CICS Transaction Server for z/OS
5.5

CICSPlex SM Managing Workloads

IBM

Note

Before using this information and the product it supports, read the information in “Notices” on page 171.

This edition applies to the IBM® CICS® Transaction Server for z/OS® Version 5 Release 5 (product number 5655-Y04) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this PDF

This PDF provides administration information for the CICSPlex® SM component of CICS Transaction Server for z/OS. It describes how to establish and maintain the CICSPlex SM definitions necessary to perform workload management.

For details of the terms and notation used, see Conventions and terminology used in the CICS documentation in IBM Knowledge Center.

Date of this PDF

This PDF was created on 2024-04-22 (Year-Month-Date).

Chapter 1. Workload management

Workload management generally means effectively distributing work around a system so that it can run in the most efficient way. In the context of CICS workload, it is the process of routing work to CICS regions so that the work can be efficiently executed. Workload management can occur ***before*** the work reaches CICS: for example, by selecting a CICS terminal-owning region (TOR) as the start point for a transaction. Or it can occur ***after*** the work arrives in CICS: for example, by selecting an appropriate region in CICS (typically, an application-owning region, or AOR) to which the transaction is routed. With CICS, there are a number of options for workload management.

What options do I have for workload management?

CICSPlex SM WLM

CICSPlex SM WLM is the CICS implementation of workload management. z/OS Workload Manager and Sysplex Distributor are concerned with sysplex efficiency, high availability and efficiency of TCP/IP, which typically occur before work reaches CICS. CICSPlex SM WLM is concerned with the efficiency of work processing after it reaches CICS. CICSPlex SM selects the most efficient region to which the work will be sent.

If you run more than about three or four AORs, you can benefit from CICSPlex SM WLM. CICSPlex SM gives the most granular control over workload execution in CICS and, for that reason, it is the most widely-adopted workload management solution. For more information, see How it works: CICSPlex SM workload management (WLM).

z/OS Workload Manager (WLM)

The Workload Manager component of z/OS monitors a sysplex and determines how much resource should be given to each item of work in the sysplex to try to meet user-defined goals. For more information, see How it works: z/OS Workload Manager (WLM).

Sysplex Distributor

Sysplex Distributor is a component of IBM z/OS Communications Server and provides TCP/IP load balancing across a sysplex (that is, multiple LPARs). It can be combined with TCP/IP port sharing. Sysplex distributor is a combination of the high availability features of distributed DVIPA and the workload optimization capabilities of WLM. The TCP/IP stacks can be configured to request workload information from WLM, enabling the distributing stack to forward those connections that are based on the workload of each of the target stacks. The WLM workload information is based on a comparison of available general CPU capacity for each target system. For more information, see Sysplex distributor in z/OS Communications Server: IP Configuration Guide.

Write your own solution

Although less commonly-used than the IBM-provided solutions, it is possible to write your own means of distributing workload. In CICS, this is done through a user-written dynamic routing program. For more information, see Writing a dynamic routing program.

Your options are not mutually-exclusive. You can use any combination of z/OS Workload Manager, Sysplex Distributor, and CICSPlex SM. For example, you might have a TCP/IP workload that uses Sysplex Distributor in a high-availability context to route incoming work into a CICS TOR. You might then use CICSPlex SM WLM to determine the most effective regions for the execution of the work. Or, you might choose to use only CICSPlex SM to control your CICS workloads.

How it works: CICSPlex SM workload management (WLM)

Typically, if you run more than three or four AORs, you can benefit from using CICSPlex SM WLM to manage workloads. In the context of CICSPlex SM, workload management (WLM) means ensuring that, when a transaction arrives in a CICS region or when a program is linked-to from a CICS region, the most optimal CICS region is dynamically selected and the work is routed to that region to be run.

This section introduces the key concepts of CICSPlex SM WLM and a high-level view of what is involved in setting up CICSPlex SM WLM.

Setting up CICSPlex SM WLM

Let's look at who is involved in setting up CICSPlex SM WLM. Figure 1 on page 2 shows the basic topology.

- The software architect determines the goals for workload management and determines any constraints to be applied to specific workloads: for example, separating - or linking - certain types of work.
- The software architect identifies the work requests that are eligible for dynamic routing using CICSPlex SM WLM.
- The system administrator configures the components of CICSPlex SM, using either CICS Explorer or the CICSPlex SM WUI. The system administrator creates groups of CICS regions to participate in CICSPlex SM workload management, and specifies how each workload will be managed.

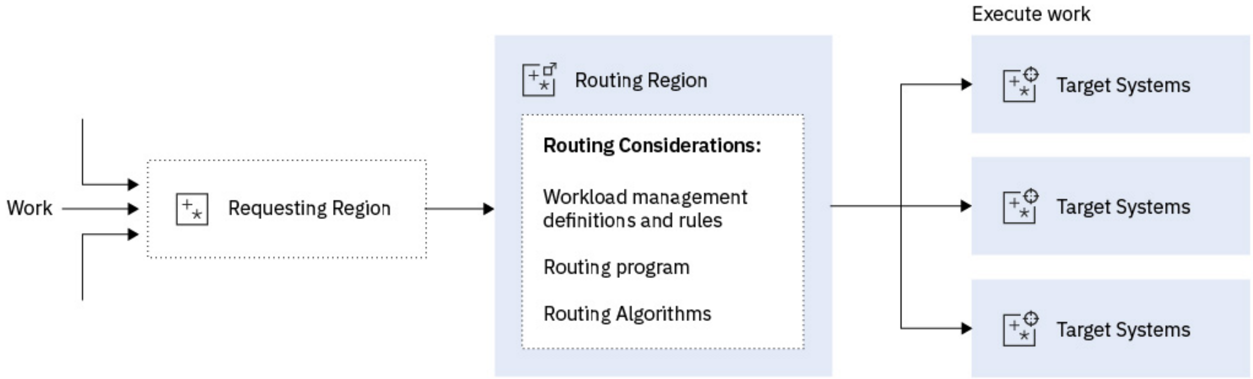


Figure 1. Basic topology of CICSPlex SM WLM

Requests for work to be managed by CICSPlex SM WLM are made from a region in a CICSplex.

A specification for each workload defines the CICS systems to be treated as routing regions and the target regions for processing the workload. Workload management rules provide the granularity that is needed to handle special workload conditions: for example, making sure that certain types of work are separated from each other, or for maintaining relationships (affinities) between certain types of work.

CICSPlex SM uses dynamic routing, so a routing program is needed to decide where to run a piece of work. In CICSPlex SM, the user-replaceable dynamic routing program EYU9XLOP provides this function,

Routing algorithms provided by CICSPlex SM WLM take into account a complex array of factors to select the optimal region for processing the incoming work request. These factors include:

- How loaded the available CICS regions are.
- How healthy the available CICS regions are.
- How responsive the available CICS regions are.
- Where the available CICS systems are and how they are connected.

The work is passed to one or more target regions for processing.

CICSPlex SM workload management: find the information that you'll need

To work with CICSPlex SM WLM, you'll need the following parts of the IBM Knowledge Center.

Table 1. Where's the information about CICSPlex SM WLM?

Topic	Where to look
Find out what you need to consider about your workload before you can manage it with CICSPlex SM WLM	Workload considerations
Find out your options for routing work between regions	Routing considerations
Find out how CICSPlex SM WLM works	Components of CICSPlex SM WLM and How CICSPlex SM WLM routes work
Set up CICSPlex SM WLM	Creating workload management definitions and Workload management tasks
Manage workloads in CICS Explorer	Managing workloads in the CICS Explorer product documentation

Workload considerations

Before you set up CICSPlex SM WLM, you need to understand whether your workload puts any constraints on its management. In workload management, the ideal situation is to have each workload capable of running in any CICS region. CICSPlex SM WLM manages the work and routes it to the most appropriate target region, taking into account factors such as region load and health. But some workloads have more complex needs that are determined by the design of the application or the CICS environment that is available to process the workload. For example, you might want certain types of workload to run only in certain regions. Or you might have to keep the processing of certain transactions together in the same region because of relationships, known as affinities, that you defined between those transactions.

Workload balancing

Workload balancing describes workloads that are under the control of CICSPlex SM WLM. It is important to understand that workload balancing does not balance the work equally between each of the available regions. It routes work to the region that is best able to process that work. It is more sophisticated than simple *round-robin routing*.

For example, let's say that your environment has nine AORs. The incoming work - for example, a transaction that is called DEMO - can be routed to any one of the nine available AORs to run. In round-robin routing, the first DEMO transaction is routed to AOR 1. The second DEMO transaction is routed to AOR 2, and the third to AOR 3, and so on. Each region gets an equal part of the incoming work.

In the same example, CICSPlex SM WLM considers the state of all nine regions: that is, their health and availability and how they are connected. Using this information, it decides which region will process the work. It might choose AOR 6 to run the DEMO work. When the next transaction comes in, it might choose AOR 4. Or it might decide that the first 20 transactions can all be routed to AOR 1. The point is that CICSPlex SM controls the routing. The routing decision is made on the overall suitability of the region to run the work, not on equally sharing work between all available regions. As such, it is substantially different to round-robin routing.

Initially, round-robin routing can appear to give a more equal - or balanced - pattern of distribution and utilization of regions. But this approach is too simplistic for complex workloads. For example, how does round-robin routing address a situation where one of the regions suddenly becomes unhealthy? For this reason, CICSPlex SM WLM does not offer round-robin routing.

Workload separation

Workload separation is used to limit certain types of work to certain target regions. A named region, or more typically, a named set of regions, are the available targets for a named set of transactions. For example, you might want only to process Payroll transactions in certain regions.

CICSPlex SM WLM supports workload separation. You can separate transaction and programs by transaction, by the terminal ID and user ID associated with a transaction or with a program occurrence, or by the process type associated with the CICS BTS activity. Work is directed to different sets of target regions, where the activity is then balanced across the regions within the set.

Workload affinities (inter-transactional affinities)

With *workload affinities*, two or more transactions have a relationship that you define and that exists for the duration of that relationship. When an affinity relationship exists between transactions, these transactions must be processed by the same target regions. The route of work from the requesting region to the target region is based on the rules that apply to the particular combination of the affinity relationship between the transactions and the duration for which those rules apply.

For example, transaction AAAA writes to a temporary storage queue. The next transaction AAAB needs to read data from that queue. Transactions AAAA and AAAB have an affinity, so they must be routed to the same CICS region for processing. If their affinity isn't taken into account by the workload manager, transaction AAAA could be routed to one AOR and transaction AAAB to another AOR where it abends when it tries to read data from a TS queue that doesn't exist in that region.

Operationally, it is always a good idea to minimize affinities. Fewer affinities mean few potential points of failure for the workload and more flexibility on workload routing. For information about detecting affinities and programming techniques that affect affinities, see Affinity.

CICSPlex SM WLM can process all five types of affinities relationships that you can define in CICS:

- Global affinity relationship: the relationship is between all instances of all the transactions that are started from any terminal, by any START command, or by any CICS BTS process.
- LUname (terminal) affinity relationship: the relationship is between all instances of all the transactions in the group that are associated with the same terminal.
- Userid affinity relationship: the relationship is between all instances of the transactions that are initiated from a terminal, by a START command, or by a CICS BTS activity, and executed on behalf of the same user ID
- BAPPL affinity relationship: the relationship is between all instances of transactions that are associated with the same BTS process.
- LOCKED affinity relationship: the relationship is between all instances of transactions in the group that are associated with dynamically-linked programs that have the same unit of work.

And the following affinity durations:

- Activity: the affinity lasts as long as the associated activity exists.
- Delimit: the affinity continues until a transaction with a pseudoconversation mode of END is encountered.
- Logon: the affinity lasts as long as the terminal remains logged-on to CICS.
- Pseudoconversation (PCONV): the affinity lasts for the whole pseudoconversation.
- Permanent: the affinity extends across all CICS restarts.
- Process: the affinity lasts as long as the associated process exists.
- Signon: the affinity lasts as long as the user is signed-on.
- System: the affinity lasts as long as the target region exists and ends when the target region terminates.
- Unit-of-work (UOW): the affinity lasts as long as the unit-of-work is active.

What type of work can work CICSPlex SM WLM handle?

CICSPlex SM WLM can route the following types of work:

- Transactions that are invoked at a terminal
- Transactions that are associated with CICS Business Transaction Services (BTS) activities
- Eligible transactions that are invoked using the **EXEC CICS START** command, either with or without an associated terminal
- Distributed Program Link (DPL) requests, including:
  - EXCI calls
  - CICS Transaction Gateway ECI calls
  - Distributed Computing Environment (DCE) remote procedure calls (RPCs)
  - Open Network Computing (ONC) remote procedure calls (RPCs)
  - Any function that issues an **EXEC CICS LINK PROGRAM** request
  - Link3270 requests

Routing considerations

You can choose to have work always run in a specified region. This is known as static routing. CICSPlex SM WLM uses dynamic routing to control where work requests are run. You can implement dynamic routing in a hub model or a distributed model.

Dynamic routing compared to static routing

With *static routing*, work always runs in a specified region. In a CICSplex or BTS-set, resources such as transactions and programs required in one region might be owned by another. For example, you might have a terminal-owning region (TOR) that requires access to transactions owned by an application-owning region (AOR). If you specify the location of a resource when you design your system (for example, in the installed resource definition) requests of that resource are always routed to the same region.

If you have relatively few CICS regions, static routing might be appropriate for you. However, there are considerations with specifying an exact SYSID for routing purposes:

- The route can only be made to the specified SYSID. If that region is unavailable, unresponsive, or unhealthy, this can cause problems.
- If the SYSID changes, the code that specifies it must also be changed.

With *dynamic routing*, the decision on where to run a piece of work is made by a dynamic routing program. In CICSPlex SM, the program is a user-replaceable dynamic routing program called EYU9XLOP.

CICSPlex SM WLM dynamic routing is application-agnostic. Values are not coded into the application. It does not require specialist application code to handle different circumstances. It provides flexibility for movement between environments and changes in names in environments. For example, a change in the number of regions can be handled with only a quick change to CICSPlex SM WLM instead of the change to application source code that static routing would require.

What are the roles of CICS regions in dynamic routing?

The CICS regions involved in dynamic routing can act as one or more of the following:

Requesting region

The CICS region where the work request is initiated. For terminal-initiated transactions and for inbound DPL client requests, the requesting region is typically a terminal-owning region (TOR). For **EXEC CICS START** commands that are associated with a terminal, for peer-to-peer DPL requests, for non-terminal related **EXEC CICS START** commands, for CICS BTS processes and activities, and for Link3270 bridge requests, the requesting region is typically an AOR.

Routing region

The CICS region that decides where to route the work request. For terminal-initiated transactions and terminal-associated **EXEC CICS START** commands, for CICS CICS BTS processes and activities, and for Link3270 bridge requests, the routing region is typically an AOR.

Target region

The CICS region where the request is executed. For all dynamically-routed transactions, programs, and BTS processes and activities, the target region is typically an AOR.

A region can be both a routing and a target region.

EYU9XLOP: the CICSPlex SM dynamic routing program

CICSPlex SM WLM uses a user-replaceable dynamic routing program called EYU9XLOP to create the environment necessary for dynamic routing and to set up the CICSPlex SM runtime environment.

For most situations, the supplied workload management capabilities are sufficient. However, if it is ever needed, you can customize the module that drives CICSPlex SM workload management processing. For more information, see Creating a user-replacement module for EYU9WRAM.

The components of workload management in CICSPlex SM

CICSPlex SM WLM uses a hierarchy of objects to define and control workloads.

In the CICSPlex SM WUI, each component has an equivalent views. Those views are shown in parentheses in the diagram. In CICS Explorer, some components are combined and the views show information from several components. For example, in CICS Explorer, a Workload (routing) Rule is the term used to describe the information from a Workload Definition (WLMDEF) and Transaction Group (TRANGRP) pair.

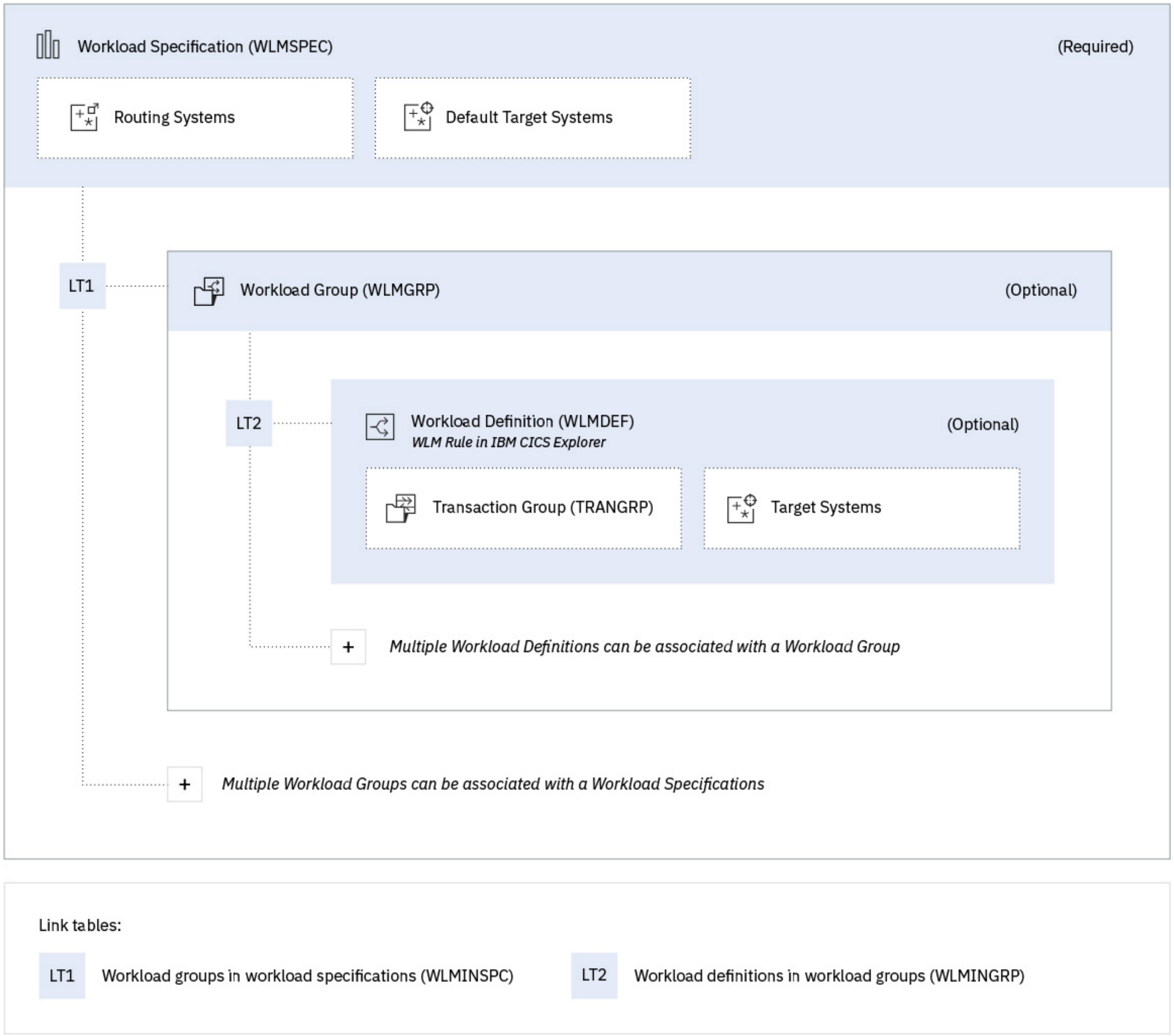

Workload Specification (WLMSPEC)
(Required)
Routing Systems
Default Target Systems
LT1
Workload Group (WLMGRP)
(Optional)
LT2
Workload Definition (WLMDEF)
WLM Rule in IBM CICS Explorer
(Optional)
Transaction Group (TRANGRP)
Target Systems
+
Multiple Workload Definitions can be associated with a Workload Group
+
Multiple Workload Groups can be associated with a Workload Specifications
Link tables:
LT1
Workload groups in workload specifications (WLMINSPC)
LT2
Workload definitions in workload groups (WLMINGRP)

Figure 2. Workload management objects and associations

Workload Management Specification (WLMSPEC)

Each workload has a workload management specification. The WLM specification:

- Names the workload
- Defines a set of CICS systems to be treated as routing regions. The CICS regions can either be associated directly with the WLMSPEC or a set of CICS regions can be defined in a CICS System group.
- Defines the default target regions for the workload
- Optionally, if a workload affinity exists, defines a default affinity relationship and duration

The workload management specification is defined in CICSPlex SM as a WLMSPEC. For more information about WMSPEC and its attributes, see WLMSPEC resource table.

Workload Management Definition (WLMDEF)

If a workload either requires work to be separated or the workload has affinities, a workload management definition (known in CICS Explorer® as a *WLM rule*) is used.

When no workload management definitions are defined and installed for a workload, all transactions and programs are routed to the target regions that are identified in the default target scope. Default target regions are specified in the Workload Management Specification (WLMSPEC).

If one or more workload management definitions is defined and installed for a workload, all transactions and programs that match the criteria, as defined on each WLMDEF, are routed to the target scope that is defined on that WLMDEF.

The workload management definition is defined in CICSPlex SM as a WLMDEF. For more information about WLMDEF and its attributes, see WLMDEF resource table.

Workload Management Group (WLMGROUP)

A workload management group is required only if you define more than one workload management definition (WLMDEF). It groups multiple workload management definitions.

If you have only one workload management definition, you can link that definition to a workload management group, or you can exclude the workload management group and link the sole workload management definition directly to the workload management specification.

The workload management group is defined in CICSPlex SM as a WLMGROUP. For more information about WLMGROUP and its attributes, see WLMGROUP resource table.

Transaction group (TRNGRP)

A transaction group (TRNGRP) is an association of logically similar transactions. The similarity can be based on affinity requirements, common shared processing requirements of the application, or any other user-determined characteristic. The transaction group is included as part of the workload management definition (WLMDEF). In turn, the WLMDEF defines the CICS system group that is to be used as the target region.

Any transaction that is not specifically associated with one or more of your transaction groups is assigned to the default transaction group.

The transaction group is defined in CICSPlex SM as a TRNGRP. For more information about TRNGRP and its attributes, see TRANGRP resource table.

Link tables

Link tables are used internally by CICS to store the associations of components of CICSPlex SM WLM with each other. The link tables are shown in this diagram for completeness and because the WUI surfaces their terminology when you associate your WLM components. You don't need to be concerned about them in the context of understanding the components of WLM and you will not see their terminology in CICS Explorer.

How CICSPlex SM WLM chooses where to route work

When a transaction is initiated, CICSPlex SM uses algorithms to weigh up the factors that contribute to a routing decision and select the target region that is likely to deliver the best performance. There are four algorithms. You specify a default routing algorithm on the workload specification. You can choose to override this default for a transaction group by specifying a different routing algorithm on the transaction group definition.

The four routing algorithms

CICSPlex SM uses one of the following four algorithms to determine which target region processes the work:

- Queue
- Goal
- Link neutral queue

- Link neutral goal

If any transaction affinities are outstanding for the transaction that is being routed, the affinity target region is selected, regardless of the algorithm used.

You specify a default routing algorithm on the workload specification (WLMSPEC). Optionally, you can specify a routing algorithm at the transaction group (TRANGRP) level. An algorithm that is specified at the transaction group level overrides the default algorithm that is associated with the workload specification.

To change the default routing algorithm on the workload specification for a running workload, you must shut down all of the CICS regions that are associated with the workload. This refreshes the workload. You can change the routing algorithm on the transaction group dynamically. You do not have to shut down the CICS region.

Note:

You might see the use of a particular algorithm described as a *routing mode*: for example, goal routing mode (or even more simply, goal mode) to describe a configuration that uses the goal algorithm.

Queue algorithm

This is the most commonly used algorithm. If the transaction that is being routed has no active affinities, this algorithm distributes the workload across a set of target regions. Selection criteria include the current task load, the health state, and the type of connection between the router and the target region.

If the transaction that is being routed has no active affinities, the queue algorithm causes CICSPlex SM to select the target region that conforms to the following set of criteria:

- The region has the shortest queue of work waiting to be processed, relative to the maximum number of tasks permitted in the target region. This queue of work, also called the *load count*, is the count of all active and queued user tasks. By default, tasks queued for both MAXTASKS and TRANCLASS attributes are included in the load count. Use of the "Task load queue mode" attribute in the CSYSDEF resource table allows sites to exclude tasks queued for the TRANCLASS attribute from the load count. For more information about the "Task load queue mode" attribute in the CSYSDEF resource table, see CICS system definitions - CSYSDEF.
- The region is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- The region is the least likely to cause the transaction to stop.
- The region is the shortest path for inter-region communication.
- The region has a z/OS WLM health value in the range 1-100 (see "The z/OS WLM health state of the region, only when the z/OS WLM health service is on." on page 10).

Goal algorithm

This algorithm selects the target region based on a predefined goal for response time. If the goal algorithm does not identify a target region, the queue algorithm is applied to the remaining target regions. Success with this algorithm depends on setting appropriate and achievable response time goals. Failing to define appropriate response time goals can result in unexpected routing patterns.

Goal mode is better suited to users with experience of CICSPlex SM WLM who use a single LPAR and have very specific requirements around response time goals.

Link neutral queue algorithm

The link neutral queue algorithm works like the queue algorithm, except that the type of connection between the routing and target region is not considered. The connection type weighting is ignored.

Like the queue algorithm, this algorithm can accommodate differences in processor power and different maximum task values in the target regions. However, it does not allow for communications link speeds between the router and a target. Compared to the queue algorithm, this algorithm might spread a workload across a target scope more evenly, but the workload might not complete as quickly.

Be cautious about setting the default routing algorithm on the workload specification to one of the link neutral algorithms because it will apply to *every* dynamic transaction. If you want the fastest transactions to continue to be processed locally but you want dynamic transactions that request services from MVS subsystems to be processed anywhere, consider the queue algorithm as the default routing algorithm on the workload specification. Then apply the link neutral algorithm to a subset of transactions at the transaction group level.

Link neutral goal algorithm

The link neutral goal algorithm works like the goal algorithm, except that the type of connection between the routing and target region is not considered. The connection type weighting is ignored.

Be cautious about setting the default routing algorithm on the workload specification to one of the link neutral algorithms because it will apply to *every* dynamic transaction. If you want the fastest transactions to continue to be processed locally but you want dynamic transactions that request services from MVS subsystems to be processed anywhere, consider the queue algorithm as the default routing algorithm on the workload specification. Then apply the link neutral algorithm to a subset of transactions at the transaction group level.

Factors that contribute to CICSPlex SM dynamic workload routing decisions

Each factor that contributes to a workload routing decision is given a weighting by CICSPlex SM. Usually, the target region with the lowest weighting after all factors are considered is selected for routing. If all available target regions have equal weightings and are, therefore, assessed as being equally capable of running the work, a target region is selected randomly from the group. So, in systems that are lightly loaded, there is no predetermined order in which work is allocated to equally capable target regions.

The number of tasks in the region

Factored as a percentile value, which is calculated by dividing the target region MAXTASKS setting with the current task count.

The health status of the region

Factored by assigning arithmetic weights, depending on whether the region is short-on-storage, taking a transaction dump, taking a system dump, running at its MAXTASKS limit, in a non-responsive state, or in a CICSPlex SM stall.

Note: Health is assessed independently of system availability monitoring (SAM) and is not influenced by SAM settings.

Connection type (link) between the routing region and the target region

Factored by assigning arithmetic weights, depending on whether the target region is linked to the target by an MRO connection, an XCF connection, an LU6.2 connection, an IPIC connection, or if the target region is the router itself. The connection type is not used by the two link-neutral algorithms.

Weight is allocated to each connection in the following order of priority:

1. Local
2. MRO/IRC and MRO/XM (local LPAR)
3. MRO/XCF (remote LPAR)
4. IPIC (local LPAR)
5. IPIC (remote LPAR)
6. LU6.2
7. Indirect

For example, if all other factors are equal, a target region that is connected to its requesting region by using MRO/XCF is preferred (that is, it has the lowest weighting allocated) to a target region that is connected by using IPIC. IPIC connections between routing and target regions in the same LPAR have a lower weighting than IPIC connections to target regions in a different LPAR, so that if other factors are equal, local IPIC connections are preferred to remote IPIC connections.

The z/OS WLM health state of the region, only when the z/OS WLM health service is on.

Factored by assigning penalizing weights, depending on the z/OS WLM health value of a region provided by the z/OS WLM health service. The z/OS WLM health state of a region is reflected by a z/OS WLM health value in the range 0 - 100:

- A region with a z/OS WLM health value of zero is not eligible to receive work. CICSPlex SM WLM does not route work to this region. Any workload affinities that are associated with this region remain and are honored; however, workload routing based on those affinities fails while the health value is zero, and results in either message EYUWR0003W or a SYSIDERR condition. Message EYUWR0003W indicates that the EYU9WRAM program was in effect and the affinity AOR was not available. To resolve such routing failures, initiate a warm-up for the region so that the health value is non-zero, and then rerun your workload. See Initiating a CICS system warm-up.
- For a region with a z/OS WLM health value in the range 1 - 99, penalizing weights are assigned in the routing algorithm. The greater the health value, the lower the penalizing weight. So, target regions with a greater health value are more favorable in the routing decision. A region that is assigned a high penalizing weight remains active and might still receive work if no other healthier regions are available in the same scope.
- A region with a z/OS WLM health value of 100 is not assigned with penalizing weights and is fully capable of receiving work.

During the CICS warm-up or cool-down process, CICS adjusts the z/OS WLM health value of the region. This affects the routing decision of CICSPlex SM WLM, which restricts workload into the region during that process. In general, when a region is warming up, its health value increases periodically, and the region might gradually receive a larger workload; when a region is cooling down, its health value decreases periodically, and the region might gradually receive less work until it becomes ineligible as a target. For more information about what happens during CICS warm-up or cool-down, see CICS warm-up and cool-down by use of z/OS Workload Manager health service.

You can deliberately make a region ineligible as a workload routing target in CICSPlex SM routing decisions by setting its z/OS WLM health open status to `CLOSE` in the **MVS workload management** (MVSWLM) view. This gradually makes the region less favorable until the region's z/OS WLM heath value drops to 0, indicating that the region becomes ineligible. Additionally, you can change a region's z/OS WLM heath value to zero immediately, by setting its z/OS WLM health open status to `IMMCLOSE`.

Outstanding CICSPlex SM Realtime Analysis (RTA) Events that are associated with the workload, if any

Factored by assigning arithmetic weights that depend on the severity of the events outstanding. These events are factored in only when Event name is specified in the WLM specification for the workload or any Transaction Group definitions that are associated with it.

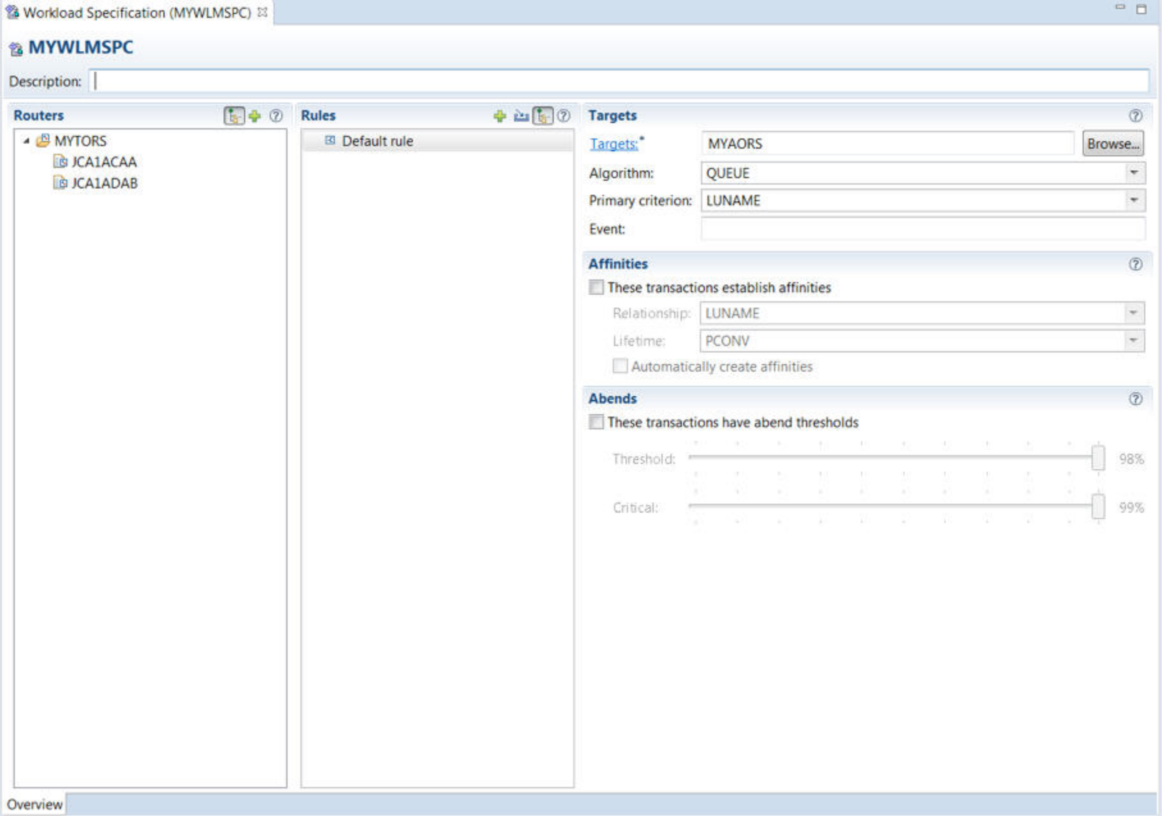
Any transaction affinities, if any, that are outstanding to override the dynamic routing decision

Regardless of any other factors, if the routing request has an outstanding affinity associated with it, that affinity always overrides the route decision.

CICSPlex SM WLM configuration at its simplest

The most basic WLM configuration provides workload balancing for all dynamic transactions and program links across CICS systems in the target scope, and optionally honors a default affinity type and lifetime. It is possible to use workload management in CICSPlex SM with a configuration that comprises simply a workload specification (WLMSPEC) to name the workload and to point it at a default set of regions to which work can be routed.

The following example shows a WLMSPEC called MYWLMSPC with a target scope of MYAORS. The target scope is a CICS systems group that contains the regions where work might be run. Here's how the WLMSPEC looks in CICS Explorer:



MYWLMSPC

Description:

Routers: MYTORS (JCA1ACAA, JCA1ADAB)

Rules: Default rule

Targets

Targets:* MYAORS

Algorithm: QUEUE

Primary criterion: LUNAME

Event:

Affinities

□ These transactions establish affinities

Relationship: LUNAME

Lifetime: PCONV

□ Automatically create affinities

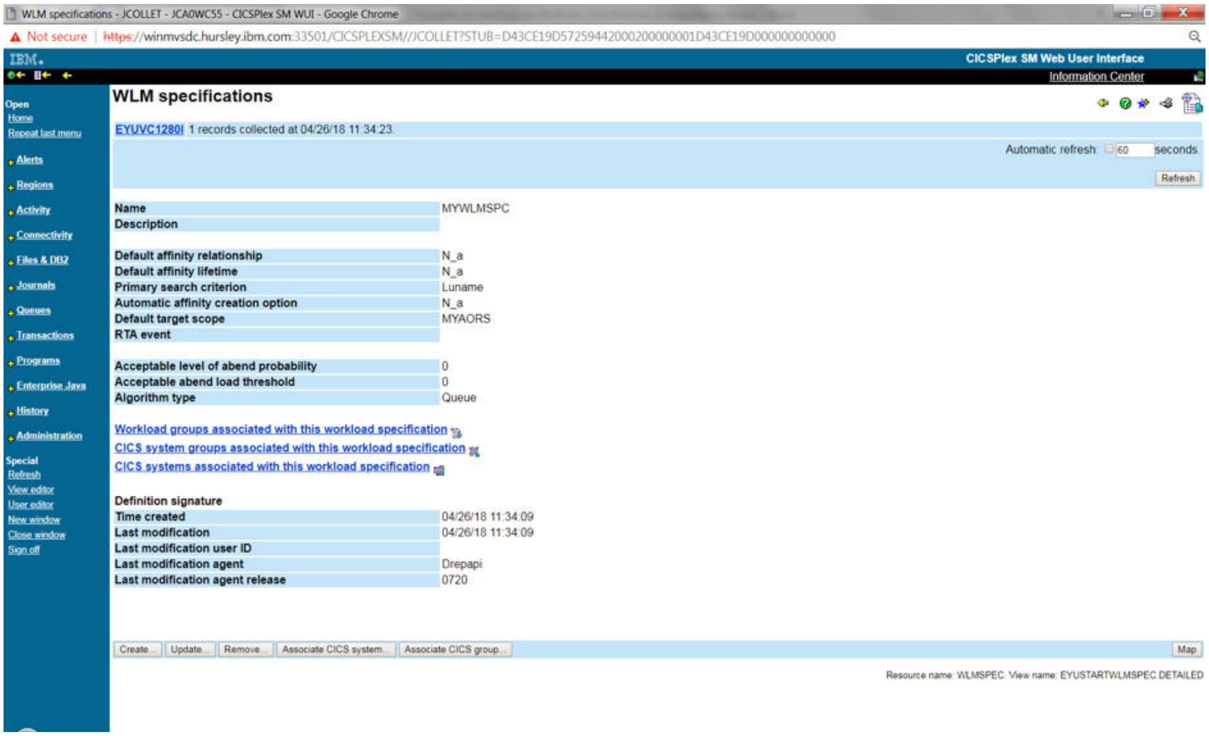
Abends

□ These transactions have abend thresholds

Threshold: 98%

Critical: 99%

Here's how it looks in the CICSPlex SM WUI:



WLM specifications

EYUVC1280I 1 records collected at 04/26/18 11:34:23.

Name	MYWLMSPC
Description	
Default affinity relationship	N_a
Default affinity lifetime	N_a
Primary search criterion	Luname
Automatic affinity creation option	N_a
Default target scope	MYAORS
RTA event	
Acceptable level of abend probability	0
Acceptable abend load threshold	0
Algorithm type	Queue

Workload groups associated with this workload specification

CICS system groups associated with this workload specification

CICS systems associated with this workload specification

Definition signature	
Time created	04/26/18 11:34:09
Last modification	04/26/18 11:34:09
Last modification user ID	
Last modification agent	Drepapi
Last modification agent release	0720

Resource name: WLMSPEC. View name: EYUSTARTWLMSPEC.DETAILED

Most situations will have more complex processing than the above example can handle. For example, you might not want all transactions to be routed to any target region. You probably want to name specific transactions for routing and you might well need some form of workload separation and workload affinities to be set up. These extensions do need some additional CICSPlex SM WLM artifacts to be configured, such as a workload management definition (WLMDEF), workload management group (WLMGROUP) and transaction group (TRNGRP) but you can build those up incrementally.

How it works: z/OS Workload Manager (WLM)

The Workload Manager component of the z/OS system (referred to as WLM) monitors a sysplex and determines how much resource should be given to each item of work in the sysplex to meet the goals that you have defined for it. It also reports data about the work.

The set of parameters that you specify to z/OS Workload Manager is called a service definition. The service definition includes a combination of policies, groupings, rules, and classes that you set up to tell z/OS Workload Manager how to manage the work within the sysplex. Some key things you can do with the service definition are:

- Identify each item of work in the sysplex that can be managed by z/OS Workload Manager (such as a CICS region, or a CICS transaction).
- Group items of work together when they have similar requirements.
- Set the performance goals that should be implemented for each type of work.
- Specify how each type of work should be reported.

You use the WLM administrative application on ISPF to set up the parameters for workload management.

A CICS region is identified in two ways to z/OS Workload Manager:

- As an address space, on the basis of the startup subsystem, which is either a JES batch job or a started task (STC). The JES or STC classification rules are set up using the job name taken from the JCL. This always needs to be done, whether you choose to manage the CICS work by region or by transaction.
- As a CICS subsystem, using the CICS subsystem classification rules. The CICS subsystem classification rules are set up using the applid of the CICS region. Sub-rules for individual transactions, or groups of transactions with similar characteristics, can be set up under the CICS subsystem classification rules. You only need to do this if you want to manage the CICS work by transaction.

z/OS Workload Manager: Service classes and report classes

Everything managed by z/OS Workload Manager has a service class, and can optionally have a report class. Service classes relate to the process of managing the work, and report classes relate to the process of collecting data about the work.

- A service class is a group of work with similar performance goals, resource requirements, and business importance. z/OS Workload Manager manages each group of work according to the performance goal assigned to the service class, and the business importance assigned to that performance goal.
- A report class is a group or item of work for which z/OS Workload Manager reports data. By default, data is reported as a total for each service class, so report classes are needed if you want to distinguish between different items within a service class.

In z/OS Workload Manager, you can assign a service class and a report class for a CICS region, or for a transaction. The service class and report class for each item can be arranged as you choose. For example, two CICS regions might be defined with the same service class, so that they have the same performance goals and the same level of access to system resources, but they can have different report classes, so that their usage of system resources is reported separately. For the best results, you should place work of the same type, with the same goals and importance, into the same service class wherever possible, but you should use as many different report classes as you need to achieve the level of granularity you require in reporting.

You set up service classes and report classes as part of setting up the service definition for the sysplex.

Service classes for CICS regions and transactions

Every CICS region and transaction that you define to z/OS Workload Manager must have a service class. Work in the same service class is managed as a single entity.

Each service class is associated with a performance goal, which specifies the target towards which z/OS Workload Manager manages the work in the service class. For example, the goal could specify an average response time for transactions in the service class. You also assign an importance level to the performance goal, which applies in case there are problems meeting the goal. The importance level tells z/OS Workload Manager how important it is to meet the goal relative to meeting the goals set for other work in the sysplex.

For service classes that apply to a CICS address space (defined under the JES or STC classification rules), you can only assign an execution velocity goal. This type of goal specifies the acceptable amount of delay for work when work is ready to run.

For service classes that apply to a CICS region applid or a transaction (defined under the CICS subsystem classification rules), you can only assign a response time goal. This type of goal specifies either an average response time for transactions in the service class, or a response time which must be met by a specified percentage of transactions (response time with a percentile). A response time with a percentile is useful if you have some long-running transactions.

You can choose which goal is used to manage the CICS workload. The next topic in the learning path explains this.

The CICS monitoring domain statistics show information about z/OS Workload Manager settings for the CICS address space (defined under the JES or STC classification rules), including the type and importance level of the performance goal for the address space.

Report classes for CICS regions and transactions

If two or more items have the same report class, the data for all the items in the report class is reported together as a total. If you place items with different service classes into the same report class, the result is called a heterogeneous report class. (A report class that contains items with the same service class is called a homogeneous report class.) Heterogeneous report classes can cause incorrect performance data, because the different service classes mean that the data collected is based on different goals, importance, or duration. If you want an accurate measurement of the resource usage for a CICS region, or meaningful data about the response time for a transaction, you need to define the region or transaction with its own individual report class.

