WLM GOAL MODE USING A QUICKSTART SERVICE POLICY

Note from Cheryl: This is an update to the Quickstart Service Policy article that was previously published in our May/June 1995 TUNING Letter and updated again in May 1999. If you're new to the Quickstart Policy, simply jump ahead to the Introduction. The 2003 changes include:

1. Changed the STC classification rules to use a default of STCLO instead of SYSSTC by using the SPM rules.
2. Added a new service class, NEWWORKV, for new work that requires a velocity goal instead of a response goal. Changed some new workloads to use NEWWORKV instead of NEWWORK.
3. Updated the description of CICS and IMS subsystems to recommend using response time goals initially.
4. Assigned transaction versus region management for production online systems, but still retained region management for test and older online systems.
5. Turned on I/O Priority Management.
6. Set the Dynamic Alias Management service option to NO (the default).
7. Added the PRDBATMD service class.
8. Changed some velocities (PRDBATHI) so that a velocity of 30% wouldn't let discretionary work run above them.
9. Added new workloads for MQSeries (MQ), NetView (NETV), Oracle (ODSI), SAP R/3 (SAP) and CB.
10. Changed all report class names to begin with 'R'. Added report classes to every service class.
11. Changed service class names ONLPRDLO and ONLPRDHI to TRANLO and TRANHI. There was too much similarity to the region managed service classes of ONLPRD and ONLTST, causing confusion. Added classification groups for these.
12. Added description fields to classification rules and groups.
13. Revised the contents of many of the classification groups.
14. Updated some of the text to provide a more thorough explanation. Added additional references to the bibliography and throughout the document.

INTRODUCTION

You've been running in WLM compatibility mode for a while and you want to move to goal mode soon. But where should you start? Here's an easy method of getting there - a "Quickstart" Service Policy. It's a simple, generalized service policy that will work in most installations as a starting point. Our Quickstart Service Policy provides a standard set of service classes and goals that you can easily use or modify for your initial policy.

Once in goal mode, you'll have access to WLM's superior resource management and a wealth of measurement data that makes tuning a real breeze!

A downloadable form of this Quickstart Policy is available for OS/390 R10 sites and later. You can obtain instructions on downloading the policy here:
www.watsonwalker.com/qspdown.html.

The migration to Workload Manager (WLM) goal mode in OS/390 and z/OS may seem intimidating at first, but it can be quite simple. In this paper, we'll introduce a simple service policy we've designed that can be used by most installations. It should ease your migration to goal mode.

Some installations have used this "Quickstart" policy with little modification and began running in goal mode within a week or so. We think you'll find it an easy way to start your own migration.

This article assumes you have a basic knowledge of WLM. If you're not familiar with service policies, service classes and goals (response, velocity, and discretionary), please refer to the bibliography at the end.

We have included several articles that might be of help to you at **www.watsonwalker.com/articles.html**:

- Positioning for Goal Mode - This article on Positioning For MVS SP 5 was written for our September/October 1994 TUNING Letter. The section of the article on preparing for Workload Manager is as useful today as it was then for sites who are preparing to move to WLM goal mode. The suggestions will help you prepare your IPS and ICS for an easy migration. If you plan to use IBM's Goal Mode Migration Aid (GMMA) from their Web site **www.ibm.com/zseries/zos/wlm**, then also read the GMMA documentation before modifying the comments of your ICS.

- Why Go to Goal Mode? - It takes only hours - at most a few days - to start a test system running in goal mode, and the benefits can be enormous. This article helps justify to management (and yourself) the effort of migration. This article was previously published in our 1999, No. 5 issue.

- Getting to Goal Mode - Step by Step - This is a simple checklist of how to get to a monoplex environment and run in goal mode.

KEEP IT SIMPLE

For years, the people maintaining IPS and ICS members have found that simplicity has its rewards. Not only is it easier to make changes to your IPS/ICS parameters, but it is also easier to analyze reports and explain the system. And of course there's the reduction in processing costs and DASD space required to maintain the data. Installations with only a couple of dozen performance groups take much less time and space to process their data than installations with several hundred groups.

The same reasons for keeping things simple hold true when defining a service policy, only more so. The performance is considerably better and WLM is much more effective with a simple definition containing a few service class periods than with a definition consisting of hundreds of periods. A simple service definition is also easier for you to maintain. Although you should try to keep the total number of service class periods to a minimum for simplicity, you should know that there is little effect on performance if you choose to add several report classes or service classes with discretionary goals.

Therefore, we have applied a single rule-of-thumb in providing these recommendations: "Use the minimum number of service definitions, policies, workloads, resource groups, service classes, service class periods, and classification rules that you require." In other words, start small and add only when necessary. Try not to have over 25 service class periods with non-discretionary and non-system goals with active work on a single system. That is, you might have over 25 service class periods, but some will have SYSSTC or discretionary goals and others will only be used on one or two systems.

You could begin with our Quickstart service definition and add to it as you find a need. Start with a single service definition and service policy. A single policy may work just fine around the clock. Define a sim-

ple set of workloads and service classes and only use multiple period service classes when you have a specific requirement, such as with TSO.

Workloads and service classes for our Quickstart service policy are shown in Figure 1. The values in the importance ("Imp") column are set by your installation based on the workload priorities, but we've started you off with a set of importance values that are common to many sites. The corresponding classification rules are shown in Figure 2, and the classification groups are in Figure 3.

