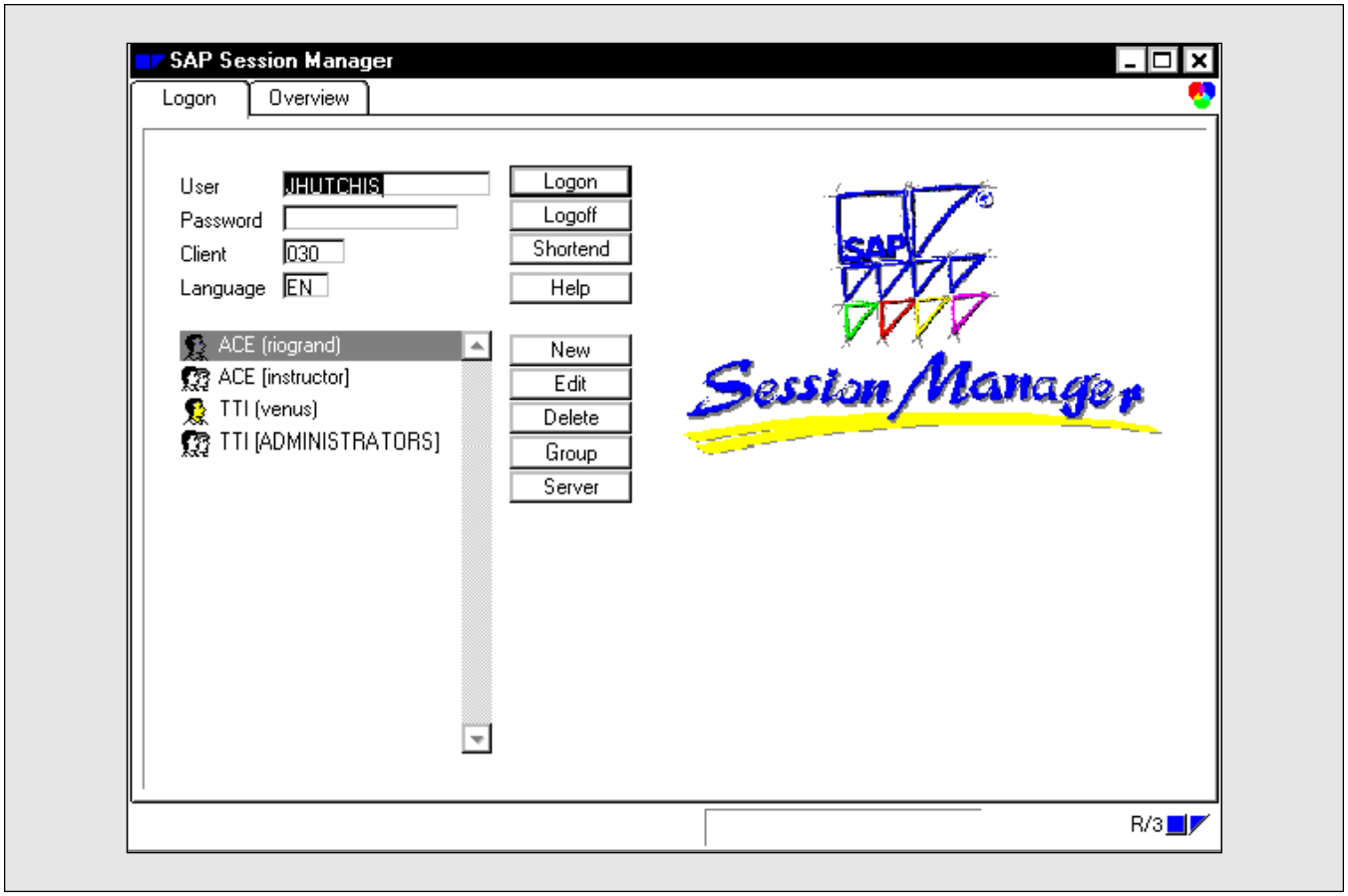


executables. So if logon groups and logon load balancing will be used in the SAP R/3 system, the presentation layer *must* be configured with either SAPLOGON, SAPLGPAD, or the Session Manager executable, GNSMX.EXE. In **Figure 3**, you can see the SAPLOGON window with four application instances in the menu list: riogrand on server ACE, instructor on server ACE, venus on server TTI, and ADMINISTRATORS on server TTI. In **Figure 4**, SAPLGPAD, you also see multiple application instances, but this time the interface that is presented to the user does not have the New, Properties, Delete, Server, and Groups buttons. It is designed this way to prevent the user from making any changes to the SAP GUI access window. (Notice that in both logon screens, a user can log on to application instances or logon groups. This makes life easier for administrators who need to support both types of frontend access.) **Figure 5** shows the Session Manager executable.

Session Manager is the newest SAP GUI access window. Session Manager gives the user the ability to establish, monitor, and navigate among multiple sessions to multiple SAP systems from within one easy-to-use window. Session Manager also helps navigation by displaying menu path selections in horizontally tiled subwindows. This helps in that the user has visibility to all menu subselection options without having to dropdown each menu entry. Using Session Manager will take some "getting used to" for veteran SAPLOGON users. However, at one quick glance, it is easy to see how it will appeal to the non-technical business user. And, for Basis administrators, technically it works exactly the same as SAPLOGON.

What makes SAPLOGON, SAPLGPAD, and Session Manager appealing from a usability standpoint is that users only need to point and click. What makes them appealing from an administrative standpoint is that all three of these options use the SAPLOGON.INI control file. As administrators add entries to the menu list via SAP GUI functionality, the SAPLOGON.INI file is automatically updated. Users whose desktops have been configured with SAPLGPAD or Session Manager can use the Add or New buttons to put new entries into the SAPLOGON system list. When the list is updated through either of the update-enabled frontends — SAPLOGON.EXE or Session Manager — the SAPLOGON.INI file is

Figure 5 *The Session Manager Executable*



updated automatically, and is kept syntactically correct.

It is more challenging to try to modify SAPLOGON.INI manually, and it is more likely that you will make syntax and content errors that will prevent access. Having SAP perform the updates without manual intervention is a marked improvement over SAPICON.EXE, which requires users to enter the system number and the server name, or the server TCP/IP address.

Once a user's desktop is configured with logon group(s), as you see in Figures 3, 4, and 5, the user selects a target group, and the logon group logic automatically directs the logon request to an application instance. But how does the logon group logic know to *which* application instance the logon request should be directed? SAP plays favorites!

SAP Plays Favorites!

At any given point in time, a logon group knows its "favorite" application instance. When a user logon request is received, the logon group will direct that request to the application instance that is currently identified as the "favorite" — the one with the best overall performance. The key components of the "favorite instance" logic are the ABAP report RSRZLLG0[1] and the data that is stored in the alert memory area of the message service.

[1] This is an SAP-delivered AUTOABAP program that executes periodically in the background. ABAP report RSRZLLG0 is run under the control of ABAP report SAPMSSY6. ABAP SAPMSSY6 is triggered to run at a frequency as defined by instance parameter rdisp/autoabaptime. The default value for this parameter is 300 seconds.

ABAP RSRZLLG0 performs the following logic:

- Confirms the availability of all application instances
- Creates an internal list of all application instances, including the host address and dispatcher service port
- Queries each application instance for the number of connected users
- Queries each application instance for its "answer time" (average dialog response time) and number of transaction step (dialog step) frequency
- Calculates a value for the "overall quality" of each application instance
- Generates a list of all available application instances in each logon group sorted by the calculated "overall quality"
- Deletes all entries from the list, except the top or favorite
- Writes the re-calculated, updated list of favorites to the message service memory

For each application instance, ABAP RSRZLLG0 refers to two threshold settings — *average dialog response time* and *connected users* — together, these are referred to as the *logon threshold profile*. The favorite instance logic reduces the quality rating for any instance where the actual *connected users* and/or *average dialog response time* value approaches or exceeds these threshold settings. These values will change as users sign on and off, and as the application instance's performance varies with the work that it is processing. Higher values for "overall performance quality" are better in that higher values indicate more available capacity to handle additional users and process additional work.

Logon Threshold Profile

Average dialog response time is a key performance measure. A high or increasing average dialog response time is a clear indication that an application instance has reduced capacity to successfully process additional work. Under these conditions, additional work needs to be directed to another application instance with a lower dialog response time.

Connected users is another key performance measure. Here again, a high or increasing number of connected users can be an indication that an application instance has reduced capacity to successfully process additional work. Just as with average dialog response time, additional work should be directed to other application instances with fewer connected users.

Logon Groups Improve Accessibility, Not Availability

Logon groups establish a logical layer between the end user and the application instances, such that end users do not have to know or understand the physical architecture of the application instances in the SAP R/3 system. If one or more application instances in the system are unavailable, the logon group will identify an available instance and will direct the user logon to it.