The data produced for the report class assigned to a CICS address space (defined under the JES or STC classification rules) gives the processor time used by the CICS region, but it does not give the number of transactions executed by the region, or their response time. Conversely, the data produced for the report class for a CICS region applid or a transaction (defined under the CICS subsystem classification rules) does give the number of transactions executed and their response time, but it does not give the processor time used by the CICS region.

To obtain a complete view of the workload for a CICS region, you need to have both the CICS address space and the transactions it runs defined with service classes and report classes. Because service classes are required by z/OS Workload Manager, even if you do not intend to make use of the service classes for the transactions, you need to define them in order to set up the report classes.

z/OS Workload Manager: Region and transaction goals

z/OS Workload Manager can manage CICS work towards a region goal or a transaction response time goal. You can choose which goal is used.

When you choose to manage towards a region goal, z/OS Workload Manager uses the goal for the service class assigned to the CICS address space under the JES or STC classification rules. When you choose to manage towards a transaction goal, z/OS Workload Manager uses the goals for the service classes assigned to transactions or groups of transactions under the CICS subsystem classification rules. You can select which mode to use when you are working with the JES or STC classification rules.

When you manage towards a transaction goal, z/OS Workload Manager does not directly manage resource allocations for the transactions. Instead, it makes calculations to assign appropriate resources to the CICS region or regions which can run the transactions. This can work less well if regions have a diverse mix of transactions and response time goals. In this situation, managing towards a region goal might work better.

Sometimes, the processing for a single work request requires more than one transaction in the CICS region. For example, up to four transactions, with different transaction identifiers, might be needed to process an inbound SOAP request in a CICS provider region. Take this into account when deciding whether to use a transaction goal or a region goal.

Where next?

To find out more information about z/OS Workload Manager, and to learn how to set up classes and goals, see the following resources:

- z/OS MVS Planning: Workload Management has a full technical description of z/OS Workload Manager and instructions for using the WLM ISPF Application.
- IBM Redbooks: System Programmer's Guide to: Workload Manager explains the functions of z/OS Workload Manager and provides information and best practice samples for implementing these functions, including some specific hints for efficiently classifying and reporting on CICS transactions and regions.
- The IBM Workload Manager for z/OS web page has more resources for z/OS Workload Manager, including articles, presentations, and a history of functions added to the component.

Chapter 2. Configuring workload management

CICSPlex SM workload management optimizes processor capacity in your enterprise by dynamically routing transactions and programs to whichever CICS region is the most appropriate at the time, taking into account any transaction affinities that exist.

Before you begin

1. Ensure that you are familiar with the concepts involved with workload management and the dynamic routing of transactions and programs. For details, see Workload management.
2. You have determined the extent to which you can use CICSPlex SM workload management in your enterprise. This includes identifying the workloads processed in your enterprise and inter-transaction affinities. Follow the instructions in Planning for WLM.
3. Ensure that you follow the implementation process described in Implementing WLM.

About this task

When you establish a workload, you are associating the work itself and the CICS systems (acting as requesting, routing, and target regions), to form a single, dynamic entity. Within this entity, you can route the work:

- To a target region selected on the basis of its availability. This type of routing, known as *workload balancing*, allows you to balance work activity across all of the target regions associated with a workload. See “Management of the work in a workload” on page 19 for additional information.
- To a subset of the target regions based on specific criteria. This type of routing, known as *workload separation*, allows you to separate transaction and program occurrences and direct them to different target region subsets, where activity is balanced across the target regions within the subset.

  The criteria you use to separate transactions or programs can be based on:

  - The terminal ID and user ID associated with a transaction or program occurrence.
  - The process type associated with the CICS BTS activity.
  - The transaction.

  See “Separating the work in a workload” on page 21 for additional information.
- To a selected target region based on its affinity relationship and lifetime. This type of routing, based on the *transaction affinity* of the target region, allows you to route specific transaction occurrences to the same target region for a designated period of time. For additional information, see “Taking affinity relations into consideration” on page 29.

Workload routing and workload separation can be active concurrently in the same or different workloads associated with a CICSplex.

Note:

1. You can use the CICSPlex SM real-time analysis functions to produce data that will help in the selection of a target region during workload management, as demonstrated in this example. However, CICSPlex SM Workload Manager (WLM) health is assessed independently and does not require RTA to be active. CICSPlex SM WLM health is not influenced by SAM settings either.
2. For additional information about how CICSPlex SM handles dynamic routing and how you can customize this facility, see Dynamic routing with CICSPlex SM.

Procedure

Workload requirements

Any CICS system can act as a requesting region, a routing region, or a target region, under certain conditions.

- All the CICS systems associated with a workload are either part of the same CICSplex, or, for BTS processes and activities, part of the same BTS-set. They do not have to reside in the same MVS™ image.
- For dynamic routing of both terminal-related and non-terminal-related **`EXEC CICS START`** commands, requesting, routing, and target regions must be a supported release of CICS Transaction Server.
- For dynamic routing of DPLs, routing regions must be a supported release of CICS Transaction Server; target regions can be any level of CICS.
- For CICS BTS activities, the routing region (which is also the requesting region) and the target region must be a supported release of CICS Transaction Server. They must also be in the same sysplex.
- For dynamic routing of transactions and static routing, the CICS system acting as the routing region must be running CICS Transaction Server. The CICS systems acting as target regions can be running any version of any CICS platform supported by CICSPlex SM.
- The routing region must be defined as local to a CMAS involved in managing the CICSplex. It must use CICSPlex SM facilities to communicate with that CMAS.
- For Link3270 bridge requests the target regions must be a supported release of CICS Transaction Server.
- The regions involved in workload management must have an in-service and `ACQUIRED` connection to each other.

A CICS system can act as a target region in one or more workloads; however, it can act as a routing region in only one workload.

Note: A CICS system can act as a routing region and a target region in the same workload.

When a CICS system acting as a target region is combined with other target regions to form a CICS system group, each of the target regions must have access to all the resources required by any transactions that might be routed to that CICS system group. In the case of a CICS BTS workload, they must all be in the same sysplex.

Establishing a workload

The criteria used to identify a workload are based on the information provided by a workload specification, one or more workload definitions and their optional transaction groups, and one or more workload groups.

- A *workload specification* identifies the CICS systems or CICS system groups in a CICSplex that are to be treated as routing regions and the default target regions. You can also use a workload specification to identify a default affinity relationship and lifetime.

  A CICS region can be associated with none, or one, workload specification (WLMSPEC). For the CICS region to be a CICSPlex SM routing region, it must be associated with a workload. For the CICS region to be a CICSPlex SM target region, it depends on the type of routing being used:

  - For dynamic routing, the target region is not associated with a workload (WLMSPEC). The target region can be a target for one or more workloads.
  - For distributed routing, the target region must be associated with a workload (WLMSPEC). The target region can be a target for one workload.

  If you currently use only dynamic routing, it is advisable that a target region is a target for only one workload. This simplifies the situation if you decide to implement distributed routing later.

  The name of a workload specification becomes the name of the workload itself.
- A *workload definition* includes specific or generic terminal and user names, or a specific or generic process type, that are used when attempting to match a transaction occurrence to a workload

definition. It also identifies the target region to which transaction occurrences matching the criteria are to be routed.

- A *transaction group* identifies one or more related transaction identifiers and indicates whether the terminal id or the user id specified in the workload definition is to be used as the primary selection criterion in determining which transactions are to be routed to the set of target regions identified in the workload definition. A transaction group is also used to define affinities.
- A *workload group* is a collection of workload definitions that are to be treated as a single entity.

Note: When no workload definitions are installed in a workload, all transactions and programs are routed to a target region in the default target scope identified by the workload specification. However, when a workload definition is installed in a workload, all transactions and programs that match the criteria in that workload definition are routed to a target region in the target scope identified by the definition itself. A dynamic routing request that is not associated with any workload definition is routed to a target region in the default target scope of the workload specification.

Management of the work in a workload

All dynamic transactions and programs that are initiated from a set of requesting regions can be routed by a routing region to a specific set of target regions in the same CICSplex.

The specific target region to which each request is routed is determined by the activity and availability of all target regions in that set.

To establish workload routing, you need to define only a workload specification.

The dynamic routing processes are explained using Figure 3 on page 19, which illustrates the Starter Set configuration. For dynamic transaction routing, any transaction initiated from a terminal associated with the requesting region EYUMAS1A (a TOR) is routed to the most appropriate target region (an AOR) in the CICS system group EYUCSG01.

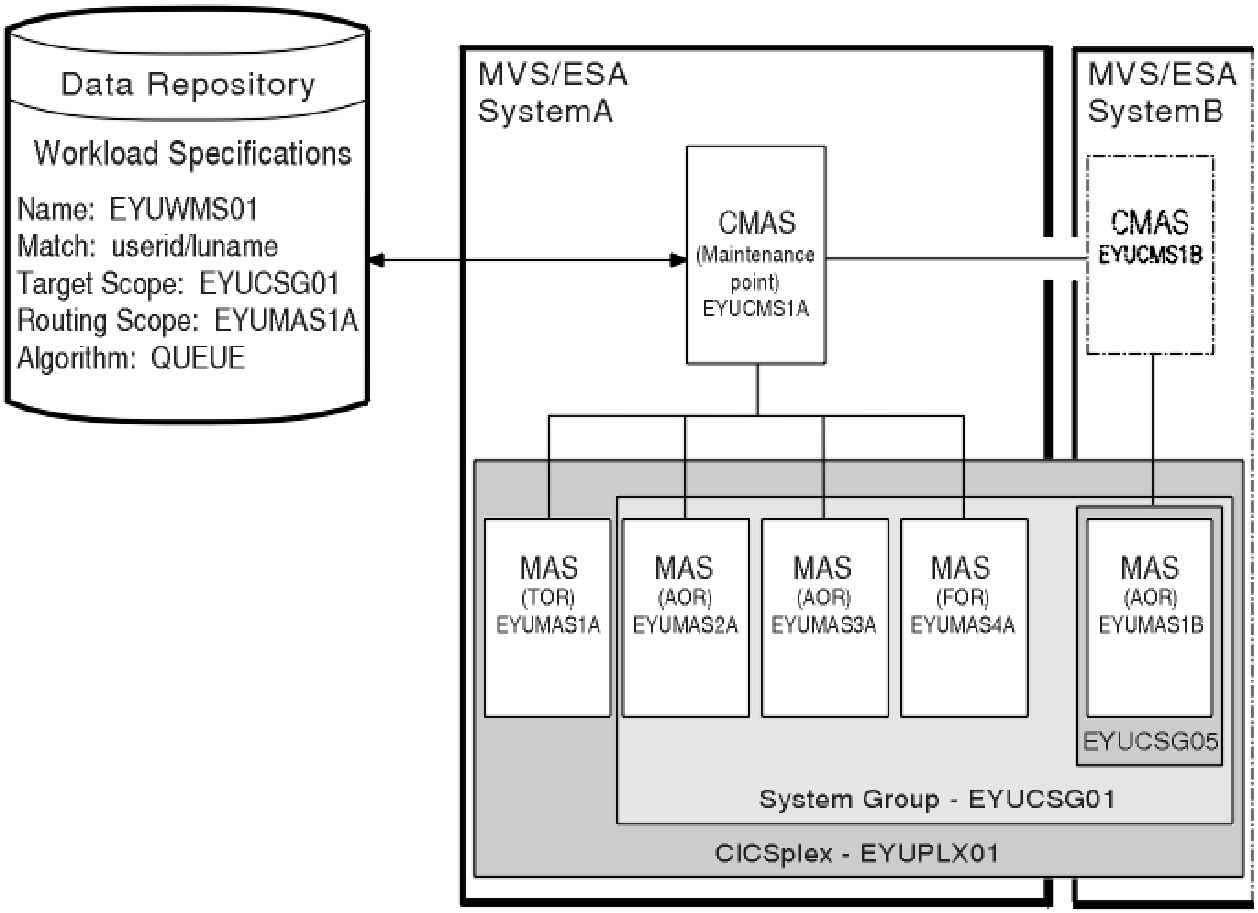


Figure 3. Sample workload definition for dynamic routing

For dynamic routing of **EXEC CICS START TRANSID TERMID** commands, any transaction initiated in the requesting region EYUMAS2A (an AOR) is sent to EYUMAS1A (a TOR), the routing region associated with the terminal identified in the TERMID option of the START command. The routing region sends the transaction to the most appropriate target region (an AOR) in the CICS system group EYUCSG01.

For dynamic program linking, there are two possible scenarios. For an inbound client request, the request is received in TOR EYUMAS1A, which acts as the requesting region and the routing region. The target region is any AOR in the CICS system group EYUCSG01. For a peer-to-peer request, the request is initiated by a transaction running in EYUMAS2A (an AOR). EYUMAS2A acts as the routing region, and the target region may be any AOR in the CICS system group EYUCSG01.

Using the queue algorithm

During workload processing using the queue algorithm, CICSPlex SM routes all transactions and programs initiated in the requesting region to the most appropriate target region in the designated set of target regions. See The queue algorithm.

Using the link neutral queue algorithm

The link neutral queue (LNQUEUE) algorithm corresponds to the queue algorithm, except that the type of connection between the routing and target region is not considered. See The link neutral queue algorithm.

Using the goal algorithm

CICSPlex SM supports the z/OS goal algorithm. The goal algorithm selects the target region that is best able to meet the defined, average, or percentile response-time goals for all work in a workload.

The goal is defined by associating transactions, using the z/OS Workload Manager, to a service class. Service classes are assigned on a transaction, LU name, and user ID basis. Service classes can define several types of response-time goals. However, CICSPlex SM recognizes average and percentile response-time goals only. If transactions are given velocity or discretionary goals, they are assumed to be meeting their goals. CICSPlex SM manages at the service-class level; that is, it has no internal knowledge of the transaction characteristics. By consistently allocating service classes to sets of target regions, it minimizes the amount of resource reallocation by the z/OS Workload Manager.

You can use goal mode to provide efficient routing decisions, where routers and targets are managed by the same CMAS, in the following scenarios:

- Dynamic routing using DTRPGM for dynamic transactions
- Dynamic routing using DTRPGM for EXEC CICS START TERMID over APPC or MRO connections
- Distributed routing using DSRTPGM for business transaction service routing

For additional information about the goal algorithm, see The goal algorithm and z/OS MVS Planning: Workload Management.

The service level administrator must define goals that are realistic for the underlying capacity of the target systems. Transactions of like attributes (for example, transactions that have similar resource consumption, or pseudoconversational transactions) must be assigned to distinct service classes. The response-time goals can be the same for several service classes. Use CICS statistics to help you define these transaction sets. See Improving performance for information about CICS statistics.

Using the link neutral goal algorithm

The link neutral goal (LNGOAL) algorithm corresponds to the goal algorithm, except that the type of connection between the routing and target region is not considered. See The link neutral goal algorithm.

Control level for workload routing

To use workload routing, you must specify a default routing algorithm for the workload at the workload specification (WLMSPEC) level. You can optionally specify a routing algorithm at the transaction group

(TRANGRP) level. An algorithm specified in a transaction group overrides the default algorithm that is associated with the workload specification.

The default routing algorithm is applied to every routed dynamic transaction in the workload, except those transactions that are associated with a transaction group that has a routing algorithm specified. You can specify one of the following routing algorithms:

- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

To change the routing algorithm specified at the workload specification level, you must close down all regions that participate in the workload so that workload is refreshed with the new algorithm specification.

At the transaction group level, you can specify a routing algorithm dynamically. The specified dynamic routing algorithm is applied to every routed dynamic transaction that is associated with the transaction group. Therefore, you can apply an alternative routing algorithm to specific transaction codes in the same workload.

If you specify an alternative routing algorithm at the transaction group level, you can change workload routing characteristics for specific target regions dynamically without stopping your routing region. If you modify an installed transaction group, you must discard its associated WLM definition (WLMDEF) and then reinstall it, so that the transaction group named by the WLM definition is also refreshed. To change the routing algorithm type immediately without discarding and reinstalling the associated WLMDEF, you can use the Active workload transaction groups (WLMATGRP) views and the **SET** command to change the ALGTYPE attribute.

You can specify one of the following routing algorithms:

- INHERIT
- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

INHERIT means that transaction group uses the routing algorithm that is associated with the workload specification for the workload.

Separating the work in a workload

You can separate the work in a workload using the name of the user, the terminal, or both that are associated with a transaction or program, the transaction itself, or the BTS process-type associated with the transaction.

You can separate a workload using either LU name and user ID, or process type. You cannot separate a workload using, for example, a generic process type and a user ID. Separation of enterprise bean workloads may be implemented only through the user ID.

Separating by terminal or user name

You can create a workload that routes requests from a set of requesting regions to different subsets of target regions based on the name of the terminal, user, or both associated with each occurrence of a transaction.

For example, you might want to route all requests initiated by certain individuals from specific terminals to a special subset of target regions.

Figure 4 on page 23 illustrates what such a workload might look like. In this case, if the user and terminal names associated with a transaction begin with SM and NET, respectively, the transaction is routed to the set of target regions identified as EYUCSG05. If either the user or terminal name begins with any other characters, the transaction is routed to the default set of target regions identified as EYUCSG01 on the workload specification.

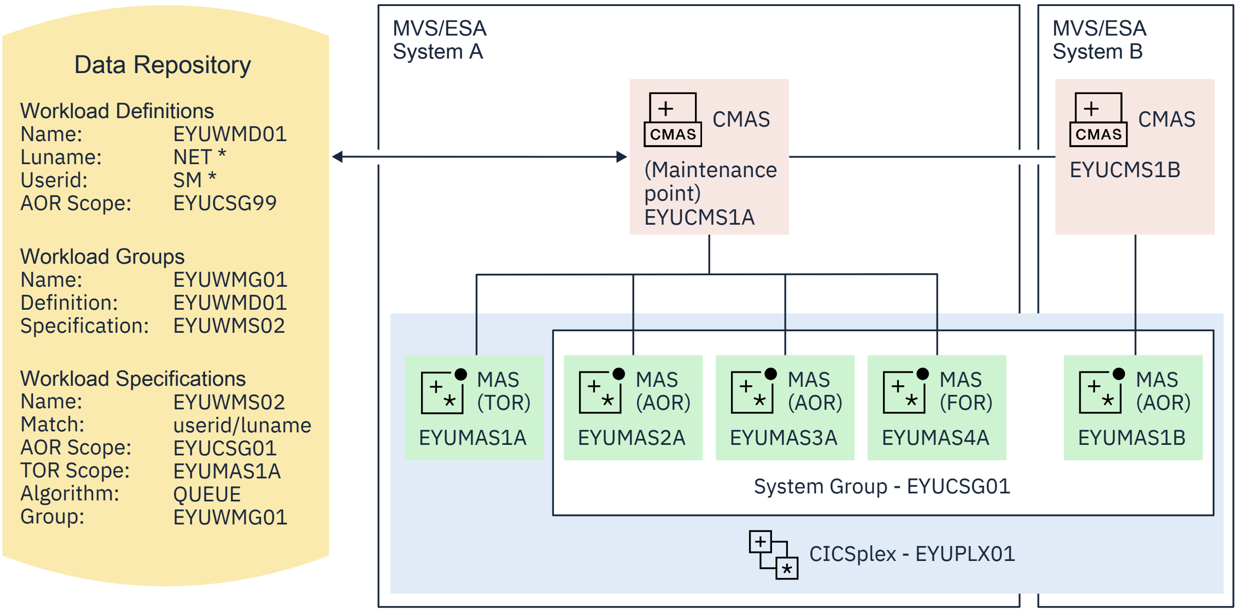


Figure 4. Sample definition separating a workload by terminal and user name

During workload processing, CICSPlex SM evaluates the terminal and user names associated with each occurrence of a request to determine where the request should be routed.

- If the terminal and user names associated with the transaction match the selection criteria specified in an installed workload definition, the request is routed to the target regions identified in that definition.
- If either the terminal or user name does not match the selection criteria, the request is routed to the default set of target regions identified in the workload specification.

After determining the appropriate set of target regions, CICSPlex SM selects one based on the status of the active target regions in that set.

Separating by process type

You can create a CICS BTS workload that routes requests associated with a certain process type to a specific target region or set of target regions. For example, you may want to route all the requests associated with the STOCK process type to a special subset of target regions.

Figure 5 on page 25 illustrates what such a workload might look like, if the process type associated with a CICS BTS transaction is STOCK, the transaction is routed to a set of target regions identified as EYUCSG05. If the process type is anything other than STOCK, the transaction is routed to the default set of target regions identified as EYUCSG01 in the workload specification.

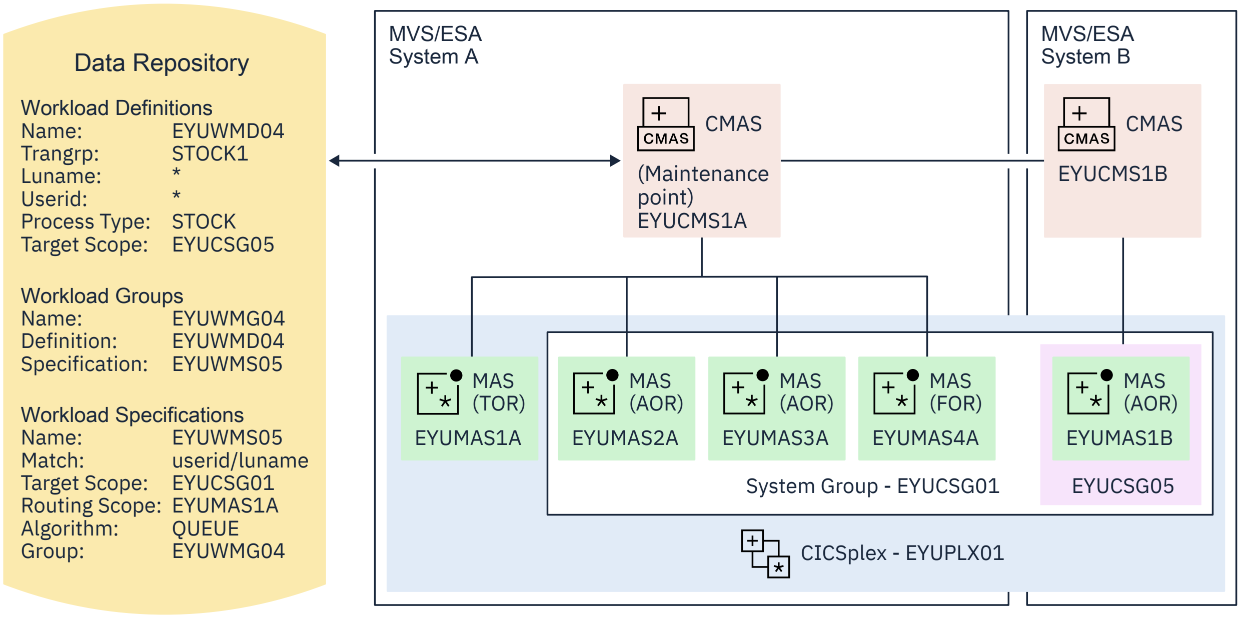


Figure 5. Sample definition separating a workload by process type

If you choose to separate a workload by process type, you must set the Luname and Userid fields to *. If you separate a workload by LU name and user ID, you must set the Process Type field to *. If you want to separate an enterprise bean workload, the Luname and Process Type fields must be set to *. You can separate a workload only either by process type or by LU name and user ID.

You can specify either a specific or a generic process type. During workload separation processing, CICSPlex SM evaluates the process type supplied by CICS to determine to where the transaction should be routed.

- If the process type matches the selection criteria specified in an installed definition, the request is routed to the target regions identified in the definition.
- If the process type does not match the selection criteria, a match may be found based on the transaction's associated LU name and user ID. If a match is found based on these criteria, the request is routed to the target regions identified in the associated definition.
- If no match is found using the process type, LU name and user ID, the request is routed to the default set of target regions identified in the specification.

Note: Separation by process type takes precedence over separation by LU name and user ID. Thus, if a transaction's associated process type, LU name and user ID mean that it satisfies the selection criteria specified in two workload definitions, one specifying separation by process type and the other separation by LU name and user ID, the transaction is routed to a region in the target scope specified in the workload definition specifying separation by process type.

Separating by transaction

You can also separate the work in a workload based on the transactions themselves. For example, you might want all occurrences of payroll-related transactions initiated from terminals in an accounting department to be routed to a specific set of target regions for processing.

Figure 6 on page 28 illustrates how you might separate the work in a workload based on transaction identifiers. In this case, if the user and terminal names associated with any transaction identified in transaction group EYUWMT01 begin with SM and NET, respectively, the transaction is routed to the target regions identified as EYUCSG05. If the transaction identifier, user name, or terminal name does not match the criteria, the transaction is routed to the default target regions identified as EYUCSG01.

During workload processing, CICSPlex SM evaluates the transaction identifier supplied by CICS to determine which transaction group to use.

- If the transaction is defined to a transaction group, CICSPlex SM notes whether the match key for that group is USERID or LUNAME.
- If the transaction is not part of a transaction group, CICSPlex SM uses the match key from the default transaction group for the specification. The match key is the value specified with the associated specification. For additional information, see Transaction group definitions - TRANGRP.

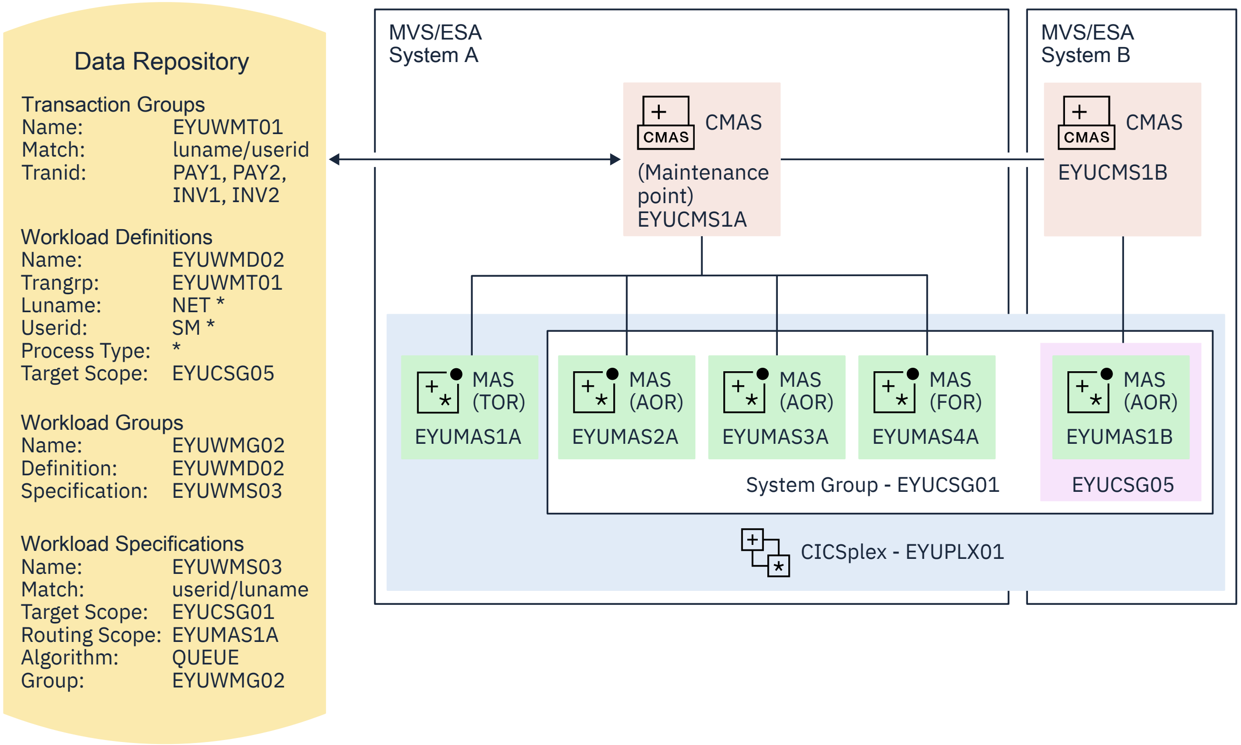
CICSPlex SM uses the match key value to establish the order in which the terminal and user names associated with the transaction are to be evaluated. The evaluation is used to determine where the transaction should be directed:

- If the terminal and user names associated with the transaction match the selection criteria specified in an installed workload definition, the request is routed to the target regions identified in that definition.
- If the terminal and user names do not match the selection criteria, the request is routed to the default set of target regions identified in the workload specification.

After determining the appropriate set of target regions, one is selected based on the status of the active target regions in that set.

Note: If you are adding new transaction codes to a transaction group, or removing transaction codes from a transaction group, you do not need to discard the WLM definition that refers to the transaction group if no other attributes of the transaction group are being changed. New transaction codes added to a transaction group can be dynamically activated within an active workload by reinstalling the WLM definition that refers to the transaction group.

A transaction (a DTRINGRP object) can be associated with one parent transaction group (TRANGRP) only, regardless of any parent workload association and irrespective of whether the transaction group is in active use or not. Changes at the transaction group level, and at the WLM definition (WLMDEF) level, can be dynamically activated without terminating an active workload. Only changes at the WLM specification (WLMSPEC) level normally require an active workload to be terminated.


Data Repository

Transaction Groups
Name: EYUWMT01
Match: luname/userid
Tranid: PAY1, PAY2, INV1, INV2

Workload Definitions
Name: EYUWMD02
Trangrp: EYUWMT01
Luname: NET *
Userid: SM *
Process Type: *
Target Scope: EYUCSG05

Workload Groups
Name: EYUWMG02
Definition: EYUWMD02
Specification: EYUWMS03

Workload Specifications
Name: EYUWMS03
Match: userid/luname
Target Scope: EYUCSG01
Routing Scope: EYUMAS1A
Algorithm: QUEUE
Group: EYUWMG02

Figure 6. Sample definition separating a workload by transaction

Separating Link3270 bridge workloads

Link3270 bridge workloads can be separated by user ID, LU name, and transaction group.

For Link3270 bridge workloads the LU name can be produced in three different ways:

1. It can be supplied by the user in the BRIH-NETNAME parameter on the Link3270 call.
2. It can be generated randomly by the Link3270 bridge facility.
3. The CICS autoinstall user replaceable program can be used in conjunction with either of the other two methods to accept, reject or modify the supplied or generated NETNAME.

You can separate Link3270 bridge workloads by LU name only if you are using methods 1 or 3 of those listed, so that the LU name is known in advance. If you are using the method 2, the LU name is not known in advance and cannot be used for workload separation.

To separate by the bridge facility NETNAME and not the name associated with the client program that started the Link3270 bridge, you must modify the EYU9WRAM module. You can use the CICS API commands **`EXEC CICS ASSIGN USERID()`** and **`EXEC CICS ASSIGN BRFACILITY NETNAME()`** to assign the user ID and LU name. You can use the NETNAME returned from the **`INQUIRE BRFACILITY()`** command rather than the NETNAME passed via the DFHDYPDS commarea parameter DYRNETNM to separate the workload.

For more information about Link3270 bridge facility definitions, see Overview of CICS external interfaces.

Taking affinity relations into consideration

An affinity is a relationship that you define between two or more transactions for the duration, or lifetime, of that relationship.

When an affinity relationship exists between transactions, those transactions must be processed by the same target region. You can use affinities to route transactions from one or more requesting regions to a specific target region based on the rules applying to a particular combination of an affinity relation and lifetime. For a list of affinity relation and lifetime values, see Affinity relations and affinity lifetimes.

When multiple CMASs in the CICSplex manage affinities for the workload, and one of these CMASs is brought down and the local MASs stay up, the workload becomes frozen. When the workload is frozen, it cannot be changed, however the current workload remains active.

When a CMAS is down, and you have any of the following affinity life times and affinity relationships, a new affinity instance cannot be created, and the transaction cannot be routed to the target MAS associated with the affinity, because the local TORs cannot be informed of the workload changes while the workload is frozen.

Affinity relation	Affinity lifetime
BAPPL	• SYSTEM • PERMANENT • ACTIVITY • PROCESS
GLOBAL	• SYSTEM • PERMANENT
LUNAME	• SYSTEM • PERMANENT
USERID	• SYSTEM • PERMANENT

When the CMAS is brought back up and reconnects to the MASs, the workload is unfrozen and is able to be changed.

You can use the IBM CICS Interdependency Analyzer for z/OS to detect existing affinities between transactions and between BTS processes and activities. The output from the Reporter component of that utility can be used as input to the CICSPlex SM batched repository-update facility. For more information, see CICS Interdependency Analyzer for z/OS Overview.

Note: You cannot use CICS Interdependency Analyzer to detect affinities between non-terminal-related EXEC CICS START commands, and between DPLs that are not associated with either a user ID or a terminal ID. For these request types, you should try to remove or avoid all affinities, and ensure that your applications can honor any remaining affinities.

CICS BTS considerations

Although BTS itself does not introduce any affinities, and discourages programming techniques that do, it does support existing code that may introduce affinities.

You must define such affinities to workload management. It is particularly important to specify each affinity's lifetime. Failure to do this may restrict unnecessarily the workload management routing options.

It is important to note that a given activity can be run both synchronously and asynchronously. Workload management is only able to honor invocations that are made asynchronously. Furthermore, you are strongly encouraged not to create these affinities, particularly activity and process affinities, because these affinities are synchronized across the BTS-set. This could have serious performance impacts on your systems.

You should also note that, with CICSPlex SM, the longest time that an affinity can be maintained is while a CMAS involved in the workload is active; that is, an affinity of PERMANENT. If there is a total system failure, or a planned shutdown, affinities will be lost, but activities in CICS will be recovered from the BTS RLS data set.

Link3270 bridge considerations

For Link3270 bridge transactions, affinities are managed by CICS and not by CICSPlex SM. Transaction affinity relation and lifetime fields in the workload management views should be left blank.

Adding affinities into a workload

For non-Link3270 bridge transactions, affinities are managed by CICSPlex SM **`Transaction affinity relation`** and **`lifetime`** fields in the workload management views. You specify these fields to isolate transactions to particular systems and to define affinity types and lifetimes.

Figure 7 on page 31 illustrates how you might separate the work in a workload based on transaction identifiers and then associate an affinity relation and lifetime with those transactions. With this example, the first occurrence of a transaction named PAY1, where the associated terminal and user names are NET1 and SMITH, respectively, is directed to the appropriate target region within the set of target regions identified as EYUCSG05. The specific target region receiving the transaction and the affinity relation and lifetime associated with the transaction group to which PAY1 belongs are noted. All subsequent occurrences of any transaction in the transaction group that meet the terminal and user name criteria are directed to the same target region for the designated period of time.

Note: If you do not specify a default affinity relation and lifetime, then you can use the same workload specification for workload routing and separation.

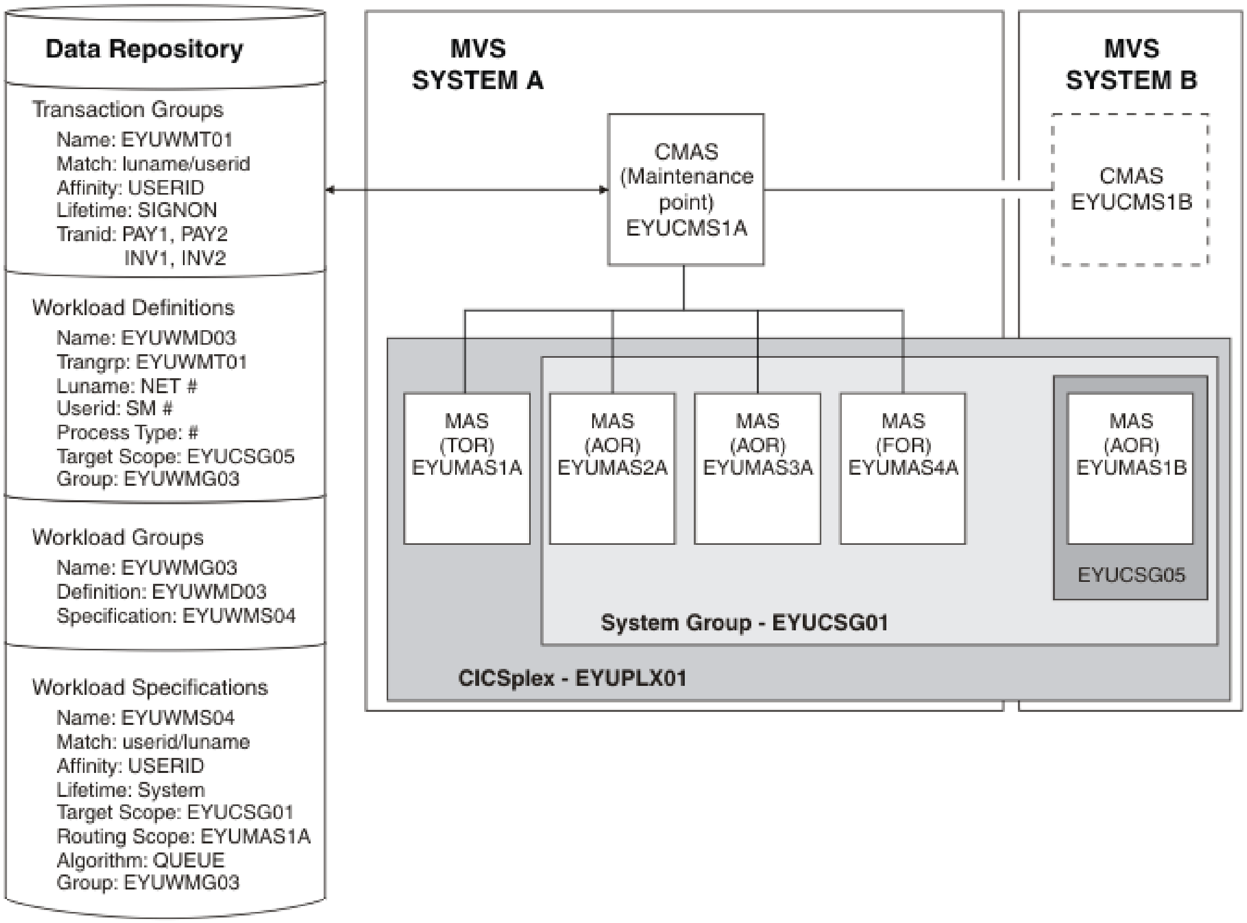


Figure 7. Sample definition adding transaction affinity to a workload

Note: During workload processing, CICSPlex SM evaluates the transaction identifier supplied by CICS to determine which transaction group to use.

- If the transaction is defined to a transaction group, CICSPlex SM notes the match key for that group.
- If the transaction is not part of a transaction group, CICSPlex SM uses the match key from the default transaction group for the workload specification.

CICSPlex SM uses this value to establish the order in which the terminal and user names associated with the transaction are to be evaluated. The evaluation is to determine where the transaction is to be directed:

- If the terminal and user names associated with the transaction match the selection criteria specified in an installed workload definition, CICSPlex SM checks to see if an affinity relation and lifetime are associated with the transaction group.
  - When an affinity relation and lifetime are associated with the transaction group:
    - If it is the first occurrence, CICSPlex SM notes the affinity relation and lifetime. Based on target region availability, CICSPlex SM then selects a target region and directs the transaction to it.
    - If it is not the first occurrence, CICSPlex SM routes the transaction to the previously selected target region. As long as the affinity relation and lifetime are applicable, subsequent occurrences of any transaction in the transaction group are directed to the same target region.
  - When no affinity relation and lifetime are associated with the transaction group, the transaction is routed to the most appropriate target region in the designated set of target regions.
- If the terminal and user names do not match the selection criteria, the transaction is routed to the default set of target regions identified in the workload specification. One is selected based on the status of the active target regions in that set.

Note: When transactions in a *series* of terminal-initiated transactions are associated with separate transaction groups, affinities for each transaction group follow the rules defined for that transaction group.

Abend probabilities and workload management

CICSPlex SM can evaluate the health and the activity (or load) of each region identified by the target scope. CICSPlex SM uses this information with the acceptable abend load threshold value (ABENDTHRESH) and the acceptable level of abend probability value (ABENDCRIT) that you provide when defining transaction groups to CICSPlex SM, in determining to which target region a transaction is routed.

In CICSPlex SM, the abend probability for transactions that are associated with a transaction group is calculated either for each transaction individually or for the entire group.

- If the transaction is associated with a transaction group that either has no defined affinity relation and lifetime, or uses the default values defined in a workload specification, the abend probability is calculated individually for that transaction.
- If the transaction is associated with a transaction group that has a defined affinity relation and lifetime, the abend probability for that transaction is calculated for the entire group of transactions.

The abend probability for transactions that are not associated with a transaction group is derived from the default abend probability values defined in the workload specification.

The acceptable abend load threshold value (ABENDTHRESH) and the acceptable level of abend probability value (ABENDCRIT) that you provide apply to both the health and the load of a potential target region. You can specify these values either in the workload specification (default rule) or a transaction group definition (associated with a specific rule) by using WUI or CICS TS Explorer, as shown in the following table:

Table 2. Fields for ABENDTHRESH and ABENDCRIT in WUI and CICS TS Explorer

Parameters	**WUI**	**CICS Explorer**
ABENDTHRESH	**Acceptable abend load threshold**	**Threshold**
ABENDCRIT	**Acceptable level of abend probability**	**Critical**

When an unhandled abend occurs in a target region:

- For the transaction, or transactions associated with the transaction, that abends, the abend probability percentage is set to 100% in the target region, which is immediately considered unhealthy as a result.
- CICSPlex SM then gradually reduces the abend probability percentage. When the abend probability percentage is below the specified acceptable level of abend probability (ABENDCRIT) yet above the specified acceptable abend load threshold (ABENDTHRESH), the target region is no longer considered unhealthy. However, the region's perceived load is doubled, making it proportionally less attractive as a target for routing than other equally loaded regions.
- CICSPlex SM continues to reduce the abend probability percentage. When the abend probability percentage falls below the acceptable abend load threshold, the target region's load is set to normal, making it equally attractive as a target for routing as other equally loaded target regions.
- If a further unhandled abend occurs at any point, the process is reset and the abend probability is recalculated from the initial starting point, where the region is considered unhealthy.
- If the target region is considered unhealthy, CICSPlex SM attempts to avoid routing the transaction to that target region. However, if there is no healthier target region, the transaction might be routed to that target region.

Notes:

1. Abend probabilities are taken into account only if you are using the dynamic routing exit DTRPGM. Abend probabilities are not considered if you are using the distributed routing exit DSRTPGM.

2. If an affinity is active, all subsequent transactions are routed to the same target region for as long as the affinity remains active, regardless of its abend probability.
3. If a transaction abends in a target region where the **EXEC CICS HANDLE ABEND** command is active, CICSPlex SM does not receive the information that the abend has occurred. Therefore, CICSPlex SM cannot take such abends into account when calculating the abend probability for the transaction in that target region. The abend probability might stay low in this situation, even though abends occurred.
4. If an application avoids making calls to resource managers because it knows the connection to the resource manager is not active, or it processes an error return code as a result of the connection being unavailable, and proceeds to issue an error message and return normally rather than abend, it could delude the workload manager into routing more work to the CICS region. This situation is called the "storm drain effect". See Avoiding the storm drain effect.