This is a very easy service definition to implement, and it contains only eighteen non-system service classes, which allows for plenty of room for growth. Most current performance groups should easily fall into one of these classes. Large online sites may need to add additional service classes for their online work.

The SYSTEM, SYSSTC and SYSOTHER service classes are standard service classes defined by WLM.

WLM GOAL MODE USING A QUICKSTART SERVICE POLICY

To be able to use this Quickstart Policy, you'll need to make some initial assignments. The Quickstart Policy is described below by subsystem, and describes the purpose of each service class and any initial decisions you might need to make.

For service classes using response time goals, we've recommended starting with percentile response time goals. Percentile goals are much better to use because they're easier to achieve and provide a more accurate picture of what the user is experiencing. A single long job or transaction won't affect a percentile response time but can greatly affect an average response time. But you may prefer to start with average response time goals simply because you can obtain average response times directly from your RMF, CMF or SMF data. After you've been running in goal mode for a short period of time, you'll be able to determine percentile response time goals to use instead of averages. Please note that response time goals (both average and percentile) can only be used when there is enough work in the service class period for WLM to make a good analysis. A rule of thumb is that there needs to be at least ten transactions completing within twenty seconds before you can use response goals. If you don't have transaction volume that high, then you need to use a velocity goal.

STARTED TASKS (STC)

We defined five service classes for started tasks (ONLPRD, ONLTST, STCMD, STCLO and SERVERS). The system automatically defines two more: SYSTEM and SYSSTC. Starting with OS/390 R4, you can directly assign any type of work (STC, batch, etc.) to either SYSTEM or SYSSTC. Prior to R4, you could not assign anything to SYSTEM, and could only indirectly assign something to SYSSTC by leaving the default service class blank on SYSSTC. For this policy, we are assuming an OS/390 R10 system. If your system predates OS/390 R4, please look at our May 1999 Quickstart policy for a recommendation.

- **SYSTEM** is automatically assigned by WLM to system address spaces such as the master scheduler and CONSOLE. These address spaces are run at dispatch priority 255 (the highest). We definitely do not recommend moving anything in or out of SYSTEM. Just because you're allowed to, doesn't mean you should do it. There is no comprehensive list of work assigned to SYSTEM. The Workload Management manual indicates that "GRS, SMF, CATALOG, MASTER, RASP, XCFAS, SMXC, CONSOLE, IOSAS, WLM" are assigned to SYSTEM. On our z/OS 1.4 system we also found the following in SYSTEM: SMSPDSE, ANTMAIN, OMVS, JESXCF, ALLOCAS, MMS, IXGLOGR, JES2MON and BPXOINIT. Only address spaces created with "ASCRE HIPRI" may be reset into the SYSTEM service class. To find a list of the high dispatching priority address spaces in your site, look

at an RMF Monitor II (or CMF Online) report and look for dispatching priorities of x'FF' or SRVCLASSes of "SYSTEM".

- **SYSSTC** is the default service class assigned by WLM to any started task that isn't otherwise assigned a service class. It's also assigned to anything with the PRIV assignment in the parmlib SCHEDxx member. Prior to OS/390 R4, you could only assign work to SYSSTC by leaving the default service class of the STC subsystem blank in the WLM screens. Your current WLM policy may still be using that method. In the Quickstart policy, we've changed our STC default to a discretionary class. SYSSTC runs at dispatch priority 254, so you really shouldn't allow low-importance started tasks to fall into this service class. (Note that prior to APAR **OW19265**, the dispatch priority for SYSSTC was 253.) You can assign any high-importance tasks here such as VTAM, JESx, VLF, LLA, auto ops packages, the OMVS kernel, APPC and ASCH. We've designated a transaction name group (TNG) in Figure 3 to be used for these called STCHI. In the STCHI group, you can place all of your high importance started tasks. There isn't a velocity goal associated with SYSSTC, but rather what the workload manager terms a "system goal." In our system, the following address spaces are assigned to SYSSTC: SMS, LLA, INIT, JES2, VLF, VTAM, ASCHINT, DLF, NFSC, RACF, TNF, VMCF, APPC, ASCH, TSO, RESOLVER, TCPIP and ZFS. INIT might be set to a different dispatch priority if you have installed APAR **OW553344** and set the IEAOPTxx parmlib parameter INITIMP to something other than 0.

- Transaction name groups (TNG) and transaction class groups (TCG) are used to easily create lists of names or classes that are to be assigned to the same service classes. We normally use the service class name as the group name for simplicity. Classification groups are assigned in the WLM ISPF dialog.

- Prior to OS/390 R4 you couldn't directly assign work to SYSSTC using classification rules, so you needed to code the rules for STCs in a special manner. SYSSTC could only be obtained as the DEFAULT for the STC subsystem. We don't recommend this method any longer, but include the technique for documentation because you might have used it originally. We previously used the following technique to assign classification groups to SYSSTC:

```
TYPE    VALUE           SVC CLASS
              DEFAULTS  ________
TNG     STCHI           ________
TNG     STCMD           STCMD
TNG     SERVERS         SERVERS
TNG     MONITORS        ________
TNG     ONLPRD          ONLPRD
TNG     ONLTST          ONLTST
TN      *               STCLO
```

STCLO is discussed later. This original technique was described in the WLM Planning manual mentioned in the bibliography.