Once logged on, if an application instance fails, users with active sessions or connections to that instance will lose those connections; there is no automatic failover or reconnect mechanism available at this time. However, with the use of logon groups, an end user can immediately issue another logon request to that same logon group. The logon group will be able to establish the user's connection to an available application instance. The user does not have to know the other possible application instances. The end user simply re-selects the original logon group and that logon group will identify and select an available application instance. If at least one application instance in the logon group is available, the user should be able to quickly establish a new connection and resume working. This eliminates users having to

step through a list of application instances until an available one is identified.

✔ Tip

End-user training should include the following behavior when logon groups are part of the SAP R/3 implementation:

> *"If your current SAP R/3 session is lost, your first step should be to try to re-logon to the system. If you cannot reconnect on subsequent attempts, then you should follow the appropriate notification procedures."*

So, the loss or failure of a single application instance should cause only a momentary interruption for the users connected to that instance. Users logging on for the first time should not have any awareness that any application instances are unavailable.

I want to make it clear that the use of logon groups and logon load balancing does not in any way improve *actual* system availability. Logon groups cannot affect the frequency or severity of hardware, software, network, or other problems that cause all or part of an SAP R/3 system to become unavailable. Logon groups do, however, increase *perceived* system availability as experienced by end users, and logon groups do contribute to the attainment of system availability objectives. I firmly recommend the use of logon groups in environments with High Availability requirements.

Logon Groups Will Never Stand in the Way of a Logon Request

Logon groups influence the direction of logon requests to application instances. They will never prevent logon requests from being completed. As described earlier, as the threshold values are approached or exceeded, the overall performance quality of the application instance will be decreased. This affects the relative position of the application instance in the comparison rating that determines the favorite instance.

When a logon request is received, the message service references the favorite table stored in its memory and directs the logon request to the application instance identified as the favorite.

In addition, the message server will trigger the re-calculation of favorite instances after every fifth logon request. This prevents one or several application instances from becoming severely overloaded in any five-minute period between time-triggered executions of RSRZLLG0.

Be advised, however, that if a system is configured for automatic logoff of idle sessions, any session that has been automatically logged off by this mechanism will not go through the logon group logic when the user requests a reconnect. A session that was logged off due to inactivity will be reconnected to the application instance from which it was disconnected. This could potentially impact the workload distribution and performance in the system. Still, automatic logoff of inactive sessions is a good policy — idle sessions can tie up some system resources, memory in particular. When the system is configured for automatic logoff of idle sessions, the system can make the system resources available to other, active users. Also, it is a generally accepted security principle that idle sessions not be left connected for long periods of time.

Adding New Logon Groups — What Every Administrator Needs to Know

Above and beyond the fact that logon groups are created and maintained through CCMS,[2] there are six things that every administrator needs to know about in order to add logon groups to an R/3 environment:

1. There is a system-enforced relationship between application instances and logon groups.

[2] The transaction code is SMLG. Note that SAP has removed the transaction code window as of SAP GUI Release 4.6.

2. Logon threshold values of *connected users* and *average dialog response time* can be used to influence the favorite instance calculation and the distribution of workload.

3. Updates that are required to the SAPLOGON.INI file are automatically made via the Group Select-Add option.

4. SAP delivers a default logon group named "SPACE."

5. End-user access to logon groups and application instances can be controlled through central management and controlled distribution of the SAPLOGON.INI file. System administrators often choose to deploy SAPLGPAD.EXE to the end users so the users do not have the ability to modify their SAPLOGON system menu list.

6. New logon groups and new assignments of application instances to logon groups take effect only after the next execution of RSRZLLG0 writes those changes to the message server memory.

The following sections provide details on each of these six issues.

The System-Enforced Relationship Between Application Instances and Logon Groups

There are three things you must understand about application instances as they relate to logon groups:

1. An application instance can be a member in none, one, or many logon groups.

2. A logon group must contain at least one, but may contain many, and even every, application instance in an SAP R/3 system.

3. An application instance must be described by one and only one logon threshold profile, which specifies threshold settings for the *average dialog response time* and maximum number of *connected users*. (There is a one-to-one correspondence between an application instance and a threshold profile).

Logon Threshold Values Can Be Used to Influence the Favorite Instance Calculation

The SAP-delivered default sets the average dialog response time and maximum number of connected user thresholds to zero. If you leave the threshold values set to zero, the favorite calculation is based only on the *actual* number of connected users and the *actual* average dialog response time. These default settings are shown in **Figure 6**, where you see the extended display format of the SMLG main window superimposed on an ABAP Dictionary (SE11) display of the contents of table RZLLITAB. Table RZLLITAB is where the data about logon groups and their threshold profiles is stored.[3] Note that the SMLG screen displays a value of zero as a blank.

In **Figure 7**, you can see that I overrode the default values of zero for these logon groups.

While I chose to assign the same logon threshold profile settings to both the application instances pictured in Figure 7, I did not have to. Different application instances in the same logon group can have different logon threshold profile settings. However, each application instance has one and only one set of threshold values, regardless of how many logon groups to which it is defined. If the logon threshold values are changed for the application instance, these values are propagated to every logon group entry for that application instance. Changing the logon threshold profile values for one application instance will not impact the settings for any other application instances. Once set and saved (via transaction SMLG), these threshold values are stored in table RZLLITAB.

By overriding the default settings, you exert influence in the favorite calculation algorithm: If the actual connected users and/or actual average dialog response time approach or exceed the threshold, then the favorite instance algorithm penalizes that instance, assigning it a lower overall quality value.

[3] The logon threshold profile values are also maintained via the Logon Group screens and functionality in CCMS. The threshold profile values are stored in table RZLLITAB in the SAP database.

Figure 6 Logon Threshold Profile with Default Settings

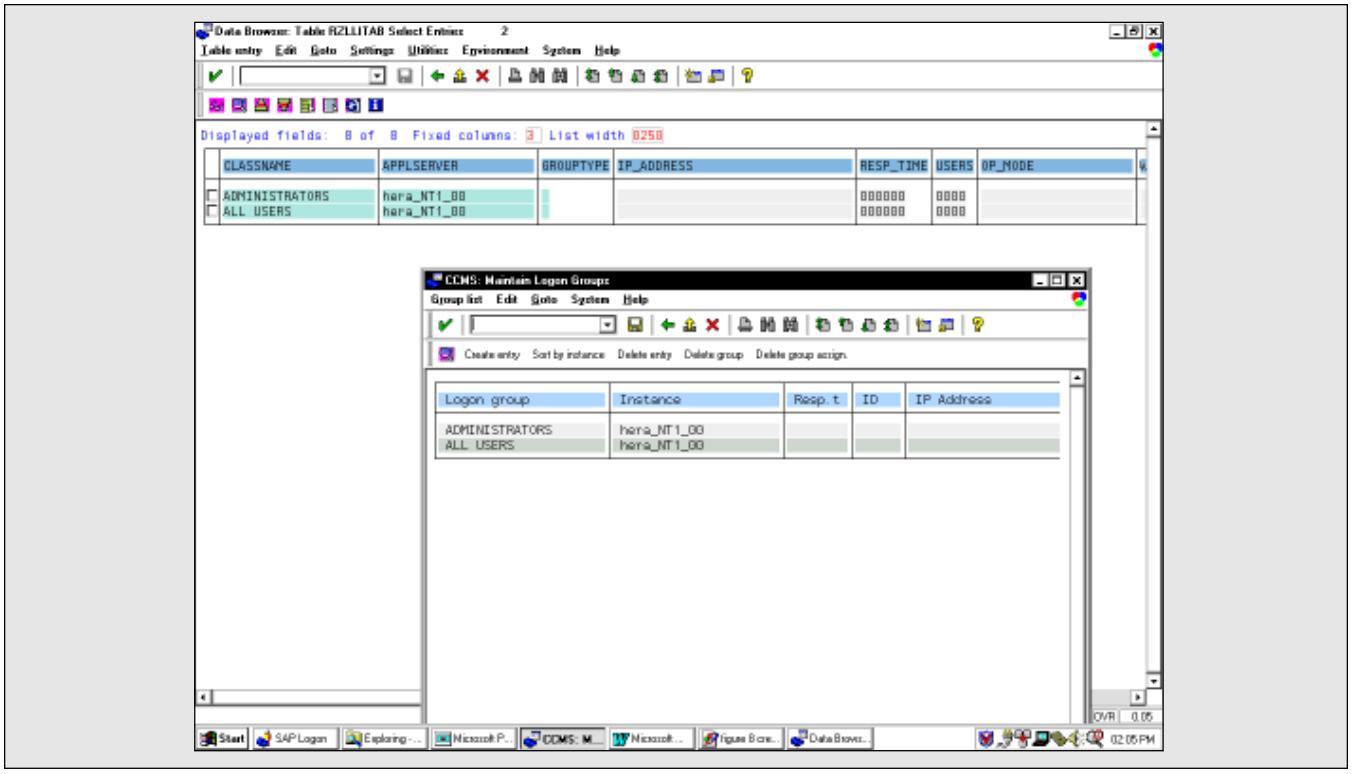
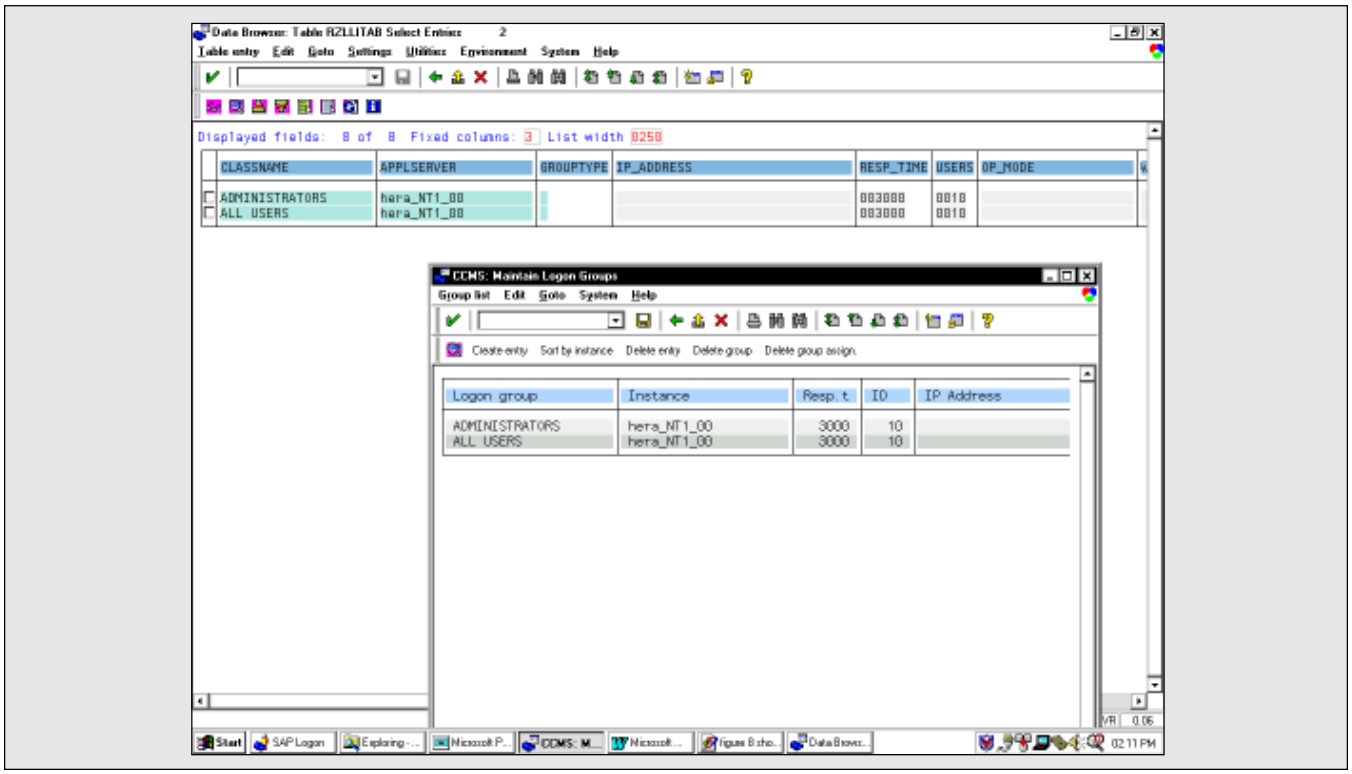