Activating workload management

To activate workload management, you must associate a workload specification with the CICS system and update the CICS system to use the dynamic routing or distributed routing program.

Before you begin

You must identify the CICS systems that you want to use for dynamic routing of your workloads. The requesting regions are where the requests are initiated, for example a terminal-owning region; the routing regions decided where to route transactions or programs; the target regions are where the transaction or program run, for example an application-owning region. Together, the requesting, routing, and target regions form a single entity within which you can route work.

About this task

For more information about dynamic routing, see Dynamic routing with CICSPlex SM.

Procedure

1. Associate a workload specification with the CICS system, using the WLMSPEC view, as described in Associating a CICS system or system group with a workload specification.
2. Identify the dynamic routing program EYU9XLOP to the CICS systems acting as requesting regions and any routing region:

   - If the CICS system is not active, you can make workload management processing available the next time the CICS system starts up. Set the DTRPGM system initialization parameter in the requesting region and in any target region that acts as a routing region:

     ```
     DTRPGM=EYU9XLOP
     ```

     For non-terminal-related **EXEC CICS START** commands, BTS activities, and for all regions in a logical server, set the DSRTPGM system initialization parameter in the region that initiates the request and in all potential target regions:

     ```
     DSRTPGM=EYU9XLOP
     ```

   - If the CICS system is active, you can use the **CICS regions** detailed view, as described in CICS regions - CICSRGN, to set the name of the dynamic routing program and the distributed routing program.
3. Activate workload management for the CICS system.

   You can do this using the following views:

   - Use the **CICS system definitions** view to change the CICS system definition in the data repository.
   - Use the **MASs known to CICSplex** view to temporarily activate workload management for an active CICS system.

Results

When either the first routing region associated with a CICSplex is started, or you use the **MASs known to CICSplex** detailed view to activate workload management for the first routing region in a CICSplex, the appropriate workload specifications are installed automatically. All CMASs involved in managing the CICSplex are also notified. All workload definitions and transaction groups, associated with the specification by workload groups, are also installed automatically. As long as the CICS system remains active, additional workload definitions can be installed manually in the CICS system.

Note: When a requesting region associated with a CICSplex, and defined with either DTRPGM=EYU9XLOP or DSRTPGM=EYU9XLOP, is starting, it connects to its target CMAS and workload management is activated. However, if the target CMAS is not active when the routing region is starting, any transactions initiated from that routing region and processed by the EYU9XLOP program wait indefinitely until the CMAS becomes active and the requesting region fully connects to it. After the requesting region connects to the CMAS and workload management is activated, the CMAS can become inactive and workload management remains active.

When a workload is active, subsequent changes to workload definitions and transaction groups are noted in the data repository. To include these changes in an active workload, you must use action commands to install or discard them.

Installed workload definitions become active immediately and remain active as long as the workload is active or until you discard them. To verify that the definition is installed in the workload and that all of the CMASs involved in managing the CICSplex using that workload know about the definition, you can use the **Active workload definitions** view.

How long a workload remains active depends on the affinity lifetime associated with the workload:

- When an affinity lifetime of PERMANENT is in effect, the workload remains active as long as any CMAS involved in managing the workload is active.
- When any other affinity lifetime, or no affinity lifetime, is associated with the workload, the workload remains active as long as any CICS system that is associated with that workload remains connected to a CMAS that manages the CICSplex to which that CICS system belongs.

What to do next

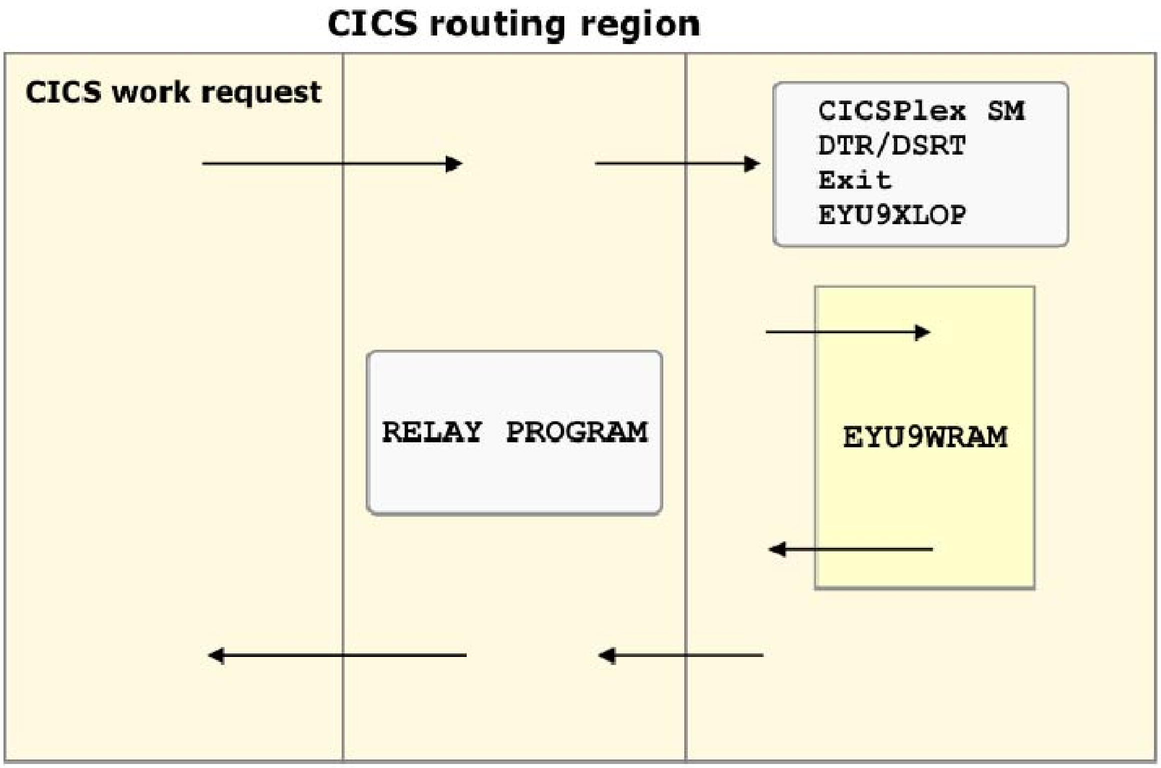
When workload management is active, do not attempt to deactivate it while any workloads are active. When CICSPlex SM is routing or separating the work in a workload, unpredictable results might occur if you attempt to deactivate workload management. Unacceptable results occur if you attempt this action when affinity relations are associated with active workloads. For details, see Discarding an active transaction from a workload.

Chapter 3. Configuring dynamic routing

You can define a transaction to CICS as either local or remote. Local transactions always run in the requesting region; remote transactions can be routed to any CICS system connected to the routing region. Routing of remote transactions can be dynamic, static, or ATI-controlled.

Dynamic routing with CICSPlex SM

When a remote transaction is initiated, the CICS relay program is invoked. The CICS relay program links to the dynamic routing program EYU9XLOP. EYU9XLOP creates the environment necessary for CICSPlex SM-based dynamic routing and sets up the CICSPlex SM runtime environment.



For a routing region, specify EYU9XLOP in the system initialization table (SIT) parameter:

DTRPGM

For static routing, ATI, and dynamic routing, set EXU9XLOP on the DTRPGM system initialization parameter that is associated with the requesting region that initiates the transaction. Any target region that is also to act as a routing region must also specify DTRPGM in the SIT and must be set up as a routing region as described in Associating a CICS system or system group with a workload specification.

DSRTPGM

For non-terminal-related **EXEC CICS START** commands, BTS activities, or enterprise beans, set EYU9XLOP on DSRTPGM system initialization parameter in:

- The requesting region that initiates the request.
- All potential target regions if the DYROPTER field in the communications area or container for the routing program (mapped by the DFHDYPDS copybook) is to be set to Y.

  If you use CICSPlex SM to route non-terminal-related **EXEC CICS START** commands, the DYROPTER field is always set to Y. You must therefore always specify *DSRTPGM*=EYU9XLOP and ensure that the target region is part of the workload. Failing to ensure that the target region is defined to the workload results in the started transaction waiting for the workload to become available and the transaction hangs.

Note: If the target region is not going to be set up as a routing region, do not specify EYU9XLOP in the DTRPGM and DSRTPGM SIT parameters. Specifying this parameter causes an endless loop for routed transactions in EYU9XLOP waiting for the workload, which never arrives.

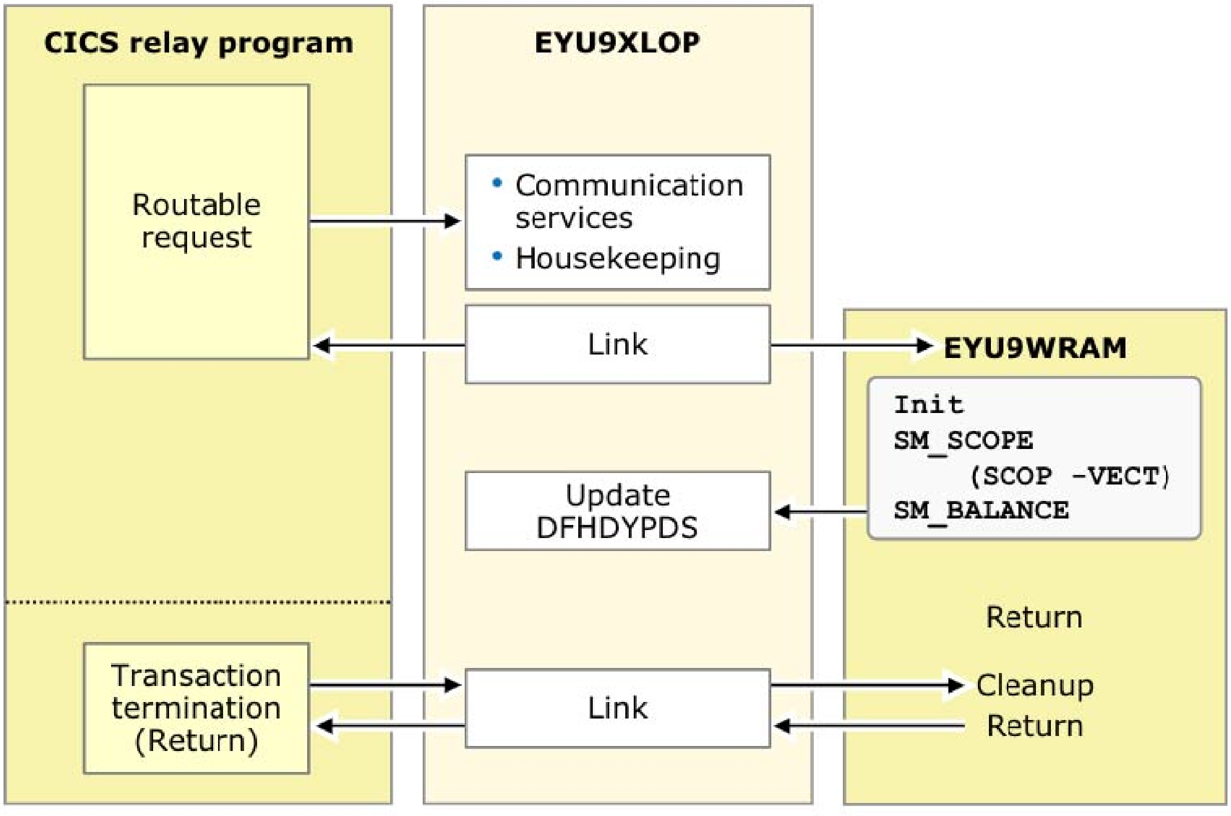
CICS notifies EYU9XLOP of all routing requests. These requests are as follows:

- Route selection, route selection error, and transaction termination.
- For BTS (all supported releases of CICS), transaction initiation, transaction abend, and routing attempt complete.
- For non-terminal **EXEC CICS START** requests (all supported releases of CICS), transaction initiation, transaction abend, and routing attempt complete.

When CICS links to EYU9XLOP, it passes the CICS communication area DFHDYPDS to it.

Control then passes to the CICSPlex SM workload management facilities. CICSPlex SM initializes the workload management MAS agent code and engages its routing action process. This process is called internally if any of the following conditions are true:

- The EYU9WRAM user-replaceable module is not defined to CICS
- EYU9WRAM is defined to CICS, but is not available
- EYU9WRAM is defined to CICS, is available, and the load module is the assembly language version of the module as distributed with CICSPlex SM.



The internal routing action process produces the same results as running the assembly language version of EYU9WRAM as it is distributed with CICSPlex SM. If no additional dynamic routing control is required at your enterprise, the internal process provides better performance.

The EYU9WRAM module, or its equivalent internal process, receives the CICSPlex SM-based communication area EYURWCOM. An entry for EYU9WRAM is added to the CICS system definition file, DFHCSD, for each CICS system during installation. As distributed with CICSPlex SM, EYU9WRAM drives

CICSPlex SM workload management processing. EYU9WRAM does this by first obtaining the appropriate list of target region candidates, based on the transaction group, and the terminal ID, LU-name, user ID, or process type. Then, EYU9WRAM selects a target region from the list of candidates.

For most situations, the supplied EYU9WRAM module provides workload management capabilities. However, if it is ever needed, you can customize CICSPlex SM workload management processing by modifying EYU9WRAM. For more information, see Creating a user-replacement module for EYU9WRAM.

CICS release requirements for dynamic routing

Here are CICS release requirements for regions involved in dynamic routing.

Table 3. Release requirements for regions involved in dynamic routing.

Release requirement for:	**Routing regions**	**Target regions**	**Requesting regions**
Dynamic transaction routing	Any supported release of CICS	Any supported release of CICS	-
EXEC CICS START commands	Any supported release of CICS	Any supported release of CICS	Any supported release of CICS
CICS Business Transaction Services (BTS)	Any supported release of CICS	Any supported release of CICS	Any supported release of CICS
Distributed program link (DPL)	Any supported release of CICS	Any supported release of CICS	-
Link3270 bridge	Any supported release of CICS	Any supported release of CICS	-

Sample source programs and copy books

Following installation, the module EYU9WRAM is loaded into CICSPlex SM.

It is an assembler-language, command-level program; its corresponding copy books are:

EYURWCOM
Defines the communication area

EYURWCOD
Defines literals for EYURWCOM

EYURWSVE
Defines each element of a target region scope list

EYURWSVD
Defines literals for EYURWSVE

To assist you in your customization effort, sample source programs and copy books for assembler, C, COBOL, and PL/I are distributed with CICSPlex SM. Copy books are located in language-specific libraries. All samples programs are located in the SEYUSAMP library. The names of the sample programs and copy books, and the CICSPlex SM libraries in which they can be found are listed in Table 4 on page 37.

Table 4. Table 1. Sample programs and copy books

Language	**Member name**	**Alias**	**Library**
Assembler:			
Program	EYUAWRAM	EYU9WRAM	SEYUSAMP
Copybook 1	EYUAWCOM	EYURWCOM	SEYUMAC

Table 4. Table 1. Sample programs and copy books (continued)

Language	Member name	Alias	Library
Copybook 2	EYUAWCOD	EYURWCOD	SEYUMAC
Copybook 3	EYUAWSVE	EYURWSVE	SEYUMAC
Copybook 4	EYUAWSVD	EYURWSVD	SEYUMAC
COBOL:			
Program	EYULWRAM	EYU9WRAM	SEYUSAMP
Copybook 1	EYULWCOM	EYURWCOM	SEYUCOB
Copybook 2	EYULWCOD	EYURWCOD	SEYUCOB
Copybook 3	EYULWSVE	EYURWSVE	SEYUCOB
Copybook 4	EYULWSVD	EYURWSVD	SEYUCOB
PL/I:			
Program	EYUPWRAM	EYU9WRAM	SEYUSAMP
Copybook 1	EYUPWCOM	EYURWCOM	SEYUPL1
Copybook 2	EYUPWCOD	EYURWCOD	SEYUPL1
Copybook 3	EYUPWSVE	EYURWSVE	SEYUPL1
Copybook 4	EYUPWSVD	EYURWSVD	SEYUPL1
C:			
Program	EYUCWRAM	EYU9WRAM	SEYUSAMP
Copybook 1	EYUCWCOM	EYURWCOM	SEYUC370
Copybook 2	EYUCWCOD	EYURWCOD	SEYUC370
Copybook 3	EYUCWSVE	EYURWSVE	SEYUC370
Copybook 4	EYUCWSVD	EYURWSVD	SEYUC370

Implementing optimized dynamic workload routing

CICSPlex SM provides dynamic workload management for CICS. CICSPlex SM can augment its workload management decisions using current status information posted directly from CICS by using a region status (RS) server.

Before you begin

Full workload optimization can be achieved with sysplex optimized workload routing. However, this capability requires that all workload regions are migrated to CICS TS for z/OS, Version 4.1 and later.

Sysplex optimized workload routing takes place automatically when all workload regions are running at CICS TS 4.1 and later, and when a region status (RS) server is started in the same z/OS image as each region in the workload in the CICSplex.

You can mix CICS TS for z/OS, Version 4.1 and later regions in a workload with regions at a level earlier than CICS TS 4.1. However, workloads will run in a nonoptimized state.

For details about sysplex optimized workload routing, see Sysplex optimized workload routing overview.

About this task

To optimize workload routing in a sysplex, you must configure and monitor a region status (RS) server, as part of a coupling facility data table. For instructions, see Setting up and running a region status server and Security for coupling facility data tables.

Procedure

In the CICSplex definition, you can set the coupling facility (CF) tuning parameters for the region status (RS) server, which provide sysplex optimized workload routing. You define and modify CICSplexes using the **EYUSTARTCPLEXDEF** view set. The coupling facility (CF) tuning parameters are in the **CPLEXDEF** detail view.

Configuring for workload routing

You can set the following configurations:

- How long region status data is cached by a routing region before requesting a refresh for sysplex optimized workload routing.

  See Region status server, read interval parameter (READRS).
- How often the coupling facility is updated with task throughput data for sysplex optimized workload routing.

  See Region status server, update frequency parameter (UPDATERS).
- The task load range that triggers workload distribution updates to the coupling facility (CF) during periods of low throughput.

  See Region status server, bottom-tier parameter (BOTRSUPD).
- A task load top tier that controls the workload when the region is close to the MAXTASK (MXT) setting during periods of high throughput.

  See Region status server, top-tier parameter (TOPRSUPD).

Viewing optimization status

- **For a workload**, view the **Active workloads list** view in the CICSPlex SM Web User Interface (WUI).
- **For a CICS region**, view the **Routing regions or Target regions** views located in the **Active workloads** menu.

  For more information, see “Optimization status” on page 42.

Region status server, read interval parameter (READRS)

You use the region status server read interval parameter in the CPLEXDEF, CSYSDEF, and MAS views to control how long region status data is cached by a routing region before requesting a refresh for sysplex optimized workload routing.

READRS={200|number}

In a workload running in optimized mode, the **READRS** value specifies the minimum interval, in milliseconds, between refreshes of a target region status from a CICS region status (RS) server. These refresh requests are issued by a routing region that is evaluating a target region for a dynamic routing request.

Note: You should only change the **READRS** value after considering the impact that the change might have on your workload and coupling facility throughput.

The value range is from 0 - 2000:

- A value of 0 means that a routing region requests a status update of a target region on every occasion that it examines the status of the target region.
- Values from 1 - 2000 specify the minimum time interval that must expire before the status of a target region can be refreshed.

A low interval value means that the RS server is polled more often for a status update. For workloads in QUEUE mode, this low value results in a smoother task load spread across the CICS regions in the workload target scope, *assuming all other health and link factors are equal*. However, the utilization of the RS server is correspondingly increased, which might result in higher utilization of your z/OS coupling facility.

The default value of the **READRS** parameter is 200 milliseconds.

The value specified in the EYUSTARTCPLEXDEF view set sets the read interval at the CICSplex level. However, you can override the interval at the CICS definition level to allow fine tuning of the value on an individual CICS target region basis.

You specify both the UPDATERS and the READRS parameters at the CICSplex definition (CPLEXDEF) level to establish default values for all regions in the CICSplex. You can override these values at the CICS system definition (CSYSDEF) level or at the MAS agent runtime (MAS) level.

At the CICS definition level, the value for **UPDATERS** and **READRS** is INHERIT, so that the values are adopted from the CICS system definition. By changing these values to explicit numeric values, the CICS system adopts the values specified on each successive restart. To implement a change to a running CICS region, the values must be applied using the MAS base tables or view sets. When the target region is restarted, it reemploys the specification from its CSYSDEF.

Region status server, update frequency parameter (UPDATERS)

You use the region status server update frequency parameter in the CPLEXDEF, CSYSDEF, and MAS views to control how often the coupling facility is updated with task throughput data for sysplex optimized workload routing.

UPDATERS={15|number}

In a workload running in optimized mode, the **UPDATERS** value indicates the frequency that the CICS region status (RS) server is called to modify the value of the task load in a target CICS region. This value is the default frequency value for all target CICS regions in the current CICSplex definition.

Note: You should only change the **UPDATERS** value after considering the impact that the change might have on your workload and coupling facility throughput.

The value range is from 0 - 25:

- A value of 0 means that the RS server is not notified of any task load count changes, because the optimized workload function for target regions in this CICSplex is not enabled.

  Note: A value of 0 does not suppress optimization of the workload routing function; a value of 0 suppresses only the target region RS domain broadcasting process.
- Values from 1 - 25 are applied as an arithmetic percentage to the MAXTASKS setting for a target. The resulting task count value is used as a numeric threshold to drive an update call to the RS server.

Specifying a value of zero effectively disables the optimized routing function for the target regions. For an optimized routing decision to be made, both the router and target must be in optimized mode.

For example, with a MAXTASKS setting of 120, and the UPDATERS set to 20, the RS server is called to update the WLM load count when the task count for a target region changes between these numbers of tasks:

- 23 and 24 tasks (20% of 120)
- 47 and 48 tasks (40% of 120)
- 71 and 72 tasks (60% of 120)
- 95 and 96 tasks (80% of 120)
- 119 and 120 tasks (100% of 120)

The RS server is updated when the task load for a target region increments or decrements across these boundaries.

If you set the **UPDATERS** parameter to a low value, the frequency of updates to the RS server increases across the task load range. For workloads in QUEUE mode, this low value results in a smoother task load spread across the target CICS regions in the workload scope, *assuming all other health and link factors are equal*. However, the utilization of the RS server is correspondingly increased, which might result in higher utilization of your z/OS coupling facility.

The default value for the **UPDATERS** parameter is 15.

The value specified in the EYUSTARTCPLEXDEF view set sets the update interval at the CICSplex level. However, you can override the interval at the CICS definition level to allow fine tuning of the value on an individual CICS target region basis.

You specify both the **UPDATERS** and the **READRS** parameters at the CICSplex definition (CPLEXDEF) level to establish default values for all target regions in the CICSplex. You can override these values at the CICS system definition (CSYSDEF) level or at the MAS agent runtime (MAS) level.

At the CICS definition level, the value for **UPDATERS** and **READRS** is INHERIT, so that the values are adopted from the CICS system definition. By changing these values to explicit numeric values, the CICS system adopts the values specified on each successive restart. To implement a change to a running CICS region, the values must be applied using the MAS base tables or view sets. When the target region is restarted, it uses the specification from its CSYSDEF.

Region status server, bottom-tier parameter (BOTRSUPD)

You use the region status server bottom-tier tuning parameter in the CPLEXDEF, CSYSDEF, and MAS views to control the workload distribution updates to the coupling facility (CF) during periods of low throughput.

BOTRSUPD={1|number}

For sysplex optimized workloads, the **BOTRSUPD** value is converted from a task load percentage to a real task count. That count is used to define the bottom-tier task load range, from zero up to this value.

Note: You should only change the **BOTRSUPD** value after considering the impact that the change might have on your workload and coupling facility throughput.

The value range is from 1 - 25. When the task load for a region falls within this range, the task load is then broadcast to the coupling facility (CF) for every change in the task load. When the load reaches this value, the RS server update frequency task rules are activated.

The default value of the **BOTRSUPD** parameter is 1%.

Change this value only after considering the impact that the change might have on your workload and coupling facility throughput. The **BOTRSUPD** parameter provides a smoother distribution of the workload during periods of low throughput. When your CICSplex consists of many target regions and you increase this value too much, you risk overloading the CF with update requests, which can result in degraded WLM and general z/OS subsystem performance.

When you need to modify the **BOTRSUPD** value, ensure that you monitor the performance of your coupling facility and WLM throughput capabilities for at least several days after modification.

Disabling updates to the CF due to BOTRSUPD

You can set the following feature toggle to disable updates to the CF when the task load falls below the lower tier threshold:

```
com.ibm.cics.cpsm.wlm.botrsupd.enabled=false
```

The lowest possible setting for **BOTRSUPD** is 1%. Therefore, regions with a high MXT setting, which spend a significant amount of time operating below 1% of their declared capacity, might perform a high number of CF updates. With this feature toggle in effect, when the task load for a region falls within the **BOTRSUPD** range, no updates are made to the CF. The task load value stored in the CF will remain zero until the first task load boundary defined by the **UPDATERS** parameter is reached.

Note that if only a subset of target regions within a workload are run with `com.ibm.cics.cpsm.wlm.botrsupd.enabled=false`, then those regions might take a greater proportion of the workload.

See Specifying feature toggles for instructions on how to specify feature toggles in your region configuration.

Region status server, top-tier parameter (TOPRSUPD)

You use the region status server top-tier tuning parameter in the CPLEXDEF, CSYSDEF, and MAS views. It controls the workload when the workload management (WLM) max task health indicator is switched off for a target region, to limit updates to the coupling facility (CF) when the region is close to the MAXTASK (MXT) setting during periods of high throughput.

TOPRSUPD={5|number}

For sysplex optimized workloads, the **TOPRSUPD** value is converted from a task load percentage to a real task count. That count is subtracted from the **MAXTASKS** value for the region to determine the top-tier task load range.

Note: You should only change the **TOPRSUPD** value after considering the impact that the change might have on your workload and coupling facility throughput.

The value range is from 1 - 25. The value is applied as an arithmetic percentage to the **MAXTASKS** setting for a region. The task count value is then subtracted from the **MAXTASKS** setting for the region to establish a task load top tier. When the task load in a region runs up to its **MAXTASKS** limit, the task load must then drop back below this value before the **MAXTASKS** state for the region is switched off and broadcast to the coupling facility.

The default value of the **TOPRSUPD** parameter is 5%.

Change the **TOPRSUPD** value only after considering the impact that the change might have on your workload and coupling facility throughput. When you increase this value too much, you see a workload batching effect in the upper load range of the workload. When you decrease the value too much, the upper-tier batching effect is reduced, but updates to the coupling facility can be significantly increased.

When you need to modify the **TOPRSUPD** value, ensure that you monitor the performance of your coupling facility and WLM throughput capabilities for at least several days after modification.

Optimization status

You can use CICS Explorer and the CICSPlex SM Web User Interface (WUI) to view the status of workloads.

Viewing optimization status of the workload

You can use the Active workloads list view in CICS Explorer or in the CICSPlex SM Web User Interface (WUI), to view the status of each workload active in the CICSplex.

The **`Optimization status`** attribute reports the sysplex optimization status of the current workload. The value is a combination of all of the workload router optimization statuses and all of the workload target optimization statuses.

These values are possible:

ACTIVE
All targets and routers in the workload are running in sysplex optimized workload state.

PARTIAL
At least one target and one router are running in sysplex optimized workload mode. Use the **Active routing regions** and **Active target regions** views to determine which regions are not running in sysplex optimized state.

INACTIVE

The workload is not running in an sysplex optimized state, for one or more of the following reasons:

- No routing regions in the workload are running in sysplex optimized state.
- No target regions in the workload are running in sysplex optimized state.
- No regions in the workload are running in sysplex optimized state.
- The workload is designated as being non-optimized by specifying a value of 0 for the RS server update frequency on the CICSplex definition or on all target CICS system definitions for this workload.

Viewing optimization status for a CICS region

If you are using CICS Explorer, follow the instructions in Viewing workloads in a CICSplex in the CICS Explorer product documentation.

If you are using the CICSPlex SM WUI, use the Routing regions or Target regions views located in the Active workloads list view to view the optimization status for a CICS region.

The **`Optimization status`** attribute reports the status of the current target region for the sysplex optimized workload routing function.

These values are possible:

ACTIVE

This target region is running in sysplex optimized workload state.

INACTIVE

This region can run in sysplex optimized workload state; however, it is not currently sysplex optimized for one or more of the following reasons:

- The region has no connection to an RS server.
- The region is connected to an RS server; however, the server cannot connect to the z/OS coupling facility.
- The optimization enablement setting for the region is set to *Disabled*. The enablement setting for the region must be set to *Enabled* before any sysplex optimized routing functions can be activated.

  Note: You can reset the sysplex optimization enablement setting in the CICS system definition view to change the value in preparation for the next region startup. The enablement setting can also be changed in an active region using the MAS view, but will revert to the setting in the CICS system definition when the region is restarted.
- The RS server update frequency value for this region is 0, which means that the optimization capabilities for this region when acting as a routing target are not enabled.

  Note: You can set the *UPDATERS* value to 0 for regions that are used as dynamic routers only. Setting this value to 0 prevents the region from making unnecessary region status broadcasts to the Region Status (RS) server.

N_A

The target region is not at a CICS release that supports region status recording. Only non-optimized WLM routing decisions can be made.

Nonoptimized workload routing

In releases before CICS TS for z/OS, Version 4.1, for nonoptimized workload management, and when a coupling facility is not available, workload routing is managed by CICSPlex SM Workload Manager using a data space owned by a CMAS to share cross-region load and status data.

Every CMAS manages a single WLM data space that it shares with every user CICS region (MAS) that it directly manages. When the CMAS initializes, it verifies and formats the data space with the structures necessary for all workloads associated with the CICS regions that it manages. When the user CICS regions begin routing dynamic traffic, the state of those CICS regions is recorded in this data space. Every 15 seconds, the CICSPLex SM agent in the user CICS region determines the task count at that time and reports to its owning CMAS. The CMAS updates the load count in the target region descriptor of its WLM data space and broadcasts that value to other CMASs participating in workloads associated with the user CICS region.

In an environment in which all user CICS regions are managed by the same CMAS, all routing and target regions refer to the same physical structures in the WLM data space. Dynamic routing decisions are based on the most current load data for a potential routing target region. A routing decision is based on a combination of factors. For details, see How CICSPlex SM WLM chooses where to route work.

Workloads are spread across multiple z/OS images, so additional CMASs are configured to manage the user CICS regions on the disparate LPARs. Each WLM data space must maintain a complete set of structures to describe every CICS region in the workload, not just the CICS regions for which each CMAS is responsible. The WLM data space owned by each CMAS must be periodically synchronized with the WLM data spaces owned by other CMASs that are participating in the workload. This synchronization occurs every 15 seconds from the MASs to their CMASs, and then out to all other CMASs in the workload.

The behavior of CICSPlex SM is different for DTRPGM and DSRTPGM requests:

- For DTRPGM requests, the routing region calls, from CICS, to decide if the target regions are synchronized with the execution of the request at the selected target, which is then followed by a call back from CICS on completion of the dynamic request. This call allows the router to increment the task load count before informing CICS of the target region system ID, and also to decrement the count on completion of the request.
- For DSRTPGM requests, the routing region calls, from CICS, to decide if targets are not synchronized with the selected target. Typically, these dynamic requests are asynchronous CICS starts. The router has no notification of when the routed transaction starts or finishes. Because of a lack of notification, CICSPlex SM stipulates that a DSRTPGM target region must also have the workload associated with the region, which transforms the targets into logical routing regions. This call allows the CICSPlex SM routing processes to determine that they are being called at the DSRTPGM target and, therefore, allows the task load count to be adjusted at transaction start and finish.

CICSPlex SM routing regions count the dynamic transaction throughput in a CICSplex, introducing an additional problem in that transactions started locally on the target regions remain unaccountable by the routing regions until a heartbeat occurs. The router transaction counts are not accurately synchronized until two heartbeats have occurred, the first to increment the count, and the second to decrement it again. This discrepancy, however, is not considered as severe as when a router and target are managed by different CMASs.

For multiple CMASs, the router regions evaluate status data for a target region as described in its local WLM data space. If that target region is managed by a different CMAS to that owned by the router, status data describing that target region can be up to 15 seconds old. For DTRPGM requests, this latency does not have a severe effect. For DSRTPGM requests, however, the effect can be significant, particularly when the workload throughput increases. The effect is known as workload batching.

For more information, see “Workload batching” on page 44.

Workload batching

Workload batching occurs in heavily used workloads in multiple CMAS environments, where dynamic distributed (DSRTPGM) routing requests are being processed.

A target region and its routing regions can be managed by different CMASes. This is always the case when the router runs in a different LPAR to the targets. In this scenario, the router is using a different descriptor structure to evaluate the target status from the real descriptor structure employed by the target itself.

The target descriptor reviewed by the router is synchronized with the real descriptor at 15-second intervals by the CICSPlex SM heartbeat. The task count for the target region, as seen by the router, is refreshed at the same time. During a 15-second heartbeat interval, the router regards the target as being relatively busy, or relatively quiet, when compared to other potential target regions in the workload. Therefore, for that 15-second duration, the router continues to divert work towards, or away from, the target, depending on how busy it seems to be, because the current task count in the target is not seen to change until the next heartbeat. This routing, based on the previous status, can make the target very busy or unused. The router reacts to this situation by more actively routing work towards, or away from, the target, so the batching cycle continues. This state continues until activity in the workload throughput declines, which settles down the batching cycle until the throughput activity increases.

If you are monitoring the task load across the CICSplex, you see some regions running at their MAXTASKS limit and being continually given dynamically routed traffic, and others remain unused. A snapshot taken 15 seconds later might show a reversal of utilization; the busy regions might be idle and the idle regions might now be at the MAXTASKS limit.

When you start your region status servers and optimize workloads, the effects of workload batching are removed.

Modifying dynamic routing

You can customize CICSPlex SM workload management processing by modifying the module EYU9WRAM.

EYU9WRAM uses the CICSPlex SM dynamic routing application programming interface (API), which is a special-purpose, call-level interface that provides the mechanism needed to request workload management actions. All calls are constructed using standard CALL statements. The CALL statement generates the linkage between the EYU9WRAM module and the CICSPlex SM Workload Manager component. The format of the CALL statement is shown here:

```
   CALL WAPIENPT(DA_TOKEN,function)
```

where:

DA_TOKEN
Identifies the dynamic routing API token supplied via the EYURWCOM communication area. This token is used by EYU9WAPI and must not be altered.

function
Is the function to be performed, specified as:

SM_SCOPE
Returns a list of eligible target regions.

SM_BALANCE
Selects a target region from the list of eligible target regions.

SM_ROUTE
Routes a transaction to a specific target region.

SM_CREAFF
Creates an affinity.

SM_DELAFF
Deletes an active affinity.

You can use SM_SCOPE and SM_BALANCE together to obtain a list of target regions and then select the target region to which a transaction is to be routed.

If you know that a transaction is always to be directed to a specific target region, you can use just SM_ROUTE.

As distributed, EYU9WRAM issues SM_SCOPE and SM_BALANCE calls. It does not include any SM_ROUTE calls. It includes unexecuted calls to SM_CREAFF and SM_DELAFF. You can modify the program to issue these calls.

Although all the examples use the assembler language form of the dynamic routing API verbs, you can also use these verbs in programs written in C, COBOL, and PL/I. Sample programs distributed with CICSPlex SM contain examples of the CALL statements for each of these languages.

Note: If you are modifying dynamic routing, be aware that:

- If they are not set, the values for application context are either zero (for the version) or nulls/low-values (for platform, application, and operation).

- String fields are padded with nulls, not spaces.

Important: As distributed, EYU9WRAM handles the workload routing and separation established via the workload management definitions by using the API verbs. Any changes you make to EYU9WRAM, therefore, might adversely impact the CICSPlex SM workload management facilities. For example, if you do not use SM_SCOPE, SM_BALANCE, or SM_ROUTE, each transaction occurrence is routed to the default target region identified when the transaction was defined to CICS. Thus, all CICSPlex SM workload management definitions are bypassed.

Nondynamic transaction considerations

You do not have to include any calls through the API if a transaction is statically routed or is started by ATI. In these cases, the target region cannot be changed. The routing function is route notify. The route notify function is intended to notify the EYU9WRAM program that such a transaction is being routed.

If you do include any of these functions, the following occurs:

- An SM_SCOPE call returns a single entry in the SCOP_VECT. The entry is the target region associated with the statically defined transaction when it was defined, or with the destination specified on the EXEC CICS START command for ATI transactions.
- An SM_BALANCE call selects the target region associated with the transaction when it was defined to CICS. The CICSPlex SM Workload Manager will not create any affinity for the transaction.
- An SM_ROUTE call returns an exception response.

When EYU9WRAM is called for:

- Notification, any existing affinity relations are ignored and none are created.
- Route selection error, any existing affinity relations are ignored and none are created. In addition, the EYU9WRAM program as delivered will write a terminal message and terminate.
- Routing attempt complete, for CICS BTS transactions only. Any existing affinities are ignored. Any dynamic routing API function call results in a bad response. See “Non-terminal-related STARTs and CICS BTS considerations” on page 46.
- Transaction initiation, for CICS BTS transactions only. Any existing affinities are ignored. Any dynamic routing API function call results in a bad response. See “Non-terminal-related STARTs and CICS BTS considerations” on page 46.
- Transaction termination, any existing affinities are ignored. Any dynamic routing API function call results in an error response.
- Transaction abend, any existing affinities are ignored. Any dynamic routing API function call results in an error response.

Non-terminal-related STARTs and CICS BTS considerations

When routing a transaction associated with either a CICS BTS activity or a non-terminal-related EXEC CICS START comand, the routing program, identified in the DSRTPGM SIT parameter, is invoked for both static and dynamic routing.

In the case of statically routed transactions, the EYU9WRAM program cannot alter the target region; see “Nondynamic transaction considerations” on page 46. In the case of dynamically routed transactions, the EYU9WRAM program may alter the target region; see “Selecting a target region from a set of target regions” on page 47. However, in both cases, the EYU9WRAM program is invoked only for:

- In the requesting region:
  - Notification
  - Route selection
  - Route selection error
  - Route attempt complete
- In the target region:

- – Transaction initiation
- – Transaction termination
- – Transaction abend

For details of the function of route initiation, see Dynamically routing DPL requests.

Selecting a target region from a set of target regions

As distributed, EYU9WRAM uses SM_SCOPE and SM_BALANCE to select a target region from the list of target regions defined for a transaction. The options that you set affect the way that the target region is chosen.

- SM_SCOPE returns a list of target regions for the current transaction occurrence. The target regions are those that are explicitly or implicitly associated with the requesting region by a workload specification.
- SM_BALANCE selects a target region from the list of candidate target regions returned by SM_SCOPE.

After an SM_SCOPE or SM_BALANCE call, the result of the operation is indicated in the response and reason fields of the EYURWCOM communication area.

Actions during route selection

A dynamic route selection occurs when a transaction or program is scheduled for routing.

- When SM_SCOPE is issued, the EYURWCOM communication area Scope Vector pointer is updated with the address of the target region scope list description area, SCOP_VECT. Also, the element count field is updated to contain the count of elements in the scope vector. Each element in the scope vector identifies a candidate target region and indicates its current status.

  During SM_BALANCE processing, the target region appearing first in the ordered list is selected for routing, unless the scope vector element EYURWSVE is marked ignore.
- When no affinity is associated with the current transaction occurrence, an SM_SCOPE call sorts the SCOP_VECT elements so that the target region that is the best candidate is placed first in the list.

  During SM_BALANCE processing, the APPLID and SYSID of the most suitable target region are placed in the fields WCOM_SEL_AOR and WCOM_SEL_SYSID, respectively. Typically, this will be the first target region represented in the scope vector. However, if the EYU9WRAM program has marked the WSVE-IGNORE field on some of the SCOP_VECT elements to denote that the target region is to be ignored, the first target region in the scope vector that is not marked to be ignored will be selected.

  Note:

  1. If the transaction identifier and the ids of the terminal and user associated with that transaction match a transaction group to which an affinity is defined, the affinity is made active during SM_BALANCE processing.
  2. With DPL, you can update the transaction id before the SM_SCOPE call.
- When an affinity is associated with the current transaction occurrence, an SM_SCOPE call normally causes only the target region with which the affinity exists to be identified in SCOP_VECT. An SM_BALANCE call then selects that target region. If the target region is not available for routing, the SM_BALANCE function will set a warning indicator. It will not attempt to select another target region.
- If the transaction is defined as a DTRTRAN, the EYURWCOM communication area will contain indicators denoting that the transaction is a DTRTRAN and that it will not be rejected by the Workload Manager. Processing is generally identical to the normal, non-DTRTRAN, case. Issuing SM_SCOPE returns a SCOP_VECT. Issuing SM_BALANCE selects a target region.

  The EYU9WRAM module might opt to reject the transaction before issuing the SM_SCOPE call. In this case, the Workload Manager returns to CICS with the reject indicator set to Y in the DFHDYPDS communication area. The sample EYU9WRAM program causes a DTRTRAN transaction to be rejected only if the transaction identifier is all blanks.

Actions during notification

Notification occurs when a static transaction or ATI transaction is being scheduled for routing.

- For statically routable transactions, ATI transactions, and BTS static routing requests, issuing SM_SCOPE causes the scope vector to contain a single target region. This is the target region associated with the transaction.

Actions during routing attempt complete

None, but you can tidy up and release any resources at this stage.

Actions during route selection error

A route selection error call occurs if the CICS link between the requesting region and target region is not available or is not defined.

- When no affinity is active: issue SM_SCOPE again. The candidate target regions identified by the SCOP_VECT it returns will not include the target region that caused the error. You can then use SM_BALANCE to select a new target region.
- When affinity became active as a result of the previous route selection:
  - If the lifetime associated with the affinity is PERMANENT, SYSTEM, ACTIVITY, or PROCESS, the target region causing the error will be included in scope list returned by SM_SCOPE. SM_BALANCE is required by affinity rules to select that target region. It also returns a warning to EYU9WRAM. EYU9WRAM should then notify the user that an error has occurred.
  - If the lifetime associated with the affinity is SIGNON, LOGON, DELIMIT, or PCONV, the active affinity status is removed before control returns to EYU9WRAM. When you reissue SM_SCOPE, the target region causing the error will not be included in the scope list. If the workload specification is defined with Create Affinity YES for the transaction group, EYU9WRAM is notified that an affinity is defined and will be activated when you issue SM_BALANCE.
- When affinity was activated by a previous transaction instance, and you issue SM_SCOPE again, the scope list returned contains the previously selected target region. Since SM_BALANCE is required by affinity rules to select that target region, a warning is returned to EYU9WRAM. EYU9WRAM should then notify the user that an error has occurred.

Actions during transaction termination

Transaction termination occurs when a transaction has terminated normally.

- EYU9WRAM should release any resources it may have acquired.
- Issuing SM_SCOPE or SM_BALANCE or SM_ROUTE causes an exception response to be returned.