The recommended technique since OS/390 R4 is:

```
TYPE    VALUE           SVC CLASS     RPT CLASS     TRANSACTION
              DEFAULTS  STCLO         RSTC
TN      GRS             SYSTEM        RGRS
TN      CATALOG         SYSTEM        RCATALOG
SPM     SYSSTC          SYSSTC        RSTC
SPM     SYSTEM          SYSTEM        RSTC
TNG     STCHI           SYSSTC        RSTC          REGION
TNG     STCMD           STCMD         RSTC          REGION
```

```
TNG     SERVERS         SERVERS       RONLINE       TRANSACTION
TNG     MONITORS        SYSSTC        RMONITOR      REGION
TNG     ONLPRD          ONLPRD        RONLINE       REGION
TNG     ONLTST          ONLTST        RONLINE       REGION
```

This method first assigns everything in your STCHI transaction name group to the system-managed SYSSTC service class. It then assigns your defined STCMD tasks to a medium velocity and importance service class of STCMD. It assigns the servers you've identified into the SERVERS service class with Transaction-managed regions and assigns monitors into SYSSTC. Production online regions managed to velocity goals instead of response goals are assigned to ONLPRD and test online regions are assigned to ONLTST. Any SYSTEM or SYSSTC address space for which you want separate reporting should then be assigned to its correct service class along with a report class. Then everything that IBM has indicated should go to SYSSTC and SYSTEM are assigned there with SPM. Finally, anything not covered by the previous rules will go to the default of STCLO.

- **SERVERS** is a service class for subsystems that provide service to transactions that are managed by transaction goals. This will normally consist of the regions for CICS V4 and higher, IMS V5 and higher and DDF V4 and higher. Work is managed to transaction goals when you have defined even one service class in that subsystem. The three most common subsystems are CICS, IMS and DDF, although you will soon be assigning server address spaces for IWEB, NetView, Corba, Oracle and others.

  In the past, we recommended that you use velocity goals when you first move to goal mode. We've changed our recommendation on that point. Because of the many benefits of using transaction goals, we suggest that you start with them. This takes a little preparation on your part, however. CICS in particular must be prepared. Because WLM uses the CICS MAXTASK parameter to determine the number of PBs (Performance Blocks) to build for collecting data, an unusually high value for the total MAXTASKs (sum of all regions) could cause severe overhead. So before running CICS in goal mode using transaction goals, you should set the MAXTASK value to no higher than absolutely needed. You should also find out from the subsystem people the names of the most critical transaction names (CICS usually) or transaction classes (IMS usually). Having done these two preparatory steps, you can then assign the production regions (either STC or batch) as being managed with transaction goals. Test regions should be managed with region goals. The main advantage of using transaction goals instead of region goals is that they will not need to be changed when hardware or software changes are made.

  The reason for a separate service class for SERVERS is to prevent these address space samples from distorting the velocities for other work. As an example, let's take a very simple case of two CICS regions being assigned to a service class called ONLPRD. You have assigned a velocity goal of 60%, which is continually met. Now, you decide to convert one of the regions to CICS V4 and assign transaction goals in the CICS subsystem. WLM will actually place the upgraded region in an internal server service class, but will still collect samples. RMF and CMF do not know anything about internal service classes, so will simply add up all of the samples for both regions to report them. If the new region ends up running at a velocity of 50%, it will appear that you missed your goal for ONLPRD, when you really didn't. If the new region ends up running at a velocity of 80%, it will appear that you exceeded your goal for ONLPRD, even if the other region missed the goal. That is, the goals of a service class may not be reported correctly if the service class contains any of the server address spaces. That's why you need to remove them from other service classes and ignore any performance index (PI) that you see on this SERVERS service class. The velocity will only be used at startup until the first transaction is recognized. From that point on, the address spaces will be managed to the transaction goal, not the velocity goal.

- **ONLPRD** is a single service class for your production CICS, IMS, DB2, and other online regions that are managed using region goals instead of transaction goals (see the SERVERS service class above). ONLPRD is used for older versions of these products or when you choose to manage your newer online systems with velocity rather than transaction goals. If you run your online system as batch jobs instead of started tasks, then simply make these assignments in the JES subsystem instead of the STC subsystem. ONLPRD is set with a velocity goal above batch. For the Quickstart Policy you can use a transaction name group called ONLPRD that would contain the started task names of your CICS, IMS, DB2, ADABAS, ROSCOE, IDMS, Oracle and WYLBUR address spaces. We'd recommend, however, that you place the test online regions in ONLTST (described below). The reason for this is that test online systems tend to get into loops and you don't want looping jobs running in high velocity service classes.

  If you're not running in a constrained environment, you can assign all of these address spaces to the same service class with a velocity goal. If you're running in a constrained environment, you may want to monitor your primary online systems during peak intervals and determine the velocity that they now receive. If there are varying velocities (e.g. 20%, 40%, and 60%), then set up different service classes for each velocity. You might end up with ONLHI, ONLMD and ONLLO service classes. In a constrained system you might want your IMS control region and CICS TOR (terminal owning region) running with a higher velocity goal than the IMS message processing regions or the CICS AORs (application owning regions).

- **ONLTST** is a service class that can be used for test online address spaces. (Again, this does not apply to online systems that are using transaction goals.) It would have a discretionary or low velocity goal, such as 10%. As we mentioned before, we would tend to keep the online test regions in a separate service class from production because the transactions in the test regions have a tendency to go into loops. We defined a separate service class in order to assign them to the ONLINE workload. If you don't need to summarize these test regions with the production regions, you could simply assign the regions to STCLO as described below.