Figure 7 Logon Threshold Profile with Adjusted Settings



Say, for example, that in my system I have five application servers. Two of them are very large processors with lots of processing power. The other three are smaller servers with modest amounts of processing power. I set the logon threshold profile values at 100 users and 2,000 milliseconds for the logon groups using the smaller servers, and set the threshold profiles at 250 users and 2,000 milliseconds for the logon groups using the larger servers. As the number of connected users increases, the logon groups using the smaller servers will be moved out of the favorite instance list sooner than will the logon groups using the larger machines. This is a reasonable approach because the larger servers, just by their configuration, have much more available processing capacity.

Assigning these thresholds non-zero values will obviously influence the calculation that is used by the logon group logic to distribute logon requests among application instances.

Let's shift our attention now to the calculations that are performed, based on the values of these threshold settings. As delivered, ABAP RSRZLLG0 assigns a weight of 1 to *connected users*, and a weight of 5 to the *average dialog response time*. The reasoning behind assigning greater consideration to the *average dialog response time* is that it is possible for an application instance to have a large number of connected users yet still be delivering excellent service as demonstrated by a low average dialog response time. By similar logic, it would be possible to have just a few connected users on an application instance where the average dialog response time is very high. Thus, from the perspective of performance and deciding where to direct subsequent logon requests, it is reasonable and desirable to give greater weight to the application instance performance as indicated by average dialog response time when calculating overall quality performance.

You can adjust the relative weights of these performance measures by modifying ABAP RSRZLLG0 for situations where *connected users* is a more strongly correlated measure of performance and available capacity than *average dialog response time*. Be advised, however, that this is not the kind of change you make on the fly. It requires modification of an SAP-delivered object. Such a change should be made and tested in the development system, and transported through the TMS landscape into INT, CON, QA, PRD systems. You also need to be aware that this change will be in effect for all future favorite instance calculations. This adjusted weighting cannot be changed selectively for only some instances or for some logon groups. So, it only makes sense to adjust the threshold weightings in favor of *connected users* over *average dialog response time* if all or part of a system workload is consistent and predictable based on the number of connected users.

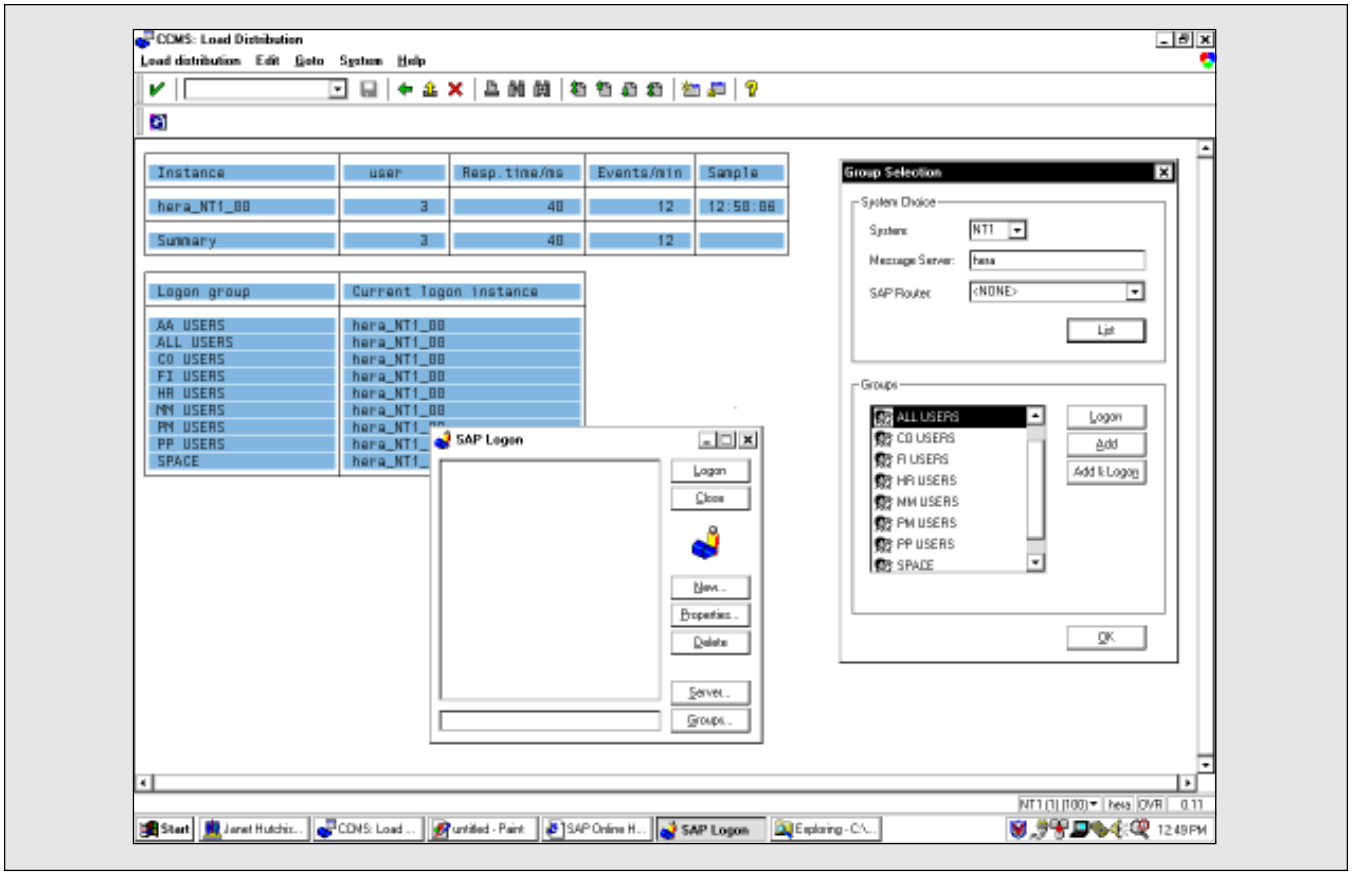
✔ Tip

One particularly creative administrator devised this approach to achieve a more even workload distribution in a large SAP system. The administrator set the *connected users* threshold to 10 on all the available application instances. The result of this action is that as users begin to sign on, the first 10 users will be connected to the current favorite instance. Then, as each instance acquires its first 10 users, the favorite instance calculation will demote these instances in the favorites list because they have reached that threshold value. Another instance will be promoted to favorite and will acquire its first 10 connected users. This instance will then be demoted by the favorite calculation, and the process will continue until all the instances have acquired the first 10 connected users. The *average dialog response time* threshold will then become the influencing factor in the favorites calculation! A very creative and effective approach to ensuring that all available application instances participate in processing the workload!!