Actions during transaction abend

Transaction abend occurs when a transaction has terminated abnormally.

- EYU9WRAM should release any resources it may have acquired.
- Issuing SM_SCOPE or SM_BALANCE or SM_ROUTE causes an exception response to be returned.

Actions during transaction initiation

Transaction initiation occurs when a CICS BTS-related transaction, or an enterprise bean-related transaction, has been routed to the target region. No specific action is taken. The call is issued for information purposes only.

Selecting a specific target region

SM_ROUTE requests that a specific target region is selected for routing. Use SM_ROUTE when you have application- or data-dependent requirements for explicit routing. For example, you might want to route a transaction associated with a specific user ID to a specific target region.

After an SM_ROUTE call, the result of the operation is stored in the response and reason fields of the EYURWCOM communication area.

A sample SM_ROUTE call is show here:

```
    CALL WAPIENPT(DA_TOKEN,SM_ROUTE)
```

DA_TOKEN identifies the dynamic routing API token supplied via the EYURWCOM communication area. This token is used by EYU9WAPI and must not be altered.

Actions during route selection

The actions during route selection are:

- If there is only one connection between a requesting region and the target region, you can supply either the SYSID or the APPLID of the target region (CICSPlex SM will determine the appropriate, corresponding ID). Place the APPLID in the WCOM_SEL_AOR field of the EYURWCOM communication area. Place the SYSID in the WCOM_SEL_SYSID field.

  If there are multiple connections between a requesting region and the target region, supply both the SYSID and the APPLID, as described previously, to ensure that the correct target region is selected. Note that when both the SYSID and APPLID are supplied, they are not validated.

  The target region need not be defined to CICSPlex SM. The affinity status is not checked. Thus, no affinity is established as a result of this call and, if an affinity was in effect, it is ignored.
- Issuing SM_ROUTE during Route Notify processing causes an exception response to be returned.

Actions during routing attempt complete

None, but you can tidy up and release any resources at this stage.

Actions during route selection error

The EYU9WRAM program may issue a message and terminate. You may then issue SM_ROUTE again specifying a different target region, or issue SM_SCOPE and SM_BALANCE.

Actions during transaction termination

- EYU9WRAM should terminate any resources it may have acquired.
- Calling any API function causes an exception response to be returned.

Actions during transaction abend

- Calling any API function causes an exception response to be returned.

Actions during transaction initiation

At transaction initiation:

- EYU9WRAM should terminate any resources it may have acquired.
- Calling any API function causes an exception response to be returned.

Creating an affinity

You can use SM_CREAFF to create an affinity if one does not already exist in the transaction group established for the transaction.

The transaction group must be defined with an affinity type and lifetime. The affinity created will have the same affinity type and lifetime as defined in the transaction group.

You should review “Affinity considerations” on page 50 before using SM_CREAFF.

Before calling SM_CREAFF, you must first call SM_SCOPE to obtain a scope list. SM_CREAFF will not create an affinity to an target region that is not in the scope list. In addition, you should set the EYUWRCOM communication area fields WCOM_SEL_AOR and WCOM_SEL_SYSID to the APPLID and SYSID, respectively, of the target region for which you want the affinity created.

EYU9WRAM contains a fragment of unexecuted code that you can use as a template for implementing the SM_CREAFF function.

The SM_CREAFF function cannot be called during:

- Route termination
- Route abend
- Route notify
- Route initiate
- Route complete

Deleting an affinity

You can use SM_DELAFF to delete an active affinity.

You should review “Affinity considerations” on page 50 before using SM_DELAFF.

Before calling SM_DELAFF, you must first call SM_SCOPE to obtain a scope list. The WCOM_AFF_STAT field in the EYURWCOM communication area contains a value indicating whether an affinity is active or committed. An active affinity can be deleted using SM_DELAFF. A committed affinity has a lifetime of SYSTEM or PERMANENT and cannot be deleted using SM_DELAFF.

EYU9WRAM contains a fragment of unexecuted code that you can use as a template for implementing the SM_DELAFF API function.

The SM_DELAFF function cannot be called during:

- Route notify
- Route initiate
- Route complete

Affinity considerations

When you define an affinity, you must specify its lifetime. The affinity normally persists until its specified lifetime expires.

You can define one of the following types of lifetime:

Activity
Expires when the CICS BTS activity ends

Delimit
Expires when the PCONV mode of the transaction is END

Logon
Expires when the terminal user logs off.

Pconv
Expires when a transaction uses EXEC CICS RETURN specifying no NEXTTRANSID or the PCONV mode of the transaction is END.

CICS does not support pseudoconversations for APPC (LUTYPE6.2) devices.

Permanent
Expires when the workload of which the target region is a part terminates

Process
Expires when the CICS BTS process ends

Signon
Expires when the terminal user signs off.

System
Expires when the target region terminates

UOW
Expires when the unit of work associated with the transaction ends. The unit of work ends either when a CICS SYNCPOINT or ROLLBACK request is run, or when the originating task terminates.

There are circumstances when SM_SCOPE generates a scope list containing a single affinity target region, but the target region is not available for routing. This happens when:

- The target region is down.
- The CICS link to the target region is down.
- The target region is currently active but it was shutdown and restarted after the affinity was created.

In these cases, the EYU9WRAM default processing issues a terminal message indicating that the affinity target region is not available and causes the transaction to terminate. If the affinity lifetime is PCONV (pseudoconversation), CICSPlex SM automatically deletes the affinity because it has expired (there is no NEXTTRANSID). However, the default EYU9WRAM processing does not delete any other affinities because the characteristics of the actual affinity to the target region are unknown. For example, a LOGON affinity may involve the use of the TCTUA to pass information to the target region. If the affinity is deleted when the target region is not available, the next transaction for the transaction group would cause a new target region to be selected. The transaction might fail upon using the TCTUA contents when routed to the new target region.

The sample EYU9WRAM program, which implements the default processing, contains a subroutine that checks on affinity status after a call to SM_SCOPE. When an affinity is active to an target region, but not committed, and the affinity target region status is not OK, the subroutine sends a message to the terminal user and then exits so that the EYU9WRAM program terminates. The subroutine contains an unexecuted code fragment that can be used to delete the affinity using the SM_DELAFF call. Before the unexecuted code fragment is a series of tests for the affinity lifetime of the active affinity. One or more of the branches can be changed to jump to the code fragment. The code fragment itself deletes the affinity, issues a message, and then returns so that the EYU9WRAM program exits, thereby causing the transaction to terminate. This processing can be modified so that the affinity is deleted, no message is issued, and the subroutine exits causing the mainline process to reexecute the SM_SCOPE call. In that case, a new set of target regions is received for use by SM_BALANCE.

The same subroutine also contains an unexecuted code fragment that can be enabled to create an affinity. In this case, the subroutine has determined that an affinity is defined but not active. It then checks the WCOM_AFF_AUTO indicator to determine whether CICSPlex SM should automatically create affinities during SM_BALANCE. The subroutine exits normally no matter what the answer is. You can enable the SM_CREAFF fragment in order to cause an affinity to be created. The SM_CREAFF call may be used regardless of whether WCOM_AFF_AUTO does or does not indicate the automatic creation of affinities during SM_BALANCE. The SM_CREAFF call can therefore be used to:

- Create an affinity when CICSPlex SM will not.
- Create an affinity to a target region that CICSPlex SM would ordinarily not select.

The SM_CREAFF code fragment creates an affinity to the target region that appears first in the scope list returned by SM_SCOPE.

The following fields in the EYURWCOM communication area provide information regarding CICSPlex SM affinity processing.

- WCOM_AFF_STAT
- WCOM_AFFAOR_STAT
- WCOM_AFF_TYPE
- WCOM_AFF_LIFE
- WCOM_AFF_AUTO
- WCOM_WORK_NAME
- WCOM_TGRP_NAME

CICSPlex SM data areas

The EYU9WRAM program uses two data areas, communication area EYURWCOM and scope vector element EYURWSVE.

Literals for EYURWCOM are defined in EYURWCOD and literals for EYURWSVE are defined in EYURWSVD.

The names of the copy books that you can use to map these data areas are identified in Sample programs and copy books.

The scope vector element, EYURWSVE, contains information about the individual target regions associated with the list of candidate target regions returned by SM_SCOPE.

Creating a user-replacement module for EYU9WRAM

You can use the sample procedures provided in CICSTS55.CPSM.SEYUPROC to create replacement modules for EYU9WRAM.

The procedure members are as follows:

Assembler	EYUEITAL
C	EYUEITDL
PL/I	EYUEITPL
COBOL	EYUEITVL

1. Copy these procedures to a cataloged system procedure library. You can use the samples in Sample user-replacement modules.
2. Use the JCL samples to run the procedure and create the replacement module for EYU9WRAM. Replace the lower case values shown in the sample JCL with the appropriate values for your site. Also, you cannot use the CICSPlex SM API in EYU9WRAM.

Requesting additional dynamic routing support

This section contains Product-sensitive Programming Interface Information.

You can access the CICSPlex SM workload management facilities directly from an application program.

Note: If you are running CICS Transaction Server for OS/390® Version 1 Release 3 and later, it is recommended that you use the CICSPlex SM-supplied program EYU9XLOP for all your dynamic routing requirements. You need the information in this section only if you want to continue to use this function during migration to CICS Transaction Server for OS/390 Version 1 Release 3.

If your routing regions are CICS Transaction Server for OS/390 Version 1 Release 3 and later, or if you are using CICS BTS or enterprise beans, you do not need the information in this section. You should use the distributed routing facilities described in Dynamic routing with CICSPlex SM.

To write such a program, you should be familiar with:

- The interface between the CICS relay program and the defined dynamic routing program.
- CICSPlex SM workload management processing, as described in Workload management.

- The CICSPlex SM workload management routing action module, EYU9WRAM, as described in Dynamic routing with CICSPlex SM.

The primary method of invoking CICSPlex SM workload management is by defining module EYU9XLOP to CICS as the dynamic routing program. This causes EYU9XLOP to be invoked by the CICS relay program, allowing CICSPlex SM workload management to make a routing decision about each work request that can be sent to another CICS system.

If necessary, CICSPlex SM workload management can be invoked directly from an application program, without going through the CICS relay program. For CICS releases prior to CICS Transaction Server for OS/390 Version 1 Release 3, you can use CICSPlex SM Workload Management to determine the best target region for a distributed program link (DPL) from a requesting region. The application program you write must follow certain guidelines:

- The program must run in a CICS system that is defined and running as a routing region. You cannot access CICSPlex SM workload management from a target region.
- The program must issue its INIT and TERM calls from the same task. CICSPlex SM expects the program to imitate the processing done by the CICS relay program, which always calls at route initiation and route termination.

Note: Accessing workload management through a program that issues an EXEC CICS START command is not recommended with an asynchronous transaction start. Furthermore, EXEC CICS START commands that are not related with either a terminal or a user should not be used for transactions that have affinity relationships, as those relationships cannot be correctly resolved using this interface.

Accessing workload management from a program using a dynamic program load is not supported. CICSPlex SM workload management uses the CICS application programming interface to gather information about the current environment in which it is running. Due to DPL restrictions some of that information is not available, and unpredictable results may occur.

In order to make a routing decision, CICSPlex SM workload management needs certain information that is normally supplied by the CICS relay program. The CICS relay program derives the information from data available to CICS for the instance of the transaction being relayed. When you invoke CICSPlex SM workload management directly, you need to provide this information, including terminal-oriented data and the equivalent of a transaction name to identify the work to be routed.

How to start CICSPlex SM workload management

To start the CICSPlex SM workload management facilities, use an EXEC CICS LINK command, specifying EYU9XLOP as the program and identifying EYURWTRA as the communication area.

The format of the command is:

```
EXEC CICS LINK
     PROGRAM(EYU9XLOP)
     COMMAREA(EYURWTRA)
     LENGTH(=AL2(WTRA_LENGTH))
```

Much of the data normally passed from the CICS relay program to the dynamic routing program in the DFHDYPDS communication area must be generated by your program and passed to CICSPlex SM workload management. Other data that is normally derived by CICSPlex SM workload management from a transaction- and terminal-oriented environment must also be provided by your program. The assembler copy book EYURWTRA provides a map of the communication area to be passed to CICSPlex SM workload management.

The corresponding copy books and the CICSPlex SM libraries in which they can be found are:

Language	Member name	Library
Assembler	EYUAWTRA	SEYUMAC
COBOL	EYULWTRA	SEYUCOB
PL/I	EYUPWTRA	SEYUPL1

Language	Member name	Library
C	EYUCWTRA	SEYUC370

Before invoking CICSPlex SM workload management, you must initialize the EYURWTRA communication area.

For a list of the possible response and reason code values that can be returned in WTRA_API_RESP and WTRA_API_REAS, refer to the assembler copy book member EYURWCOD.

Processing considerations

After CICSPlex SM workload management processing, the WTRA_RESPONSE field contains a return code of either zero (0) or 8.

If the return code is 0, WTRA_SYSID contains the SYSID of the selected target region, and WTRA_APPLID contains its APPLID. With that information, you can proceed to route the unit of work to the target region.

If the return code is 8, you should inspect the WTRA_API_RESP and WTRA_API_REAS fields for more information. These fields contain the last response and reason codes returned to the CICSPlex SM workload management routing action module (EYU9WRAM). In most cases, the response and reason codes describe what caused CICSPlex SM workload management to generate the return code of 8. Based on that information, you can decide how to proceed.

In some cases, however, the WTRA_RESPONSE field may contain a return code of 8, while the WTRA_API_RESP and WTRA_API_REAS fields have not been set by CICSPlex SM workload management. In those cases, an error was detected before invoking the routing action process. CICSPlex SM user trace records are written to the CICS trace data set for those types of errors. When this happens, your program should terminate processing.

When the return code is 8, you should also inspect the WTRA_OPTER field. If its value is WTRA_CALLYES:

1. Set the WTRA_FUNC field to WTRA_FUNCTRM for route termination.
2. Reinvoke CICSPlex SM workload management.
3. Terminate processing for the proposed unit of work.

If the WTRA_OPTER field contains a value of WTRA_CALLNO, terminate further processing for the proposed unit of work.

After the unit of work has completed in the target region, set the WTRA_FUNC field according to how the unit of work completed. If the unit of work completed successfully:

1. Set the WTRA_FUNC field to WTRA_FUNCTRM for route termination.
2. Optionally, set the WTRA_NEXTTRAN field to the ID of the next transaction to be used in the process. If there is no next transaction, set the field to all blanks.
3. Reinvoke CICSPlex SM workload management to request termination processing for the transaction.

If the unit of work abended or generated an error that you consider abnormal:

1. Set the WTRA_FUNC field to WTRA_FUNCABD to notify CICSPlex SM workload management that the routed transaction abended.
2. Reinvoke CICSPlex SM workload management to request abnormal termination processing for the transaction.

Route error considerations

After you receive the SYSID and APPLID of a target region, you can attempt to route the unit of work to the target region.

However, the routing attempt may result in an error condition if the system is unavailable. When this happens, you should:

1. Set the WTRA_FUNC field to WTRA_FUNCERR to indicate a routing error.

2. Specify the reason for the error in the WTRA_ERR field.
3. Reinvoke CICSPlex SM workload management.

CICSPlex SM workload management provides another target from the scope list, if possible. If no other systems are available, the WTRA_RESPONSE field is set to 8 and the WTRA_API_RESP and WTRA_API_REAS fields describe the reason for the failure. You may have to invoke CICSPlex SM workload management more than once for routing errors until you receive a valid target region.

Transaction affinity considerations

If the WTRA_TRANID, WTRA_USERID and WTRA_LUNAME fields cause CICSPlex SM workload management to select a transaction group that includes an affinity, the rules of transaction affinities are obeyed.

In this case, you should ensure that the WTRA_NEXTTRAN field is initialized with the ID of the next transaction before calling the route termination function.

If your units of work, or any subset of them, have an affinity relation defined, be careful in how you define that affinity. Since your program cannot provide SIGNOFF or LOGOFF processing, an affinity lifetime of SIGNON or LOGON, though valid, may cause an affinity relation to be built that is not removed until either the requesting region or the target region terminates.

The WTRA_NEXTTRAN field is most useful for pseudo-conversation affinities. For this affinity lifetime, CICSPlex SM workload management normally issues the EXEC CICS ASSIGN NEXTTRANSID command to retrieve the next transaction ID, if any, for the facility. Since that command is not available when CICSPlex SM workload management is invoked by your program, it is your responsibility to provide the ID of the next transaction.

For a description of CICSPlex SM workload management affinity processing, see Taking affinity relations into consideration. For details on specifying affinity relation and lifetime values, see Creating a transaction group.

Abend compensation considerations

If the target region is running CICS TS, transactions initiated by your program can participate in abend compensation processing, provided that the active CICSPlex SM workload specifies it.

You should note that, if the work requests are either non-terminal-related EXEC CICS START commands, or BTS activities, the routing and target regions must be within the same MVS image for the routing region to detect that an abend has occurred.

Your program should notify CICSPlex SM workload management that a particular unit of work abended by specifying WTRA_FUNCABD in the WTRA_FUNC field. Then, if possible, CICSPlex SM workload management will tend to avoid selecting the same target region for the transaction (or set of transactions, if an affinity is defined) on subsequent route select functions.

For a description of CICSPlex SM workload management abend compensation processing, see Abend probabilities and workload management. For details on requesting abend compensation, see Creating a workload specification.

CMAS availability considerations

If your program issues an EXEC CICS LINK for the EYU9XLOP program, but the requesting region in which your program is running has not yet fully connected to its target CMAS, you see message EYUXL0020I, indicating that an ESSS connection is in progress.

Your program waits indefinitely until the CMAS becomes available and the requesting region is joined to a workload. If you have connected successfully to the CMAS and the MAS, but the CMAS has not installed any workload definitions to the MAS, you see a message indicating that the requesting region is waiting for a workload. You may need to set up and install workloads to rectify this situation.

Note: Once the requesting region has successfully connected to the CMAS, the CMAS can become inactive and workload management remains active.

Sample calling sequence

This code example illustrates a sample calling sequence for a program that accesses CICSPlex SM workload management facilities. It is not intended to be used as a sample program and is, therefore, not complete in every detail.

```
*----------------------------------------------------------------------*
         DFHEISTG  ,                   Define Workarea
WRK_WTRA       DS   CL(WTRA_LENGTH)
               DS   0D
WRK_UOWCOMM    DS   0C
WRK_UOW_RESP   DS   F
WRK_UOWCOMM_L  EQU *-WRK_UOWCOMM
         COPY  EYURWTRA                Include DSECT to map WTRA
SRVPGM   DFHEIENT EIBREG=R11,DATAREG=R13,CODEREG=R12

*----------------------------------------------------------------------*
* Initialize the WTRA COMMAREA.                                        *
*----------------------------------------------------------------------*
         LA    R8,WRK_WTRA             --> WTRA
         USING EYURWTRA,R8             *** USING WTRA ***
         MVC   WTRA_SLENGTH,=AL2(WTRA_LENGTH)
                                       Set length of block.
         MVI   WTRA_ARROW,C'>'         Set arrow.
         MVC   WTRA_NAME,=C'EYURWTRA'  Set the name.
         MVI   WTRA_BLANK,C' '         Set blank delimit.
         MVC   WTRA_PGMNAME,=CL8'SRVPGM  '
                                       Set program name.
         MVC   WTRA_TERMID,=CL4'TRM1'  Set TermID.
         MVC   WTRA_USERID,=CL8'USR1'  Set USERID.
         MVC   WTRA_LUNAME(8),=CL8'.NET1'
                                       Set LUNAME.
         MVC   WTRA_TRANID,=CL8'TRN1'  Set TRANID.
         MVC   WTRA_SYSID,=C'SYS1'     Set SYSID.
         MVC   WTRA_APPLID,=C'APPLID1' Set Applid.
*
* Set Application Context
*
         MVC   WTRA_PLATFORM,=CL64'PLATFORM_1.0.0'
         MVC   WTRA_APPLICATION,=CL64'APPLICATION_1.0.2'
         MVC   WTRA_MAJORVER,=F'1'
         MVC   WTRA_MINORVER,=F'1'
         MVC   WTRA_MICROVER,=F'1'
         MVC   WTRA_OPERATION,=CL64'PLEASE_ROUTE'
         MVI   WTRA_FUNC,WTRA_FUNCSEL  Set the Route Select Function.
         MVI   WTRA_DYRTYPE,WTRA_DYRTYPE_DYN
*----------------------------------------------------------------------*
* Invoke the WLM MAS Agent for Route Select.                           *
*----------------------------------------------------------------------*
ROUTE_SELECT   DS  0H
         BAS   R5,LINK_WLM             Go do it.
         CLC   WTRA_RESPONSE,=F'0'     Call go OK?
         BNE   CHECK_OPTER             ..no.
         BAS   R5,START_UOW            Go Start the UOW.
         LTR   R15,R15                 Work Completed?
         BZ    ROUTE_TERM              ..yes.
         BP    ROUTE_ABND              UOW gave non zero return code.
         C     R15,=F'-4'              SYSID error?
         BNE   ROUTE_ABND              ..no.
```

```
*----------------------------------------------------------------------*
* Invoke the WLM MAS Agent for Route Error.                            *
*----------------------------------------------------------------------*
ROUTE_ERROR    DS  0H
         MVI   WTRA_FUNC,WTRA_FUNCERR  Set the Termination Function.
         MVI   WTRA_ERR,WTRA_ERROUT    Say out of service.
         B     ROUTE_SELECT            Go Get another system.

*----------------------------------------------------------------------*
* Invoke the WLM MAS Agent for Route Abend.                            *
*----------------------------------------------------------------------*
ROUTE_ABND     DS  0H
         MVI   WTRA_FUNC,WTRA_FUNCABD  Set the Abend function.
         BAS   R5,LINK_WLM             Go terminate.
         CLC   WTRA_RESPONSE,=F'0'     OK?
         BE    EXIT_ABD                ..yes.
CHECK_OPTER    DS  0H
```

```
         CLI   WTRA_OPTER,WTRA_CALLYES Call WLM for Term?
         BNE   WLM_CALLERR             ..no.

*---------------------------------------------------------------------*
* Invoke the WLM MAS Agent for Route Termination                      *
*---------------------------------------------------------------------*
ROUTE_TERM     DS  0H
         MVI   WTRA_FUNC,WTRA_FUNCTRM  Set the Termination Function.
         MVC   WTRA_NEXTTRAN,=CL8'TRN2'                                 X
                                       Set the next TRANID.
         BAS   R5,LINK_WLM             Go terminate.
         CLC   WTRA_RESPONSE,=F'0'     OK?
         BE    EXIT_TERMOK             ..yes. .

*---------------------------------------------------------------------*
* The WLM MAS AGENT returned an 8 in WTRA_RESPONSE                    *
*---------------------------------------------------------------------*
WLM_CALLERR    DS  0H

*---------------------------------------------------------------------*
* Process completed successfully                                      *
*---------------------------------------------------------------------*
EXIT_TERMOK    DS  0H

*---------------------------------------------------------------------*
* Route Abend Call returned a 0                                       *
*---------------------------------------------------------------------*
EXIT_ABD       DS  0H
EXIT           DS  0H
         EXEC  CICS RETURN
*---------------------------------------------------------------------*
* LINK_WLM      : Link to the WLM MAS AGENT                           *
*---------------------------------------------------------------------*
LINK_WLM       DS  0H
         EXEC  CICS LINK PROGRAM(EYU9XLOP)                              X
               COMMAREA(EYURWTRA) LENGTH(=AL2(WTRA_LENGTH))             X
         BR    R5                      Exit routine.
```

```
*---------------------------------------------------------------------*
* START_UOW     : Start the Unit Work.                                *
*---------------------------------------------------------------------*
START_UOW      DS  0H
         EXEC  CICS LINK PROGRAM(UOWPGM)                                X
               COMMAREA(WRK_UOWCOMM) LENGTH(=AL2(WRK_UOWCOMM_L))        X
               SYSID(WTRA_SYSID)                                        X
               RESP(WRK_EIBRESP)
         CLC   WRK_EIBRESP,DFHRESP(NORMAL)                              X
                                       Did call go ok?
         BE    START_UOWL              ..yes.
         L     R15,=F'-4'              Assume SYSIDERR
         CLC   WRK_EIBRESP,DFHRESP(SYSIDERR)                            X
                                       Is there one?
         BE    START_UOWX              ..yes.
         L     R15,=F'-8'              Load Failure code.
         B     START_UOWX              Go exit.
START_UOWL     DS  0H
         L     R15,WORK_UOW_RESP       Load Response Code.
START_UOWX     DS  0H
         BR    R5                      Exit routine.
```

Chapter 4. Administering workloads with CICSPlex SM

The CICSPlex SM workload management (WLM) function provides the capability to control where your work requests are run, in order to optimize performance and workload throughput. WLM achieves this by *dynamic routing*. WLM uses the CICSPlex SM dynamic routing program EYU9XLOP to route work requests to the most suitable target region from a predefined set of target regions.

For an introduction to the concepts and architecture of CICSPlex SM workload management, see How it works: CICSPlex SM workload management (WLM).

Workload routing

Workload routing is the directing of transactions or programs among a group of target regions according to the availability, health, and activity levels of those target regions. You can use workload routing in addition to, or in place of, workload separation.

CICSPlex SM can route a workload among a defined group of target regions by selecting, when each transaction is initiated, the target region that is likely to deliver the best performance.

Workload routing is statistical. CICSPlex SM uses one of the following four algorithms to determine which target region processes the work:

- Goal
- Queue
- Link neutral goal
- Link neutral queue

If any transaction affinities are outstanding for the transaction being routed, the affinity target region is selected, regardless of the algorithm used.

With the goal and link neutral goal algorithms, the appropriate target region is selected based on the ability of that region to achieve the expected response time. With the queue and link neutral queue algorithms, the appropriate target region is selected such that the load is distributed across a set of target regions.

If the goal or link neutral goal algorithm does not identify a specific target region, the queue or link neutral queue algorithm, respectively, is applied to the remaining set of target regions.

If all the available target regions are equally capable of handling the work, a target region is selected randomly from the group. Therefore, in systems that are lightly loaded, there is no predetermined order in which work is allocated to equally capable target regions.

The queue algorithm and the goal algorithm allow for the type of connection between a target region and its routing region. A weighting factor is allocated to each type of connection, such that links are put in the following order of priority:

- Local
- MRO/IRC and MRO/XM (local LPAR)
- MRO/XCF (remote LPAR)
- IPIC (local LPAR)
- IPIC (remote LPAR)
- LU6.2
- Indirect

CICSPlex SM uses the weighting factor of the link as a multiplier against the task load and other factors to determine an overall routing weight. At the end of the evaluation, the region with the lightest weight is usually selected as the target region.

For example, if all other factors are equal, a target region that is connected to its requesting region by using MRO/XCF is preferred to a target region that is connected by using IPIC. IPIC connections between routing and target regions in the same LPAR have a lower weighting than IPIC connections to target regions in a different LPAR, so that if other factors are equal, local IPIC connections are preferred to remote IPIC connections.

Control level for workload routing

To use workload routing, you must specify a default routing algorithm for the workload at the workload specification (WLMSPEC) level. You can optionally specify a routing algorithm at the transaction group (TRANGRP) level. An algorithm specified in a transaction group overrides the default algorithm that is associated with the workload specification.

The default routing algorithm is applied to every routed dynamic transaction in the workload, except those transactions that are associated with a transaction group that has a routing algorithm specified. You can specify one of the following routing algorithms:

- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

To change the routing algorithm specified at the workload specification level, you must close down all regions that participate in the workload so that workload is refreshed with the new algorithm specification.

At the transaction group level, you can specify a routing algorithm dynamically. The specified dynamic routing algorithm is applied to every routed dynamic transaction that is associated with the transaction group. Therefore, you can apply an alternative routing algorithm to specific transaction codes in the same workload.

If you specify an alternative routing algorithm at the transaction group level, you can change workload routing characteristics for specific target regions dynamically without stopping your routing region. If you modify an installed transaction group, you must discard its associated WLM definition (WLMDEF) and then reinstall it, so that the transaction group named by the WLM definition is also refreshed. To change the routing algorithm type immediately without discarding and reinstalling the associated WLMDEF, you can use the Active workload transaction groups (WLMATGRP) views and the **SET** command to change the ALGTYPE attribute.

You can specify one of the following routing algorithms:

- INHERIT
- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

INHERIT means that transaction group uses the routing algorithm that is associated with the workload specification for the workload.

Link neutral workload routing

In some situations, link weighting has a strong impact on the routing behavior and can prevent CICS transaction routing across system boundaries in a single-site parallel sysplex environment. To use workload routing without link weighting, you can use the link neutral goal and link neutral queue algorithms. These algorithms are identical to the goal and queue algorithms, respectively, except that the type of connection between the routing and target region is not considered.

Link neutral algorithms can be useful to route dynamic transactions, for example those that might require services from MVS subsystems. With queue and goal algorithms, routers focus dynamic traffic on the systems with the fastest links, which probably reside in the same LPAR. This behavior might overload subsystems in the local MVS image, and under use the remote MVS images that participate in the workload. If you assign these transactions to a transaction group that uses a link neutral algorithm, the routed dynamic traffic is routed more evenly between the local and remote LPARs, and the load on those subsystems is spread.

However, remember that with a link neutral algorithm, if other factors are equal, all target regions have equal preference. The most remote target regions, connected with the slowest telecommunications links, might have equal preference with a locally connected MRO region, or the routing region itself if it is part of the routing target scope. Therefore, consider carefully whether to specify a link neutral algorithm at the workload specification (WLMSPEC) level, because it might affect every dynamically routed transaction. As a result, the workload manager might not select the best target region for dynamic routing traffic and the overall workload throughput might deteriorate.

If you require a link neutral algorithm for a specific transaction set, assign the algorithm to the workload management transaction group (TRANGRP) that identifies those transactions.

The queue algorithm

When CICSPlex SM uses the queue algorithm, if the transaction being routed has no active affinities, the load is distributed across a set of target regions. Selection criteria include the current task load, the health state, and the type of connection between the router and the target region.

If the transaction being routed has no active affinities, the queue algorithm causes CICSPlex SM to select the target region that conforms to the following set of criteria:

- The region has the shortest queue of work waiting to be processed, relative to the maximum number of tasks permitted in the target region. This queue of work, also called the *load count*, is the count of all active and queued user tasks. By default, tasks queued for both MAXTASKS and TRANCLASS attributes are included in the load count. Use of the "Task load queue mode" attribute in the CSYSDEF resource table allows sites to exclude tasks queued for the TRANCLASS attribute from the load count. For more information about the "Task load queue mode" attribute in the CSYSDEF resource table, see CICS system definitions - CSYSDEF.
- The region is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- The region is the least likely to cause the transaction to stop.
- The region is the shortest path for interregion communication.
- The region has a z/OS WLM health value in the range 1-100.

The queue algorithm maximizes work throughput and standardizes response times across the CICSplex. This algorithm is very robust; it can accommodate differences in processor power, different maximum task values in the target regions, asymmetric target region configurations, and an unpredictable workload.

The link neutral queue algorithm

The link neutral queue (LNQUEUE) algorithm corresponds to the queue algorithm, except that the type of connection between the routing and target region is not considered.

If a transaction being routed has no active affinities, the link neutral queue algorithm causes CICSPlex SM to select the target region that conforms the following set of criteria:

- The region has the shortest queue of work waiting to be processed, relative to the maximum number of tasks permitted in the target region. This queue of work, also called the *load count*, is the count of all active and queued user tasks. By default, tasks queued for both MAXTASKS and TRANCLASS attributes are included in the load count. Use of the "Task load queue mode" attribute in the CSYSDEF resource table allows sites to exclude tasks queued for the TRANCLASS attribute from the load count.

For more information about the "Task load queue mode" attribute in the CSYSDEF resource table, see CICS system definitions - CSYSDEF.

- The region is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- The region is the least likely to cause the transaction to stop.
- The region has a z/OS WLM health value in the range 1-100.

The link neutral queue algorithm provides efficient work throughput and response times across the CICSplex. Similarly to the queue algorithm, this algorithm can accommodate differences in processor power and different maximum task values in the target regions. However, it does not allow for communications link speeds between the router and a target. Compared to the queue algorithm, this algorithm might spread a workload across a target scope more evenly, but the workload might not complete as quickly.

The goal algorithm

The Goal algorithm may be used when dynamic routing decisions are primarily concerned with the ability of a region to achieve response time targets.

Goal mode routing should only be employed where routers and targets are managed by the same CMAS in the following scenarios:

- Dynamic routing using DTRPGM for dynamic transactions.
- Dynamic routing using DTRPGM for EXEC CICS START TERMID over APPC or MRO connections.
- Distributed routing using DSRTPGM for business transaction service routing.

In any other dynamic routing scenario, the QUEUE or LNQUEUE algorithms should be used.

If a transaction being routed has no active affinities then the goal algorithm causes CICSPlex SM to select the target region that conforms to the following set of criteria:

- Is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- Is the least likely to cause the transaction to stop.
- Is the most likely to meet the transaction response-time goals (either average or percentile) set for it (and for other transactions in its class) using the Workload Manager component of z/OS.
- Is the shortest path for interregion communication.
- Is the AOR with the shortest queue of transactions, relative to MAXTASK, waiting to be processed.
- Has a z/OS WLM health value in the range 1 - 100.

The link neutral goal algorithm

The link neutral goal (LNGOAL) algorithm corresponds to the goal algorithm, except that the type of connection between the routing and target region is not considered.

If a transaction being routed has no active affinities, the link neutral goal algorithm causes CICSPlex SM to select the target region that conforms the following set of criteria:

- The region is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- The region is the least likely to cause the transaction to stop.
- The region is the most likely to meet the transaction response-time goals, either average or percentile, set for it (and for other transactions in its class) using the Workload Manager component of z/OS.
- The region is the AOR with the shortest queue of transactions, relative to MAXTASK, waiting to be processed.
- The region has a z/OS WLM health value in the range 1-100.

Effect of the z/OS WLM health service on CICSPlex SM workload routing

In a CICS region, if the z/OS Workload Manager (z/OS WLM) health service is active, CICSPlex SM WLM takes account of the region's z/OS WLM health state in the routing decision.

Prerequisite:

For CICSPlex SM WLM to take into account the z/OS WLM health state of the target regions, the following conditions apply:

- Both the routing and target regions must be at a minimum of CICS TS V5.4 (CICS and CICSPlex SM libraries); otherwise, the z/OS WLM health state will be ignored, and the target regions are deemed as ready to receive work.
- The z/OS WLM health service must be enabled in the CICS region. It is enabled by default. If the service is disabled, the region is also deemed as ready to receive work. To activate this service, you must set the **`WLMHEALTH`** system initialization parameter.

How the z/OS WLM health state of CICS regions affects CICSPlex SM routing decisions

The z/OS WLM health state of a region is reflected by a z/OS WLM health value in the range 0 - 100. When deciding where to route work, CICSPlex SM WLM might assign an additional weight in the routing algorithm for the target region, based on its health value:

- If the target region has a z/OS WLM health value of zero, it is not eligible to receive work. CICSPlex SM WLM will not route work to this region. Any workload affinities that are associated with this region will remain and be honored; however, workload routing based on those affinities will fail while the health value is zero, and will result in either message EYUWR0003W or a SYSIDERR condition. Message EYUWR0003W indicates that the EYU9WRAM program was in effect and the affinity AOR was not available. To resolve such routing failures, initiate a warm-up for the region so that the health value is non-zero, and then rerun your workload. See Initiating a CICS system warm-up.
- For any target region with a health value between 1 and 99, CICSPlex SM WLM assigns an additional weight in the routing algorithm, based on the health value. The greater the health value, the lower the weight, which makes this region more favorable in the routing decision. Likewise, the lower the health value, the greater the weight, which reduces the chances of this region to receive more work. Note that a region that is assigned a high additional weight remains active and might still receive work if no other healthier regions are available in the same scope.
- When a target region has a health value of 100, no additional weight is assigned.

During the CICS warm-up or cool-down process, CICS adjusts the z/OS WLM health value of the region. As a result, this affects the routing decision of CICSPlex SM WLM, which restricts workload into the region during that process. In general, when a region is warming up, its health value increases periodically, and the region might gradually receive a larger workload; when a region is cooling down, its health value decreases periodically, and the region might gradually receive less work until it becomes ineligible as a target. For more information about what happens during CICS warm-up or cool-down, see CICS warm-up and cool-down by use of z/OS Workload Manager health service.

Making a region ineligible as a workload routing target

You can deliberately make a region ineligible as a workload routing target in CICSPlex SM routing decisions by setting its z/OS WLM health open status to `CLOSE` in the **MVS workload management** (MVSWLM) view. This will gradually make the region less favorable until the region's z/OS WLM heath value drops to 0, indicating that the region becomes ineligible. Additionally, you can change a region's z/OS WLM heath value to zero immediately, by setting its z/OS WLM health open status to `IMMCLOSE`.

Workload separation

Workload separation routes work from a requesting or routing region among a set of target regions, but the requirements of users, terminals, the transactions themselves, and their BTS process type, can influence which target region set is used.

The routing of particular transactions or programs to a particular group of target regions is based on any combination either of user ID, terminal ID, and transaction group. For BTS processes, routing is based on a combination of transaction group and process type. For enterprise bean-related transactions, routing is based on transaction group. For Link3270 bridge requests, separating by terminal luname is subject to some restrictions. See Separating Link3270 bridge workloads for more information.

For example, you can specify that:

- Any transaction initiated by users whose user IDs begin with the characters "PAY" must be routed to a target region in CICS system group AORPAY1.
- Any transaction initiated from a terminal whose LU name begins with the characters "NYORK" must be routed to a target region in CICS system group AORNYORK.
- Any transaction belonging to the transaction group ACCOUNTS must be routed to a target region in CICS system group AORACCNT.
- All processes associated with a BTS process type TRAVEL are routed to a target region in one BTS-set, and all processes associated with BTS process-type PAYROLL are routed to another BTS-set.
- All enterprise bean-related transactions belonging to transaction group STOCK are routed to any target region in CICS system group STOCKTG, if the user id matches that in the workload definition.

The decision as to which region is selected from the group of target regions that meet the workload separation criteria is based on the same routing criteria as described in “Workload routing” on page 59. The algorithm that is applied when making the routing decision can either be specified at the individual workload separation level or can be left to default to the algorithm used by basic workload routing.

Inter-transaction affinity

An inter-transaction affinity is a relationship between transactions, of a specified duration, that requires them to be processed by the same target region. For example, you might have a pseudoconversation made up of three separate transactions, and each transaction passes data to the next transaction in the sequence via a temporary storage queue (which may be shared in the distributed model). You would then specify that all three transactions must be processed by the same target region, and that this affinity lasts for the duration of the pseudoconversation.

If you did not define this affinity to CICSPlex SM, each transaction could be routed to a different target region and would therefore be unable to access temporary-storage data left by the previous transaction. The target region itself is selected by CICSPlex SM from the specified target scope.

Workload management and the IBM CICS Interdependency Analyzer for z/OS understand affinities between BTS processes and activities. BTS itself does not introduce affinities, and discourages programming techniques that do, but it does handle existing code that may introduce affinities. You should define such affinities to workload management, so that it can make sensible routing decisions. It is particularly important to specify each affinity's lifetime; failure to do so may restrict unnecessarily workload management's routing options.

Workload management and the IBM CICS Interdependency Analyzer for z/OS do not understand affinities between routable non-terminal-related EXEC CICS START commands, or between DPLs not associated with a user id or a terminal. You should take steps either to remove any affinities from your applications, or to ensure that your applications honor any affinities.

Note that, if data is passed between transactions via the COMMAREA on the **`EXEC CICS RETURN`** command, no such affinity exists: the COMMAREA is passed back to the requesting region, and so can be passed to the target selected to process the next transaction in the sequence.

How inter-transaction affinities are honored

When the first transaction from a group of related transactions is started, CICSPlex SM selects an appropriate region from the specified target scope.

If there is more than one suitable region in the target scope, CICSPlex SM selects one using the current workload routing algorithm. Subsequent transactions in the same group that meet the affinity criteria are directed to the same region as the first transaction. If subsequent transactions do not meet the affinity criteria (for example, if the same pseudoconversation is started from a different user ID), the selection process for a suitable region starts again.

Managing workloads with CICSPlex SM workload management

CICSPlex SM workload management optimizes processor capacity in your enterprise. Workload management achieves this by dynamically routing transactions and programs to whichever CICS region is the most appropriate at the time, taking into account any transaction affinities that exist.

When you have identified your workload management requirements, define them to CICSPlex SM as described in Configuring workload management.

Planning for WLM

This section provides some instructions to help you determine the extent to which you can use CICSPlex SM workload management in your enterprise.

Choosing between workload routing and workload separation

Use workload routing wherever possible because it makes the best use of the available CICS systems, and provides opportunities for increased throughput and improved performance. Implement workload separation (by process type, user, terminal, and transaction) only where strictly necessary because it prevents full exploitation of CICSPlex SM workload routing functions.

If you have defined more than one CICSplex, and have made the division to reflect use of CICS systems by different groups of users, for example, it is possible that you will be able to use simple workload routing rather than workload separation in the CICSplex. For example, if you have separated the CICS systems used by group A from the CICS systems used by group B by defining two CICSplexes, you can implement workload routing in each CICSplex. If you have not taken this approach, you might consider it necessary to recognize such groupings by implementing workload separation.

Identifying the workloads

To plan for WLM, you shall begin by identifying the workloads processed in your enterprise. The workloads are certain to be apparent in any underlying TOR–AOR–FOR configurations. Confirm that the current CICSPlex SM configuration of CICS systems supports the identified workloads.

In particular:

- Routing regions and target regions from a single workload must be in the same CICSplex. That is, the supplied dynamic routing program cannot route transactions beyond the confines of the CICSplex. (It is possible to route transactions outside of the CICSplex by customizing the supplied dynamic routing program. Customization of the supplied dynamic routing program is described in Modifying dynamic routing.)
- A routing region must be:
  - A CICS TS region.
  - A local MAS, that is, the routing region cannot be running on an MVS image on which there is no CMAS.
  - In only one workload, that is, the routing region can be associated with only one active workload specification at a time.
- A target region can be:

  - a local MAS
  - in multiple workloads
  - any CICS system managed by CICSPlex SM

Identifying inter-transaction affinities

Inter-transaction affinities, which require related transactions to be processed by the same target region, prevent optimum workload distribution. In general, they arise either because of the way in which one transaction passes data to another, or because of a requirement to coordinate the processing of two or more transactions. Identifying affinities isn't always easy, but there are some methods you can use.

For example, you can review application design documentation or source code; you can run CICS traces; or you can use the IBM CICS Interdependency Analyzer for z/OS.

Having identified any affinities in a workload, you should make every attempt to remove them. If you cannot remove them completely, gauge the duration of the affinity and try to minimize it. You can define an inter-transaction affinity to CICSPlex SM as lasting:

- While the user's session is active
- For the duration of the terminal session
- While the target region remains active
- While the workload is active
- For the duration of a pseudoconversation
- While the BTS activity is active
- While the BTS process is active

Be aware that CICSPlex SM must honor an active affinity: if an affinity is active but the target region becomes unavailable, the transaction isn't routed. In the case of a BTS transaction, BTS will wait for the region to start.