- **STCMD** should be used for relatively high-importance started tasks that should be run below your online regions. We provided a STCMD transaction name group that can be used to contain all of the STC names in this service class. This service class will be unique between installations, but might include such products as your scheduling products, printer management products and high importance operations jobs.

- **STCLO** is the service class used for all other started tasks and should be assigned a discretionary goal. See STCHI for the technique used in assigning this. It will contain all started tasks not assigned to any of the other service classes. You may implement this scheme only to find a few address spaces that aren't performing as well under goal mode. If so, you can move them and later define them in one of the other started task classification groups. As mentioned above, you could also include any online test regions in this service class. Some installations run the default STCs at a very low velocity (e.g. 5%) instead of discretionary.

- **MONITORS** is a report class, not a service class. We added this because many sites want to know how much of their machine is spent monitoring the machine itself. We assign this to SYSSTC because we believe that monitors need to run at least as high in dispatch priority as the work they monitor. Looking at the MONITORS report class resource consumption will tell you how much of the machine is spent in monitoring. You can use this same technique for any other type of work in your site.

BATCH JOBS

The Quickstart Policy has seven service classes defined for batch jobs, divided between test and production because most installations tend to manage their batch work in those two categories.

- **PRDBATHI** - You may not need to assign anything to this service class initially. It's meant for high importance production batch jobs to help them complete on time, and isn't needed if you can easily meet your production deadlines (i.e. an unconstrained system at night). It can also be used for operators who need to increase the importance and velocity of a job with the RESET command.

  In a constrained system, you could identify critical path jobs and give them a unique jobclass or accounting code in order to assign them to this service class. We defined a TCG (Transaction Class Group) of PRDBATHI for use with unique jobclasses, but you could easily change it to a TNG (Transaction Name Group). The PRDBATHI service class runs with a velocity goal.

- **PRDBATLO** is for any production job that is considered a "hog." This is the type of job that can run for hours and you want all other work to run above it. You usually identify these jobs by running them in a unique job class. This service class runs at discretionary.

- **PRDBATMD** is for all of the other production batch jobs and is assigned a low velocity, but one that will run above discretionary.

  We did not define PRDBATMD in our previous Quickstart Policy, and PRDBATLO was used for all production batch with the exception of PRDBATHI jobs. But so many installations needed to add a PRDBATMD in order to keep their production from dying in discretionary, that we decided to add one to the standard Quickstart Policy.

- **TSTBATHI** is the service class used for short turnaround batch job classes. For most installations the turnaround time from job submission to execution completion is between 10 and 20 minutes. Because most installations set up a separate jobclass for these short jobs, we've included a classification group for transaction classes called TSTBATHI where you can define these job classes. You need to have a rate of at least ten completions in a twenty minute period to be able to use response time goals. You can determine the rate of completions by looking at any RMF workload activity report at the ENDED field, which provides the number of transactions (or jobs) that ended during the interval. Divide that number by the number of minutes in the interval to determine if you have enough activity to use a response time goal. If you don't have this volume of activity, you'll need to change the goal to a velocity goal.

  You can determine the current average response time by looking at RMF, CMF or SMF data for these jobs. In an RMF Workload Activity report, for example, the ACTUAL (or TOTAL for RMF releases prior to V5) response time is the average response time from job submission to termination. We've defined this with a percentile response time goal, but you could start with average and change it after you've collected more response data.

- **TSTBATMD** can be used for longer test batch jobs, typically with a twenty to thirty minute turnaround. If you have several of these jobs (at least ten completing every twenty minutes so that WLM can obtain adequate sampling) you can use a percentile response time goal. If the activity is lighter than that, you'll need to change the goal of this service class to a velocity goal.
- **TSTBATLO** should be used for the remainder of the test batch jobs. This service class uses a discretionary goal and contains all jobs not assigned to the other test batch service classes. This is the default service class for the JES subsystem instead of the previous default of PRDBATLO.

It's possible that your installation doesn't use jobclasses to differentiate test jobs because you use multiple periods to determine short, medium and long. In that case, you can simply define one service class, such as TSTBAT, containing three periods with different goals for each period (similar to the three classes we've just described).

- **HOTBATCH** is a service class to be used by the operators when they need to move something through the system quickly. Its use should be minimal. It's defined with a velocity goal because it shouldn't have enough activity to support a response goal.

TSO

The Quickstart Policy defines a single TSO service class, **TSOPRD**, with two periods. It's set up using percentile response time goals, but you could start by using average response times corresponding to the current response times you see reported by performance group period. Percentile response times are highly recommended. The second period uses a velocity goal because there may not be enough transactions ending to use a response time goal.

Most sites currently use three or four periods for TSO, but after more detailed analysis you will generally find that second period accounts for few transactions and a small amount of CPU time. In goal mode, it is more important to reduce the number of periods with goals than to try to manage this small amount of work. Installations that have used two periods have been very happy with it. Try it!

TSOPRD is the default service class for all TSO users in this Quickstart Policy. We do not recommend having a special TSO service class for the system programmers. If an emergency occurs, an operator command can move a TSO session for a system programmer into SYSSTC.