Required Updates to the SAPLOGON.INI File Are Made Automatically via the Group Select Option

Creating a logon group or application instance in CCMS does *not* make it available to users. Making

Figure 8 SAP GUI Group Select Function Retrieves Available Logon Groups



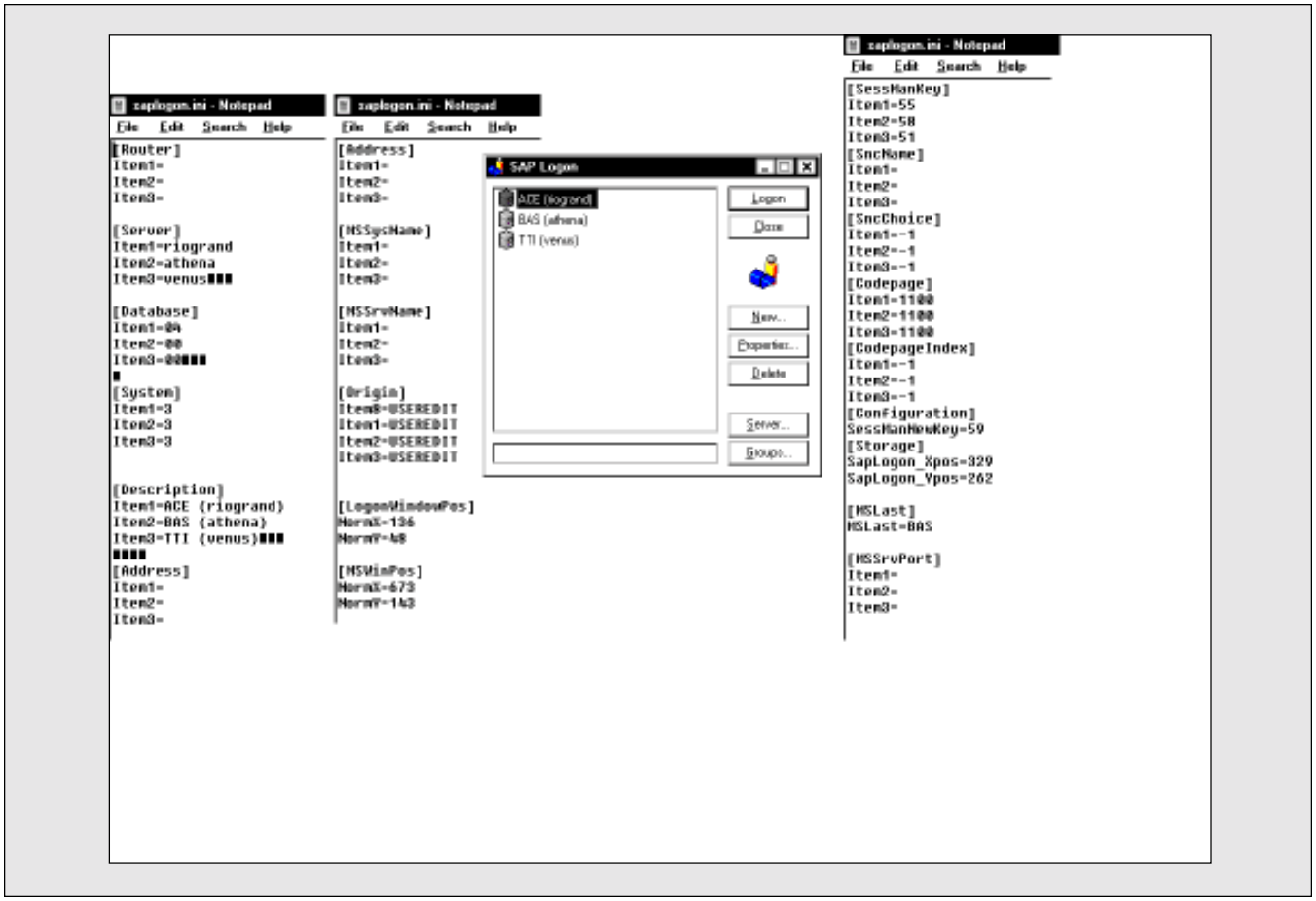
these entries or access points available to end users requires updates to the presentation layer, specifically the SAPLOGON.INI file. It is the contents of the SAPLOGON.INI file that determines what entries are displayed in a user's SAPLOGON menu list of available application instances and logon groups (access points). It is also where the SAP GUI gets the technical information (the SAP system number, the host name or TCP/IP address of the server, and, for logon groups, the identification of the server where the message service resides) needed to make the connection.

New application instances and logon groups, therefore, both require updates to the SAPLOGON.INI file in order to make them available to end users. Don't worry. You don't have to make these entries manually! The Group Select option will make the updates for you. When you click on the "Groupsel" button,[4] the Group Select option establishes a connection to the message service and retrieves the list of all available logon groups in that system. You can select an entry from this list, and then click on the Add button to add that logon group to your SAPLOGON menu list. This is shown in **Figure 8**, where I have superimposed the SAPLOGON and Group Selection windows on top of the SMLG Load Distribution screen.

When the Add button is selected, the SAPLOGON.INI file is *automatically* updated with the appropriate entries. I always add logon groups

[4] As of Release 4.5, the SAP GUI makes the Group Select option available via a button that is labeled "Groups…" Earlier SAP GUI releases make the option available via the "Groupsel" button.

Figure 9 SAPLOGON.INI Application Instance Entries



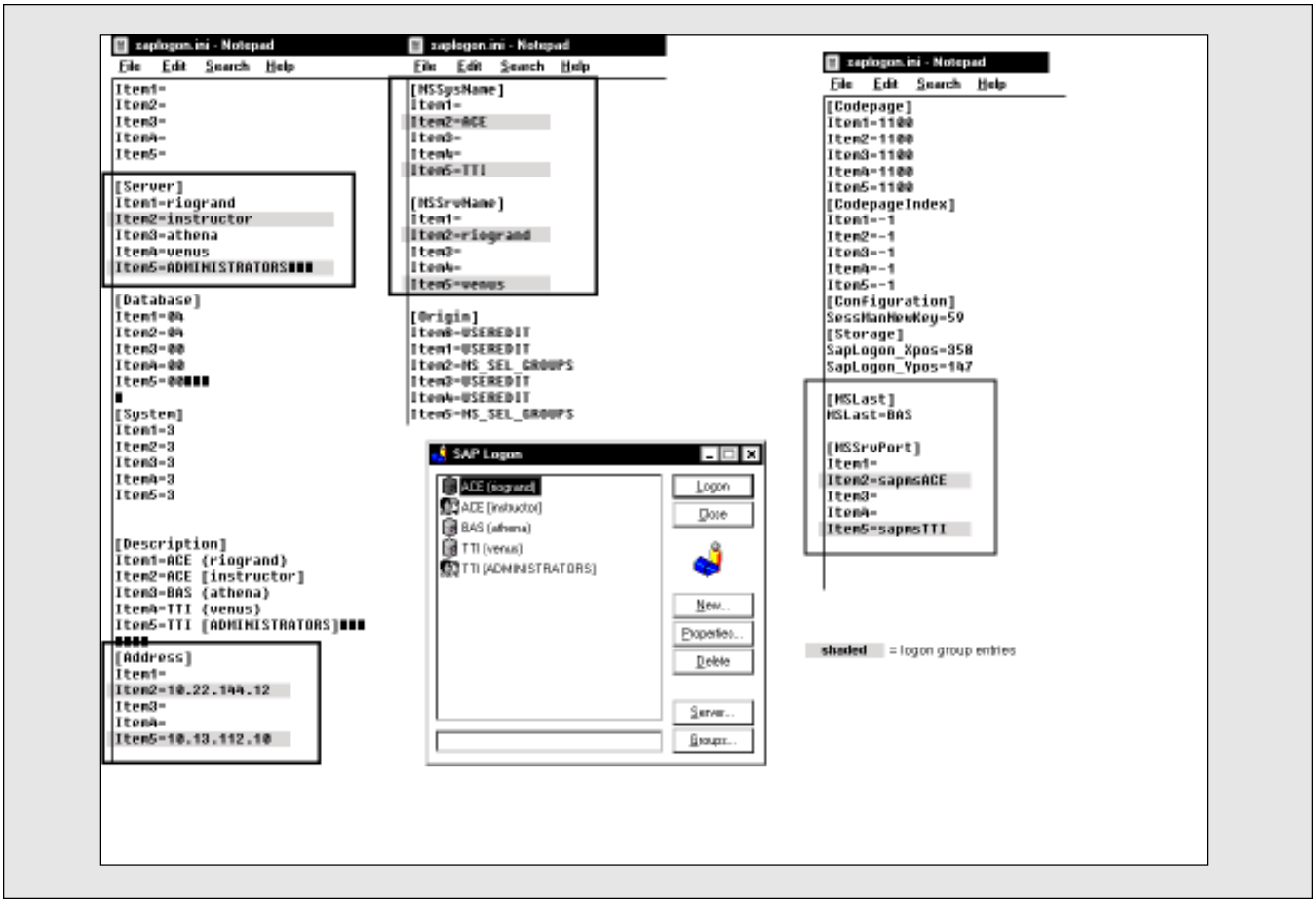
to SAPLOGON via this method. It's the best way for administrators to build the SAPLOGON menus and the SAPLOGON.INI files behind them. The alternative is to manually edit the SAPLOGON.INI file, which is more difficult and introduces the possibility of making a syntax error or skipping a critical entry in the file.