Implementing WLM

What you have to do to use CICSPlex SM workload management functions varies depending on whether you want to implement workload routing only, or whether you also want to implement some workload separation, and possibly define one or more transaction affinities in the same CICSplex.

You will get the best results if you follow this process:

1. Identify candidates for dynamic routing
2. Implement workload routing for each workload in the CICSplex where dynamic routing is possible.
3. Add any necessary workload separation requirements.
4. Define any inter-transaction affinities.

Identifying dynamic routing candidates

Not all work requests are candidates for dynamic routing.

The conditions that have to apply for a work request to be eligible are listed in Dynamic Routing.

For further details:

- See EXEC CICS LINK and EXEC CICS START commands.
- See EXEC CICS RUN ASYNCHRONOUS command.
- See CICS resources and Administering BAS for information about CICS and CICSPlex SM resources.
- See Deploying applications to a JVM server for information about Java™.
- See Overview of CICS external interfaces for more information about the Link3270 bridge.

Implementing workload routing

To implement workload routing for a group of target regions, you must associate a routing region or a routing system group with a workload specification. Optionally, you can apply alternative routing to a specified group of transaction in the workload.

About this task

To use workload routing, you specify routing characteristics in the workload specification (WLMSPEC object). These characteristics apply to all of the transactions in the workload.

However, you can override these routing characteristics by supplying associated pairs of WLM definitions and transaction group definitions (WLMDEF and TRANGRP objects) that specify a specialized routing evaluation to be applied to one or more transactions. For workload routing, you specify a different routing algorithm in a transaction group definition. If you specify an alternative routing algorithm at the transaction group level, you can change workload routing characteristics dynamically without stopping routing regions.

Procedure

1. If there is more than one routing region in a single workload, create a CICS system group of those routing regions, using the CICS system group (CSYSGRP) object.
2. Create a CICS system group of the target regions to which the routing regions can route transactions, again using CSYSGRP.

   This group could include every CICS system in the CICSplex.
3. Create one workload specification for each workload in the CICSplex, using the workload management specification (WLMSPEC) object.

   In the workload specification:

   a) Specify the name of the target system group as the target scope value.

   b) Identify which routing algorithm to use for workload routing.

      You can use one of the following algorithms:

      - QUEUE
      - LNQUEUE
      - GOAL
      - LNGOAL
4. Associate the routing region, or the routing system group, with the workload specification.

   You can have multiple workloads in a single CICSplex, but a routing region or group of routing regions can belong to only one active workload specification at a time.
5. Optional: Override the routing algorithm for specific transactions in the workload by specifying a different routing algorithm in a transaction group definition.

   See Creating a transaction group.

   If you modify an installed transaction group, you must discard its associated WLM definition (WLMDEF) and then reinstall it, so that the transaction group named by the WLM definition is also refreshed. To change the routing algorithm type immediately without discarding and reinstalling the associated WLMDEF, you can use the Active workload transaction groups (WLMATGRP) views and the **SET** command to change the ALGTYPE attribute.

Adding workload separation requirements

To implement workload separation in the same workload, by any combination either of user id, terminal id, and transaction group, or of process type and transaction group, do the following:

1. Create a CICS system group *for each set of target regions*, using the **System group definitions** view (CSYSGRP object). For example, if:

   - Transactions initiated by users ABC* from terminals NETA* must be routed to target regions 2 through 7

   and

   - Transactions initiated by users HIJ* from any terminal must be routed to target regions 5 through 9

   two target system groups must be defined: one group includes target regions 2 through 7, and one group includes target regions 5 through 9.
2. Define a transaction group, using the **Transaction groups** view (TRANGRP object), if workload separation by transaction is to be implemented. For example, if transactions EFG1, EFG2, and EFG3 must be routed to a specific group of target regions, you must define a transaction group to include those transactions.
3. Create one workload definition, using the **Workload definitions** view (WLMDEF object), for each user, terminal, and transaction group combination. For the example in item "1" on page 68 (above):

   - One workload definition would be required for the combination of any transaction, users ABC*, and terminals NETA*.
   - One workload definition would be required for the combination of any transaction, users HIJ*, and any terminal.

   A third workload definition would be required to support the workload separation by transaction described in item "2" on page 68. In each workload definition, include the name of the appropriate target region system group to which eligible transactions are to be routed.
4. If the workload definitions are to be installed automatically when the first requesting region in the workload starts, you should use the **Workload groups** view (WLMGROUP object) to:

   a. Create a workload group.

   b. Add the workload definitions to the workload group.

   c. Associate the workload group with the workload specification.

   Automatic installation of the workload definitions is likely to be required, because workload separation is usually a regular and consistent requirement. However, you can also install a workload group or individual workload definitions manually for immediate effect.

Defining inter-transaction affinities

If there is a requirement for multiple transactions in the workload to be processed by the *same* target region, then do the following:

1. Create a transaction group, using the **Transaction groups** view (TRANGRP object), for each set of transactions within the workload that share an affinity. For example, if transactions MNO1, MNO2, and MNO3 must be processed by the same target region (perhaps because they run in sequence and leave data for the next transaction), define a transaction group to include those three transactions specifying the affinity type and its duration.
2. If you want to implement workload separation, create a workload definition for each transaction group and user or terminal pattern, using the **Workload definitions** view (WLMDEF object). For example, if the inter-transaction affinity described in item "1" on page 68 (above) applies to any user from terminals NYORK*, the workload definition must include that information. In addition, the workload definition identifies the set of target regions from which CICSPlex SM can select one. This could be every target region in the CICSplex, or some subset of target regions.

   Note that you can define an inter-transaction affinity to CICSPlex SM without also implementing workload separation. In this case, no workload definition is required and CICSPlex SM routes the first transaction in the group to a suitable target region from the set identified in the workload specification. Subsequent transactions in the transaction group are routed to the same target region while the affinity is active.

3. If the workload definitions are to be installed automatically when the first routing region in the workload starts, you should use the **Workload groups** view (WLMGROUP object) to:

   a. Create a workload group.

   b. Add the workload definitions to the workload group.

   c. Associate the workload group with the workload specification.

   You can also install a workload group or individual workload definitions manually for immediate effect.

Note: It's important that you define routing regions and target regions in groups, to ensure that changes to the composition of those groups are automatically reflected in CICSPlex SM workload management.

A recommended approach

You can click the **Map** button to display a visual map of the WLM definitions in your data repository. When you decide to implement WLM functions in your enterprise, you should begin by defining the appropriate objects.

You should:

- Use the **WLM definitions** view to create a workload definition (WLMDEF object).
- Use the **WLM groups** view to create a workload group (WLMGROUP object).
- Use the **WLM specifications** view to create a workload specification (WLMSPEC object).

Then, to create associations between these objects:

- Use the **WLM definitions** view to add the workload definition to the workload group (WLMINGRP object).
- Use the **Workload groups** view to add the workload group to the workload specification (WLMINSPC object).
- Use the **Workload specifications** view to associate the workload specification with a CICS system (LNKSWSCS object) or CICS system group (LNKSWSCG object). For more information about the LNKSxSCG parameters, see LNKSxSCG Records (LNKSMSCG, LNKSRSCG, LNKSWSCG).

If you want to see a list of the objects and associations you have already defined, use one of the following views:

- The **Workload definitions in groups** view (WLMINGRP object)
- The **Workload groups in workload specifications** view (WLMINSPC object)
- The **WLM specifications to CICS system links** view (LNKSWSCS object)
- The **WLM specifications to CICS system group links** view (LNKSWSCG object).

Workload management resources

You can set up and manage your workload environment using the workload management WUI view sets and resource objects.

For a diagram of the WLM components, see Components of CICSPlex SM WLM.

Table 5. WUI view sets and resource objects for managing workloads

WUI view set	Object name	Description
Transactions in transaction groups	DTRINGRP	This object displays information about the transactions associated with one or more transaction groups.
WLM specifications to system group links	LNKSWSCG	CICS system groups associated with workload specifications

Table 5. WUI view sets and resource objects for managing workloads (continued)

WUI view set	**Object name**	**Description**
WLM specifications to CICS system links	LNKSWSCS	CICS systems associated with workload specifications
Transaction groups	TRANGRP	A transaction group is an association of logically similar transactions. The similarity may be based on affinity requirements, common shared processing requirements, or any other user-determined characteristics. The transaction group is included as part of the workload definition, which in turn defines the CICS system group to be used as the target region. Any transaction that is not specifically associated with a transaction group is assigned to the default transaction group.
Active workload transaction group affinities	WLMATAFF	This object shows information about the active affinities for a transaction group associated with a workload within the CICSplex identified as the context. An affinity becomes active when the first transaction associated with the transaction group is dynamically routed to a target region.
Active workload transaction groups	WLMATGRP	This object shows information about transaction groups associated with a workload that is within the CICSplex identified as the context.
Active workload dynamic transactions	WLMATRAN	This object shows information about all active transactions associated with a workload that is within the CICSplex identified as the context.
Active workload target distribution factors	WLMAWAOR	This object shows information about all target regions that are associated with a workload that is within the CICSplex identified as the context.
Active workload definitions	WLMAWDEF	This object shows information about active workload definitions associated with a workload that is within the CICSplex identified as the context.
Active workloads	WLMAWORK	This object shows information about active workloads within the CICSplex identified as the context. A workload is active within a CICSplex as long as: • A CICS system that is acting as a routing region or target region, and is participating in the workload, is connected to that CICSplex. • Any transaction causes an affinity lifetime of PERMANENT to be established.
Routing regions in an active workload	WLMAWTOR	This object shows information about routing regions that are associated with a workload that is within the CICSplex identified as the context.
Workload definitions	WLMDEF	This object identifies which transactions are to be routed to which CICS system group. You identify the transactions by transaction group, and within the group either by LU name and user id, or by process type.

Table 5. WUI view sets and resource objects for managing workloads (continued)

WUI view set	**Object name**	**Description**
Workload groups	WLMGROUP	This object is used to associate one or more workload definitions.
Workload definitions in groups	WLMINGRP	This object displays the names of workload groups and the workload definitions associated with them.
Workload groups in workload specifications	WLMINSPC	This object displays the names of workload specifications and the workload groups associated with them.
Workload specifications	WLMSPEC	A workload specification identifies a workload and one or more CICS system groups acting as target regions. It also defines the attributes of the default transaction group.

Workload management definitions and their related views

You use the Web User Interface (WUI) workload management administration views to define a variety of workload management attributes.

Figure 8 on page 72 provides an overview of the workload management views from the perspective of the CICSPlex SM object model. In addition to these views, you can use the views to display information about and manage active workloads. You can also display a visual map of your workload management definitions by using the MAP button.

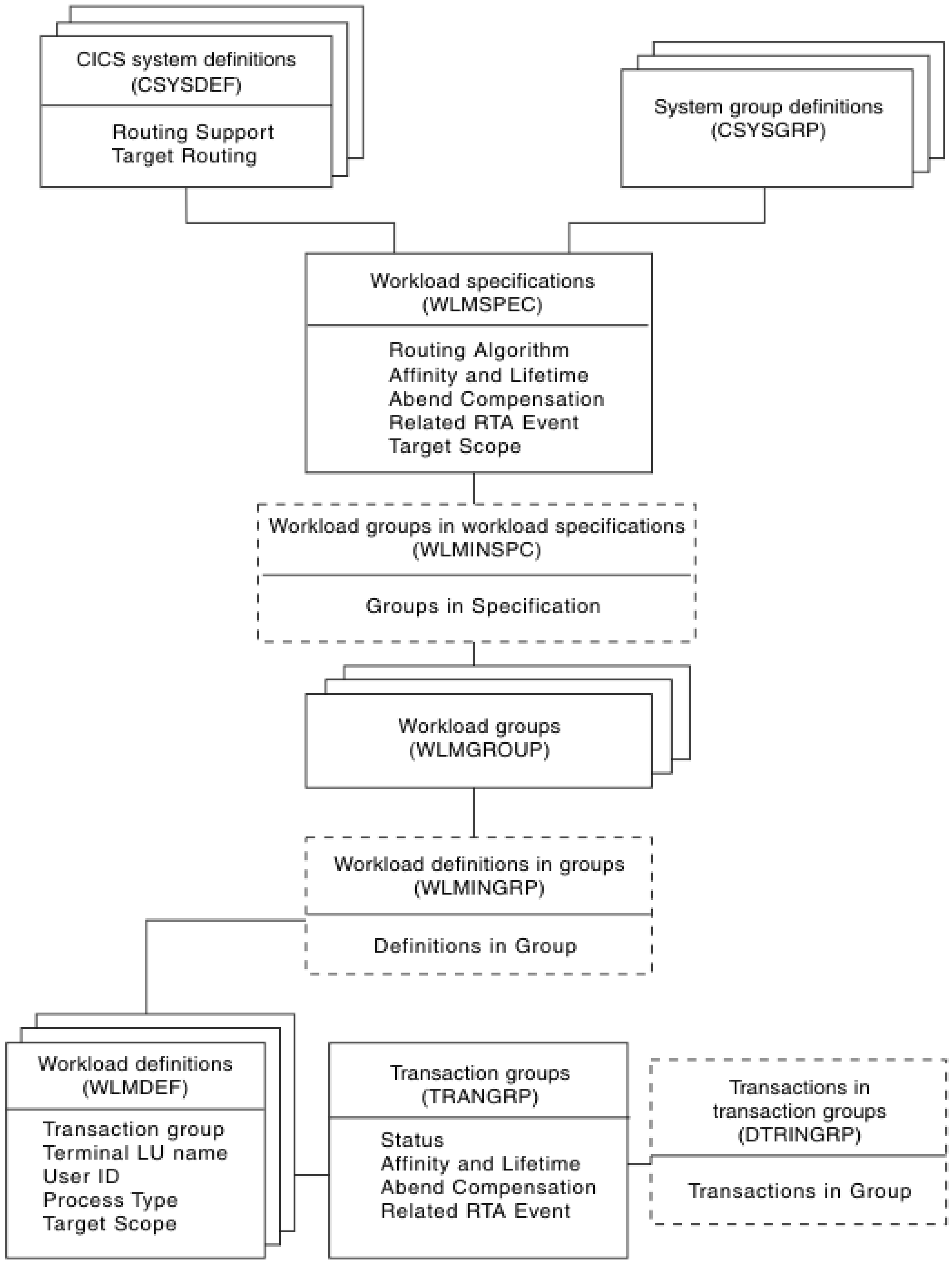
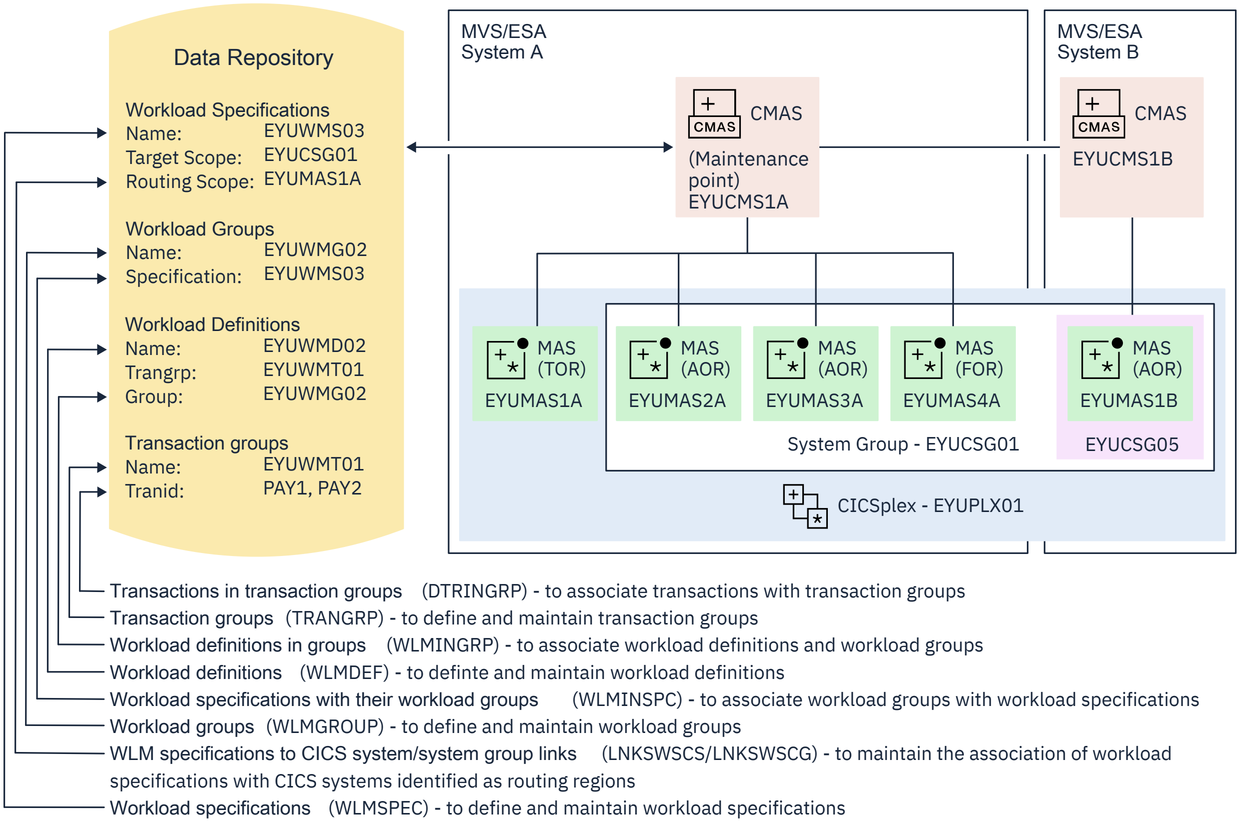


Figure 8. Views for creating workload management objects and associations

Figure 9 on page 73 illustrates the relationship between the components of a workload in a CICSplex and the views used to establish the workload.



Transactions in transaction groups (DTRINGRP) - to associate transactions with transaction groups

Transaction groups (TRANGRP) - to define and maintain transaction groups

Workload definitions in groups (WLMINGRP) - to associate workload definitions and workload groups

Workload definitions (WLMDEF) - to definte and maintain workload definitions

Workload specifications with their workload groups (WLMINSPC) - to associate workload groups with workload specifications

Workload groups (WLMGROUP) - to define and maintain workload groups

WLM specifications to CICS system/system group links (LNKSWSCS/LNKSWSCG) - to maintain the association of workload specifications with CICS systems identified as routing regions

Workload specifications (WLMSPEC) - to define and maintain workload specifications

Figure 9. The relationship between a workload and the workload management views

Table 6 on page 74 shows the views you can use to create workload management definitions. It also indicates the information you can display and the actions you can perform using these views. To access these views in the Web User Interface, from the main menu, click **Administration** > **Workload manager administration**. For details of these views, see CICPlex SM administration views.

Table 6. Views to create and maintain workload management definitions

WUI view	Object name	Tasks Supported
Transactions in transaction groups	DTRINGRP	• Display all transaction groups within the current context. • Remove a transaction from a transaction group. • Display a map of workload definitions using the designated transaction group as a starting point.
CICS system groups associated with workload specifications	LNKSWSCG	• Display, create or remove the association between a workload specification and a CICS system group.
CICS systems associated with workload specifications	LNKSWSCS	• Display, create, update or remove a link between a workload specification and a CICS system. • Display a map of workload definitions using the designated definition as a starting point.
Transaction groups	TRANGRP	• Display, create, display, update, remove, or update a transaction group. • Add a transaction to a transaction group. • Install • Display a map of workload definitions using the designated transaction group as a starting point.
Workload definitions	WLMDEF	• Create, change, or remove a workload definition. • Add an association between a workload definition and a workload group. • Install a workload definition into a workload.

Table 6. Views to create and maintain workload management definitions (continued)

WUI view	**Object name**	**Tasks Supported**
Workload groups	WLMGROUP	• Display, create, update, or remove a workload group. • Add an association between a workload group and a workload specification. • Install workload definitions associated with a workload group into a workload. • Display a map of workload definitions using the designated workload group as the starting point.
Workload definitions in groups	WLMINGRP	• Add or remove the association between a workload definition and a workload group.
Workload specifications	WLMSPEC	• Display, create, update, or remove a workload specification. • Add an association between a workload specification and a CICS system. • Add an association between a workload specification and a CICS system group. • Display a map of workload definitions using the designated workload specification as a starting point.
Workload groups in workload specifications	WLMINSPC	• Remove the association between a workload group and a workload specification. • Display a map of workload definitions using the designated definition as a starting point.

Table 7 on page 75 shows the views you can use to manage active workloads. It also indicates the information you can display and the actions you can perform using these views. To access these views in the Web User Interface, from the main menu, click **Active workloads (WLM)**.

Table 7. Views to manage active workloads

WUI view	**Object name**	**Tasks Supported**
Active workloads	WLMAWORK	• Display active workloads within the current context.
Routing regions in an active workload	WLMAWTOR	• Display active routing regions that are associated with a workload.

Table 7. Views to manage active workloads (continued)

WUI view	Object name	Tasks Supported
Active workload target distribution factors	WLMAWAOR	• Display target regions that are associated with a workload. • Activate a target region associated with an active workload. • Quiesce a target region associated with an active workload.
Active workload definitions	WLMAWDEF	• Display active workload definitions associated with a workload. • Discard an active workload definition.
Active workload transaction groups	WLMATGRP	• Display active transaction groups associated with a workload. • Change the status of a transaction group to active. • Change the status of an active transaction group to dormant.
Active workload dynamic transactions	WLMATRAN	• Display active transactions associated with a workload. • Discard a transaction from a transaction group.
Active workload transaction group affinities	WLMATAFF	• Display the active affinities for a transaction group associated with a workload. • Discard an affinity entity.

Creating workload management definitions

This section guides you through the steps to create and manage workload management definitions by using the WUI or the CICS Explorer. Unless noted otherwise, only the context is recognized when you are creating and maintaining workload management definitions.

Creating a workload specification

A WLM specification identifies a workload and one or more CICS systems acting as target regions. The specification also defines the attributes of the default transaction group. You can use the **Workload specifications** view to create a workload specification.

About this task

You can use either the CICSPlex SM WUI or the CICS Explorer to create a workload specification and add it to the data repository.

For CICS Explorer users:

Follow the procedure in Creating a workload specification in the CICS Explorer product documentation.

For CICSPlex SM WUI users:

Procedure

1. From the WUI main menu, click **Administration** > **Workload manager administration** > **Specifications** to open the **WLM specifications** tabular view.

   This view displays a list of existing workload specifications. It has action buttons that allow you to create, update, and remove workload specifications, and to associate CICS systems and CICS system groups with a workload specification.
2. Optional: To use some of the information from an existing definition in the creation of your new definition, select an existing definition by selecting an adjacent check box in the **Record** column.
3. Click **Create**.
4. Complete the fields and click **Yes** to create the new specification. Otherwise, click **No** to abandon the process.

   The **Workload specification** view is redisplayed and includes an entry for the new definition.

Associating a CICS system or system group with a workload specification

Each workload specification has a default target scope and one or more CICS systems or CICS system groups associated with it.

These associated CICS systems or groups are known as the routing scope. The default target scope is identified when you create the specification. Associated CICS systems and system groups are identified when you add the routing region to the specification.

Follow this procedure to associate a CICS system group to an existing workload specification:

1. Click **Administration > Workload manager administration—>Specifications** to open the **WLM specifications** view.

   This view displays a list of existing workload specifications.
2. Select a workload specification and click the **Associate CICS group...** button. This opens the **Associate CICS group...** view.
3. In the **CICS system group field**, enter the specific or generic name of an existing CICS system group that represents one or more routing regions.

   Note: A CICS system or CICS system group acting as a routing region can be associated with only one workload specification at a time. A specification, however, can be associated with any number of CICS systems and CICS system groups.
4. Select one of the following options to indicate how the CICS systems that make up the CICS system group are to use the workload specification associated with the CICS system group:

   FORCE

   All CICS systems currently associated with the CICS system group are to use the workload specification. (The workload specification attribute for each CICS system changes to INHERIT, indicating that the specification was acquired from the CICS system group). If you add a new CICS system to the CICS system group, it does not automatically inherit the FORCE option. You must specify NULL or FORCE when adding a CICS system to a CICS system group.

   NULL

   Those CICS systems in the CICS system group that are not associated with a workload specification are to use this workload specification. (The workload specification attribute for those CICS systems changes to INHERIT, indicating that the specification was acquired from the CICS system group.)

NONE

Only the CICS system group is to be associated with the workload specification. The CICS systems in the CICS system group are not affected. That is, if there is no association between a CICS system and a workload specification, none is established; if there is an association, either explicitly established or inherited from another CICS system group, it is unchanged.

Note:

a. If the CICS system group includes other CICS system groups, all of the CICS systems, including those in subordinate CICS system groups, are affected by the value specified in this field.
b. The CICS systems designated as the default target scope when you created the workload specification must be accessible to the CICS systems identified as the routing scope. Therefore, you should verify that the appropriate CICS connections exist between each routing region and all of the CICS systems in the target scope identified here.

5. Click **Yes** to update the workload specification in the data repository. Otherwise, click **No** to abandon the process.

Once you associate a CICS system or system group to a workload specification, the specification is automatically installed whenever an associated CICS system is started. Any workload definitions associated with the specification through workload groups are also automatically installed.

However, if you associate the workload specification with a CICS system that is already active, the new specification is not immediately available. To turn workload management on:

1. Click **Administration > Topology administration > System definitions** to display a list of active CICS systems.
2. Select the CICS system and click the **Update...** button
3. In the **Workload manager status** field, specify **YES**, and click the **Yes** button to turn on workload management. Otherwise, click **No** to abandon the process.

Updating a link between a WLM specification and a CICS system

Procedure

1. From the main menu, click **Administration** > **Workload manager administration**.
2. From the **Workload manager administration views** menu, click **Specifications to system links**.

   The **WLM specifications to CICS system links** view (LNKSWSCS object) is displayed.
3. Check the box for the WLM specification to be updated.

   You can check more than one box.
4. Click the **Change Spec Association** button to display the **Change Spec Association (EYUSTARTLNKSWSCS.CHGSPEC)** view.
5. Type the new name in the **New WLM specification name** field and select an appropriate inherit option.

   - Click **Yes** to update the link between a monitor specification and a CICS system group.
   - Click **No** to abandon the process.

Updating a link between a WLM specification and a CICS system group

Procedure

1. From the main menu, click **Administration** > **Workload manager administration**.
2. From the **Workload manager administration views** menu, click **Specifications to system group links**.

   The **WLM specifications to CICS system links** view (LNKSWSCG object) is displayed. For more information about the LNKSxSCG parameters, see LNKSxSCG Records (LNKSMSCG, LNKSRSCG, LNKSWSCG).

3. Check the box for the WLM specification to be updated.

   You can check more than one box.
4. Click the **Change Spec Association** button to display the **Change Spec Association (EYUSTARTLNKSWSCG.CHGSPEC)** view.
5. Type the new name in the **New WLM specification name** field and select an appropriate inherit option.
   - Click **Yes** to update the link between a monitor specification and a CICS system group.
   - Click **No** to abandon the process.

Creating a transaction group

A transaction group is an association of logically similar transactions. The similarity can be based on workload management (WLM) or affinity requirements, common shared processing requirements, or any other user-determined characteristic. You can use the **Transaction group definitions** view to create a transaction group.

About this task

For workload management, any WLM information you specify in the transaction group, including the routing algorithm type, overrides the defaults supplied in the associated workload specification (WLMSPEC). If you specify alternative WLM attributes in a transaction group, you can change workload routing characteristics for the associated transactions dynamically without stopping your routing region.

You can use either the CICSPlex SM WUI or the CICS Explorer to create a transaction group.

For CICS Explorer users:
Refer to the CICS Explorer product documentation for instructions.

For CICSPlex SM WUI users:

Procedure

1. From the WUI main menu, click **Administration** > **Workload manager administration** > **Transaction group definitions** to open the **Transaction group definitions** view.

   This view displays a list of existing transaction group definitions. You can create, update, and remove transaction group definitions, and add a transaction to a transaction group.
2. Optional: To use information from an existing definition when you create a new definition, select the check box in the Record column next to the required existing definition.
3. Click **Create**.
4. Provide the appropriate information to create your transaction group definition.

   See Transaction group definitions - TRANGRP for a description of the fields in this view.
5. Click **Yes** to create the new definition and add it to the data repository.

   The **Transaction group definitions** view is displayed again with an entry for the new definition.

What to do next

If you modify the WLM information for an installed transaction group, you must discard its associated WLM definition (WLMDEF) and then reinstall it, so that the transaction group named by the WLM definition is also refreshed. However, if you want to change the routing algorithm type (the ALGTYPE attribute), you can change it immediately without discarding and reinstalling the associated WLMDEF by using the **Active workload transaction groups** (WLMATGRP) views and the **SET** command.

Creating a workload definition

You can use workload definitions to route work requests to a specific set of target regions based on the terminal and user names, or the process types, associated with those work requests. This section describes how to create a workload definition and add it to the data repository.

The terminal and user names may be either specific or generic. For example, you can create a workload definition that causes all transactions initiated by any user from terminals with logical unit names starting with NET to be routed to the target scope identified as EYUCSG01.

The transaction match criteria that CICSPlex SM uses to determine if a transaction should be routed are as follows:

- Is the transaction identifier part of a transaction group associated with the workload definition?
- Is there a definition specifying separation by process type? Is there a match? If so, use this definition. If not, is there a definition specifying separation by terminal logical unit name and user ID?
- In what order should the terminal and user names associated with the transaction be evaluated; that is, which name is to be used as the primary filter?
- Do the user and terminal names associated with the transaction match the name patterns specified with the workload definition?

The rules for establishing terminal and user name patterns are:

- When there are multiple workload definitions, the field containing the name identified as the primary filter can contain the same specific or generic pattern; the contents of the field used as the secondary filter must always be unique. For example, the following definitions are valid when USERID is the primary filter and LUNAME is the secondary filter. They are not valid, however, when LUNAME is the primary filter because the user identifiers are not unique.

  ```
  Entry Name     :  EYUWMD01  EYUWMD02  EYUWMD03
  Trangroup Name :  EYUTNG01  EYUTNG01  EYUTNG01
  Terminal Luname:  LUR*      LUL*      LUT*
  User ID        :  PIE*      PIE*      PIE*
  ```

- When a generic name is specified, the pattern with the most matching leading characters has precedence. For example, with the following patterns:

  ```
  A37AR*
  A37+R*
  ```

  Pattern A37AR* is selected when the input is A37AR123. Pattern A37+R* is selected when the input is A37TRAP.

- Terminal LU name patterns are matched on the concatenated values of network name and logical unit name. The following are valid terminal LU name patterns:

  ```
  NETWORK1.LU12345678   - A specific luname
  NETWORK1.LU1*         - Generic lunames in the network
  NETWORK1.*            - All lunames in the network
  NET*                  - All lunames in a generic network
  .LU12345678           - A specific luname within all networks
  .* or *               - All lunames within all networks
  ```

To create a workload definition and add it to the data repository:

1. Click **Administration** > **Workload manager administration** > **Definitions** to open the **Workload management definition** view.

   This view displays a list of existing workload management definitions. It has action buttons that allow you to create, update, remove and install workload definitions, and to add a workload definition to a workload group.

2. If you want to use some of the information from an existing definition in the creation of your new definition, select an existing definition by selection by selecting an adjacent check box in the **Record** column.
3. Click the **Create** action button.
4. Provide the following information, as appropriate:

   Workload management definition name

   Specify a 1- to 8-character name for the workload definition. The name can contain alphabetic, numeric, or national characters. However, the first character must be alphabetic.

Description

(Optional) Specify a 1- to 30-character description of the definition.

Trangroup group name

(Optional) Enter the specific or generic name of a transaction group. If you enter a generic value, a list of valid transaction groups is displayed.

If you do not identify a transaction group, the default transaction group for the specification is assumed.

Terminal LU name

Enter a specific logical unit name or a pattern, using the format:

```
    [network_qualifier_name][.luname]
```

where both the `network_qualifier_name` and the `luname` are between 1 and 8 characters in length. Both the `network_qualifier_name` and the `luname` are optional, however one must be specified. If the `luname` is specified it must be prefixed by a period sign.

A pattern can include the characters + (plus sign), * (asterisk), or both; where the plus sign represents any non-blank character and can be specified anywhere in the pattern and the asterisk represents a range of zero or more characters that may include blanks. The asterisk must only be used at the end of the pattern.

User ID

Enter a specific user ID or a pattern. A pattern can include the characters + (plus sign), * (asterisk), or both.

Note: The Terminal Luname and User ID values are used when CICSPlex SM attempts to match a transaction with this definition.

BTS process type

Enter a specific process type or a pattern. A pattern may include + (plus sign), * (asterisk), or both.

Note: If you specify anything other than an * in this field, the luname and user id fields must be set to an *. Similarly, if there is anything other than an * in either luname and user id, this field must be set to *. You cannot separate a workload by process type, luname and user id.

Scope name of set of target systems

Enter the specific or generic name of a CICS system or CICS system group to be used as target regions for dynamic routing requests. If you enter a generic value, a list of valid CICS systems and CICS system groups is displayed.

Note: The specified CICS system or CICS system group must be accessible to the CICS systems identified as the routing scope with the workload specification to which this definition is associated. Therefore, you should verify that the appropriate CICS connections exist between each routing region and all of the CICS systems identified here as the target scope.

5. Click **Yes** to create the new definition.

   The **Workload management definition** view is redisplayed and includes an entry for the new definition.

Creating a workload group

A workload group is used to associate one or more related workload definitions.

An example of how to use this view can be found in “Routing a specific transaction to a specific target region” on page 88. This section describes how to create a workload group definition and add it to the data repository.

- Click **Administration views—>Workload manager administration views—>Groups** to open the **WLM groups** view.

This view displays a list of your existing workload management group definitions. It has action buttons that allow you to create, update, remove and install workload groups, and to add a group to a workload specification.

- If you want to use some of the information from an existing definition in the creation of your new definition, select an existing definition by selecting an adjacent check box in the **Record** column.
- Click the **Create** action button.
- Provide the following information, as appropriate:

  Workload management group name
  Specify a 1- to 8-character name for the workload group. The name can contain alphabetic, numeric, or national characters. However, the first character must be alphabetic.

  Description
  (Optional.) Specify a 1- to 30-character description of the workload group.
- Click **Yes** to create the new definition.

  The **Workload management group** view is redisplayed and includes an entry for the new definition.

About workload view route fields

Use the "Active workload target distribution factors" and "Summarized active target region distribution trends" views together with route fields statistics displayed in the active workload views to understand what is happening in workload management.

You can use the route fields statistics information in the "Summarized active target region distribution trends" view to verify that your existing workload is running correctly, and that it is not losing transactions. The "Summarized active target region distribution trends" view also provides a new perspective on events inside WLM during the routing process.

Using the "Active workload views", you can highlight potential issues with your setup. For example, if routing is called and continually retrying to discover other available targets.

You can see counts of what is happening inside WLM; for example, the number of `Route completes` versus the number of `Route selects`. An example of this is the `Route notifies` field which shows you when you are linking to a program that is defined as **`DYNAMIC(YES)`** and you have provided a SYSID.

Using the "Summarized active target region distribution trends" view with the "Active workload target distribution factors" view can help you understand WLM routing decisions and behavior of CICSPlex SM. You might want to understand why some regions are receiving more work than others, and an analysis of `Route Selections` shows which regions are receiving more work than others.

Figure 10 on page 83 illustrates the route field processes involved in distributed routing.

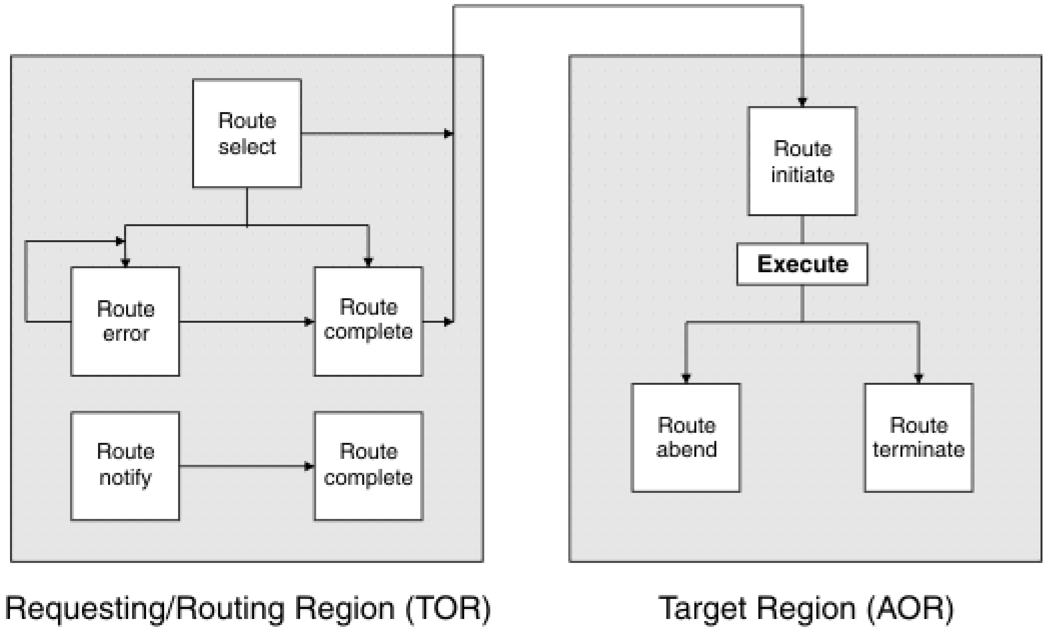


Figure 10. Conceptual view of distributed routing

Figure 11 on page 83 illustrates the route field processes involved in dynamic routing.

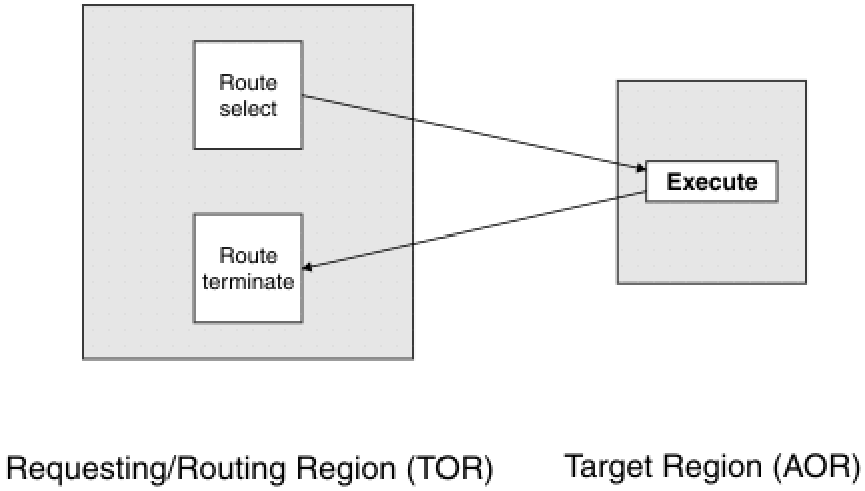


Figure 11. Conceptual view of dynamic routing

These definitions describe the route fields:

Route select
Route select is updated when this region is selected as a target by CICSPlex SM workload management.

Route initiate
Route initiate is updated when a task routed using the distributed workload model starts on a target system.

Route error
Route error is updated when the region selected by CICSPlex SM is unreachable; for example, the connection is out of service, or no sessions are available. CICSPlex SM then attempts to select another region. This reselection is not counted as a `Route select`, and the `Route select` count for any subsequent region is not updated. A high number of `route errors` can indicate that there are

insufficient sessions to run the workload. The lack of sessions can cause increased response time and CPU time because CICSPlex SM must try the routing again, potentially several times.

Route complete

Route complete is updated when a task that is routed using the distributed workload model starts on the target system. Route complete indicates to the routing system that its responsibility for this task is complete.

Route notify

Route notify is updated when a task is routed to a target region that is selected by the SYSID option on the LINK or START command. Route notify informs CICSPlex SM of the routing; CICSPlex SM does not choose the region. Route notify is a variant of static routing, in which the program or transaction is defined as Dynamic, but the API that calls it specifies a SYSID.

Route abend

Route abend is updated when a task is routed to a target region and subsequently abends; for example, because the transaction is not defined in that system or the program is unavailable.

Route terminate

Route terminate is updated when a task has completed in a target region, whether at the end of a distributed program link (DPL) or a dynamic transaction routing (DTR) request, or when a task that is routed using the distributed workload model finishes.

Example CICSPlex SM workload management tasks

All the workload management examples are based on the same configuration. The CICSplex, PLXPROD1 consists of one TOR, CICSPT01, and three AORs, CICSPA01, CICSPA02, and CICSPA03. You will see how the roles played by these CICS regions vary, depending on the nature of the individual task.

These sections include a number of typical workload management example tasks and illustrates how these tasks can be carried out using the Web User Interface (WUI).

For any task, you must be aware of the scope, that is, of the CICS systems, with which you are working. If the scope is a single CICS system, any data you retrieve from CICSPlex SM relates to that single system. If the scope is a group of CICS systems, the data relates to all of the systems in the group. If the scope is a CICSplex, the data relates to every system in that CICSplex. For all of the examples in these sections, the initial scope is CICSplex PLXPROD1.

Before you begin

To support optimized workload management, a CICS region status server must be set up and running in the CICSplex to share CICS region status data rapidly. For detailed instructions, see Setting up and running a region status server.

Managing a workload

This example describes how to use the Web User Interface (WUI) to get CICSPlex SM to manage a workload: that is, to route all work from a single routing region to a suitable target region.

For this task, a "suitable" target region is the one with the shortest queue of work requests, relative to the maximum number of tasks permitted in the target region. This queue of work, also called the *load count*, is the queue of all active and queued user tasks. By default, tasks queued for both MAXTASKS and TRANCLASS are included in the load count. Use of the "Task load queue mode" attribute in the CSYSDEF resource table allows sites to exclude tasks queued for TRANCLASS from the load count. For more information about the "Task load queue mode" attribute in the CSYSDEF resource table, see CICS system definitions - CSYSDEF.

The TOR (CICSPT01) is the requesting region and the routing region, and the three AORs (CICSPA01, CICSPA02, and CICSPA03) are all target regions. None of the CICS regions are currently running.

If some of your work requests can't be routed freely (perhaps you want work requests from a particular user always to go to the same target region, for example), don't worry. You'll see how to add that requirement in a later example task ("Routing a specific transaction to a specific target region" on page 88).

1. Create a CICS system group that will include all of the target regions among which the work requests ("the workload") can be routed dynamically.

   - From the main menu, click **Administration > Topology administration > System groups** to open the **System group definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click **Create** to open the **System group definitions** create view.
   - Type the following:

System group name	`CSGTGTS1`
Description	`All target regions in CICSplex PLXPROD1`

   - Click **Yes** to create the system group

     The **System group definitions** tabular view is redisplayed, this time showing an entry for CICS system group CSGTGTS1.