ASCH

Most installations don't run many APPC/MVS workloads. The earlier releases of OpenEdition MVS (now called UNIX System Services) used APPC/MVS for its implementation. APPC/MVS has several started tasks that can be assigned to service classes. The APPC and ASCH (scheduler) started tasks (you can change the name, of course) intercept and manage all of the APPC/MVS work. Therefore, these tasks should be run in the SYSSTC service class.

The transaction programs (TPs) are the applications that process the requests coming from APPC/MVS clients. The characteristics of these programs will determine the actual goals. Many are long-lived started tasks with little resource usage and others are short-lived tasks with very high resource usage. You can monitor each one and place them in appropriate service classes, such as STCMD or STCLO. Because of their nature, the TPs are best managed with a two period service class that defines a short first period with a response goal and a longer second period with a velocity goal.

The APPC and ASCH address spaces are assigned in the STC subsystem, but the actual APPC/MVS TPs are assigned in the ASCH subsystem.

The Quickstart Policy service class **NEWWORK** is defined with two periods: a percentile response for short transactions and a velocity goal for longer work. This is really meant to be used by any of the unknown work on your system for which you want to assign a response-based goal. You can assign transactions in the ASCH subsystem to the NEWWORK service class and also assign them to the RASCH report class. When (and if) the ASCH workload (as seen in the report class) increases to a significant part of the system (over 10%, for example), you can assign a separate service class to ASCH.

OMVS

UNIX System Services (previously called OpenEdition MVS (OMVS)) transactions will be new to most installations. The Kernel for UNIX is similar to the master scheduler for MVS so the started task for the OMVS Kernel should be run at a very high priority. You can assign it to the SYSSTC service class. OMVS daemons are long-living started tasks that perform continuous or periodic system-wide functions such as network control. These started tasks should be assigned to STCMD in the STC subsystem.

Actual OMVS transactions, the forked children, are classified in the OMVS subsystem. (Don't you just love UNIX terminology?) Due to the characteristics of OMVS transactions, some of which have short response times and high resource usage and others which have long response times and low resource usage, a service class assignment similar to ASCH is the best place to start. The Quickstart Policy assigns the ASCH subsystem default to **NEWWORK** (as described under ASCH above) and a report class of ROMVS. As with ASCH, when the volume of activity becomes high enough, you can create a unique service class for this work.

CICS V4+ and IMS V5+

These subsystems provide additional WLM support to allow you to assign response goals by transaction, subsystem, transaction class or userid. If you install either of these versions while still running in compatibility mode, turn on collection of average response times. This will give you better data to let you set response time goals when you migrate to goal mode. The sample service classes that we've shown in the Quickstart Policy can be used as a starting point. Caution: you may see overhead (5%-10%) while collecting this data. For CICS, overhead can be reduced significantly by reducing MAXTASK to the minimum possible for each CICS region.

An important item to keep in mind is that there are two possible users of the transaction response time data. WLM will look at the goals and the actual response times of the transactions being processed by an address space. If goals aren't being met, WLM can respond by finding additional resources for the address space. But WLM can't really divide its resources among the transactions - it only manages address spaces. If you have short transactions with high importance and long transactions with low importance in the same address space, WLM can't help the high importance transactions without giving a free ride to the low importance transactions.

That's where other products come into play. As an example, CICSPlex/SM can route transactions to multiple address spaces. CICSPlex/SM uses one of two routing methods: Queue (simplified - which AOR has the shortest queue) and Goal Mode (which AOR provides the correct response time). In most installations, however, CICSPlex/SM is set up to use the Queue method. If you set up CICSPlex/SM to use the Goal Mode method, you will need to change our Quickstart Policy's CICS service classes to use average response times (the only method supported by CICSPlex/SM) rather than percentile response times.

It will do you little good to separate your transactions into multiple service classes if you'll end up letting WLM manage a single address space. Our recommendation is that you define a single service class for your most important transactions and set a response goal for them. If they don't make their goal, WLM can try to find more resources for the address space to help the short transactions. Other transactions in that address space will see some benefit as well.

- **TRANHI** is a transaction based service class for the transactions that should be completed in the shortest response time. Percentile response times are the best type to use, so 80% completed in .5 seconds would be a reasonable starting point for short online transactions. Many sites have already divided their important work by region, so you can use the subsystem instance (SI) classification to identify the

particular regions that will use this service class. You can also use a transaction name group, ONLPRDHI, to allow easy classification by transaction names. You may already have them divided by classes, userids or even subsystem instances (e.g. VTAM applid), so just change the group to correspond to your classification. What's important to note here is that you don't need to analyze all 5,000 of your transactions. Ask the application developers. They can tell you the one, two or three most important transactions in the shop. Simply classify those few transactions and the rest will follow.

- **TRANLO** would contain the transactions that don't need to get wonderful service, and it's the default for the CICS and IMS subsystems. You can also use it for all the transactions from a specific subsystem instance (such as a development region). You must assign a response goal (not discretionary or a velocity goal), so set this up with a very easily attainable goal, such as 50% complete within 10 minutes. By assigning low importance and long-running transactions to a separate service class like this, you're eliminating them from the service class that is trying to meet a response goal. Because WLM doesn't really manage transactions, but manages address spaces instead, this low response time goal will allow these long running transactions to live in regions that are really being managed to provide response times for more important and shorter transactions. It essentially eliminates them from consideration. You can't use velocity goals with the CICS or IMS subsystems. If the entire region will never need good service (such as a test region), then assign the region to be managed to a region goal instead of a transaction goal (as described in SERVERS previously).