If, for some reason, you do need to manually edit the file, you need to be aware of the following:

- For both application instance and logon group entries, the Description section contains the descriptive text that is displayed in the SAPLOGON or Session Manager list of access points.
- For both types of entries, the Database section of the file contains the SAP system number of the database instance.
- Both types of entries require an entry in the Server section. For an application instance, the entry needs to be the name of the server where the application instance resides. For a logon group entry, this needs to be the name of the logon group.
- For logon groups only, the Address section contains the TCP/IP address of the message service host, the MSSysName section stores the SAP SID, and the MSSrvName stores the host name of the server where the system's message service resides.

A comparative look at **Figure 9** and **Figure 10** will give you a feel for the different way information is entered into the INI file to support logon groups versus application instances. The focus of Figure 9 is

Figure 10 SAPLOGON.INI Application Instance and Logon Group Entries



on application instance entries. Figure 10 shows entries for both application instances and logon groups.

Notice that SAP displays a "server" icon for application instance entries, and a "people" icon for logon groups. The text descriptions of these entries look similar, but here again, there is a difference. While the default text description for both starts with the SAP SID, only an entry for an application instance will show the server name (within the parenthetical text). Logon group entries show the logon group name. (You can modify the text description by using the Properties button.)

For these examples, I selected and added three application instances. Then I added the two logon groups. SAP merged each of these entries into the list in alphabetical order, according to the text description. You can see that the list items were reordered based on the inclusion of the new entries. You can also see in the SAPLOGON.INI file how the entries for Item2 and Item5 are different from the others. Item2 and Item5 are the entries for the two logon groups. In the Server section, Item2 and Item5 contain the names of the logon group, whereas the entries for the other items contain a server name. In the Address section of the file, Item2 and Item5 contain the TCP/IP address of the message service host. The other item entries remain blank in this section. In the MSSysName section, Item2 and Item5 contain the SAP SIDs, and in the MSSrvName section, these items contain the server name of the server where the message service runs.

You might ask, "Why is both the server name and the TCP/IP address needed? Isn't one sufficient for

making the connection?" It is the TCP/IP address that is the key in establishing the user connection. However, SAP uses the data from SAPLOGON.INI to construct the instance name of the central instance where the message service is found. That is why the host name of the message service and the SAP SID are needed.

The instance name is constructed as follows:

<host name>_<SAPSID>_<sap system number>

For example:

athena_BAS_00
hera_NT1_00
riograndACE_04

Just as a point of interest: the SAPLOGON.INI file has a section called Origin. The entries in this section of the file tell you how the entry was added. A value here of "USEREDIT" tells that the entry was added using the Add functionality of SAPLOGON or Session Manager. A value of "MS-SEL_GROUPS" tells that the entry was added as a result of using the Group Select function of SAPLOGON or Session Manager. This allows the administrator to easily identify entries that were made by the SAPGUI functionality as opposed to those made manually.

"SPACE" and Other Logon Group Naming Considerations

The "SPACE" logon group you see at the very bottom of the list of available logon groups shown back in Figure 8 was introduced in Release 4.0. It has four unique characteristics:

- It automatically contains every application instance in the SAP R/3 system.
- It does not appear on the logon group list on the first display screen of the SMLG transaction.
- It does appear when you execute the Group Select option.
- It is created by RSRZLLG0 and the message service, so no system administrator can (permanently) remove or delete this logon group. If it is deleted, the system (RSRZLLG0 and the message service) will re-create it.

This is the one default logon group that SAP delivers. You should *not* create a logon group with this name.

I suspect SAP established this default logon group for use in large systems. If you have a really large system and some application instances are unavailable, you could always use the SPACE logon group and if at least one instance is available, you will be able to connect to the system without searching through the entire list of application instances trying to find one that is available.

It offers some other benefits as well:

- As of Release 4.0, anyone should be able to connect to any SAP system without knowing too much about the system or its application instances. Granted, you still have to know the SAP SID, the system number, and the server on which the message service runs, but this represents a relatively small portion of the data that characterizes an SAP system and its architecture.
- When creating RFC connections to other SAP systems, if the target system is at least Release 4.0, you know that you can always connect via the SPACE logon group. This may not be the best design for your RFC connections, but it's a design that lets you get started with your development and testing.

Other than knowing and understanding about the logon group SPACE, and understanding that a logon group name is limited to 20 characters in length, you will find that there are no naming restrictions.[5] I would, however, urge you to exercise care and planning when choosing names for your logon groups. Departments, job roles, titles, and even physical locations can be transient in these days of business mergers, acquisitions, reorganization, and reengin-

[5] There are some issues with regard to spaces, dashes, and numeric characters in logon group names with SAP GUI Release 4.0. There is an OSS note and a downloadable fix for this problem.

eering. It will be a challenge for both users and Basis administrators if the logon group structure and names have to be revised due to the organization changes.

Personally, I try to keep logon group names as generic as possible. Having witnessed a situation where logon group names were tied to job titles, only to have those job titles reengineered, I'm very wary of any scheme that relates logon names to things that have the potential to change. Logon group redesign, redeployment, and retraining is an arduous affair.

Controlling End-User Access to Application Instances and Logon Groups

The relationship between logon groups and application instances is the only system-enforced relationship in the definition of logon groups. What I mean by a "system-enforced relationship" is that the relationship that defines application instances belonging to logon groups is the only one that is captured and stored in the SAP database. So, by viewing the information stored in the system, you can see what logon groups exist and what application instances belong to them.

There are no *enforced* relationships between logon groups and users. There is no data stored in the SAP database that defines which users belong to which logon groups. You have to document this outside of the SAP system. Nor is there any data relationship or security authorization objects stored *within* an SAP R/3 system that can be used to control user access to logon groups.

If the SAPLOGON.EXE frontend is distributed to end users, they can use the Group Select option to identify logon groups, and then add the entries to their individual SAP GUIs. (Recall that the SAPLOGON.EXE window includes the Add, Properties, Delete, and Group buttons through which the user can modify her or his own SAPLOGON menu list of access points.) If a user has either an application instance or a logon group entry displayed in the SAPLOGON menu, that user will have access to that resource. (Of course, the user must have a valid SAP user account on the SAP system/client.)