   At this point, the group CSGTGTS1 exists, but is empty.
2. Specify to CICSPlex SM which target regions belong to the group. To add target regions to CICS system group CSGTGTS1:

   - From the main menu click **Administration > Topology administration > System definitions** to open the **CICS system definitions** tabular view
   - Select the entries for the CICS systems CICSPA01, CICSPA02 and CICSPA03 and click **Add to CICS system group** .
   - In the **Group which member will join** field, type CSGTGTS1 and click **Yes to 3 remaining**. This adds the three selected systems to your new group.
3. Create a workload specification:

   - From the main menu click **Administration > Workload manager administration > Specifications** to open the **WLM specifications** tabular view.
   - Click **Create**, and type the following information:

WLM specification name	`WLSPAY01`
Description	`Manage payroll workload in PLXPROD1`
Primary search criterion	`USERID`
Default target scope	`CSGTGTS1`
Algorithm type	`QUEUE`

     Leave the remaining fields empty or accept the defaults.

     Note:

     – A **Primary search criterion** value of USERID is shown in this example, though in fact it does not matter whether you specify USERID or LUNAME, because the **Primary search criterion** value has no effect on simple workload searching. It is used only for some kinds of workload separation, which is the subject of a later example. You must supply a value because CICSPlex SM doesn't know, at this stage, that you aren't going to use this workload specification for workload separation.

     – The **Default target scope** value is the name of the single target region, or group of target regions, to which work requests can be routed.

     – The **Algorithm type** value, QUEUE, tells CICSPlex SM to select the target region that has the shortest queue of work requests.

- The **Description** is optional, but is worth providing. (When you are confronted with a list of specification names, and have no way of telling one from another, you will appreciate its value.)

Click **Yes**. The **WLM specification** tabular view is redisplayed, this time showing an entry for the new WLM Specification, WLSPAY01.

4. Tell CICSPlex SM about the region that is going to be routing the work requests to the target regions in group CSGTGTS1. To associate the workload specification with a routing region:

   - In the **WLM specifications** view, select the entry for the WLSPAY01 specification, and click **Associate CICS system**.
   - In the **CICS system** field, enter the name of the routing region (CICSPT01, in this example) and click **Yes**.

   To check that the association between the routing region and the workload specification has worked:

   - On the **WLM specifications** tabular view, click the name **WLSPAY01** to show the details of this specification.
   - Click the link **CICS systems associated with this workload specification** to show the **WLM specifications to CICS system links** tabular view (LNKSWSCS object).

   Note: You can also access this view from the main menu by clicking **Administration > Workload manager administration > Specifications to system links**.

5. Activate workload management in the routing region:

   - From the main menu click **Administration > Topology administration > System definitions** to open the **CICS system definitions** tabular view.
   - Select the entry for CICSPT01 and click **Update**, to open a detail view of CICSPT01.
   - Change the **AOR dynamic routing mode** field to **YES**. Click **Yes** to return to the **CICS system definition** tabular view.

     This change takes effect when the target region CICSPT01 is next started.

6. Activate workload management in the target regions:

   - From the **CICS system definitions** tabular view, select the entry for CICSPA01 and click **Update** to open a detail view of CICSPA01.
   - Change the **Workload manager status** field to **YES**. Click **Yes** to return to the **CICS system definitions** tabular view.

     This change takes effect when the target region CICSPT01 is next started.

   Repeat this step for target regions CICSPA02 and CICSPA03.

When the routing region and the target regions have been started, you can check that the workload is active from the main menu by clicking **Active workloads (WLM) > Active workloads**. This opens the **Active workloads** tabular view, showing the workload specification WLSPAY01 as active.

To see which target regions are being routed to, click **Active workloads (WLM) > Active workload target distribution factors**. The displayed view shows all active target regions being routed to by workload WLSPAY01. All of the target regions listed belong to the CICS system group CSGTGTS1.

Adding a region to an existing target region scope

This example describes how to use the Web User Interface (WUI) to increase the number of regions in an existing target scope without disrupting an existing workload.

Assume that you have implemented workload routing in CICSplex PLXPROD1, via workload specification WLSPAY01, and that work requests are being routed among target regions CICSPA01, CICSPA02, and CICSPA03. These target regions all belong to CICS system group CSGTGTS1. Now you want to add a fourth region CICSPA04 to group CSGTGTS1. Region CICSPA04 has been defined to CICSplex PLXPROD1 and is running and is linked to the routing region CICSPT01

1. Add target region CICSPA04 to CICS system group CSGTGTS1:

a. From the Web User Interface main menu click **Administration > Topology administration > System definitions** to open the **CICS system definitions** tabular view.

   If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.

b. Select the entry for CICSPA04, and click the **Add to CICS system group** button. The **Add to CICS system group** view is displayed.

c. In the **Group which member will join** field, enter the name CSGTGTS1 and click **Yes**.

2. Update the CICSPlex SM definition of target region CICSPA04:

   a. From the **CICS system definitions** tabular view, select the entry for CICSPA04 and click **Update**.

   b. Change the **Workload manager status** field to **YES** from the menu. Click **Yes** to confirm the update.

   This change takes effect when the target region CICSPA04 is next started.

When the target region CICSPA04 has been started, you can check that the workload is active by clicking **Active workloads (WLM) > Active workload target distribution factors**. This opens the **Active workload target distribution factors** view showing all target regions (including CICSPA04) to which work requests in this workload can be routed.

Removing a region from a target region scope

CICS system group CSGTGTS1 contains four target regions (CICSPA01, CICSPA02, CICSPA03, and CICSPA04) in CICSplex PLXPROD1. Work requests are routed among these target regions, and the routing is controlled by workload specification WLSPAY01.

This example describes how to use the Web User Interface (WUI) to remove region CICSPA04 from the group CSGTGTS1, without disrupting the active workload.

1. Remove target region CICSPA04 from CICS system group CSGTGTS1:

   - From the Web User Interface main menu click **Administration > Topology administration > System groups** to open the **System group definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.

   - Click CSGTGTS1 in the **System group name** field to open a detail view and click the **CICS systems in this CICS system group** link. This opens the **CICS system to CICS system group links** tabular view.
   - Click the Record check box beside the entry for CICSPA04 and click the **Remove...** button. This opens the **Remove** confirmation view.
   - Click **Yes** to confirm the update.

2. To check that CICSPA04 has been removed, from the main menu click **Active workloads (WLM) > Active workload target distribution factors**. This opens the **Active workload target distribution factors** view showing all target regions to which the workload can be routed. CICSPA04 should not appear in the list.

Adding a routing region to an active workload

Use the Web User Interface (WUI) to add a second routing region to an active workload, without disrupting that workload.

Assume that you are still routing the workload in CICSplex PLXPROD1, via workload specification WLSPAY01, and that work requests are being routed by CICSPT01 among the target regions in CICS system group CSGTGTS1. Now you want to add a second routing region - CICSPT02 - to the workload. Region CICSPT02 has been defined to CICSplex PLXPROD1 (using the **CICS system definitions** views), is running, and is linked to the target regions CICSPA01, CICSPA02, and CICSPA03.

1. Update the CICSPlex SM definition for CICS system CICSPT02:

   - From the Web User Interface main menu click **Administration > Topology administration > System definitions** to open the **CICS system definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.

- Select the record for CICSPT02 and click the **Update...** button.
- Change the **AOR dynamic routing mode** field to YES.
- Click **Yes** to confirm the update. This change takes effect when routing region CICSPT02 is next started.

2. Associate CICSPT02 with workload specification WLSPAY01:

   - From the main menu, click **Administration > Workload manager administration > Specifications**.
   - On the **WLM specifications** view, click the Record check box beside the entry for WLSPAY01 and click **Associate CICS system...**.
   - In the **CICS system** field, type in CICSPT02 and click **Yes** to confirm the update.

When CICSPT02 has been restarted, you can verify that it has been added to WLSPAY01 as follows:

- Open the **WLM specification** tabular view and click the entry for WLSPAY01. This opens a detail view.
- Click the **CICS systems associated with this workload specification** link. Both CICSPT01 and CICSPT02 should be listed in the **WLM specifications to CICS system links** tabular view.

To check that CICSPT02 is part of the active workload, from the main menu, click **Active workloads (WLM) > Routing regions in an active workload** This opens a tabular view showing entries for both CICSPT01 and CICSPT02.

Quiescing a target region in an active workload

This example describes how to use the Web User Interface (WUI) to quiesce the target region CICSPA03, which belongs to the active workload WLSPAY01.

You might need to do this so that you can apply maintenance to a region, for example. The regions CICSPT01 and CICSPT02 are routing work requests among three target regions (CICSPA01, CICSPA02, and CICSPA03) when you perform this task.

If you want to prevent work routing to a region, regardless of whether work is already running there, you can set the region's z/OS WLM health open status to CLOSE or IMMCLOSE. For more information, see "Effect of the z/OS WLM health service on CICSPlex SM workload routing" on page 63.

1. List the target regions associated with workload WLSPAY01.

   - From the Web User Interface main menu click **Active workloads (WLM) > Active workload target distribution factors** to open the **Active workload target distribution factors** view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - In the **Workload name** field type WLSPAY01 and click **Refresh**. This redisplays the **Active workload target distribution factors** view. The three CICS regions CICSPA01, CICSPA02 and CICSPA03 are listed. Their status is active.

2. Quiesce target region CICSPA03:

   - On the **Active workload target distribution factors** view, select the entry for target region CICSPA03 and click the **Quiesce...** button. This opens the **Quiesce** confirmation view.
   - Click **Yes** to confirm the action. The **Active workload target distribution factors** view is redisplayed, showing the **Target region status** value for CICSPA03 changed to QUIESCED. No new work requests are routed to the target region, though any work already running there is allowed to complete.

Routing a specific transaction to a specific target region

Use the Web User Interface (WUI) to define some workload separation requirements to CICSPlex SM.

This example describes how to use the Web User Interface (WUI) to always route the transaction PAY1 to the target region CICSPA02, which belongs to CICS system group CSGTGTS1. You are still working in CICSplex PLXPROD1, and workload routing, from a single CICSPT01 among target regions in the group CSGTGTS1, is in effect.

1. If the transaction to be routed is to be started with EXEC CICS START, it should be defined as ROUTABLE.
2. Create a transaction group:
   - From the main menu, click **Administration > Workload manager administration > Transaction group definitions**. This opens the **Transaction group definitions** view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the **Create...** button, and provide the following information:

Name	TRGPAY01
Description	Send PAY1 to CICSPA02
Affinity relation and lifetime checking status	ACTIVE
Primary search criterion	USERID

     You can leave the remaining fields blank or accept the defaults.

   Click **Yes**. The **Transaction group definitions** view is redisplayed, now showing the name of the transaction group TRGPAY01.
3. Add transaction PAY1 to transaction group TRGPAY01:
   - In the **Transaction group definitions** view, select the entry for TRGPAY01, and click the **Add transaction...** button.
   - In the **Transaction name** field, type the name PAY1 and click **Yes** to confirm. The **Transaction group definitions** view is redisplayed.
4. Create a workload definition:
   - From the main menu, click **Administration > Workload manager administration > Definitions**. This opens the **WLM definitions** view, listing any workload definitions already created in PLXPROD1.
   - Click the **Create...** button, and provide the following information:

Workload management definition	WLDPAY01
Description	Separate TRGPAY01 to CICSPA02
Transaction group name	TRGPAY01
Scope name of set of target systems	CICSPA02

   - Click **Yes** to confirm.
5. Create a workload group. (A workload group is essential if you want a workload definition to be installed automatically when the routing region that's routing the transactions is started.)
   - From the main menu, click **Administration > Workload manager administration > Groups**. This open the **WLM groups** view, listing any workload groups already created in PLXPROD1.
   - Click the **Create...** button, and type in the following:

Workload management group	WLGPAY01
Description	Workload group for WLDPAY01

   - Click **Yes** to confirm. The **WLM groups** view is redisplayed.
6. Add the workload group WLGPAY01 to the existing workload specification WLSPAY01:
   - In the **WLM groups** view, select the entry for WLGPAY01, and click the **Add to WLM specification...** button.
   - In the **Specification name** field, enter the name of the existing workload specification, WLSPAY01 and click **Yes**. The **WLM groups** view is redisplayed.
7. Add the workload definition to the workload group:

- From the main menu, click **Administration > Workload manager administration > Definitions** to open the **WLM definitions** view.
- Select the entry for WLDPAY01, and click the **Add to WLM group...** button.
- In the **Resource group name** field, type WLGPAY01 and click **Yes**. The **WLM definition** view is redisplayed.

8. Install the workload group into the active workload.

   Because the workload WLSPAY01 is already active, you have to install the new workload group WLGPAY01 explicitly. If you did not install WLGPAY01, it would not take effect until the routing region CICSPT01 and the target regions in CICS system group CSGTGTS1 were next started.

   - From the main menu, click , **Administration > Workload manager administration > Groups** to open the **WLM groups** tabular view.
   - Select the entry for WLGPAY01 and click the **Install...** button.
   - In the **Workload name** field, type WLSPAY01, and, in the **Workload owner** field, type in the SYSID of the system on which the workload specification WLSPAY01 was created.

     Click **Yes** to confirm

Because you are reusing a workload specification that is already active in CICSplex PLXPROD1, and have installed the workload group, the workload separation you have defined in this example takes immediate effect.

You can check that the new workload definition WLDPAY01 is active by opening the **WLM definitions** view. This should include an entry for WLDPAY01 in workload WLSPAY01. When transaction PAY1 is next started, by any user and from any terminal, CICSPlex SM will route it to target region CICSPA02.

Routing particular transactions from a given user to a specific target region

This example describes how to use the Web User Interface (WUI) to route particular transactions from a given user to a specific target region.

This example task again describes how to use the Web User Interface in a refinement of the previous example (“Routing a specific transaction to a specific target region” on page 88). This time, the user ID value is to be an additional factor in determining where transactions are to be routed: transactions PAY6, PAY7, PAY8, and PAY9, when started from user ID USRPAY03, must be routed to target region CICSPA03 in CICSplex PLXPROD1.

1. Create a transaction group.

   - From the Web User Interface main menu, click **Administration > Workload manager administration > Transaction group definitions**. This opens the **Transaction group definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the **Create...** button, and provide the following information:

Name	TRGPAY02
Description	Send USRPAY03's transactions to CICSPA03
Affinity relation and lifetime checking status	ACTIVE
Primary search criterion	USERID

     You can leave the remaining fields or accept the defaults.
   - Click **Yes** to confirm. The **Transaction group definitions** tabular view is redisplayed.

2. Identify the transactions that belong to group TRGPAY02:

   - In the **Transaction group definitions** tabular view, select the entry for TRGPAY02, and click the **Add transaction...** button.

- In the **Transaction name** field, type in the name of the first transaction PAY6, and click **Yes** to confirm.

  Repeat this step for three more transactions: PAY7, PAY8, and PAY9.

3. Create a workload definition:
   - From the main menu, click , **Administration > Workload manager administration > Definitions**. This opens the **WLM definitions** tabular view, listing any workload definitions already created in PLXPROD1.
   - Click the **Create...** button, and provide the following information:

Workload management definition	WLDPAY02
Description	Separate TRGPAY02 to CICSPA03
Transaction group name	TRGPAY02
User ID	USRPAY03
Scope name of set of target systems	CICSPA03

   - Click **Yes** to confirm.
4. Add the workload definition to the existing workload group:
   - From the main menu, click , **Administration > Workload manager administration—>Definitions** to open the **WLM definitions** view.
   - Select the entry for WLDPAY0 and click the **Add to WLM group** button.
   - In the **Resource group name** field, type WLGPAY01 (the name of the workload group you created in the previous example). Workload group WLGPAY01 is already associated with the active workload specification WLSPAY01.
   - Click **Yes** to confirm.
5. Install the new workload definition into the active workload:

   Because group WLGPAY01 is already associated with the active workload WLSPAY01, changes you make to that group will not take effect until the routing region CICSPT01, and the target regions in CICS system group CSGTGTS1, are next started. To make the new workload definition take effect immediately, you must install it explicitly in WLSPAY01:

   - Open the **WLM definitions** view again, select the entry for WLDPAY02, and click the **Install...** button.
   - In the **Workload name** field, type in WLSPAY01 and in the **Workload owner** field, type in the SYSID of the system on which the workload specification WLSPAY01 was created.

Because you have explicitly installed the workload definition WLDPAY02 in the active workload WLSPAY01, the workload separation requirements you have defined in this example take immediate effect.

Honoring a pseudoconversational transaction

This example describes how to use the Web User Interface (WUI) to ensure that multiple transactions, among which there is an affinity, are routed to the same target region.

As before, you are working in CICSplex PLXPROD1 and are routing transactions from CICSPT01 to the target regions in CICS system group CSGTGTS1.

1. Create a transaction group:
   - From the main menu, click **Administration > Workload manager administration > Transaction group definitions**. This opens the **Transaction group definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the **Create...** button, and provide the following information:

Name	TRGPAY03
Description	Pseudoconversational transaction
Affinity relation and lifetime checking status	ACTIVE
Primary search criterion	USERID
Affinity relationship	USERID
Affinity lifetime	PCONV

Notice that the **Affinity relationship** and **Affinity lifetime** fields must be completed. These values tell CICSPlex SM that the transactions in this group constitute a pseudoconversational transaction (PCONV), and that this affinity lasts while those transactions are coming from the same user ID (USERID). If one of them is initiated from a different user ID, CICSPlex SM can select a different target region. Of course, the same type of affinity will then come into play in that second target region. You can ignore any fields that have been left blank or accept the defaults.

- Click **Yes** to confirm. The **Transaction group definitions** tabular view is redisplayed.

2. Identify the transactions that belong to group TRGPAY03:

   - In the **Transaction group definitions** tabular view, select the entry for TRGPAY03 and click the **Add transaction...** button.
   - In the **Transaction name** field, type in the name of the first transaction PAY2 and click **Yes** to confirm.

     Repeat this step for three more transactions: PAY3, PAY4, and PAY5.

     You can see that PAY2 starts the pseudoconversation and PAY5 ends it.

3. Create a workload definition:

   - From the main menu, click **Administration > Workload manager administration > Definitions**. This opens the **WLM definitions** view, listing any workload definitions already created in PLXPROD1.
   - Click the **Create...** button, and provide the following information:

Workload management definition	WLDPAY03
Description	TRGPAY03 to the same target region
Transaction group name	TRGPAY03
Scope name of set of target systems	CSGTGTS1

   - Click **Yes** to confirm.

4. Add the workload definition to the workload group:

   - From the main menu, click **Administration > Workload manager administration > Definitions** to open the **WLM definitions** tabular view.
   - Select the entry for WLDPAY03 and click the **Add to WLM group...** button.
   - In the **Resource group name** field, type WLGPAY01.
   - Click **Yes** to confirm.

5. Because group WLGPAY01 is already associated with the active workload WLSPAY01, changes you make to that group will not take effect until the routing region CICSPT01, and the target regions in CICS system group CSGTGTS1, are next started. To make the new workload definition take effect immediately, you must install it explicitly in WLSPAY01. To install the new workload definition into the active workload:

   - Open the **WLM definitions** view again, select the entry for WLDPAY03, and click the **Install...** button.
   - In the **Workload name** field, type in WLSPAY01, and in the **Workload owner** field, type in the SYSID of the system on which the workload specification WLSPAY01 was created.

Because you have explicitly installed the workload definition WLDPAY03 in the active workload WLSPAY01, CICSPlex SM is able to honor this pseudoconversational transaction immediately. Be aware that you are able to use the single workload specification WLSPAY01 for both workload routing and workload separation because you did *not* specify default **Affinity relationship** and **Affinity lifetime** values in WLSPAY01. Had you done so, you would have had to create different workload specifications for workload routing and workload separation.

Deactivating a workload definition

This example describes how to use the Web User Interface (WUI) to deactivate a workload definition.

This example describes how to use the Web User Interface (WUI) to deactivate the workload definition WLDPAY02 created in the example "Routing particular transactions from a given user to a specific target region" on page 90.

1. Display active workload definitions:
   - From the main menu, click **Active workloads (WLM) > Definitions**. This opens the **Active workload definitions** view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - In the **Active workload definitions** view, type WLSPAY01 and click **Refresh**. The active workload definitions associated with workload specification WLSPAY01 are listed.
2. Discard workload definition WLDPAY02:
   a. Select the entry for WLDPAY02, and click the **Discard...** button. This opens the **Discard** confirmation view.
   b. Click **Yes** to confirm the action.

Be aware that, when you deactivate an active workload definition, you also deactivate any transaction groups associated with it if they aren't referenced by another workload definition in the same workload.

Updating an active workload definition

This example describes how to use the Web User Interface (WUI) to update an active workload definition and then reinstall it in the active workload.

In the example "Routing particular transactions from a given user to a specific target region" on page 90, you created the transaction group TRGPAY02 and named it in the workload definition WLDPAY02. In this example, you'll see how to remove TRGPAY02 and replace it with a new transaction group, TRGPAY04, which has already been created.

1. Display active workload definitions:
   - From the main menu, click **Active workloads (WLM) > Definitions**. This opens the **Active workload definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - In the **Active workload definitions** view, type WLSPAY01 and click **Refresh**. The active workload definitions associated with workload specification WLSPAY01 are listed.

   Make a note of the value in the **System ID of workload owner** field for WLDPAY02 (CM1B, in this example).
2. Update workload definition WLDPAY02.
   - From the main menu, click **Administration > Workload manager administration > Definitions** to open the **WLM definition** tabular view.
   - Select the entry for WLDPAY02 and click the **Update...** button. The **WLM definitions** create view is displayed.
   - In the **WLM definitions** create view, change the **Transaction group name** to TRGPAY04 and change the **Description** text to "Separate TRGPAY04 to CICSPA03"

- Click **Yes**. The **WLM definitions** tabular view is redisplayed.

3. Install the updated workload definition in WLSPAY01:

   In the **WLM definitions** tabular view, select the entry for WLDPAY02 and click the **Install...** view. The **Install** view is displayed. In the **Workload name** field, type WLSPAY01. In the **Workload owner** field, type the 4-character ID of the workload owner that you made a note of in step "1" on page 93. Click **Yes**. The updated workload definition is installed in workload WLSPAY01.

4. Check that the updated workload definition has been installed:

   - From the main menu, click **Active workloads (WLM) > Definitions**. This opens the **Active workload definitions** tabular view showing the updated definition. Check that TRGPAY02 has been replaced by TRGPAY04.

5. Check that transaction group TRGPAY02 is inactive and that transaction group TRGPAY04 is now active:

   - From the main menu, click **Active workloads (WLM) > Transaction groups** to view the status of the two transaction groups.

Note:

Updating the **User ID**, **Terminal LU name**, **BTS process type**, or **Scope name of set of target systems** fields in a workload definition prevents the workload definition from being reinstalled dynamically. To reinstall the workload definition into an active workload, you must:

- From the **Active workload definitions** tabular view, click the **Discard...** button to discard the active workload definition
- From the workload management **Definitions** view, click the **Install...** button to install the workload definition into the workload.

Alternatively, to reinstall workload definitions with changed attributes, use a batch API program to disable access to the affected application, discard the **Active workload definitions** view, install the **WLM definitions** view, and re-enable the affected application.

Discarding an active transaction from a workload

This example describes how to use the Web User Interface (WUI) to discard an active transaction from a workload.

1. Display active transactions.

   - From the main menu, click **Active workloads (WLM) > Dynamic transactions**.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - The **Active workload dynamic transactions** tabular view shows the active transactions associated with workload specification EYUWLS02

2. Discard transaction PAY2:

   - From the **Active workload dynamic transactions** view, select transaction PAY2, and click the **Force...** button. The **Force** confirmation view is displayed.
   - Click **Yes** to confirm the action.

Updating a workload specification

This example describes how to use the Web User Interface to update a workload specification.

About this task

In this example, the workload specification to be updated is WLSPAY01, which you created in the example "Managing a workload" on page 84.

Consider the effects of this task carefully. In particular, the updated workload specification cannot take effect immediately. For the update to take immediate effect, you must stop and then restart both the routing region with which the workload specification is associated and the target regions to which the routing region routes transactions.

Procedure

1. Update workload specification WLSPAY01.
   a. From the Web User Interface main menu, click **Administration** > **Workload manager administration** > **Specifications** to open the **WLM specifications** (WLMSPEC) tabular view.

      If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   b. Select the entry for WLSPAY01 and click the **Update** button.
   c. Change the **Algorithm type** field from QUEUE to GOAL.
   d. Click **Yes**. The view is redisplayed showing the updated entry for WLSPAY01.
2. Display the **Active workload** view:
   a. From the main menu, click **Active workloads (WLM)** > **Active workloads**.
   b. In the **Active workloads** tabular view, type in the workload name WLSPAY01 and click **Refresh**.

   The **Active workloads** tabular view is redisplayed. You will see that the algorithm type for WLM specification WLSPAY01 is still QUEUE. This is because region CICSPT01 doesn't know about the changes you've made.

What to do next

If you want the updated workload specification to take effect immediately, use the **CICS regions** view to stop the routing regions and the target regions to which they route, and then restart them. The regions must all be shut down together and not restarted until each CMAS that had CICS elements participating in the named workload issues message EYUWM0410I confirming that the workload specification has terminated successfully. If CICS regions that are members of the workload restart before receiving message EYUWM0410I, the existing workload specification will continue to be used and the updates to the workload specification will not be in effect. After the routing regions are restarted, display the **Active workloads** tabular view and you will see that the **Algorithm type** field has been updated to GOAL.

Using real-time analysis to select a target region for workload routing

Use CICSPlex SM's real-time analysis functions to produce data that will help in the selection of a target region during workload routing.

The number of items on the temporary storage queue will be monitored for each target region to which work can be routed. When the number of items rises above 50, a real-time analysis event notification (severity HS) and an external message will be issued. When an event notification is issued, CICSPlex SM's workload-routing function is notified and uses the information, in addition to the standard queue algorithm criteria, in selecting the best target region.

A CICS system group CSGTGTS3 has already been created and contains four target regions (CICSPA01, CICSPA02, CICSPA03, and CICSPA04). The target regions are currently running. Work is currently being routed among these target regions by the routing region CICSPT03.

1. Create a real-time analysis evaluation definition:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Evaluations**. This opens the **Evaluation definitions** tabular view, listing any evaluation definitions already created in PLXPROD1.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the **Create...** button, and provide the following information:

Name	RTEPAY15
Description	TSQ NUMITEMS > 50

Sample interval	300
Resource table name	MTSQGBL
Instance identifier of evaluated resource	*
Method of evaluating results in result set	ANY
Separate task indicator	NO
Name of field being evaluated	PUTQAUX
Evaluation type	VALUE
Evaluation logical operator	GT
Evaluation data value	50
Severity assigned when result meets criteria	HS
Name of a view that may provide extra information	MTSQGBL

- Click **Yes** to create the new evaluation definition.

2. Create a real-time analysis action definition:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Actions**. This opens the **Action definitions** view.
   - Click the **Create...** button, and provide the following information:

Action	RTAPAY15
Description	NUMITEMS IN TSQ > 50
Generate event option	YES
Name of view that may provide useful information	TSQ
Action priority	255
Message to send when event occurs	AUX TSQUEUE PUTQ ITEMS > 50
Generate external message option	YES
External message sent when event occurs	AUX TSQUEUE PUTQ ITEMS > 50
External message sent when event is cleared	AUX TSQUEUE PUTQ ITEMS < 50
Generate SNA generic alert option	NO
MVS automatic restart	NO

   - Click **Yes** to create the new RTA action definition. The **Action definitions** tabular view is redisplayed.

3. Create an analysis definition:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Definitions**. This opens the **RTA definitions** view.
   - Click the **Create...** button, and provide the following information:

Name	RTDPAY15
Description	TSQ NUMITEMS FOR WLM /RTA
Execute evaluation modification string	NO
Analysis interval	300
Action definition name	RTAPAY15
Count of true evaluations before VLS raised	0001

Count of false evaluations before VLS resolved	0001
Count of true evaluations before LS raised	001
Count of false evaluations before LS resolved	001
Count of true evaluations before LW raised	001
Count of false evaluations before LW resolved	001
Count of true evaluations before HW raised	001
Count of false evaluations before HW resolved	001
Count of true evaluations before HS raised	001
Count of false evaluations before HS resolved	0001
Count of true evaluations before VHS raised	0001
Count of false evaluations before VHS resolved	0001
Count of false evaluations before VHS resolved	0001
Evaluation expression	RTEPAY15

- Click **Yes** to create the new analysis definition. The **Analysis definitions** tabular view is redisplayed.

4. Create an analysis group:
   - From the main menu, click**Administration** > **RTA MAS resource monitoring** > **Groups**. This opens the **RTA groups** tabular view.
   - Click the **Create...** button, and provide the following information:

RTA group	RTGPAY09
Description	RTA/WLM Group

   - Click **Yes** to create the new analysis group. The **RTA groups** tabular view is redisplayed.
5. Add the analysis definition to the analysis group:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Definitions**. This opens the **RTA definitions** view.
   - Click the record check box beside the entry for RTDPAY15 and click the **Add to RTA group...** button. The **Add to RTA group** view is displayed
   - In the **Resource group name** field type RTGPAY09 and click **Yes**. The **RTA definitions** tabular view is redisplayed
6. Create an analysis specification:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Specifications**. This opens the **RTA specifications** tabular view.
   - Click the **Create...** button, and provide the following information:

RTA specification name	RTSPAY09
Description	Workload routing with RTA

   - Click **Yes** and the **RTA specifications** tabular view is redisplayed, listing the new RTA specification.
7. Add the analysis group to the analysis specification:
   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Groups**. This opens the **RTA groups** tabular view.
   - Click the record check box beside the entry for RTGPAY09 and click the **Add to RTA specification...** button. The **Add to RTA specification** view is displayed

- In the **RTA specname** field, type RTSPAY09 and click **Yes**. The **RTA groups** tabular view is redisplayed

8. Set the scope of the analysis specification.:

   - From the main menu, click **Administration** > **RTA MAS resource monitoring** > **Specifications**. This opens the **RTA specifications** tabular view.
   - Click the record check box beside the entry for RTGPAY09 and click the **Associate CICS group...** button. The **Associate CICS group** view is displayed
   - In the **CICS system group** field, type CSGTGTS3 and select the **Force** option. Click **Yes** and the **RTA specifications** tabular view is redisplayed.

9. Activate real-time analysis in the target regions in CSGTGTS3:

   - From the main menu, click **Administration** > **RTA system availability monitoring** > **CICS system definitions**. This opens the **CICS system definitions** tabular view.
   - Click the record check box beside the entry for CICS system CICSPA01, click the **Update...** button, and provide the following information:

CICS system definition name	CICSPA01
Description	Target region 1 on System A
Real time analysis status	YES
Severity for system availability monitoring event	NO
Severity for short-on-storage (SOS) event	NO
Severity for system dump event	NO
Severity for transaction dump event	NO
Severity for CICS-at-maximum-tasks event	NO
Severity for CICS-stalled event	NO

   - Click **Yes** to confirm the change. The CICS system definition is updated and the change, which is permanent, takes immediate effect. You do not have to restart CICS system CICSPA01. Repeat this step for the target regions CICSPA02, CICSPA03, and CICSPA04.

10. The workload specification WLSPAY02 is already being used for workload routing among the target regions in CSGTGTS3. You must update the specification to add the real-time analysis data to the standard target region selection criteria. To update the workload specification:

    - From the main menu, click **Administration** > **Workload manager administration** > **Specifications**. This opens the**WLM specifications** tabular view.
    - Click the record check box beside the entry for WLSPAY02, click the **Update...** button, and provide the following information:

Name	WLSPAY02
Description	Workload using RTA for target region selection
Primary search criterion	USERID
Default target scope	CSGTGTS3
RTA event name	RTDPAY15
Acceptable level of abend probability	0
Acceptable abend load threshold	0
Algorithm type	QUEUE

    - Click **Yes** to update the specification.

Notice that the **RTA event** field value is the name of the analysis definition you created in step "3" on page 96.

When the routing region CICSPT03 and the target regions in CICS system group CSGTGTS3 are next started, the routing region routes transactions among the target regions using both the standard queue algorithm criteria and the analysis definition RTDPAY15 to select a target region.

Dynamic routing with EXEC CICS START TERMID

Use the Web User Interface (WUI) to set up dynamic routing capability for a transaction started with **EXEC CICS START**, that specifies a terminal ID and a transaction ID, and to use the GOAL algorithm to select the target region.

Note: You should check the system requirements before trying to route **EXEC CICS START TERMID** dynamically. See CICSPlex SM workload requirements.

In this example, a program running in CICSPA01 begins with **EXEC CICS START**, which is associated with terminal TRM1, to run transaction PAY1, for which you require a 2-second response time. Terminal TRM1 is associated with region CICSPT01. Transaction PAY1 may execute in any region connected to CICSPT01, that is, in CICSPA01, CICSPA02, or CICSPA03.

This example uses the environment that has already been created as part of earlier examples. You are working in CICSplex PLXPROD1, which comprises TOR CICSPT01 and AORs CICSPA01, CICSPA02, and CICSPA03 in CICS system group CSGTGTS1. Transaction group TRGPAY03 has transactions PAY1, PAY2, PAY3, and PAY4 associated with it. Workload definition WLDPAY03 was defined to tell CICSPlex SM that transactions in group TRGPAY03 must be routed to a target region in CICS system group CSGTGS1.

As you want to use the GOAL algorithm, you need to define, in MVS Workload Manager, a Service Class with the required response time and allocate that Service Class to transaction PAY1. For example, you could specify:

- A Service Class of Fast that has an average response time of 2 seconds.
- A classification rule that associates Classification Subsystem CICS with transaction ID PAY1 and Service Class Fast.

For more information about Service Classes, see the information about the goal algorithm in Management of the work in a workload.

In CICSPlex SM:

1. Create a workload specification.
   - From the main menu, click **Administration** > **Workload manager administration** > **Specifications**. This opens the **WLM specifications** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the **Create** button, and provide the following information:

Name	WLSDYN01
Description	Dynamic routing START
Primary search criterion	USERID
Default target scope	CSGTGTS1
Acceptable level of abend probability	0
Acceptable abend load threshold	0
Algorithm type	GOAL

   - Click **Yes** to create the specification.

   You must supply:

- A name for the specification (WLSDYN01 in this example).
- A **Primary search criterion** value. USERID is shown in this example, though in fact it doesn't matter whether you specify USERID or LUNAME, because the **Primary search criterion** value has no effect on simple workload routing. It is used only for some kinds of workload separation, but you have to supply a value because CICSPlex SM doesn't know, at this stage, that you aren't going to use this workload specification for workload separation.
- A **Default target scope** value, which is the name of the single target region, or group of target regions, to which transactions can be routed.
- An **Algorithm type** value. For this example, use GOAL, because the criteria for selecting the target region are based on the requirement that the transaction has a response time of 2 seconds.

When you click **Yes**, the **WLM specifications** tabular view is redisplayed, this time showing an entry for the new workload specification, WLSDYN01.

2. The next step is to tell CICSPlex SM about the region that's going to be routing the work requests to the target regions in group CSGTGTS1. To associate the workload specification with a routing region:

   - On the **WLM specifications** tabular view, click the Record check box beside the entry for the WLSDYN01 specification and click the **Associate CICS system** button.
   - Type WLSDYN01 in the **CICS system** field and click **Yes**. The **WLM specifications** tabular view is redisplayed and you can check that the association between the routing region and the workload specification has been created.

3. Activate workload routing in the routing region:

   - From the main menu, click **Administration** > **Topology administration** > **System definitions**. This opens the **CICS system definitions** tabular view.
   - Click the Record check box beside the entry for CICSPT01 and click the **Update** button.
   - In the **Workload manager status** field, select YES from the drop-down menu. This change takes effect when CICSPT01 is next started.

4. Activate workload routing in the target regions.

   - From the main menu, click **Administration** > **Topology administration** > **System definitions**. This opens the **CICS system definitions** tabular view.
   - Click the Record check box beside the entry for CICSPA01 and click the **Update** button.
   - In the **AOR dynamic routing mode** field, select YES from the drop-down menu. This change takes effect when the target region CICSPA01 is next started.

   Repeat this step for target regions CICSPA02 and CICSPA03.

5. Install program definitions:

   - From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Program definitions**. This opens the **Program definitions** tabular view.
   - Click the Record check box beside the entry for CICSPA01 and click the **Update** button.
   - Set the **Dynamic routing status** field to NO from the drop-down box.
   - Click **Yes** to confirm. The **Program definitions** tabular view is redisplayed.
   - Click the Record check box beside the entry for CICSPA01 and click the **Install** button.
   - Type CICSPA01 in the **Target scope value** field and click Yes to confirm.

   Repeat this step for CICSPA02 and CICSPA03.

6. Install transaction definitions:

   - From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Transaction definitions**. This opens the **Transaction definitions** tabular view.
   - From the **Transaction definitions** view, select PAY1 and click the **Install** button. Type CICSPA01 in the **Target scope value** field.
   - Click Yes to confirm.

You can either update the transaction definitions to specify these field values, or you can specify override values when you install the transaction definitions.

In this example task, a program running in CICSPA01, the requesting region, issues an **EXEC CICS START** command that specifies a transaction ID of PAY1 and a terminal ID TRM1. The START command is function shipped to CICSPT01, the TOR that owns the specified terminal. CICSPT01 acts as the routing region and invokes the dynamic routing exit, which selects the target region. All the AORs in CICS system group CSGTGS1 are possible target regions; the actual target region is selected on the basis of the GOAL criterion of a response time of 2 seconds. CICSPlex SM obtains the Service Class of transaction PAY1 and the identity of the target region to which that Service Class is allocated, from a CICSPlex SM-maintained table.

Dynamic routing of an inbound client DPL request

This example describes how to use the Web User Interface (WUI) to set up dynamic routing capability for a DPL request from a CICS client, and to use the GOAL algorithm to select the target region.

Before you begin

You should check the system requirements before trying to dynamically route inbound client DPL requests. See CICSPlex SM workload requirements.

About this task

In this example, a request is received to run transaction PAY1 to invoke client program PAYPROG1. PAYPROG1 issues an **EXEC CICS LINK** command to server program PAYPROG2. You require transaction PAY1 to have a 2-second response time.

This example uses the environment that was used for "Dynamic routing with EXEC CICS START TERMID" on page 99.

As you want to use the GOAL algorithm, you need to define, in MVS Workload Manager, a Service Class with the required response time and allocate that Service Class to the transaction. For example, you could specify:

- A Service Class of Fast with a average response time of 2 seconds.
- A classification rule that associates Classification subsystem CICS with transaction ID PAY1 and Service Class Fast.

For more information about Service Classes, see the information about the goal algorithm in Management of the work in a workload.

Perform the following procedure in CICSPlex SM.

Procedure

1. Set the dynamic routing status value for program PAYPROG1.
   a) From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Program definitions** > .
      This opens the **Program definitions** tabular view.
   b) If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   c) Click the Record check box beside the entry for PAYPROG1 and click the **Update** button. From the drop-down box, set the **Dynamic routing status** field to NO.
   d) Click **Yes** to confirm.
2. Install program definitions.
   a) From the **Program definitions** tabular view, click the Record check box beside the entry for PAYPROG1 and click the **Install** button.
   b) Type `CICSPA01` in the **Target scope value** field.

c) Click **Yes** to confirm.

Repeat this step to install PAYPROG1 in target scopes CICSPA02 and CICSPA03.

For details of defining and installing program definitions, see Creating resources with BAS: PROGRAM resource definitions.

3. Install the transaction definition.

   a) From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Transaction definitions**.

   This opens the **Transaction definitions** tabular view.

   b) If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.

   c) From the **Transaction definitions** view, click the Record check box beside the entry for PAY1 and click the **Install** button.

   d) Type `CICSPA01` in the **Target scope value** field.

   e) Click **Yes** to confirm.

   Repeat this step to install PAY1 in target scopes CICSPA02 and CICSPA03.

   The transaction definition should point to the mirror program DFHMIRS. For details of defining and installing transaction definitions, see Creating resources with BAS: Transaction resource definitions.

Dynamic routing of a peer-to-peer DPL request

This example describes how to use the Web User Interface (WUI) to set up dynamic routing for a peer-to-peer DPL request, and to use the GOAL algorithm to select the target region.

Note: You should check the system requirements before trying to dynamically route peer-to-peer DPL requests. See CICSPlex SM workload requirements.

In this example, transaction PAY1 runs program PAYPROG1, which issues an **`EXEC CICS LINK`** command to program PAYPROG2. You require transaction PAY1 to have a response time of 4 seconds.

This example uses the environment that was used for "Dynamic routing with EXEC CICS START TERMID" on page 99. However, because this is a peer-to-peer dynamic linking request, only the AORs are involved and each AOR may act as a requesting, routing, or target region.

As you want to use the GOAL algorithm, you need to define, in MVS Workload Manager, a Service Class with the required response time and allocate that Service Class to the transaction. For example, you could specify:

- A Service Class of Medium with an average response time of 4 seconds.
- A classification rule associating Classification Subsystem CICS with transaction ID PAY1 and Service Class Medium.

For more information about Service Classes, see the information about the goal algorithm in Management of the work in a workload.

In CICSPlex SM:

1. The first step is to set the dynamic routing status value for program PAYPROG1. To do this:

   - From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Program definitions**. This opens the **Program definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the Record check box beside the entry for PAYPROG1 and click the **Update...** button.
   - Set the **Dynamic routing status** field to NO from the drop-down box.
   - Click **Yes** to confirm.

2. Install program definitions:

- From the **Program definitions** tabular view, click the Record check box beside the entry for PAYPROG1 and click the **Install...** button.
- Type CICSPA01 in the **Target scope value** field.
- Click **Yes** to confirm.

Repeat this step to install PAYPROG1 in target scopes CICSPA02 and CICSPA03.

For details of defining and installing program definitions, see Creating resources with BAS: PROGRAM resource definitions.

You don't need to install program definition PAYPROG2 in the regions, but if you do, PAYPROG2 should be defined as dynamic.

3. Install the transaction definition:

   - From the main menu, click **Administration** > **Basic CICS resource administration** > **Resource definitions** > **Transaction definitions**. This opens the **Transaction definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - From the **Transaction definitions** tabular view, Click the Record check box beside the entry for PAY2 and click the **Install...** button.
   - Type CICSPA01 in the **Target scope value** field.
   - Click **Yes** to confirm.

   The transaction definition should point to the mirror program DFHMIRS.

   Repeat this step to install PAY2 in target scopes CICSPA02 and CICSPA03.

Routing CICS BTS activities

Use the Web User Interface (WUI) to route a CICS BTS-related workload.

This example uses the configuration that has been used for earlier examples. You are working in CICSplex PLXPROD1, which comprises AORs CICSPA01, CICSPA02, and CICSPA03 in CICS system group CSGTGTS1. These three systems may act as routing and target regions. In addition, it is assumed that the RLS file for the BTS process type is accessible to all CICS system in CICS system group CSGTGTS1, and that all the systems in CICS system group CSGTGTS1 are interconnected.