OTHER NEW WORK

Each release of OS/390 and z/OS provides additional subsystems. It may be years before you are using these subsystems, but they should still be assigned to a service class for management and a report class for reporting. You can assign any of these new subsystems to **NEWWORK** (as described above under ASCH) or NEWWORKV and a corresponding report class. **NEWWORKV** was recently added to the Quickstart Policy because some new subsystems can only use velocity goals, and this service class provides that. Rather than describe future assignments for these new subsystems, please see our TUNING Letters or IBM's WLM Migration Guide (at **www.ibm.com/servers/eserver/zseries/zos/wlm/**) for recommendations. IBM's WSC team has produced a sample service definition, similar to our Quickstart Policy, to show examples of every type of workload and application environment. This is a great source for examples for each subsystem, but use extreme caution when using their sample definition for a production policy. There are seventy service class periods with non-system goals, far too many for practical use. In most environments, you will not need to get as complex as these examples. With that said, I use their policy often to find their latest recommendations on new workloads.

The new subsystems include:

CB - WebSphere Application Server (formerly Component Broker)
Assign the CB subsystem default to NEWWORK with a report class called RCB.

DB2 - Sysplex Queries
This subsystem is used to classify parallelized transactions that have entered this system from another DB2 system in the sysplex. We would recommend running these transactions in NEWWORK with a report class of RDB2SQ until you gain a better understanding of them.

DDF - DB2 V4+ Distributed Data Facility
Assign the DDF address space (usually DSNDIST) to the SERVERS service class. Assign the DDF subsystem default to NEWWORK with a report class called RDDF. See TUNING Letter 1999, No. 2.

IWEB - HTTP Server (running in Scalable or Enclave mode)
Assign the IWEB subsystem default to NEWWORK with a report class called RIWEB.

LSFM - Lan Server for MVS

Assign the LSFM subsystem default to NEWWORKV with a report class called RLSFM.

MQ - MQSeries

Assign the MQ subsystem default to NEWWORKV with a report class called RMQ.

NETV - NetView

At your option, you can tell NetView to create different enclaves for automation tasks and for network tasks. You specify this option in the DSIPARM member CNMSTYLE by uncommenting the line: WLM.SubSystemsName=<subsystem instance>. By specifying the option, you can set the automation enclaves to a higher velocity than the network enclaves. Both enclaves should be set to a single period velocity goal (e.g. 40% and 30%). To keep these tasks from falling into SYSOTHER, we've assigned them to NEWWORKV and report class RNETV. See TUNING Letter 2002, No. 3 and No. 5.

OSDI - Oracle

Assign the Oracle address space to the SERVERS service class. Assign the OSDI subsystem default to NEWWORK and report class ROSDI. See TUNING Letter 2003, No. 2.

SAP - SAP R/3

Assign the SAP address space to the SERVERS service class. Assign the SAP subsystem default to NEWWORK and report class RSAP.

SOM - System Object Model

Assign the SOM address space to the SERVERS service class. Assign the SOM subsystem default to NEWWORKV with a report class called RSOM.

When the volume of the report class becomes significant (normally 10% of the machine or more), then assign that subsystem to its own service class, or a service class with similar goals.

KILLIT

The Quickstart Policy contains a new service class that people have found especially useful. The KILLIT service class can be used by operators to (almost) stop non-swappable work. Swappable work can be swapped out by "e job,quiesce," but the quiesce command has little effect on non-swappable work (other than to lower its priority to be less than discretionary). The Quickstart Policy has defined a resource group called KILLIT that has one service unit defined as the maximum for all service classes assigned to it. The KILLIT service class is assigned to that resource group. Moving a job to KILLIT will really slow down its progress, even if it won't stop it completely. Swappable work can still get cycles under KILLIT, but not a lot. Non-swappable work will still get cycles, but far fewer than using quiesce. The command will simply be "e job,srvclass=killit."

MODIFICATIONS TO THE QUICKSTART POLICY

There are several modifications that you'll want to make to the Quickstart Policy either initially or after you've collected data under goal mode.

Initial Definitions

1. Determine the batch job classes for production and test and modify the transaction class groups as defined in the Quickstart Policy.

2. Determine the started task names and online transaction names for the transaction name groups as described earlier. This includes your high importance started tasks, online regions, monitors and operations jobs.

3. If you run test batch as multi-period work in a single jobclass rather than different jobclasses, then set up a single test batch service class with multiple periods to use instead of TSTBATHI, TSTBATMD and TSTBATLO. We would HIGHLY recommend using single period batch jobs, however.

4. If you currently run TSO with more than two periods, you may want to experiment in compat mode with two periods before going to goal mode. Use your current average response time from RMF or CMF to set response goals for each period if you've been collecting the data or have already set response objectives.

5. If you've been running CICS V4+, IMS V5+ or DDF in compatibility mode, use the average response times you collected from the performance groups in compatibility mode to set percentile response time goals for the service classes in goal mode.

6. If you're running many online systems in a constrained system, then consider setting up multiple service classes instead of assigning them all to ONLPRD. You could set up three service classes, for example, with velocities of 31%, 40% and 50% depending on the velocities that are seen during compatibility mode. Use a minimum number of velocities. That is, if four performance groups currently have measured velocities of 46%, 49%, 52% and 54%, assign them all to a single velocity goal of 50%. The 31% mentioned above is to prevent this service class from contributing to discretionary if the work is exceeding its goal significantly.