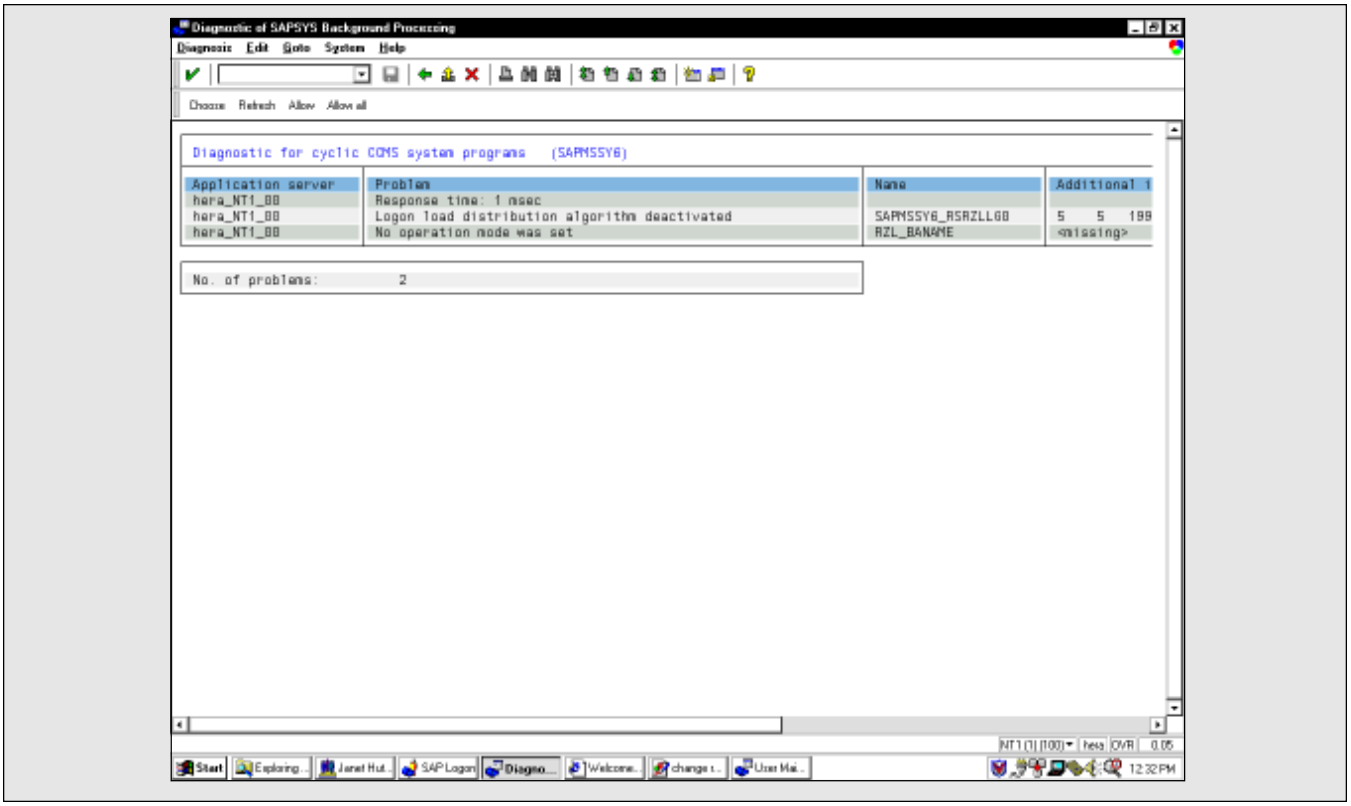
The only way to control how users interact with logon groups is to control access to the logon groups themselves. This can only be accomplished by controlling the contents and distribution of the SAPLOGON.INI file. I explained earlier how using SAPLGPAD.EXE removes the user's ability to modify the SAPLOGON menu list. Using SAPLGPAD.EXE also means that the end users cannot modify the SAPLOGON.INI file through SAP GUI functionality. (Keep in mind that a user will lose the ability to connect to SAP if the SAPLOGON.INI file is corrupted or lost through other update methods.)

Rest assured, users will exercise the ability to modify their menu list, if given the chance! Let me tell you the tale of Dan, Mary, and Sally, SAP users of a large system…

Dan and Mary are coworkers. Dan focuses primarily in the FI area and has been instructed to use the FI logon group. Mary's responsibilities are in the SD area. She uses the SD logon group. One day, over a cup of coffee, Mary tells Dan, "I was experimenting with the different SAP options the other day and I found a list of all kinds of different logon groups. You know, my transactions always take so long to process, and Jamie from HR says that his stuff always gets done really fast. I saw that there was an HR group in the list so I decided to try that one. It didn't seem to be much different from the one that I usually use. So, I guess that it doesn't really matter."

The next day, Sally, the Basis administrator, received five calls from the HR department complaining about slow response time. When Sally investigated, she found FI and SD work mixed in with the HR work on the application instances that were supposed to be dedicated to HR. She thought to herself, "That logon group stuff really doesn't work very well at all. Now what do I have to do to tune the HR servers and eliminate this performance problem?" Meanwhile, she did not notice that there was very little work being processed by the FI and SD instances.

Figure 11 Viewing the Execution Status of RSRZLLG0 Through CCMS



If it is desirable to isolate all or certain parts of the system's workload, and this needs to happen on a consistent and predictable basis, then you must eliminate the end users' ability to choose where they will sign on. In such a case, the end users will use SAPLGPAD.EXE, and the SAPLOGON.INI file will be created and maintained by a central administrator, and distributed to end users as changes need to be made.

On the other hand, if it is decided to create one or more large pools of application servers, and the approach is that work will be distributed as it arrives and not based on any other criteria, then it is less problematic if users have some choice over the logon group used for any particular sign-on.

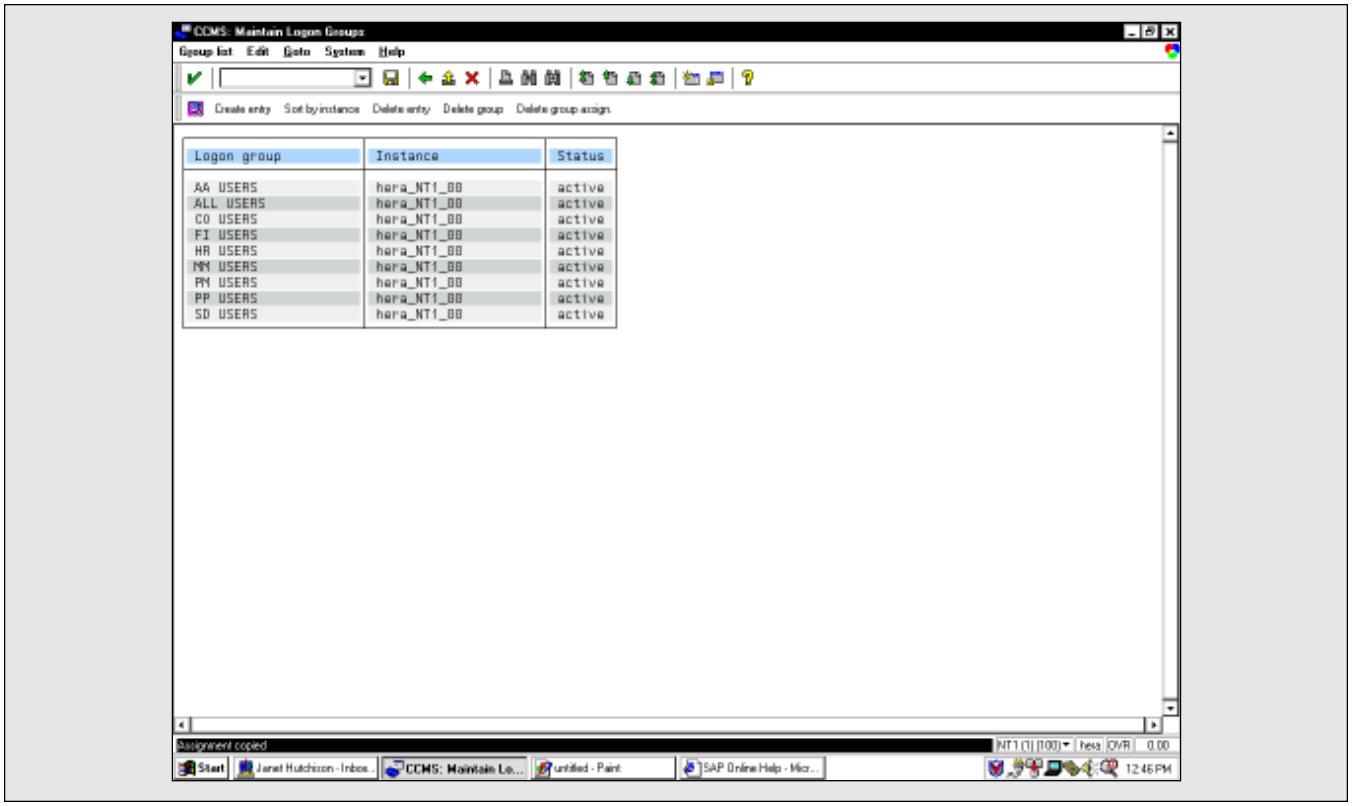
New Logon Groups, Group Assignments, and Deletes Take Effect After a Short Delay

Any and all changes made to the logon group architecture take effect with the next execution of program RSRZLLG0. As there is actually a time delay between the executions of this ABAP program, it may appear, at times, as though changes have been saved but not activated. Monitor the execution of RSRZLLG0. If the changes do not take effect after about 5 minutes or the next execution of ABAP RSRZLLG0, then go back and verify that the changes were indeed made and saved correctly. The problem might be that the regular, periodic execution of RSRZLLG0 has been deactivated, as shown in **Figure 11**.

Three Strategies for Designing Logon Groups

The goal of your logon group implementation is workload, and therefore performance, predictability. At the same time, you want to minimize the effort to design, create, maintain, train for, and use logon groups.