Note: You should check the system requirements before trying to dynamically route CICS BTS activities. See “Separating CICS BTS activities” on page 104 and CICS release requirements for dynamic routing.

1. Activate workload routing for the systems in CSGTGTS1.

   - From the main menu, click **Administration > Topology administration > System definitions**. This opens the **System Definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
   - Click the record check box beside the entry for CICSPA01 and click the **Update...** button.
   - Select YES from the drop-down box for both the **Workload manager status** field and in the **AOR dynamic routing mode** field . These changes takes effect when the target region CICSPA01 is next started.

   Repeat this step for target regions CICSPA02 and CICSPA03.

2. Create a workload specification:

   - From the main menu, click **Administration > Workload manager administration > Specifications**. This opens the **WLM specifications** tabular view.
   - Click the **Create...** button, and provide the following information:

Name	WLSCBTS
Description	Routing CICS BTS activity

Primary search criterion	USERID
Default target scope	CSGTGTS1
Acceptable level of abend probability	0
Acceptable abend load threshold	0
Algorithm type	LNQUEUE

- Click **Yes** to confirm. The **WLM specifications** tabular view is redisplayed.

You must supply :

- A name for the specification (WLSCBTS in this example).
- A primary search criterion value. USERID is shown in this example, though in fact it does not matter whether you specify USERID or LUNAME, because the primary search criterion value has no effect on simple workload routing. It is used only for some kinds of workload separation, but you have to supply a value because CICSPlex SM cannot determine, at this stage, that you are not going to use this workload specification for workload separation.
- A default target scope value, which is the name of the single target region, or group of target regions, to which transactions can be routed. In this example, use CSGTGTS1.
- An algorithm Type value. For this example, use LNQUEUE. The LNQUEUE (link neutral queue) algorithm ignores the type of link between routing and target regions but otherwise uses the same routing criteria as the QUEUE algorithm.

3. Associate the workload specification with routing region scope CSGTGTS1.
   - Click the Record check box beside the entry for WLSCBTS and click the **Associate CICS system...** button.
   - In the **CICS system** field, type CSGTGTS1.
   - Click **Yes** to confirm. The **WLM specifications** tabular view is redisplayed.
4. Specify the system initialization parameter DSRTPGM is set to EYU9XLOP.

   You can do this either in the system initialization parameter of each region in the CICS system group, or by using the **CICS system definitions** view.
5. Activate workload management:
   - From the main menu, click **CICSPlex SM operations > MASs known to CICSplex**. This opens the **MASs known to CICSplex** tabular view.
   - Click CICS system name CICSPA01 open the **MASs known to CICSplex** detailed view for CICSPA01.
   - In the **Workload manager status** field, select YES from the drop-down list to start managing workloads for this CICS system.
   - Click **Apply changes**.

   Repeat this step to activate workload management for CICSPA02 and CICSPA03.
6. Program your BTS activities to run asynchronously.

When the routing and target regions in system group CSGTGS1 are next started, BTS activities are routed among the target regions.

Separating CICS BTS activities

This example describes how to use the Web User Interface (WUI) to separate a CICS BTS-related workload.

In this example, BTS activity BTSACT1 has an affinity of LIFETIME and runs under transaction ID BTS1 and process type SALES. This example uses the configuration that has been used for the example in "Routing CICS BTS activities" on page 103. Similarly, the system initialization parameter DSRTPGM must be set to EYU9XLOP.

Note: You should check the system requirements before trying to dynamically route CICS BTS activities. See CICS release requirements for dynamic routing.

1. Create a transaction group:

   - From the main menu, click **Administration views > Workload manager administration views > Transaction groups definitions**. This opens the **Transaction group definitions** tabular view.

     If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.

   - Click the **Create...** and provide the following information:

Name	TRGCBTS1
Description	CICS BTS transaction group
Affinity relation and lifetime checking status	ACTIVE
Primary search criterion	USERID
Affinity relationship	BAPPL
Affinity lifetime	ACTIVITY
Acceptable level of abend probability	0
Acceptable abend load threshold	0
Algorithm type	QUEUE

     You can ignore any fields that have been left blank.

   - Click **Yes** to redisplay the **Transaction group definitions** tabular view, now showing the name of the transaction group TRGCBTS1.

     Notice that the **Affinity relationship** and **Affinity lifetime** fields must be completed. These values tell CICSPlex SM that the transactions in this group constitute a BTS affinity, and that this affinity lasts while those transactions are coming from the same BTS application. If one of them is initiated from a different BTS application, CICSPlex SM can select a different target region. Of course, the same type of affinity will then come into play in that second target region.

2. Identify the transactions in group TRGCBTS1:

   - From the **Transaction group definitions** view select the entry for TRGCBTS1 and click the **Add transaction...** to open the **Add transaction** tabular view.
   - In the **Transaction name** field, type BTS1 and click **Yes** to add the transaction to the Transaction group. The **Transaction group definition** view is redisplayed.

3. Create a workload definition:

   - From the main menu, click **Administration > Workload manager administration > Definitions**. This opens the **WLM definitions** tabular view.
   - Click the **Create...** button and provide the following information:

Workload management definition	WLDCBTS3
Description	Separate CICS BTS activities
Transaction group name	TRGCBTS1
BTS process type	SALES
Scope name of set of target systems	CSGTGTS1

   - Click **Yes** and the **WLM definitions** tabular view is redisplayed.

   These values tell CICSPlex SM that transactions in group TRGCBTS1, and of process type SALES, must go to a target region in group CSGTGTS1. CICSPlex SM can select the most appropriate target region at the time the transaction is initiated.

4. Add the workload definition to the workload group:
   - In the **WLM definition** tabular view, click the Record check box beside the entry for the WLDCBTS1 and click the **Add to WLM group...** button. The **Add to WLM group** view is displayed.
   - In the **Resource group name** field, type WLGCBTS1 and click **Yes**. The **WLM definitions** tabular view is redisplayed.
5. Add the workload group to a workload specification:
   - From the main menu, click **Administration views > Workload manager administration views > Groups**. This opens the **WLM group** tabular view.
   - In the **WLM group** view, click the Record check box beside the entry for the WLDCBTS1 and click the **Add to WLM specification...** button. The **Add to WLM specification** view is displayed.
   - In the **Specification name** field, type WLGCBTS1 and click **Yes**. The **WLM groups** tabular view is redisplayed.

Managing a Link3270 bridge workload

These examples describe how to use the Web User Interface (WUI) to manage a Link3270 bridge workload.

In the case of Link3270 bridge requests, the client application in the requester region calls the Link3270 bridge using **EXEC CICS LINK**, EXCI or ECI, and passes a communication area to the bridge router program, DFHL3270, which runs in the router region. The transaction ID that is used for Link3270 bridge workload routing is the name that is passed to DFHL3270 in the communication area and is not necessarily the same as the transaction name entered at the terminal or workstation. The target regions contain the bridge environment in which the target transaction runs.

When CICSPlex SM is in use and the CICSPlex SM routing exit, EYU9XLOP, is specified as the **DTRPGM** system initialization parameter, DFHL3270 uses CICS distributed program linking to pass control to EYU9XLOP. The mirror program, DFHMIRS, passes the following information to EYU9XLOP in the DFHDYPDS communication area:

- A DYRTYPE value of 8, which indicates a Link3270 bridge request type
- An eight-character bridge token in the DYRBRTK field
- The transaction ID of the transaction that is to run in the target region in the DYRTRAN field.

For more information about the Link3270 bridge, see Introduction to the 3270 bridge.

For more information about CICS distributed program linking, see CICS distributed program link.

Routing a Link3270 bridge workload

In this example CICSPT01 is the router region and the target regions are CICSPA01, CICSPA02 and CICSPA03.

1. If the current context is not PLXPROD1, specify PLXPROD1 in the **Context** field and click **Refresh**.
2. Set the DTRPGM system initialization parameter to EYU9XLOP in all regions.
3. Update definitions
   a. From the Web User Interface main menu, click **Administration > Topology administration > System definitions** to open the **CICS system definition** tabular view.
   b. Select the entry for CICSPT01 and click **Update** to open a detail view of CICSPT01.
   c. Scroll down to the **Workload manager status** field and select **YES** from the menu.
   d. Scroll down to the **AOR dynamic routing mode** field and select **YES** from the menu .
   e. Click **Yes** at the bottom of the view to return to the **CICS system definitions** tabular view. See Administering CICSPlex SM for a description of the CICS system definition view.

      This change takes effect when the target region CICSPA01 is next started.

Repeat this for the regions CICSPA01, CICSPA02 and CICSPA03.

4. Create a CICS system group

   a. From the main menu click **Administration > Topology administration > System groups** to open the **System group definitions** tabular view.

   b. Scroll to the bottom of the view and click **Create**, to create a system group called CSGTGTS1

      CSGTGTS1, in this example, is to contain the regions that are to act as targets. See Administering CICSPlex SM for a description of the **System group definitions** view.)

5. Add the target regions to CSGTGTS1.

6. Create a workload specification

   a. From the main menu click **Administration > Workload manager administration > Specifications** to open the **WLM specifications** tabular view (WLMSPEC object).

   b. Scroll to the bottom of the view, click **Create**, and provide the following information:

      WLM specification name
      BRSPEC01

      Description
      Link3270 Bridge Workload

      Primary search criterion
      USERID

      Automatic affinity creation option
      N/A

      Default target scope
      BRITGTS1

      Algorithm type
      QUEUE

      Leave the remaining fields empty or accept the defaults.

   You must supply:

   - The name of the workload specification. In this example it is called BRSPEC01.
   - A **Primary search criterion** value. USERID is shown in this example, though in fact it doesn't matter whether you specify USERID or LUNAME, because the **Primary search criterion** value has no effect on simple workload routing. It is used only for some kinds of workload separation, but you have to supply a value because CICSPlex SM doesn't know, at this stage, that you aren't going to use this workload specification for workload separation.
   - A **Default target scope**, which is the name of the region or group of regions (BRITGTS1 in this example) to which work is to be routed.
   - An **Algorithm type**. This example uses QUEUE but you can specify either QUEUE, LNQUEUE, GOAL, or LNGOAL for Link3270 bridge requests.

7. Associate the workload specification with the routing region scope

   - On the **WLM specifications** tabular view, select the record for BRSPEC01 and click the **Associate CICS system...** button.
   - Enter the routing region scope into the **CICS system** field and click **Yes**.

When the regions are next restarted, the workload will be routed across the target regions. You can use the **Active workloads** view (WLMAWORK object) to check that workload specification BRSPEC01 is active. You can use the **Active workload target distribution factors** view (WLMAWAOR object) to see to which target regions workloads are being routed.

Separating a Link3270 bridge workload

You can separate Link3270 bridge workloads by user ID, transaction group, or LU name.

About this task

You can separate by LU name only if you are overriding the NETNAME that the bridge generates automatically. Be aware of the restrictions on the use of LUNAME when separating Link3270 bridge workloads. For more information, see Separating Link3270 bridge workloads. For Link3270 bridge workloads, the LU name is the eight–character NETNAME of the terminal running the client transaction that started the Link3270 bridge, rather than the NETNAME of the bridge facility itself.

To separate by bridge facility NETNAME, you must modify the EYU9WRAM module. For more information, see Separating Link3270 bridge workloads. CICS routes all transactions running under the same bridge facility to the same target region. You cannot force them to go to different regions.

The following example describes how to separate a bridge workload by userid and transaction group. The example uses the same configuration as that described in "Creating workload management definitions" on page 76.

Add the following definitions to the definitions that you created in "Creating workload management definitions" on page 76.

Procedure

1. Create a transaction group.
   a) From the Web User Interface main menu, click **Administration** > **Workload manager administration** > **Transaction group definitions**.

      The **Transaction group definition** view opens, listing any transaction groups already defined in PLXPROD1.

      If the current context is not PLXPROD1, specify PLXPR0D1 in the **Context** field and click **Refresh**.
   b) Scroll to the bottom of the view, click **Create**, and provide the following information:

      Transaction group name
      TRGBRI01

      Description
      Link3270 bridge transaction group

      Affinity relation and lifetime checking status
      ACTIVE

      Primary search criterion
      User ID

      Automatic affinity creation option
      N/A

      You can leave the remaining fields or accept the defaults.
   c) Click **Yes**.

      The **Transaction group definition** view is redisplayed, now showing the name of the transaction group TRGBRI01.

      Leave the **Affinity relationship** and **Affinity lifetime** fields blank and the **Automatic affinity creation option** field set to **N/A** because CICSPlex SM does not handle affinities between Link3270 bridge transactions.

   Note: If you want to separate by LUNAME, you must enter LUNAME in the **Primary search criterion** field of the **Transaction group definition** create view.
2. Identify the transactions in group TRGBRI01.
   a) In the **Transaction group definition** view, select the entry for TRGPAY03, and click **Add transaction**.
   b) Type the name BRI1 in the **Transaction name** field, and click **Yes** to confirm.

      The **Transaction group definition** view is redisplayed.

3. Repeat the previous two steps for any further transactions that you want to be routed to different regions.
4. Create a workload definition.
   a) From the main menu, click **Administration** > **Workload manager administration** > **Definitions**.

      The **Workload management definition** view opens.
   b) Scroll to the bottom of the view, click **Create**, and provide the following information:

      Workload management definition name
      WLDBRI01

      Description
      Separate Link3270 bridge activities

      Transaction group name
      TRGBRI01

      Terminal LU name
      *

      User ID
      BRIUSER1

      BTS process type
      *

      Scope name of set of target systems
      CICSPA01

      These values indicate that transactions in group TRGBRI01 entered by BRIUSER1 are to be routed to CICSPA01.
   c) Click **Yes** to confirm.

   Note: If you want to separate by LUNAME, you must enter the LUNAME in the **Terminal LU name** field of the **Workload management definition** create view.
5. Repeat the previous step for any further transaction groups you have created for transactions that are to be routed to different regions.
6. Create a workload group.

   A workload group is essential if you want a workload definition to be installed automatically when the routing region that is routing the transactions is started.
   a) From the main menu, click **Administration** > **Workload manager administration** > **Groups**.

      The **Workload management group** view opens, listing any workload groups already created in PLXPROD1.
   b) Click **Create** and type in the following information:

      Workload management group name
      WLGBRI01

      Description
      Workload Group for WLDBRI01
   c) Click **Yes** to confirm.

      The **Workload management group** view is redisplayed.
7. Add the workload definition WLDBRI01 and any other workload definitions you have created to WLGBRI01.
   a) From the main menu, click **Administration** > **Workload manager administration** > **Definitions**.

      The **Workload management definition** view opens.
   b) Select the entry for WLDBRI01, click **Add to WLM group**, and provide the following information:

      Workload management definition name
      WLDPAY01

Description
Link3270 bridge workload definition

Resource group name
WLGBRI01

c) Click **Yes** to confirm.

d) Repeat for any additional workload definitions that you created earlier.

8. Add workload group WLGBRI01 to workload specification BRSPEC01.

   a) From the main menu, click **Administration** > **Workload manager administration** > **Groups**.

      The **Workload management (WLM) group** view opens.

   b) Select the entry for WLGBRI01, click **Add to WLM specification**, and type `BRSPEC01` in the **Specification name** field.

   c) Click **Yes** to confirm.

9. Install definition into the active workload.

   a) From the main menu, click **Administration** > **Workload manager administration** > **Definitions**.

      The **Workload management definition** view opens.

   b) Select the entry for WLDBRI01, and click **Install** to install it into the active workload. Click **Yes** to confirm.

      The workload separation definitions you created take effect immediately.

   c) Click **Yes** to confirm.

What to do next

You can check that the workload definition is active using the **Workload definition installed in active workload** view.

Chapter 5. Active workload views

The active workload views display information about active workloads and the transactions and transaction groups comprising those workloads. Note: Workload management is performed on a CICSplex-wide basis. These views ignore any scope that may be in effect.

Active workloads - WLMAWORK

The **Active workloads** (WLMAWORK) views display information about an active workload within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Active workloads**

*Table 8. Views in the supplied **Active workloads** (WLMAWORK) view set*

View	Notes
Active workloads EYUSTARTWLMAWORK.DETAILED	Detailed information about a selected active workload.
Active workloads EYUSTARTWLMAWORK.SET	Set active workload attributes.
Active workloads EYUSTARTWLMAWORK.TABULAR	Tabular information about all active workloads within the current context.

Actions

Table 9. Actions available for WLMAWORK views

Action	Description
GET	Optional parameter. YES requests records from every CMAS managing the workload. NO requests records from a single CMAS managing the workload. NO is the default value if the parameter is not specified.
SET	Set active workload attributes.

Fields

Table 10. Fields in WLMAWORK views

Field	Attribute name	Description
Acceptable level of abend probability	ABENDCRIT	The abend probability for a transaction associated with the default transaction group that should cause a target region to be considered unhealthy. A value of 0 means WLM is not calculating abend probabilities for the workload.

Table 10. Fields in WLMAWORK views (continued)

Field	Attribute name	Description
Acceptable target region load level	ABENDTHRESH	The abend probability for a transaction associated with the default transaction group that should cause a target region's load level to be doubled. A value of 0 means WLM is not calculating abend loads for the workload.
Automatic affinity creation	AFFAUTO	Indicates whether CICSPlex SM is to automatically create an affinity relationship for transactions that are not associated with any installed transaction group. • YES – An affinity is created using the values in the Affinity Relation and Affinity Lifetime fields. • NO – An affinity is not automatically created (but can be created by a customized version of the dynamic routing program EYU9WRAM). • N/A – There are no values in the **Affinity Relation** and **Affinity Lifetime** fields, therefore, no affinity is created.
Default affinity relation	AFFINITY	The default affinity relation applied to transactions not associated with any installed transaction group. The affinity relation values are: • GLOBAL - All users at all terminals • LUNAME - Terminal logical unit name • USERID - User ID • BAPPL - Business Application • Locked - LOCKED affinities can arise only between dynamically linked programs. A LOCKED affinity is created when a called program retains state data that is to be preserved after returning to its caller. Programs with this type of affinity are routed to the same target region until end of unit of work occurs. LOCKED can be used only for dynamic program link (DPL) requests with an associated affinity lifetime of UOW. • NONE - No affinity relation was defined
Default affinity lifetime	AFFLIFE	The default affinity lifetime used with the default affinity relation. This value is applied to transactions that are not associated with any installed transaction group. After the first instance of a transaction, subsequent instances of the transaction are run on the same target region: • DELIMIT - Until the pseudoconversation mode is END • LOGON - For the duration of the terminal session • PCONV - For the duration of the pseudoconversation • PERMANENT - For as long as the workload is active • SIGNON - Until the terminal user signs off • SYSTEM - Until the target region terminates • ACTIVITY - For as long as the CICS BTS activity is active • PROCESS - For as long as the CICS BTS process is active • UOW - For as long as the unit-of-work is active • NONE - No affinity lifetime was defined

Table 10. Fields in WLMAWORK views (continued)

Field	Attribute name	Description
Default algorithm type	ALGTYPE	The algorithm to be applied to all transactions that are not associated with any installed transaction group: • **QUEUE** - Routes transactions to the target region that has the lowest routing weight at the time of the routing evaluation. The routing weight of a target region is a combination of factors that include: – The current task load – The current target health state – The link speed from the router to the target – The probability of transaction abend at the target – The effect of any outstanding RTA events at the target • **LNQUEUE** - Routes transactions to the target region that has the lowest routing weight at the time of the routing evaluation. The routing weight of a target region is a combination of factors that include: – The current task load – The current target health state – The effect of any outstanding RTA events at the target – The probability of transaction abend at the target **Note** - The link speed from the router to the target is not factored into the routing weight calculation for the **LNQUEUE** algorithm • **GOAL** - Routes transactions to the target region which is best able to meet the transaction's average response time goal, as predefined using z/OS Workload Manager. – If a specific target region cannot be resolved from the evaluated GOAL set, then the **QUEUE** algorithm is applied to the remaining target set. • **LNGOAL** - Routes transactions to the target region which is best able to meet the transaction's average response time goal, as predefined using z/OS Workload Manager. – If a specific target region cannot be resolved from the evaluated LNGOAL set, then the **LNQUEUE** algorithm is applied to the remaining target set.
Active target regions	AORCNT	The number of active CICS systems, identified as target region for the workload.
Default target scope	AORSCOPE	The name of the CICS system or CICS system group that is to be the target for any dynamic transactions or programs not associated with a transaction or program group.
Description	DESC	The description of the workload.
Event name	EVENTNAME	The name of a real-time analysis event that will affect the routing of transactions in this workload. If a real-time analysis event is generated, WLM uses the information as part of the target selection criteria. An event can optionally be associated with a workload when the workload specification is created. If this field is blank, no event is associated with the workload. The name of an event is the same as an installed real-time analysis definition (RTADEF) or status definition (STATDEF)

Table 10. Fields in WLMAWORK views (continued)

Field	Attribute name	Description
Optimization status	OWSTATE	This field reports the sysplex optimization status of the current workload. The value is an amalgamation of all of the workload router optimization statuses, and all of the workload target optimization statuses. The possible values are: • ACTIVE – All targets and routers in the workload are executing in optimized workload state. • PARTIAL – At least one target and one router are executing in optimized workload mode. Use the Active routing regions and Active target regions hyperlinks to determine which regions are not running in an optimized state. • INACTIVE – The workload is not running in an optimized state, for one or more of the following reasons: - No routing regions in the workload are running in an optimized state - No target regions in the workload are running in an optimized state - No regions in the workload are running in an optimized state - The optimised workload routing function is DISABLED for the current target region - The workload was designated as being non optimized by specifying a value of 0 for the RS server update frequency on the CICSplex definition or CICS system definition for this workload.
Reporting CMAS	RPTCMAS	The name of the CMAS that provided the information in this record. If the GET operation optional parameter EXPLODE is not specified or is specified as NO, then this attribute will be set to blanks and only one record will be returned for the workload. If the GET operation optional parameter EXPLODE is specified as YES, then this attribute will be set with the name of the CMAS that provided the information in this record. A separate record will be returned for the workload from each CMAS that manages the workload. When the IBM distributed WUI menus and views are used to request the WLMAWORK tabular view, the EXPLODE parameter will be set to YES.
Shared status	SHARED	Indicates whether the workload has been shared with a pre-CICS TS 1.3 CMAS. If it has, the installation of CICS BTS related definitions are inhibited.

Table 10. Fields in WLMAWORK views (continued)

Field	Attribute name	Description
Workload status	STATUS	The status of the workload: • ACTIVE - All the target and routing region connections that this workload depends upon are available • FROZEN - A target or routing region connection that this workload depends upon has been lost As long as a workload remains frozen: • Transactions will continue to be routed according to any existing affinities • New affinities can be created as long as they do not involve other CMASs. Global or permanent affinities, which require other CMASs to be notified, cannot be created • Install and discard action commands are not allowed for any aspect of the workload The cause of a lost connection could be the target or routing region, one of the CMASs that manage the workload or a connection between the CMASs. To determine why a workload is frozen: • Check the contact status in the Target region in active workload (WLMAWAOR) and Active workload routing regions (WLMAWTOR) views to determine which target or routing region is affected • Check the MAS view to determine the status of the target or routing region • Check the status of all connections for the target or routing region Determine which connection(s) were lost and take appropriate action to re-establish the connections.
Active routing regions	TORCNT	The number of active CICS systems, identified as routing regions, that are associated with the workload.
Active transaction group affinities	WAFFCNT	The number of active transaction group affinities that are associated with the workload.
Active workload definitions	WDEFCNT	The number of active workload definitions that are associated with the workload.
Name	WORKLOAD	The name of the workload. This is also the name of the installed workload specification definition.
Owning system ID	WRKLOWNER	The CICS system ID of the CMAS that created the workload.
Active transaction groups	WTGPCNT	The number of active transactions groups that are associated with the workload.
Active dynamic transactions	WTRNCNT	The number of active dynamic transactions that are associated with the workload.

Active routing regions - WLMAWTOR

The **Routing regions in an active workload** (WLMAWTOR) views display information about all active routing regions that are associated with a workload that is within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Active routing regions**

*Table 11. Views in the supplied **Active workload routing regions** (WLMAWTOR) view set*

View	Notes
Active workload routing regions EYUSTARTWLMAWTOR.DETAILED	Detailed information about a selected active routing region.
Active workload routing regions EYUSTARTWLMAWTOR.TABULAR	Tabular information about all active routing regions that are associated with a workload.

Actions

Table 12. Actions available for WLMAWTOR views

Action	Description
GET	The name of the workload specification. You can specify a workload name of '*', which returns details of all routing regions in all workloads. No other type of generic workload name may be specified.

Fields

Table 13. Fields in WLMAWTOR views

Field	Attribute name	Description
Routing region contact status	CON_STATUS	The status of the connection between the routing region and the CMAS. Values are: • LOSTCON - The connection between the routing region and the CMASs has been lost. • N_A - The connection between the routing region and the CMASes managing the workload is available.
Optimization status	OWSTATE	This field reports the status of the current routing region for the optimized workload routing function. The possible values are: • ACTIVE – This router region is executing in an optimized workload state. • INACTIVE – This region is capable of executing in an optimized workload state, however, is not currently optimized for one or more of the following reasons: - The region has no connection to an RS Server - The region is connected to an RS Server, however, the server has not been able to connect to the coupling facility - The optimised workload routing function is DISABLED for the current target region - The RS server update frequency value for this region is 0, which means that the optimization capabilities for this region are not enabled • N_A – The router region is not configured to provide optimized workload management. Only non optimized WLM routing decisions can be made. If the CMAS that your WUI is using is not connected to a Region Status (RS) server, the optimization status for remote regions will not be updated and defaults to INACTIVE. Where this situation occurs, either connect all reporting CMASs to the RS server or use the Active target regions view to check the optimization state.

Table 13. Fields in WLMAWTOR views (continued)

Field	Attribute name	Description
Reporting CMAS	RPTCMAS	The name of the CMAS that provided the information in this record. If the GET operation optional parameter EXPLODE is not specified or is specified as NO, then this attribute will be set to blanks and only one record will be returned for the routing region. If the GET operation optional parameter EXPLODE is specified as YES, then this attribute will be set with the name of the CMAS that provided the information in this record. A separate record will be returned for the routing region from each CMAS that manages the workload associated with the routing region. When the IBM distributed WUI menus and views are used to request the WLMAWTOR tabular view, the EXPLODE parameter will be set to YES.
Routing region name	TOR	The name of an active CICS system, acting as a routing region, to which the workload is associated.
Name of controlling CMAS	TOROWNER	The name of the controlling CMAS.
Workload name	WORKLOAD	The name of the workload specification.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Active workload target distribution factors - WLMAWAOR

The **Target regions in an active workload** (WLMAWAOR) views display information about all target regions that are associated with a workload that is within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Active workload target distribution factors**

*Table 14. Views in the supplied **Target region in active workload** (WLMAWAOR) view set*

View	Notes
Target region in active workload EYUSTARTWLMAWAOR.ACTIVATE	Cause the specified target region to be recognized so that it can participate in workload management.
Target region in active workload EYUSTARTWLMAWAOR.DETAILED	Detailed information about a selected target region.
Target region in active workload EYUSTARTWLMAWAOR.QUIESCE	Cause the specified target region to be removed from workload management. This means that no further transactions are routed to the target region unless they share an affinity with earlier transactions. For example, if you are engaged in a pseudoconversation when a target region begins quiescing, the transactions that constitute the pseudoconversation continue to be routed to the same target region until the end of the affinity lifetime. When the affinity is no longer active, the target region is fully quiesced and subsequent transactions are routed to a different target region in the target scope.
Target region in active workload EYUSTARTWLMAWAOR.TABULAR	Tabular information about target regions that are associated with a workload that is within the CICSplex identified as the context.
Target region in active workload EYUSTARTWLMAWAOR.TABULAR2	Tabular information about target regions that are associated with a workload that is within the CICSplex identified as the context.

Actions

Table 15. Actions available for WLMAWAOR views

Action	Description
ACTIVATE	Cause the specified target region to be recognized so that it can participate in workload management.
QUIESCE	Cause the specified target region to be removed from workload management. This means that no further transactions are routed to the target region unless they share an affinity with earlier transactions. For example, if you are engaged in a pseudoconversation when a target region begins quiescing, the transactions that constitute the pseudoconversation continue to be routed to the same target region until the end of the affinity lifetime. When the affinity is no longer active, the target region is fully quiesced and subsequent transactions are routed to a different target region in the target scope.

Fields

Table 16. Fields in WLMAWAOR views

Field	Attribute name	Description
Target region name	AOR	The name of an active CICS system, acting as a target region, to which the workload is associated.
Coupling Facility update count	CFUPDCNT	This is the number of update calls that the RS Server has made to the z/OS Coupling Facility for this target region. This value is reset to zero whenever the RS Server or the CICS region is started.
Target region contact status	CON_STATUS	The status of the connection between the target region and the CMAS. Values are: • LOSTCON - The connection between the target region and the CMASs has been lost. • N_A - The connection between the target region and the CMASes managing the workload is available.
CICSPlex SM version	CPSMVER	This is the CICSPlex SM version of the target region.
RTA event active	EVENTS	This is the indicator of an outstanding RTA event for the current region. It is associated with either the workload specification or if the Transaction name (TRANNAME) parameter is specified and does not have a value of '*', then the TRANGRP in the current workload which manages that specific transaction. This value may vary for each workload that targets the current region, and will cause corresponding variations of the routing weight for that region.
Region is dumping	HLTHDUMP	This indicates whether the routing target is currently issuing a transaction or system dump.
Region is at MAXTASKS	HLTHMAXT	This indicates whether the routing target is currently running at a task rate that has reached its maximum tasks threshold.
MAS non responsive	HLTHNRM	This field indicates the routing target is non responsive at the current time.
Short on storage	HLTHSOS	This field indicates the SOS status of this routing target at the current time. Note that a CICS region can change from being in and out of SOS status very rapidly.
Region has stalled	HLTHSTALL	This indicates whether the routing target is in a stall condition. If the region is stalled, use the Active task views to determine which tasks are suspended, and determine why. Note: Stall here means a task has been waiting for the same resource for a specific time.
Maximum task count for region	MAXTASKS	This is the defined maximum number of active tasks that may concurrently execute in this routing target.

Table 16. Fields in WLMAWAOR views (continued)

Field	Attribute name	Description
Optimization status	OWSTATE	This field reports the status of the current target region for the optimized workload routing function. The possible values are: • ACTIVE – This target region is executing in an optimized workload state. • INACTIVE – This region is capable of executing in an optimized workload state, however, it is not currently optimized for one or more of the following reasons: - The region has no connection to an RS server - The region is connected to an RS server, however, the server has not been able to connect to the z/OS coupling facility - The optimised workload routing function is DISABLED for the current target region - The RS server update frequency value for this region is 0, which means that the optimization capabilities for this region are not enabled • N_A – The target region is not configured to provide optimized workload management. Only non optimized WLM routing decisions can be made. If the CMAS that your WUI is using is not connected to a Region Status (RS) server, the optimization status for remote regions will not be updated and defaults to INACTIVE. Where this situation occurs, either connect all reporting CMASs to the RS server or use the Active target regions view to check the optimization state.
RS server read interval	READRS	Within a workload running in optimized mode, this value specifies the minimum interval between refreshes of a target region status from a CICS CFDT Server. These refresh requests will be issued by a routing region that is evaluating this region as a possible target for a dynamic routing request. The value range is from 0 to 2000, and represents units of milliseconds: • When the Optimization status is not set to Active, then this value will be set to 0, and WLM will ignore it. When the Optimization status is set to Active, then a value of 0 means that a routing region will request a status update of a target region on every occasion that it examines this region's status. • Values between 1 and 2000 specify the minimum time interval that must expire before the status of this region can be refreshed. A low interval value means that the CFDT Server will be polled more often for a status update, than for a higher value. For workloads in QUEUE mode, this will result in a task load more evenly distributed across the CICS regions in the workload target scope **(assuming all other health and link factors are equal)** . However, the utilization of the RS server will be correspondingly increased, which may consequently result in a higher utilization of your z/OS Coupling Facility. A value of 1000 represents a one second interval. The default value is 200 milliseconds This value may be modified for this specific region using the MASs known to CICSplex views, or the CICS system definition views.

Table 16. Fields in WLMAWAOR views (continued)

Field	Attribute name	Description
WLM routing weight for region	ROUTEWGHT	This is the calculated routing weight for the current routing target according to a specified transaction name as specified in the Transaction name (TRANNAME) parameter. If the transaction name has a value of '*', the routing weight is that of the default route suggested by the workload specification. This value will be combined with the link weight from a Router to determine an overall normalized routing weight for this CICS region. This value will be compared with the same value for other potential routing targets. Assuming no affinities are outstanding, then the target with the lowest routing weight will be chosen as the actual route target. If the workload associated with this target region specifies that RTA MRM events should influence a route decision for certain transaction codes, then different transaction names will result in different routing weights because of the effects of outstanding RTA events. Note: 2147483647 is a special value indicating that the target is quiesced or z/OS WLM health is 0. Target regions with this value have been removed from workload management. This means that no further transactions are routed to the target region. Generally speaking, the Link weight is used to prefer faster connections over slower ones. Weighting is as follows; • MRO is considered faster than XCF (cross coupling-facility) connections. • XCF is considered faster than IPIC. • IPIC is considered faster than VTAM. • VTAM is considered faster than indirect connections.
Routing Load	ROUTINGLOAD	This is the routing load used by WLM route decision making for the target region. Note that this will not match the real CICS task count at the time of this query. For an ACTIVE System Optimized WLM target region this will be the approximate task load in the target region, taking into account 'Task Count Increment Value', 'RS Server Bottom Tier' and 'RS Server Top Tier'. The accuracy of this value is also affected by the RS Read Interval and RS Update frequency settings. For target regions that are not connected to a RS Server, and are therefore in a non-optimized WLM state, this value will appear to be less accurate, when compared to the actual task count. This is because this is the normalized routing load for this region. It should be noted that when running in a non-optimized WLM state, this value can exceed the MAX TASK value, without the MAX TASK flag being set on, and is correct behaviour as the MAX TASK indicator is related to the target region's health status taken from actual task counts. It is possible that a region could be the target for multiple workloads and show different routing weight values. This is not abnormal behaviour - it is a symptom of the synchronization latencies within workload manager. Workload optimization will nullify this effect. The workload manager runtime processes will continue to execute normally in spite of this disparity in the load values. To suppress these observed load differences, try to ensure that all MASes and CMASes that participate in the workload have a connectable region status server available. Where a workload contains target regions with different CICS TS releases, differences in the default number of active CICSPlex SM tasks in a CICS region when no other tasks are running may cause changes in the distribution of work to the target regions. Target regions at CICS TS V5.4 will have 0 CICSPlex SM tasks by default. This is compared to target regions at CICS TS V5.3 or earlier, which can have anywhere between 3 and 19 CICSPlex SM tasks active by default. This may lead to target regions earlier than the CICS TS V5.4 release being treated as less desirable by the routing algorithm.

Table 16. Fields in WLMAWAOR views (continued)

Field	Attribute name	Description
Reporting CMAS	RPTCMAS	The name of the CMAS that provided the information in this record. If the GET operation optional parameter EXPLODE is not specified or is specified as NO, then this attribute will be set to blanks and only one record will be returned for the target region. If the GET operation optional parameter EXPLODE is specified as YES, then this attribute will be set with the name of the CMAS that provided the information in this record. A separate record will be returned for the target region from each CMAS that manages the workload associated with the target region. When the IBM distributed WUI menus and views are used to request the WLMAWAOR tabular view, the EXPLODE parameter will be set to YES.
RS server pool name	RSPOOLID	For sysplex optimized workloads, region status data is maintained within a coupling facility data table. That table will be contained within a CFDT pool identified by this attribute. For an optimized workload to activate, there must be an active Region Status server which manages data for the pool name identified here. If your CICSplex identifiers are not unique within your sysplex, you must ensure that the RS server pool names are **unique.** If your Sysplex comprises unique CICSplex identifiers, then they may all share the same RS server pool. This attribute is defined in the parent CICSplex definition of the current target region. The default name is DFHRSTAT. You may choose to employ an existing CFDT pool for containing your CICSplex data tables. If you do, be aware that the throughput of your optimized workloads may be impeded by any user application activity to the specified pool name. Likewise, any application throughput to the pool may be impacted by sysplex optimized workloads. It is recommended that a discrete RS server and pool name is defined for the optimized workload function.
Target region status	STATUS	The current status of the target region associated with the workload, as one of the following: • ACTIVE - The target is available to participate in workload management. • QUIESCEING - The target is being quiesced. No new transactions are routed to this target. Transactions currently being routed to this target are routed to another target, unless an affinity relation exists. If there is an affinity relation, transactions continue to be routed to this target until the affinity lifetime expires. Note that if a target has an affinity lifetime of PERMANENT, it will remain in a QUIESCING state indefinitely. • QUIESCED - The target is not available to participate in workload management. Input Values: ACTIVE | QUIESCE
Task count increment value	TASKINC	This is the translation of the RS Server update frequency value to an actual task count value for the current target region. Each time the current task count reaches a boundary that is a factor of this value, then an update is sent to the RS server.
Task load percentage	TASKLOAD	This is the normalized task load for this target region. It is calculated from dividing the current task routing count by the MAXTASKS value for the region, and multiplying by 100.

Table 16. Fields in WLMAWAOR views (continued)

Field	Attribute name	Description
RS Server update frequency	UPDATERS	Within a workload running in optimized mode, this value specifies the frequency with which the CICS CFDT (RS) server will be called to modify the value of the task load within this target region. The value range is from 0 to 25: • When the Optimization status is not set to Active, then this value will be set to 0, and WLM will ignore it. When the Optimization status is set to Active, then a value of 0 means that the RS Server is not notified of any task load count changes, which disables the optimized workload function for this target region. • Values between 1 and 25 are applied as an arithmetic percentage to a region's MAXTASKS setting. The resultant task count value is used as a numeric threshold to drive an update call to the RS Server. **For example** , with a MAXTASKS setting of 120, and with this attribute set to 20, the RS Server will be called to update the WLM load count when the regions task count changes between: • 23 and 24 tasks - (20%), • 47 and 48 tasks - (40%), • 71 and 72 tasks - (60%), • 95 and 96 tasks - (80%), • 119 and 120 tasks - (100%). The RS Server would be updated when the task load increments or decrements across these boundaries. If a value is specified that is at the lower end of the 1-25 scale, then that will cause an increase in the frequency of updates to the RS Server across the task load range. For workloads in QUEUE mode, this will result in a task load more evenly distributed across the CICS regions in the workload target scope **(assuming all other health and link factors are equal)** . However, the utilization of the RS server will be correspondingly increased, which may consequently result in a higher utilization of your z/OS Coupling Facility. This value may be modified for this specific region using the MASs known to CICSplex views, or the CICS system definition views.
z/OS WLM health	WLMHLTH	The percentage value that represents the z/OS WLM health open state for the region. This value may influence the amount of work a CPSM WLM routing region will send to the region. A value of zero (0) means that no work will be routed to the region. Any workload affinities associated with this region will remain and be honored, but routes based upon those affinities will fail while the value is zero. Note that the route failure seen depends upon how the route is attempted and whether a custom WRAM program is in effect. Some users may receive message EYUWR0003W indicating that the affinity AOR is not available, while others may receive a SYSIDERR. A value of 1 to 99 inclusive means that an additional weight will be added to the region that is inversely proportional to the WLMHLTH value. That is, the lower the WLMHLTH value, the greater the additional weight assigned. This may have the effect of increasing or decreasing workload being received by the region as its WLMHLTH value increases or decreases, depending upon the weight assigned to other regions within the AORSCOPE. A value of 100 means either that a region has now achieved a fully open z/OS WLM health open state, or that the region is running with WLMHEALTH=OFF in the SIT, or that the region is running CICS TS V5.3 or lower. A region with a value of 100 is treated as healthy and no additional weight is added for the region.

Table 16. Fields in WLMAWAOR views (continued)

Field	Attribute name	Description
Task load queue mode	WLMQMODE	This attribute specifies how the queued task load of a target CICS region is to be evaluated: • **MAXTASK** - specifies that both active and MAXTASK queued tasks are to be included in the task load evaluation for the region. • **ALL** - specifies that the task load evaluation for the region will include active tasks, tasks queued for the MAXTASK limit **and** tasks that are queued because of a TRANCLASS limit. • **N_a** - specifies that the current region is not at a high enough level to support the use of this attribute. You will not be able to change it, and the WLM route decision process will ignore it. The default value is **ALL.** This value is specified within the CICSPlex SM CICS system definition object. If you want this value to be changed, you must modify the system definition for the current region and then restart it. You may change the value in an active CICS region by using the CICSPlex SM MAS object. When the region restarts it will revert to the value specified in the system definition.
Task load health threshold	WLMTHRSH	This specifies a percentage threshold of the current region's task load, which is calculated by dividing the current task count by the maximum task count. When the load for a target region reaches this threshold, then WLM considers the region to be relatively unhealthy. This will cause higher link weights to be applied to the WLM routing algorithm when evaluating this region. When a target scope covers local and remote regions relative to the router, then WLM will favour local targets over remote ones. The effect of this attribute is that when this load threshold in the local regions is achieved, then WLM will start to favour remote targets instead. When all targets in the scope achieve this load threshold, then WLM will revert to favouring local targets again. The value range is from 1 to 100, and the default value is 60. This value is specified within the CICSPlex SM CICS System Definition object. If you want this value to be changed, you must modify the system definition for the current region and then restart it. You may change the value in an active CICS region by using the CICSPlex SM MAS object. When the region restarts it will revert to the value specified in the system definition. If this value is set to 0, then the current region is not at a high enough level to support the use of this attribute. You will not be able to change it, and the WLM route decision process will ignore it. Note: this value is nullified when applied to the routing factor of link neutral dynamic routing requests. This is because the link weight itself is ignored for the LNQUEUE and LNGOAL algorithms.
Workload name	WORKLOAD	The name of the workload specification.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Active target regions - WLMATARG

The **Target regions in an active workload** (WLMATARG) views display information about all target regions that are associated with workloads within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Active target regions**

*Table 17. Views in the supplied **Active target region** (WLMATARG) view set*

View	Notes
Active target region EYUSTARTWLMATARG.DETAILED	Detailed information about a selected target region.
Active target region EYUSTARTWLMATARG.RESET	Causes the Dynamic Routing statistics for the current target region to be reset to zero. After resetting the statistics, for performance and accuracy purposes, it is recommended that the refresh button on this viewset is used to reflect the changes to the statistics.
Active target region EYUSTARTWLMATARG.TABULAR	Tabular information about target regions that are associated with a workload that is within the CICSplex identified as the context.
Active target region EYUSTARTWLMATARG.TABULAR2	Tabular information about target regions that are associated with a workload that is within the CICSplex identified as the context. Routing statistics can only be collected in a region connected to a CICS TS V4.1 CMAS (or greater). For targets or routers that are connected to a pre-CICS TS V4.1 CMAS, some statistics may appear as zero.
Active target region EYUSTARTWLMATARG.TABULAR3	Tabular information about target regions that are associated with a workload that is within the CICSplex identified as the context. Routing statistics can only be collected in a region connected to a CICS TS V4.1 CMAS (or greater). For targets or routers that are connected to a pre-CICS TS V4.1 CMAS, some statistics may appear as zero.