7. We have set our service definition coefficients (SDCs) to:

   CPU=1.0, SRB=1.0, IOC=0.1, MSO=0.0.

   If you don't currently have a reporting system that uses service units, you can start with our recommendations. If you report by service units, you may want to change these coefficients back to your own values until you can convert your reporting programs. Once you've converted your report programs (we'd suggest changing them to use base units; i.e. divide by the SDC), you are ready to change the SDCs. If you've been using other SDCs, you may also need to alter your duration values for multi-period service classes. This is because the duration is the sum of all service units (CPU, SRB, IOC and MSO) and if you change the coefficients, you'll change the rate at which jobs accumulate service units. If you move to 64-bit mode, it is imperative you reduce the MSO to 0.0 (or at least 0.0001).

   Our article called "Positioning for Goal Mode" on our Web site describes how to calculate the new durations when you alter the SDCs.

8. Immediately after activating goal mode, review the NEWWORK and NEWWORKV service classes. If they are taking much CPU time (over 5% or 10% of the machine), then investigate the report classes to see what type of work is predominate. That work may need to be moved into its own service class with unique goals.

9. If you are running on a machine with very few logical CPUs, then you may need to lower all of the velocities. A velocity of 70 on a 6-way machine might translate to a velocity of 30 on a 2-way machine. We've given you starting values, but they'll need to be monitored closely.

OTHER MODIFICATIONS

Here's a list of some other changes you might want to make to the Quickstart Policy as you need them.

1. First, you should make sure that you've set the current response times accurately. Compare the response times after activating goal mode to those during compatibility mode. Be sure you understand any differences. If some workloads are getting poorer service in goal mode than compatibility mode, perhaps it's because another workload was given too good a goal. Make goal adjustments before attempting other changes, but review the following modifications in case they could help solve a problem.

2. After you've been collecting response times for a short time (a day to a week) in goal mode, you can change any average response time goals you might have defined to percentile response time goals. The RMF/CMF reports will show you what percentiles are currently being met. Percentile response goals allow you a greater possibility of meeting your goals consistently because a single long-running transaction (job in loop, etc.) will simply fall into the percent of jobs that don't complete within the goal. On the flip side, however, if you're missing the goal for percentile responses, you're probably severely constrained (or have set the goals wrong). Another possibility is that you don't have a high enough volume of work in the service class period.

3. If you have problems quickly starting your online system in the middle of the day after an unscheduled outage, consider this suggestion from Gary Hall of Washington Systems Center. He suggests defining a two period service class for these regions with the first period long enough to complete initialization and set with a very high velocity goal. Second period would then run at the normal velocity for your online region. This technique isn't necessary for online systems that are assigned to SERVERS and that will be managed by transaction goals.

4. DB2 is a very resource intensive subsystem, and some installations will want to keep DB2 in its own service class. It will need to be set up in a service class with a velocity goal. Remember that much of the DB2 activity is reflected in the caller's address space (CPU time and paging). Starting with MVS SP 5.2 and DB2 V4 (with the Distributed Data Facility, DDF), users can set response goals for distributed DB2 transactions.

5. Determine whether you need to add application environments for WebSphere, IBM's HTTP Server or other subsystems. If so, add them at this time. We did not feel like we could add a generic application environment in the Quickstart Policy that would apply to multiple sites. For the same reason we did not add any scheduling environments. These are facilities to add later if you need them.

SUMMARY

We've just described a service policy that can get you started into goal mode. Just remember the original concept of "keep it simple," and you'll find your migration will be easy and produce a very usable service policy.

BIBLIOGRAPHY

Basics:

IBM, "MVS/ESA Planning: Workload Management," GC28-1761 (OS/390) or SA22-7602 (z/OS).

C. Watson, "WLM Goal Mode," Cheryl Watson's TUNING Letter March/April 1995.

IBM, "WLM Migration Guide and Checklist V3.0," at **www.ibm.com/zseries/zos/wlm**.

Steve Samson, "MVS Performance Management - z/OS Version 1 Edition," **www.setsystems.com/cgi-bin/buy-mvs**.

P. Enrico and E. Berkel, "Effective Use of MVS WLM Controls," Cheryl Watson's TUNING Letter March/April 1995 and CMG Trans. 87, 21-38 (1995).

C. Watson, "WLM Update," Cheryl Watson's TUNING Letter 1998, No. 2.

For further reading:

C. Watson, "WLM/SRM Update," Cheryl Watson's TUNING Letter 2003, No. 1.

C. Watson, "WLM in R10," Cheryl Watson's TUNING Letter 2000, No. 4.

C. Watson, "WLM Update," Cheryl Watson's TUNING Letter 1999, No. 4.

C. Watson, "WLM Classification," "A Quickstart Policy," and "WLM Measurements," Cheryl Watson's TUNING Letter May/June 1995.

C. Watson, "How to Set Velocities," Cheryl Watson's TUNING Letter 1996, #3.

IBM, "WLM Performance Studies," GG24-4352.