Figure 12 Logon Groups Set Up According to SAP Function Module



Strategy #1: Devising Logon Groups Along the Lines of Similar Workloads

One strategy for achieving this is to group like workloads together along the lines of:

- SAP module
- Business role or function (within or across SAP module areas)
- Business group or organization
- Geography or physical location

The main benefit of grouping like workloads together is the improvement of SAP buffer quality, especially when workgroups/logon groups are defined along SAP module or business functional lines. Users in the same business functional area tend to use a common set of transactions. The repetitive use of a finite set of transactions will tend to improve the buffer quality in the program buffer. It has also been demonstrated that online, operational applications tend to use and reuse data, which further improves the data buffer quality. Grouping of like workloads also helps to eliminate competition for resources from workloads with different characteristics or objectives. Segregating work components with similar processing attributes is a long-standing, proven approach to systems performance. It avoids situations where a few resource-intensive, long-running transactions cause performance problems when introduced into an environment that is tuned and intended to process numerous small and quick transactions. FI transactions, for instance, tend not to be resource-intensive, whereas HR transactions can be very resource-intensive. **Figure 12** shows an example of logon group design based on SAP function modules as the differentiating criteria.

Like all good implementation strategies, there is a tradeoff. Grouping like workloads requires a good

deal of time to analyze, understand, and categorize the various workload components.

Strategy #2: A Single, Mixed Workload Grouping

In a mixed workload implementation, you define just one (or just a few) logon group(s). The idea behind having all users use just one logon group is that incoming workload requests are always assigned to the application instance with the best performance at that point in time. The underlying assumption here is that the instance with the best performance is also the instance having the most available processing capacity. (You would still use the profile threshold values to set performance targets from instance to instance.)

The key benefit to this strategy is its simplicity. There is just one logon group to define and maintain. You don't have to understand, analyze, and categorize workloads. As new users are added, there is no effort required to determine which logon group(s) the user should access. There would only be one SAPLOGON.INI file that all users could use. In fact, you could use the SAP-delivered SPACE group — although this is not a very user-friendly name.

Other practitioners may disagree with this strategy, thinking that the improved buffer-quality performance gains of like workload grouping are a must. However, it depends on the SAP environment, its resources (both system and human), and its workload. A simple logon group strategy with mixed workload may be a good strategy in some cases.

Strategy #3: Establishing Logon Groups by Time Zones

In an SAP R/3 system environment that spans several time zones, the logon groups may be designed so that a consistent level of work is directed to application instances around the clock. This approach can be used to optimize the use of available processing capacity.

Helpful Hints

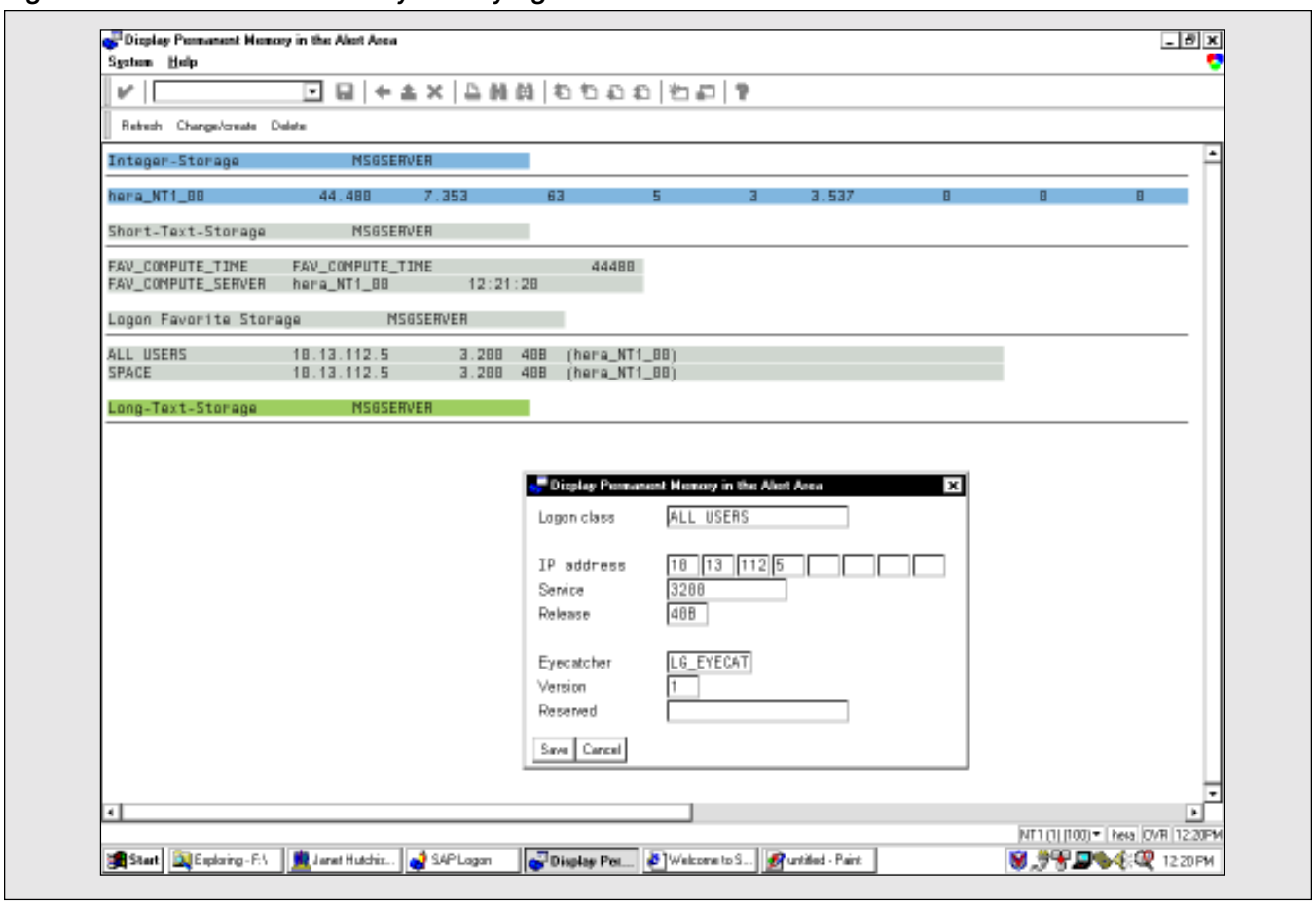
✔ **Logon load balancing is *not* dynamic workload distribution.** There is no way that work can be re-directed or redistributed to another application instance once that work has been started. Once a user is logged on to an application instance, that application instance will be responsible for processing all work requested by that user during that session. Logon groups and logon load balancing provide a way for us to influence, but not control, workload distribution among the application instances in the R/3 system.

The term "load limits" can cause some confusion. Reaching or exceeding these limits does not prevent additional users from being connected. It only results in adjustments in the calculation of the favorite application instance. A favorite application instance will still be identified for every logon group, even if every application instance has exceeded its load limits. Presumably, other alerts to Basis administrators will be triggered under such conditions.

It cannot be expected that individual user logon requests will be distributed in a round-robin fashion among all available instances. The recalculation of the favorite instance does not occur frequently enough, nor is the logon load distribution logic founded on that basis.

✔ **It is possible for RSRZLLG0 to become disabled and cease to run.** If this occurs, the favorite list will not be updated and all subsequent logon requests will be directed to the same application instance. There are two parameter values stored in memory that can cause this and other cyclic ABAP programs to become disabled. These parameters can be viewed via CCMS transaction SMLG. These parameters set the threshold values for maximum runtime of a single execution and maximum runtime for each of two successive executions. Generally, the maximum value for a single execution would be set to a higher value than that for two successive executions. If either of these two values is reached or exceeded, RSRZLLG0 will be disabled for the rest

Figure 13 Manually Modifying the Contents of the Favorites Table



of that day. The system will re-activate the program at 00:00 hours.

You can trigger the execution of RSRZLLG0 and the recalculation of the favorite instance. You can trigger the immediate execution of RSRZLLG0 from SMLG → Goto → Load distribution. Once this screen is displayed, double-click on any instance entry in the top section of the display. RSRZLLG0 will run, and you will see that the time displayed in the Sample Time column will be updated. (Note: It is not possible to run ABAP RSRZLLG0 from the ABAP execution window. You will not have the appropriate authorizations for this activity.) There are several situations in which you might want or need to do this:

- RSRZLLG0 has been removed from the automatic processing cycle due to excessive runtimes.
- One or more application instances have become unavailable, and you want the favorites list to be updated without the delay of waiting for the next time-triggered execution of RSZLLG0.

✔ **You can modify the favorites list stored in the message service memory area via transaction SMLG.** This is shown in **Figure 13**. Here, you manually modify the contents of the favorites table in the message service memory, and thus assign the favorite instance! These changes will be overwritten the next time that RSRZLLG0 is executed.

Why would you want to do this? If the automatic execution of RSRZLLG0 becomes deactivated, *and* manually forcing RSRZLLG0 to execute by the method described above does not produce the desired result in terms of assigning a favorite instance, then you, as the administrator, can enter the specific TCP/IP address of the server that you want to be the favorite. *Just be careful!* When I tried this, I was able to put in "any old" TCP/IP address, and there was no checking to see that the server belonging to this TCP/IP address was actually part of *this* SAP system. In fact, I was able to enter the TCP/IP address of an application server from a *different* SAP system, and logons were directed there! (It happens that I have accounts on the same client in both of these systems.) Presumably your users would not have the same valid accounts and passwords in multiple SAP systems in your environment. You face a very remote chance of this situation in your system, but it is a caution to keep in mind.

✔ **The characteristics of a logon group are conceptual and documentation-only based.** Remember, there is no way to define and enforce user logons based on any data attributes of users or their intended workloads. As this cannot be enforced or captured in an R/3 system, good documentation about the logon groups, their intended purposes, performance attributes, and users must be created. This should include any changes to the execution time parameters for RSRZLLG0.

✔ **When designing a logon group strategy, be sure to account for work requests received via RFC calls from bolt-on applications and remote systems.** It is possible for a significant amount of work to come via RFC connections, and it is possible that this work could impact application instance performance. It may even be possible that separate application instances and/or logon groups would be established if the remote workload is very large.

When the RFC destination is defined in the source system, there is the option to specify either a specific application instance or a logon group in the target system.

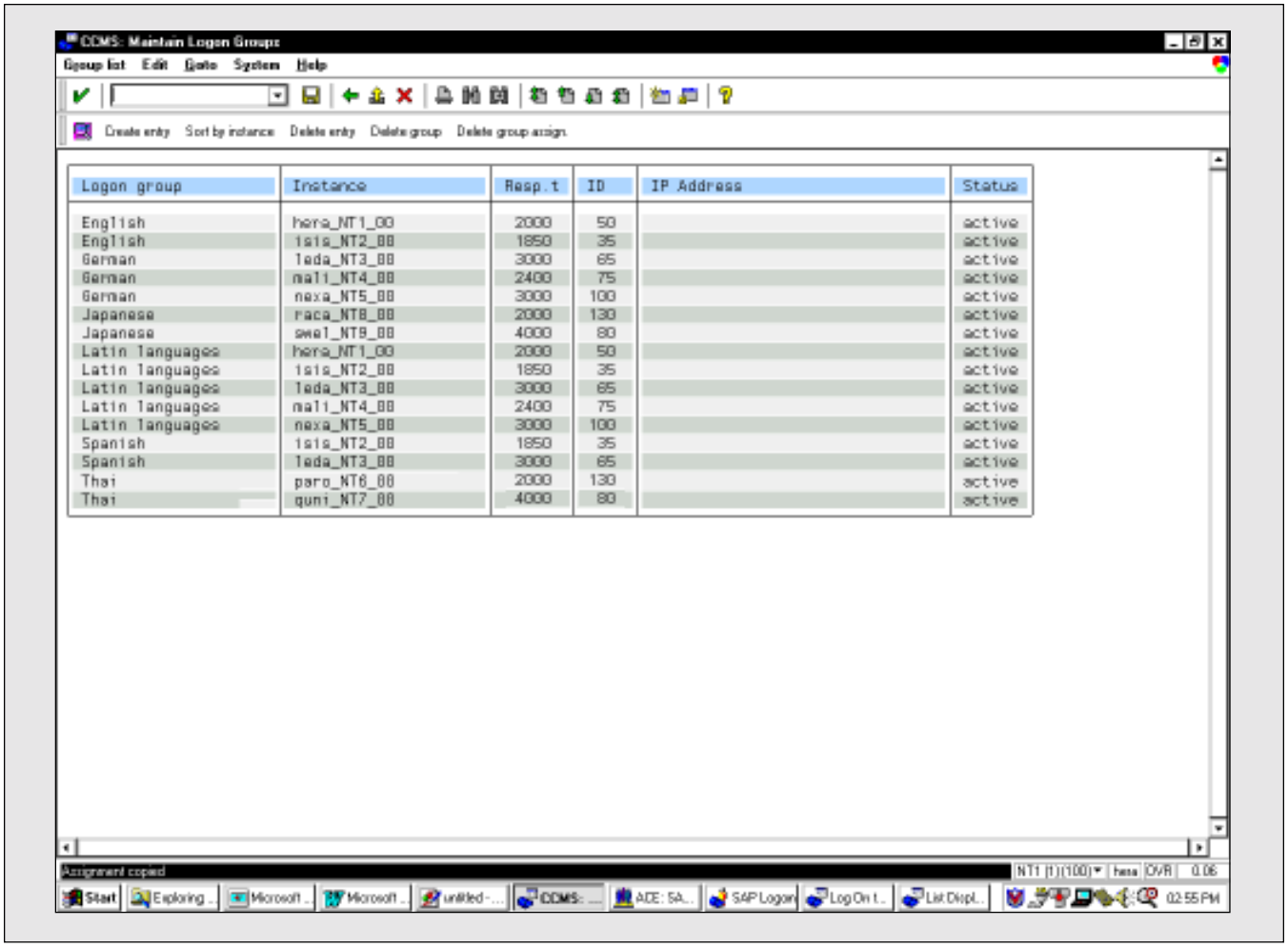
Logon groups deliver similar benefit to RFC connections as they do for direct user logon requests: accessibility to the system is maintained as long as there is one application instance available for the logon group; bolt-on applications and remote systems are insulated from application instance architecture changes.

✔ **SAP reports that relative transaction resource consumption and performance has changed noticeably from Release 3.x to 4.x.** Some transactions are more resource-intensive than in earlier releases and some transactions now consume far fewer resources. As a result, a logon group design that was effective under Release 3.x may not produce the desired workload distribution after an upgrade to Release 4.x. Logon group design and logon load balancing goals and objectives must be considered and tested as part of the upgrade project.

✔ **If you have to support double-byte languages at the presentation level, you *must* set up logon groups along the lines of similar workloads.** When double-byte languages are being used at the presentation layer (SAP GUI) a double-byte code page must also be running there. Double-byte presentation layers have to connect to double-byte application instances. You must not mix double-byte and single-byte application instances in one logon group. This would run the risk of having a single-byte application being identified as the favorite for a logon group being used by double-byte users. So in multi-language environments that require multiple code pages or any combination of languages that cannot all be supported by the same code page, there will have to be separate application instances for the different code pages. And there will have to be separate logon groups to access the different groups of application instances.

Figure 14 shows a system with nine application instances and six logon groups. The basis of this design is that there are five languages that must be supported. (Some language combinations are supported by a blended code page.) Each language

Figure 14 A Multiple-Language Logon Group Design



is supported by at least two application instances. You can also see in this figure that the threshold profile is consistent for an application instance, even if it is used by more than one logon group.

✔ **You can't just design and implement a logon group strategy and then ignore it!** You won't know that there are problems with your logon strategy unless you include it in the monitoring process. With CCMS, you can set and trigger alerts for the two logon load balancing thresholds: *connected users* and *average dialog response time*. If you have set values in the logon load balancing thresholds, you should probably also set CCMS alerts for these as well! If you don't configure CCMS monitors to coincide with your logon load balancing objectives, the first clue about a problem may be calls from users who are experiencing poor system performance. In an administrator's job, success is to see the problem coming before it occurs!

✔ **In OSS note 39507, SAP supplies a pre-production checklist.** Verifying the use of logon groups in the system architecture is one of the items on this checklist. The evaluator is instructed to review the non-use of logon groups.

Conclusion

Clearly, there are benefits to be had in SAP environments by the use of logon groups that are large enough to require multiple application instances. But those of you who support small R/3 environments would realize benefits, too. I would argue that even in smaller installations, it is still a sound strategy to implement logon groups in anticipation that future growth will lead to more application instances. If a logon group structure is already in place, the new instances can be added with no impact to end users.

In logon groups, SAP delivers logic and functionality that is difficult to duplicate. There is very little cost and great benefit to be attained. Some of the most significant benefits — use of logon groups for achieving system performance objectives, High Availability solutions, system usability, availability benefits for end users, administration flexibility, and system performance objectives — are not well advertised by SAP.

Logon groups and logon load balancing are stable and effective features of SAP R/3. It is clear that SAP feels that logon groups and logon load balancing are valuable in attaining highly satisfactory, highly available, and well-performing SAP R/3 systems. You should, too.

Janet Hutchison wrote her first computer program while listening to disco music. The computer was a Wang "desktop" with 10K of ferrite core, and it accepted input via punched paper tape. Janet gained her first experience with real-time systems in the same decade. Since then, she has held a variety of IT positions with several large, multi-national corporations as DBA, Systems Administrator, and as a Systems Performance and Capacity Analyst. With over four years of SAP experience, Janet has worked with both SAP R/2 and SAP R/3 systems. Janet joined PricewaterhouseCoopers in 1998 as a Principal Consultant, and is based out of PricewaterhouseCoopers' ERP Technologies Global Training Center in Philadelphia, Pennsylvania. Janet develops and instructs training classes in advanced ERP Technical Infrastructure topics, and is responsible for courses such as SAP Basis Administration, SAP System Performance, Oracle Database Administration, and Backup and Recovery. She was recently a presenter at SAP TechEd 99, where she delivered a session entitled "Logon Load Balancing and Its Role in SAP R/3 System Performance." Janet can be reached at janet.hutchison@us.pwcglobal.com.