Workload	Service Class	Imp	Response Type	Goal
SYSTEM	SYSTEM		System goal	.
SYSTEM	SYSSTC		System goal	.
SYSTEM	SYSOTHER		Discretionary	.
STC	STCMD	3	Velocity	40%
STC	STCLO		Discretionary	.
ONLINE	ONLPRD	1	Velocity	50%
ONLINE	ONLTST	5	Velocity	10%
ONLINE	TRANHI	1	Percentile response	80% < .5 sec
ONLINE	TRANLO	3	Percentile response	50% < 10 min
ONLINE	SERVERS	1	Velocity	70%
TSO	TSOPRD - Period 1, Dur=800	2	Percentile response	80% < .3 sec
TSO	- Period 2	4	Velocity	40%
PRDBAT	PRDBATHI	3	Velocity	40%
PRDBAT	PRDBATMD	4	Velocity	5%
PRDBAT	PRDBATLO	5	Discretionary	.
PRDBAT	HOTBATCH	2	Velocity	50%
TSTBAT	TSTBATHI	3	Percentile response	90% < 10 min
TSTBAT	TSTBATMD	4	Percentile response	80% < 30 min
TSTBAT	TSTBATLO		Discretionary	.
NEWWORK	NEWWORKV	3	Velocity	20%
NEWWORK	NEWWORK - Period 1, Dur=500	3	Percentile response	80% < .5 sec
NEWWORK	- Period 2	4	Velocity	20%

Figure 1 - Quickstart Service Policy (IMP=0 is system; IMP=6 is discretionary)

Figure 2 - Classification Rules

Subsystem	Rule	Value	Mgmt. Goals	Service Class	Report Class
ASCH	Default	.		NEWWORK	RASCH
CB	Default	.		NEWWORK	RCB
CICS	Default	.		TRANLO	RCICS
CICS	SI	CICSP*		TRANHI	RCICS
CICS	TNG	TRANCIC		TRANHI	RCICS
DB2	Default	.		NEWWORK	RDB2SQ
DDF	Default	.		NEWWORK	RDDF
IMS	Default	.		TRANLO	RIMS
IMS	SI	IMSP*		TRANHI	RIMS
IMS	TCG	TRANIMS		TRANHI	RIMS
IWEB	Default	.		NEWWORK	RIWEB
JES	Default	.		TSTBATLO	RTBATLO
JES	TCG	PRDBATMD		PRDBATMD	RPBATMD
JES	TCG	PRDBATHI		PRDBATHI	RPBATHI
JES	TCG	TSTBATHI		TSTBATHI	RTBATHI
JES	TCG	TSTBATMD		TSTBATMD	RTBATMD
LSFM	Default	.		NEWWORKV	RLSFM
MQ	Default	.		NEWWORKV	RMQ
NETV	Default	.		NEWWORKV	RNETV
OMVS	Default	.		NEWWORK	ROMVS
OSDI	Default	.		NEWWORK	ROSDI
SAP	Default	.		NEWWORK	RSAP
SOM	Default	.		NEWWORKV	RSOM
STC	Default	.		STCLO	RSTCLO
STC	TN	GRS		SYSTEM	RGRS
STC	TN	CATALOG		SYSTEM	RCATALOG
STC	SPM	SYSSTC		SYSSTC	RSTC
STC	SPM	SYSTEM		SYSTEM	RSTC
STC	TNG	STCHI	Region	SYSSTC	RSYSSTC
STC	TNG	STCMD	Region	STCMD	RSTCMD
STC	TNG	SERVERS		SERVERS	RSERV
STC	TNG	MONITORS	Region	MONITORS	RMON
STC	TNG	ONLPRD	Region	ONLPRD	RONLINE
STC	TNG	ONLTST	Region	ONLTST	RONLINE
TSO	Default	.		TSOPRD	RTSO

Classification Group	Type	Group Entries
STCHI	TNG	VTAM, NPM, NFS*
		JES*
		TSO (TCAS)
		RACF/CA-ACF2/TOPSECRET
		Auto ops package
		APPC & ASCH address spaces
		DLF, LLA, VLF
		IRLM
		MIM, CB*
		SMS, SYSBMAS
		JES2AUX, ANTMAIN, ANTAS000
		TRACE, PCAUTH
		OMVS kernel
ONLPRD	TNG	DSN* (DB2 production regions)
		ADABAS*
		IDMS*
		*DIST (if not using DDF subsystem)
		ORA* (if not using OSDI subsystem)
		MQ* (if not using MQ subsystem)
STCMD	TNG	your scheduler
		your spooler programs
		your important operations STCs
		OMVS daemons
ONLTST	TNG	your online test regions (CICS, IMS, DB2)
MONITORS	TNG	RMF*
		OMON*
		NETV*
		TMON*
		CMF*
		CA*
		SDSF
SERVERS	TNG	CICS4*
		CICSTS*
		IMS5*
		IMS6*
		IMS7*
		*DIST (if using DDF subsystem)
		ORA* (if using OSDI subsystem)
		MQ* (if using MQ subsystem)
PRDBATHI	TCG	your hot prod batch jobclasses
PRDBATMD	TCG	Your normal prod batch jobclasses
PRDBATLO	TCG	Your hog prod batch jobclasses
TSTBATHI	TCG	your hot test batch jobclasses
TSTBATMD	TCG	your medium test batch jobclasses
TRANCIC	TNG	important CICS transactions
TRANIMS	TCG	Important IMS classes

Figure 3 - Quickstart Classification Groups