Actions

Table 18. Actions available for WLMATARG views

Action	Description
RESET	Causes the Dynamic Routing statistics for the current target region to be reset to zero. After resetting the statistics, for performance and accuracy purposes, it is recommended that the refresh button on this viewset is used to reflect the changes to the statistics.

Fields

Table 19. Fields in WLMATARG views

Field	Attribute name	Description
Target region name	AOR	The name of an active CICS system, acting as a WLM target region.
RS server bottom tier	BOTRSUPD	This is the bottom tier task load range (from zero up to this value). The value range is from 1 to 25. When a region's task load falls within this range, then the task load will be broadcast to the coupling facility for every change in the task load. Once the load reaches this value, then the 'RS server update frequency' task rules will be activated. The value shown here at the CICSplex level may be overridden at the CICS definition level to allow fine tuning of the value on an individual CICS region basis.
Coupling Facility read count	CFREADCT	This is the number of read calls that have been made by routing regions to the z/OS Coupling Facility for this region's status.
Coupling Facility update count	CFUPDCNT	This is the number of update calls that the RS Server has made to the coupling facility for this target region. This value is reset to zero whenever the RS Server or the CICS region is started.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
Target region contact status	CON_STATUS	The status of the connection between the target region and the CMAS. Values are: • LOSTCON - The connection between the target region and the CMASs has been lost. • N_A - The connection between the target region and the CMASes managing the workload is available.
CICSPlex SM version	CPSMVER	This is the CICSPlex SM version of the target region.
RTA event count for target region	EVENTS	This is the count of all outstanding RTA events that are associated with the current region. This value is set to zero (0) if the reporting CMAS is not the owning CMAS.
Region is dumping	HLTHDUMP	This indicates whether the routing target is currently issuing a transaction or system dump.
Region is at MAXTASKS	HLTHMAXT	This indicates whether the routing target is currently running at a task rate that has reached its maximum tasks threshold.
MAS non responsive	HLTHNRM	This field indicates the routing target is non responsive at the current time.
Short on storage	HLTHSOS	This field indicates the SOS status of this routing target at the current time. Note that a CICS region can change from being in and out of SOS status very rapidly.
Region has stalled	HLTHSTALL	This indicates whether the routing target is in a stall condition. If the region is stalled, use the Active task views to determine which tasks are suspended, and determine why. Note: Stall here means a task has been waiting for the same resource for a specific time.
Maximum task count for region	MAXTASKS	This is the defined maximum number of active tasks that may concurrently execute in this routing target.
Owning CMAS name	OWNINGCMAS	The name of a CMAS that participates in the management of the target region.
Optimization status	OWSTATE	This field reports the status of the current target region for the optimized workload routing function. The possible values are: • ACTIVE – This target region is executing in an optimized workload state. • INACTIVE – This region is capable of executing in an optimized workload state, however, it is not currently optimized for one or more of the following reasons: - The region has no connection to an RS server - The region is connected to an RS server, however, the server has not been able to connect to the z/OS coupling facility - The optimised workload routing function is DISABLED for the current target region facility - The RS server update frequency value for this region is 0, which means that the optimization capabilities for this region are not enabled • N_A – The target region is not configured to provide optimized workload management. Only non optimized WLM routing decisions can be made. This value is set to INACTIVE if the reporting CMAS is not connected to a Region Status (RS) server.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
RS server read interval	READRS	Within a region running in optimized mode, this value specifies the minimum interval between refreshes of a target region status from a CICS CFDT Server. These refresh requests will be issued by a routing region that is evaluating this region as a possible target for a dynamic routing request. The value range is from 0 to 2000, and represents units of milliseconds: • When the Optimization status is not set to 'Active', then this value will be set to 0, and WLM will ignore it. When the Optimization status is set to 'Active', then a value of 0 means that a routing region will request a status update of a target region on every occasion that it examines this region's status. • Values between 1 and 2000 specify the minimum time interval that must expire before the status of this region can be refreshed. A low interval value means that the CFDT Server will be polled more often for a status update, than for a higher value. For workloads in QUEUE mode, this will result in a task load more evenly balanced across the CICS regions in the workload target scope **(assuming all other health and link factors are equal)** . However, the utilization of the RS server will be correspondingly increased, which may consequently result in a higher utilization of your z/OS Coupling Facility. A value of 1000 represents a one second interval. The default value is 200 milliseconds This value may be modified for this specific region using the 'MASs known to CICSplex' views, or the 'CICS system definition' views.
WLM routing weight for region	ROUTEWGHT	This is the calculated routing weight for the current routing target without accounting for outstanding RTA MRM events, router region link speeds and transaction abend probabilities. Only the transaction load and health state of the region are calculated into this value. The health state is a combination of the SOS, Dump, Stall, MAXTASK and z/OS WLM health conditions. The transaction load is the value indicated by the 'Task load percentage' attribute. No workload associated parameters are included in this weight value. To determine routing weights for specific workloads, then use the corresponding attribute returned in the WLMAWAOR 'Target region in active workload' views. When no affinities are applicable to a dynamic routing request, then the routing weights of all target regions in the routing scope are compared and the region with the lowest weight value will be the one selected. If several regions have the same lowest weight value, then one of that set is randomly selected as the routing target. Note: 2147483647 is a special value indicating that the target is quiesced or z/OS WLM health is 0. Target regions with this value have been removed from workload management. This means that no further transactions are routed to the target region.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
Routing Load	ROUTINGLOAD	This is the routing load used by WLM route decision making for the target region. Note that this will not match the real CICS task count at the time of this query. For an ACTIVE System Optimized WLM target region this will be the approximate task load in the target region, taking into account 'Task Count Increment Value', 'RS Server Bottom Tier' and 'RS Server Top Tier'. The accuracy of this value is also affected by the RS Read Interval and RS Update frequency settings. For target regions that are not connected to a RS Server, and are therefore in a non-optimized WLM state, this value will appear to be less accurate, when compared to the actual task count. This is because this is the normalized routing load for this region. It should be noted that when running in a non-optimized WLM state, this value can exceed the MAX TASK value, without the MAX TASK flag being set on, and is correct behaviour as the MAX TASK indicator is related to the target region's health status taken from actual task counts. It is possible that a region could be the target for multiple workloads and show different routing weight values. This is not abnormal behaviour - it is a symptom of the synchronization latencies within workload manager. Workload optimization will nullify this effect. The workload manager runtime processes will continue to execute normally in spite of this disparity in the load values. To suppress these observed load differences, try to ensure that all MASes and CMASes that participate in the workload have a connectable region status server available. Where a workload contains target regions with different CICS TS releases, differences in the default number of active CICSPlex SM tasks in a CICS region when no other tasks are running may cause changes in the distribution of work to the target regions. Target regions at CICS TS V5.4 will have 0 CICSPlex SM tasks by default. This is compared to target regions at CICS TS V5.3 or earlier, which can have anywhere between 3 and 19 CICSPlex SM tasks active by default. This may lead to target regions earlier than the CICS TS V5.4 release being treated as less desirable by the routing algorithm.
Reporting CMAS name	RPTINGCMAS	The name of a CMAS that provided this record.
RS server pool name	RSPOOLID	For sysplex optimized workloads, region status data is maintained within a coupling facility data table. That table will be contained within a CFDT pool identified by this attribute. For an optimized workload to activate, there must be an active Region Status server which manages data for the pool name identified here. If your CICSplex identifiers are not unique within your sysplex, you must ensure that the RS server pool names are **unique.** If your Sysplex comprises unique CICSplex identifiers, then they may all share the same RS server pool. This attribute is defined in the parent CICSplex definition of the current target region. The default name is DFHRSTAT. You may choose to employ an existing CFDT pool for containing your CICSplex data tables. If you do, be aware that the throughput of your optimized workloads may be impeded by any user application activity to the specified pool name. Likewise, any application throughput to the pool may be impacted by sysplex optimized workloads. It is recommended that a discrete RS server and pool name is defined for the optimized workload function.
Route abends	RTABEND	The number of times that route abend has been returned after selecting this region. This value is the total for all requests processed by routing regions (dynamic routing) and all requests processed by the target region (distributed routing) while they were connected to the reporting CMAS.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
Route completes	RTCOMPLETE	The number of times that route complete has been driven for this region. Route complete occurs after the transaction request is dispatched to the waiting queue of the target region. This value is the total for all requests processed by routing regions while they were connected to the reporting CMAS.
Route errors	RTERROR	The number of times that the selection of the current region has caused a route error to occur. Route errors occur when CICSPlex SM has selected a target region, and CICS has experienced an error in routing the work to it. For example, the connection is out of service, or no sessions are immediately available for an LU 6.2 connection. CICSPlex SM will attempt to select another target when this occurs. This value is the total for all requests processed by routing regions while they were connected to the reporting CMAS.
Route initiates	RTINITIATE	The number of times that routed transactions have been actually started on the target region. This value is the total for all requests processed by the target region while it was connected to the reporting CMAS.
Route notifies	RTNOTIFY	The number of times that route notify has been driven for this region. Route notify normally occurs when a static routing starts. This value is the total for all requests processed by routing regions while they were connected to the reporting CMAS.
Route selections	RTSELECT	The number of times that the current region has been chosen as a routing target by route selection processing. This value is the total for all requests processed by routing regions while they were connected to the reporting CMAS.
Route terminates	RTTERMINATE	The number of times that route terminate has been driven for this region. Route terminate occurs when a transaction finishes on a target region. This value is the total for all requests processed by routing regions (dynamic routing) and all requests processed by the target region (distributed routing) while they were connected to the reporting CMAS.
Target region status	STATUS	The current status of the target region associated with the workload, as one of the following: • ACTIVE - The target is available to participate in workload management. • QUIESCEING - The target is being quiesced. No new transactions are routed to this target. Transactions currently being routed to this target are routed to another target, unless an affinity relation exists. If there is an affinity relation, transactions continue to be routed to this target until the affinity lifetime expires. Note that if a target has an affinity lifetime of PERMANENT, it will remain in a QUIESCING state indefinitely. • QUIESCED - The target is not available to participate in workload management. Input Values: ACTIVE | QUIESCE
Task count increment value	TASKINC	This is the translation of the RS Server update frequency value to an actual task count value for the current target region. Each time the current task count reaches a boundary that is a factor of this value, then an update is sent to the RS server.
Task load percentage	TASKLOAD	This is the normalized task load for this target region. It is calculated from dividing the current task routing count by the MAXTASKS value for the region, and multiplying by 100.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
RS server top tier	TOPRSUPD	This is the arithmetic percentage to a region's MAXTASKS setting. The resultant task count value is subtracted from the region's MAXTASKS setting to establish a task load 'top tier'. If the task load in a region runs up to its MAXTASKS limit, then the task load must drop back below this value before the MAXTASKS state for the region is switched off and broadcast to the coupling facility. The value shown here at the CICSplex level may be overridden at the CICS definition level to allow fine tuning of the value on an individual CICS region basis.
RS Server update frequency	UPDATERS	Within a region running in optimized mode, this value specifies the frequency with which the CICS CFDT (RS) server will be called to modify the value of the task load within this target region. The value range is from 0 to 25: • When the Optimization status is not set to 'Active', then this value will be set to 0, and WLM will ignore it. When the Optimization status is set to 'Active', then a value of 0 means that the RS Server is not notified of any task load count changes, which disables the optimized workload function for this target region. • Values between 1 and 25 are applied as an arithmetic percentage to a region's MAXTASKS setting. The resultant task count value is used as a numeric threshold to drive an update call to the RS Server. **For example** , with a MAXTASKS setting of 120, and with this attribute set to 20, the RS Server will be called to update the WLM load count when the regions task count changes between: • 23 and 24 tasks - (20%), • 47 and 48 tasks - (40%), • 71 and 72 tasks - (60%), • 95 and 96 tasks - (80%). • 119 and 120 tasks - (100%). The RS Server would be updated when the task load increments or decrements across these boundaries. If a value is specified that is at the lower end of the 1-25 scale, then that will cause an increase in the frequency of updates to the RS Server across the task load range. For workloads in QUEUE mode, this will result in a task load more evenly balanced across the CICS regions in the workload target scope **(assuming all other health and link factors are equal)** . However, the utilization of the RS server will be correspondingly increased, which may consequently result in a higher utilization of your z/OS Coupling Facility. This value may be modified for this specific region using the 'MASs known to CICSplex' views, or the 'CICS system definition' views.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
z/OS WLM health	WLMHLTH	The percentage value that represents the z/OS WLM health open state for the region. This value may influence the amount of work a CPSM WLM routing region will send to the region. A value of zero (0) means that no work will be routed to the region. Any workload affinities associated with this region will remain and be honored, but routes based upon those affinities will fail while the value is zero. Note that the route failure seen depends upon how the route is attempted and whether a custom WRAM program is in effect. Some users may receive message EYUWR0003W indicating that the affinity AOR is not available, while others may receive a SYSIDERR. A value of 1 to 99 inclusive means that an additional weight will be added to the region that is inversely proportional to the WLMHLTH value. That is, the lower the WLMHLTH value, the greater the additional weight assigned. This may have the effect of increasing or decreasing workload being received by the region as its WLMHLTH value increases or decreases, depending upon the weight assigned to other regions within the AORSCOPE. A value of 100 means either that a region has now achieved a fully open z/OS WLM health open state, or that the region is running with WLMHEALTH=OFF in the SIT, or that the region is running CICS TS V5.3 or lower. A region with a value of 100 is treated as healthy and no additional weight is added for the region.
Optimization enablement state	WLMOPTEN	This field reports whether the current target region will support the optimized workload routing function. The possible values are: • ENABLED – This target region will support the optimized workload routing requests. • DISABLED – This region will not allow optimized workload routing requests, and will not contact the Region Status server for querying or broadcasting the current region state. This value may be modified for this specific region using the 'MASs known to CICSplex' views, or the 'CICS system definition' views.
Task load queue mode	WLMQMODE	This attribute specifies how the queued task load of a target CICS region is to be evaluated: • **MAXTASK** - specifies that both active and MAXTASK queued tasks are to be included in the task load evaluation for the region. • **ALL** - specifies that the task load evaluation for the region will include active tasks, tasks queued for the MAXTASK limit **and** tasks that are queued because of a TRANCLASS limit. • **N_a** - specifies that the current region is not at a high enough level to support the use of this attribute. You will not be able to change it, and the WLM route decision process will ignore it. The default value is **ALL.** This value is specified within the CICSPlex SM CICS system definition object. If you want this value to be changed, you must modify the system definition for the current region and then restart it. You may change the value in an active CICS region by using the CICSPlex SM MAS object. When the region restarts it will revert to the value specified in the system definition.

Table 19. Fields in WLMATARG views (continued)

Field	Attribute name	Description
Task load health threshold	WLMTHRSH	This specifies a percentage threshold of the current region's task load, which is calculated by dividing the current task count by the maximum task count. When the load for a target region reaches this threshold, then WLM considers the region to be relatively unhealthy. This will cause higher link weights to be applied to the WLM routing algorithm when evaluating this region. When a target scope covers local and remote regions relative to the router, then WLM will favour local targets over remote ones. The effect of this attribute is that when this load threshold in the local regions is achieved, then WLM will start to favour remote targets instead. When all targets in the scope achieve this load threshold, then WLM will revert to favouring local targets again. The value range is from 1 to 100, and the default value is 60. This value is specified within the CICSPlex SM CICS System Definition object. If you want this value to be changed, you must modify the system definition for the current region and then restart it. You may change the value in an active CICS region by using the CICSPlex SM MAS object. When the region restarts it will revert to the value specified in the system definition. If this value is set to 0, then the current region is not at a high enough level to support the use of this attribute. You will not be able to change it, and the WLM route decision process will ignore it. Note: this value is nullified when applied to the routing factor of link neutral dynamic routing requests. This is because the link weight itself is ignored for the LNQUEUE and LNGOAL algorithms.
Workload count	WRKLDCNT	This is a count of the workloads which have designated this region as a dynamic routing target. The workload count is set to zero (0) if: • the reporting CMAS is not the owning CMAS. • the owning CMAS is a version prior to CICS TS 4.1. • the owning CMAS is participating in a workload but has no routers connecting to it.

Definitions - WLMAWDEF

The **Active workload definition** (WLMAWDEF) views display information about active workload definitions installed in a workload that is within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Definitions**

*Table 20. Views in the supplied **Active workload definitions** (WLMAWDEF) view set*

View	Notes
Active workload definitions EYUSTARTWLMAWDEF.DETAILED	Detailed information about a selected workload definition.
Active workload definitions EYUSTARTWLMAWDEF.DISCARD	Discard an active workload definition.
Active workload definitions EYUSTARTWLMAWDEF.TABULAR	Tabular information about active workload definitions associated with a workload that is within the CICSplex identified as the context.

Actions

Table 21. Actions available for WLMAWDEF views

Action	Description
DISCARD	Discard an active workload definition.

Fields

Table 22. Fields in WLMAWDEF views

Field	Attribute name	Description
Scope name of set of target systems	AORSCOPE	The name of the CICS system or CICS system group to which transactions associated with the workload definition are directed.
Description	DESC	A description of the workload definition.
Terminal LU name	LUNAME	The logical unit name used in matching a transaction with the workload definition.
Workload definition	NAME	The name of the workload definition.
Process type	PROCESSTYPE	The specific or generic process type used in matching a transaction with the workload definition. Terminal LU name and User ID must both be set to '*'. A Transaction group of blanks or a specific name can be specified. You may only separate a workload by Transaction group and Process type or by Transaction group, Terminal LU name and User ID.
Transaction group	TRANGRP	The name of the transaction group associated with the workload definition. If this field is blank, it means no transaction group was defined; the workload definition is using the default transaction group associated with its workload specification. **Note:** The hyperlink to the WLMATGRP view is not valid when this field is blank.
User ID	USERID	The user ID used in matching a transaction with the workload definition.
Workload name	WORKLOAD	The name of the workload specification to which the workload definition is associated.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Transaction groups - WLMATGRP

The **Active workload transaction groups** (WLMATGRP) views display information about transaction groups installed in a workload that is within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Transaction groups**

*Table 23. Views in the supplied **Active workload transaction groups** (WLMATGRP) view set*

View	Notes
Active workload transaction groups EYUSTARTWLMATGRP.ACTIVE	Set the workload active.
Active workload transaction groups EYUSTARTWLMATGRP.DETAILED	Detailed information about a selected transaction group.

Table 23. Views in the supplied **Active workload transaction groups** (WLMATGRP) view set (continued)

View	Notes
Active workload transaction groups EYUSTARTWLMATGRP.DORMANT	Set the workload dormant.
Active workload transaction groups EYUSTARTWLMATGRP.SET	Set transaction group attributes.
Active workload transaction groups EYUSTARTWLMATGRP.TABULAR	Tabular information about transaction groups associated with a workload that is within the CICSplex identified as the context.

Actions

Table 24. Actions available for WLMATGRP views

Action	Description
ACTIVE	Set the workload active.
DORMANT	Set the workload dormant.
GET	(Required) The name of an active workload.
SET	Set transaction group attributes.

Fields

Table 25. Fields in WLMATGRP views

Field	Attribute name	Description
Acceptable abend level	ABENDCRIT	The abend probability for a transaction associated with the default transaction group that should cause a target region to be considered unhealthy.
Acceptable target region load level	ABENDTHRESH	The abend probability for a transaction associated with the default transaction group that should cause a target region's load level to be doubled.
Automatic affinity creation	AFFAUTO	Indicates whether CICSPlex SM is to automatically create an affinity relationship for transactions associated with the transaction group. • YES – An affinity is created using the values in the Affinity relationship and Affinity lifetime fields. • NO – An affinity is not automatically created (but can be created by a customized version of the dynamic routing program EYU9WRAM). • N/A – There are no values in the **Affinity relationship** and **Affinity lifetime** fields, therefore, no affinity is created.

Table 25. Fields in WLMATGRP views (continued)

Field	Attribute name	Description
Affinity relationship	AFFINITY	The affinity relationship used when processing occurrences of the transaction group. The affinity relationship values are: • GLOBAL - All users at all terminals • LUNAME - Terminal logical unit name • NONE - No affinity was defined • N/A - This field is not applicable • USERID - User ID • BAPPL - CICS BTS Business Application • Locked - LOCKED affinities can arise only between dynamically linked programs. A LOCKED affinity is created when a called program retains state data that is to be preserved after returning to its caller. Programs with this type of affinity are routed to the same target region until end of unit of work occurs. LOCKED can be used only for dynamic program link (DPL) requests with an associated affinity lifetime of UOW.
Affinity lifetime	AFFLIFE	The affinity lifetime to be used when processing occurrences of the transaction group. The affinity lifetime values are: • DELIMIT - Until the pseudoconversation mode is END • LOGON - For the duration of the terminal session • PCONV - For the duration of the pseudoconversation • PERMANENT - As long as the workload is active • SIGNON - As long as the user session is active • SYSTEM - As long as the AOR to which transactions are routed is active • ACTIVITY - As long as the CICS BTS activity is active • PROCESS - As long as the CICS BTS process is active • UOW - For as long as the unit-of-work is active • NONE - No affinity lifetime was defined • N/A - This field is not applicable

Table 25. Fields in WLMATGRP views (continued)

Field	Attribute name	Description
Algorithm type	ALGTYPE	The algorithm to be used when selecting the best target region in the Target Scope to which a transaction in the current transaction group should be routed. Valid options are: • **QUEUE** - Route the transaction to the target region with best combination of: – Health (MaxTask, Short-on-storage, Dumping, Stalled) – Task queue depth (or load) – Link speed from the routing region – Abend probability, when calculated – RTA event impact, when defined • **LNQUEUE** - Route the transaction to the target region with best combination of: – Health (MaxTask, Short-on-storage, Dumping, Stalled) – Task queue depth (or load) – Abend probability, when calculated – RTA event impact, when defined **Note** - The link speed from the router to the target is not factored into the routing weight calculation for the **LNQUEUE** algorithm • **GOAL** - Route the transaction to the target region that: – Is the most likely to allow the transaction to meet the response time goal set for it and other transactions in its MVS workload management class - When a specific target cannot be identified, apply the **QUEUE** algorithm to the remaining set of target regions • **LNGOAL** - Route the transaction to the target region that: – Is the most likely to allow the transaction to meet the response time goal set for it and other transactions in its MVS workload management class - When a specific target cannot be identified, apply the **LNQUEUE** algorithm to the remaining set of target regions • **INHERIT** - Use the algorithm type specified in the WLM specification (WLMSPEC) to which this transaction group is associated.
Event name	EVENTNAME	The name of an analysis definition (RTADEF) or status definition (STATDEF) that may affect the routing of transactions in this workload. If a real-time analysis event is generated by this definition, WLM uses the information as part of the AOR selection criteria. An analysis or status definition can optionally be associated with a workload when the workload specification is created. If this field is blank, no analysis or status definition is associated with the workload. **Note:** The EVENT view is displayed only if an event of the specified type has occurred.
Primary search criterion	FILTER	Identifies whether the user name (USERID) or the logical unit name (LUNAME) is to be used as the primary search criteria for transactions associated with the transaction group.
Transaction group status	STATUS	Indicates how an AOR is to be selected for transactions associated with the transaction group: • ACTIVE - Selects an AOR from the AOR scope identified with the associated workload definition. • DORMANT - Uses the AOR associated with the transaction when it was defined to CICS.
Active transaction count	TRANCNT	The number of active transactions that are associated with this transaction group.
Transaction group	TRANGRP	The name of the transaction group.

Table 25. Fields in WLMATGRP views (continued)

Field	Attribute name	Description
Workload	WORKLOAD	The name of the workload in which the transaction group is active.
Workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Dynamic transactions - WLMATRAN

The **Active workload dynamic transactions** (WLMATRAN) views display information about installed dynamic transaction definitions associated with a workload that is within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Dynamic transactions**

*Table 26. Views in the supplied **Active workload dynamic transactions** (WLMATRAN) view set*

View	Notes
Active workload dynamic transactions EYUSTARTWLMATRAN.DETAILED	Detailed information about a selected active transaction.
Active workload dynamic transactions EYUSTARTWLMATRAN.FORCE	Discard a transaction from a transaction group.
Active workload dynamic transactions EYUSTARTWLMATRAN.TABULAR	Tabular information about active transactions associated with a workload that is within the CICSplex identified as the context.

Actions

Table 27. Actions available for WLMATRAN views

Action	Description
FORCE	Discard a transaction from a transaction group.

Fields

Table 28. Fields in WLMATRAN views

Field	Attribute name	Description
Start or end of pseudo-conversation	PSEUDOMODE	Indicates whether the transaction is the first (START) or last (END) transaction in a pseudo-conversation. If this field is blank, it means the transaction is neither the first nor the last in a pseudo-conversation.
Transaction group	TRANGRP	The name of the transaction group.
Transaction	TRANID	The identifier of an active transaction.
Workload name	WORKLOAD	The name of the workload in which the transaction group is active.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Transaction group affinities - WLMATAFF

The **Active workload transaction group affinities** (WLMATAFF) views display information about the active affinities for a transaction group installed in a workload within the CICSplex identified as the context. An affinity becomes active when the first transaction associated with the transaction group is routed to a target region.

Supplied views

To access from the main menu, click:

Active workload views > **Transaction group affinities**

*Table 29. Views in the supplied **Active workload transaction group affinities** (WLMATAFF) view set*

View	Notes
Active workload transaction group affinities EYUSTARTWLMATAFF.DETAILED	Detailed information about a selected active affinity.
Active workload transaction group affinities EYUSTARTWLMATAFF.FORCE	Discard an affinity entry. The affinity is reestablished when the next transaction within the affected transaction group is encountered.
Active workload transaction group affinities EYUSTARTWLMATAFF.TABULAR	Tabular information about the active affinities for a transaction group associated with a workload within the CICSplex identified as the context.

Actions

Table 30. Actions available for WLMATAFF views

Action	Description
FORCE	Discard an affinity entry. The affinity is reestablished when the next transaction within the affected transaction group is encountered.

Fields

Table 31. Fields in WLMATAFF views

Field	Attribute name	Description
Affinity relationship	AFFINITY	The affinity relationship used when processing a transaction associated with the transaction group. The affinity relationship values are: • GLOBAL - All users at all terminals. • LUNAME - Terminal logical unit name. • USERID - User ID. • BAPPL - CICS BTS Business application. • Locked - LOCKED affinities can arise only between dynamically linked programs. A LOCKED affinity is created when a called program retains state data that is to be preserved after returning to its caller. Programs with this type of affinity are routed to the same target region until end of unit of work occurs. LOCKED can be used only for dynamic program link (DPL) requests with an associated affinity lifetime of UOW. • NONE - No affinity relation was defined.

Table 31. Fields in WLMATAFF views (continued)

Field	Attribute name	Description
Key that identifies creator of affinity	AFFKEY	Up to 64 characters identifying who invoked a transaction, comprising the transaction group and their location. Depending on the affinity relation and lifetime values, this key will be some combination of the user ID, logical unit name, routing region name, and terminal ID, in the following format: userid.luname.rtrname.termid. If one or more of these values does not apply to the affinity, that part of the key is left blank. For BAPPL affinities, this field contains the Activity ID of the CICS BTS activity. For LOCKED affinities, this field contains the combination of the routing region name with the network and local unit-of-work identifiers. A value of **GA** is displayed for global affinities.
Target region	AOR	The name of the CICS system to which transactions associated with the transaction group are to be routed.
BTS activity ID	CBTSKEY	The hexadecimal representation of the Affinity Key. It is shown in this additional format because part of the key may not be displayable in normal character representation.
Affinity lifetime	LIFETIME	The affinity lifetime used when processing a transaction associated with the transaction group. The affinity lifetime values are: • DELIMIT - Until the pseudoconversation mode is END. • LOGON - For the duration of the terminal session. • NONE - No affinity lifetime was defined. • PCONV - For the duration of the pseudoconversation. • PERMANENT - As long as the workload is active. • SIGNON - As long as the user session is active. • SYSTEM - As long as the target region is active. • ACTIVITY - As long as the CICS BTS activity is active. • PROCESS - As long as the CICS BTS process is active. • UOW - For as long as the unit-of-work is active
Local unit-of-work ID	LOCUOWID	The hexadecimal representation of the CICS local unit-of-work identifier. It is shown in this additional format because part of the key may not be displayable in normal character representation.
Terminal LU name	LUNAME	The logical unit name of the terminal of the CICS system to which transactions associated with the transaction group are to be routed.
Network unit-of-work ID	NETUOWID	The hexadecimal representation of the CICS network unit-of-work identifier. It is shown in this additional format because part of the key may not be displayable in normal character representation.
The activity ID of the CBTS activity	RESERVD2	The ID of the CBTS activity.
Routing region terminal	TERMID	The terminal ID of the CICS system to which transactions associated with the transaction group are to be routed.
Routing region	TORNAME	The name of the CICS system to which transactions associated with the transaction group are to be routed.
Transaction group	TRANGRP	The name of an active transaction group. A value of **$$DTGA$$** represents the default transaction group, which is used for any transactions that are not associated with a transaction group.
User identifier	USERID	The user identifier ID.
Workload name	WORKLOAD	The name of the workload in which the transaction group is active.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Complete active routing region details - WLMAROUT

The **Complete active routing regions** (WLMAROUT) views display information about all active routing regions within the CICSplex identified as the context.

Supplied views

To access from the main menu, click:

Active workload views > **Complete active routing region details**

*Table 32. Views in the supplied **Active routing region** (WLMAROUT) view set*

View	Notes
Active routing region EYUSTARTWLMAROUT.DETAILED	Detailed information about a selected active routing region.
Active routing region EYUSTARTWLMAROUT.TABULAR	Tabular information about all active routing regions that are associated with a workload.

Actions

Table 33. Actions available for WLMAROUT views

Action	Description
GET	The name of the workload specification. You can specify a workload name of '*', which returns details of all routing regions in all workloads. No other type of generic workload name may be specified.

Fields

Table 34. Fields in WLMAROUT views

Field	Attribute name	Description
Abend compensation status	ABDCMPSTATE	This field reports the status of abend compensation for the transaction that the TOR is currently routing. The possible values are: • YES – This router region is executing a transaction with abend compensation in effect. • NO – This router region is executing a transaction without abend compensation in effect.
Routing region contact status	CON_STATUS	The status of the connection between the routing region and the CMAS. Values are: • LOSTCON - The connection between the routing region and the CMASs has been lost. • N_A - The connection between the routing region and the CMASes managing the workload is available.
FMID sequence number	FMIDSEQ	The sequence number portion of the FMID value for the operating system version that the current region is executing under.
MVS system name	MVSSYSNAME	The operating system identifier that this routing region is executing under.

Table 34. Fields in WLMAROUT views (continued)		
Field	**Attribute name**	**Description**
Optimization status	OWSTATE	This field reports the status of the current routing region for the optimized workload routing function. The possible values are: • ACTIVE – This router region is executing in an optimized workload state. • INACTIVE – This region is capable of executing in an optimized workload state, however, is not currently optimized for one or more of the following reasons: - The region has no connection to an RS Server - The region is connected to an RS Server, however, the server has not been able to connect to the coupling facility - The optimised workload routing function is DISABLED for the current target region - The RS server update frequency value for this region is 0, which means that the optimization capabilities for this region are not enabled • N_A – The router region is not configured to provide optimized workload management. Only non optimized WLM routing decisions can be made. If the CMAS that your WUI is using is not connected to a Region Status (RS) server, the optimization status for remote regions will not be updated and defaults to INACTIVE. Where this situation occurs, either connect all reporting CMASs to the RS server or use the Active target regions view to check the optimization state.
Reporting CMAS name	RPTINGCMAS	The name of a CMAS that provided this record.
Unit of work completes	RTUOWEND	The number of times that the calling program of a dynamic program link has completed a unit-of-work. These completions may be a result of a CICS SYNCPOINT command or normal task termination.
Local time at which router was started	STRTTIME	The time at which this run of CICS started.
Routing region name	TOR	The name of an active CICS system, acting as a routing region, to which the workload is associated.
Name of controlling CMAS	TOROWNER	The name of the controlling CMAS.
Workload name	WORKLOAD	The name of the workload specification.
System ID of workload owner	WRKLOWNER	The 4-character CICS system ID of the CMAS that created the workload.

Chapter 6. CICSPlex SM API sample programs

CICSPlex SM provides several sample programs.

Each sample program is shown here in one of the languages in which it is distributed. For a list of the sample programs provided in each language and the libraries where they are distributed, see Sample programs.

Note: Additional sample CICSPlex SM API programs are available through the IBM CICS SupportPacs system at https://www.ibm.com/support/docview.wss?uid=swg27007241

EYU#API1

Program EYU#API1 is written in REXX for the TSO environment.

About EYU#API1

This program does the following processing:

- Establishes a connection to the API.
- Creates a result set containing all PROGRAM resource table records that do not begin with DFH, EYU, or IBM.
- Retrieves each record in the result set.
- Translates any CICS CVDA attributes into meaningful character values.
- Displays each record on the terminal, showing the program name, language, enable status, and CEDF status.
- Terminates the API connection.

Commands used: CONNECT, FETCH, GET, TERMINATE, TRANSLATE

```
/* REXX */
/********************************************************************/
/*                                                                  */
/* MODULE NAME = EYU#API1                                           */
/*                                                                  */
/* DESCRIPTIVE NAME = CPSM Sample API Program 1                     */
/*                    (Sample REXX Version)                         */
/*                                                                  */
/*    5695-081                                                      */
/*    COPYRIGHT = NONE                                              */
/*                                                                  */
/* STATUS = %CP00                                                   */
/*                                                                  */
/* FUNCTION =                                                       */
/*                                                                  */
/*  To provide an example of the use of the following EXEC CPSM     */
/*  commands: CONNECT, GET, FETCH, TRANSLATE, TERMINATE.            */
/*                                                                  */
/*  When invoked, the program depends upon the values held in the   */
/*  W_CONTEXT and W_SCOPE declarations when establishing a          */
/*  connection with CICSPlex SM. They must take the following       */
/*  values:                                                         */
/*                                                                  */
/*  W_CONTEXT  = The name of a CMAS or CICSplex. Refer to the       */
/*               description of the EXEC CPSM CONNECT command       */
/*               for further information regarding the CONTEXT      */
/*               option.                                            */
/*                                                                  */
/*  W_SCOPE    = The name of a CICSplex, CICS system, or CICS       */
/*               system group within the CICSplex. Refer to the     */
/*               description of the EXEC CPSM CONNECT command       */
/*               for further information regarding the SCOPE        */
/*               option.                                            */
/*                                                                  */
/*  This sample requires no parameters at invocation time.          */
/*                                                                  */
/*  The sample establishes an API connection and issues a GET       */
/*  command to create a result set containing program resource      */
/*  table records which match the criteria.                         */
/*                                                                  */
/*  Using the FETCH command each record in the result set is        */
/*  retrieved. Once retrieved the TRANSLATE command is used to      */
/*  convert those attributes of each record which are EYUDA or      */
/*  CVDA values into meaningful character representations. A       */
/*  record is then displayed on the terminal showing the program    */
/*  name, language, program status, and CEDF status.                */
/*                                                                  */
/*  Finally, the API connection is terminated.                      */
/*                                                                  */
/*------------------------------------------------------------------*/
/*NOTES :                                                           */
/*  DEPENDENCIES = S/390, TSO                                       */
/*  RESTRICTIONS = None                                             */
/*  REGISTER CONVENTIONS =                                          */
/*  MODULE TYPE  = Executable                                       */
/*  PROCESSOR    = REXX                                             */
/*  ATTRIBUTES   = Read only, Serially Reusable                     */
/*                                                                  */
```

```
/*------------------------------------------------------------------*/
/*                                                                  */
/*ENTRY POINT = EYU#API1                                            */
/*                                                                  */
/*   PURPOSE = All Functions                                        */
/*                                                                  */
/*   LINKAGE = From TSO as a REXX EXEC.                             */
/*                                                                  */
/*   INPUT   = None.                                                */
/*                                                                  */
/*------------------------------------------------------------------*/
/*                                                                  */
```

```

Address 'TSO'
Parse Value 0 0 With W_RESPONSE W_REASON .
/*------------------------------------------------------------------*/
/*    CHANGE W_CONTEXT AND W_SCOPE TO MATCH YOUR INSTALLATION       */
/*------------------------------------------------------------------*/
W_CONTEXT = 'RTGA'
W_SCOPE = 'RTGA'
/*------------------------------------------------------------------*/
/*    OBTAIN A CPSM API CONNECTION.                                 */
/*                                                                  */
/*    THE API WILL RETURN A TOKEN IDENTIFYING THE THREAD IN         */
/*    VARIABLE W_THREAD.                                            */
/*------------------------------------------------------------------*/
Say 'Initializing API...'
XX = EYUINIT()
If XX <> 0 Then Signal UNEXPECTED
Say 'Establishing connection...'
XX = EYUAPI('CONNECT' ,
            'CONTEXT('W_CONTEXT')' ,
            'SCOPE('W_SCOPE')' ,
            'VERSION(0310)' ,
            'THREAD(W_THREAD)' ,
            'RESPONSE(W_RESPONSE)' ,
            'REASON(W_REASON)')
If XX <> 0 Then Signal UNEXPECTED
If W_RESPONSE <> EYURESP(OK) Then Signal NO_CONNECT
/*------------------------------------------------------------------*/
/*    GET THE PROGRAM RESOURCE TABLE.                               */
/*                                                                  */
/*    CREATE A RESULT SET CONTAINING ENTRIES FOR ALL PROGRAMS       */
/*    WITH NAMES NOT BEGINNING DFH, EYU or IBM.                     */
/*    THE NUMBER OF ENTRIES MEETING THE CRITERIA IS RETURNED IN     */
/*    VARIABLE W_RECCNT.                                            */
/*------------------------------------------------------------------*/
Say 'Get the PROGRAM resource table...'
W_CRITERIA = 'NOT (PROGRAM=DFH* OR PROGRAM=EYU* OR PROGRAM=IBM*).'
W_CRITERIALEN = 'LENGTH'(W_CRITERIA)
XX = EYUAPI('GET OBJECT(PROGRAM)' ,
                'CRITERIA(W_CRITERIA)' ,
                'LENGTH('W_CRITERIALEN')' ,
                'COUNT(W_RECCNT)' ,
                'RESULT(W_RESULT)' ,
                'THREAD(W_THREAD)' ,
                'RESPONSE(W_RESPONSE)' ,
                'REASON(W_REASON)')
If XX <> 0 Then Signal UNEXPECTED
If W_RESPONSE <> EYURESP(OK) Then Signal NO_GET
```

```
/*-------------------------------------------------------------------*/
/*    RETRIEVE INFORMATION ABOUT EACH PROGRAM.                       */
/*                                                                   */
/*    FETCH EACH ENTRY AND USE TPARSE TO OBTAIN EACH ATTRIBUTE.      */
/*    DISPLAY DETAILS OF EACH PROGRAM TO THE USER.                   */
/*-------------------------------------------------------------------*/
Say 'Fetching' W_RECCNT 'PROGRAM entries...'
Say 'Program  Language     Status       CEDF Status'
W_INTO_OBJECTLEN = 136                   /* LENGTH OF PROGRAM TABLE */
Do III = 1 To W_RECCNT
  XX = EYUAPI('FETCH INTO(W_INTO_OBJECT)' ,
                    'LENGTH(W_INTO_OBJECTLEN)' ,
                    'RESULT(W_RESULT)' ,
                    'THREAD(W_THREAD)' ,
                    'RESPONSE(W_RESPONSE)' ,
                    'REASON(W_REASON)')
  If XX <> 0 Then Signal UNEXPECTED
  If W_RESPONSE <> EYURESP(OK) Then Signal NO_FETCH
  XX = EYUAPI('TPARSE OBJECT(PROGRAM)' ,
                     'PREFIX(PGM)' ,
                     'STATUS(W_RESPONSE)' ,
                     'VAR(W_INTO_OBJECT.1)' ,
                     'THREAD(W_THREAD)')
  If W_RESPONSE <> 'OK' Then Signal UNEXPECTED
  W_TEXT = PGM_PROGRAM
  W_TEXT = 'OVERLAY'(PGM_LANGUAGE,W_TEXT,10)
  W_TEXT = 'OVERLAY'(PGM_STATUS,W_TEXT,23)
  W_TEXT = 'OVERLAY'(PGM_CEDFSTATUS,W_TEXT,36)
  Say W_TEXT
End III
Signal ENDIT
```

```
/*-------------------------------------------------------------------*/
/*    PROCESSING FOR API FAILURES.                                   */
/*-------------------------------------------------------------------*/
UNEXPECTED:
      W_MSG_TEXT = 'UNEXPECTED ERROR.'
      Signal SCRNLOG
NO_CONNECT:
      W_MSG_TEXT = 'ERROR CONNECTING TO API.'
      Signal SCRNLOG
NO_GET:
      W_MSG_TEXT = 'ERROR GETTING RESOURCE TABLE.'
      Signal SCRNLOG
NO_FETCH:
      W_MSG_TEXT = 'ERROR FETCHING RESULT SET.'
      Signal SCRNLOG
SCRNLOG:
      Say W_MSG_TEXT
      Say 'RESPONSE='||W_RESPONSE ,
          'REASON='||W_REASON 'RESULT='XX
ENDIT:
/*-------------------------------------------------------------------*/
/*    TERMINATE API CONNECTION.                                      */
/*-------------------------------------------------------------------*/
XX = EYUAPI('TERMINATE RESPONSE(W_RESPONSE) REASON(W_REASON)')
XX = EYUTERM()
Exit
```

The C/370, COBOL, and PL/I versions of EYUxAPI1 are written for the CICS environment and can be converted to run in the MVS batch environment by commenting the **`EXEC CICS SEND`** commands and uncommenting the preceding language specific output statements.

EYUCAPI2

Program EYUCAPI2 is written in C for the CICS enviornment.

About EYUxAPI2

This program does the following processing:

- Establishes a connection to the API.
- Defines a filter to identify PROGRAM resource table records with a language attribute of Assembler.
- Creates a result set containing all PROGRAM resource table records that do not begin with DFH, EYU, or IBM.
- Marks those records in the result set that match the specified filter (LANGUAGE=ASSEMBLER).
- Copies the marked records to a new result set.
- Deletes the marked records from the original result set.
- For each result set (LANGUAGE=ASSEMBLER and LANGUAGE≠ASSEMBLER):
  - Retrieves each record.
  - Translates any CICS CVDA attributes.
  - Displays each record on the terminal.
- Terminates the API connection.

Commands used: CONNECT, COPY, DELETE, FETCH, GET, LOCATE, MARK, SPECIFY FILTER, TERMINATE, TRANSLATE

```
/********************************************************************/
/*                                                                  */
/* MODULE NAME = EYUCAPI2                                           */
/*                                                                  */
/* DESCRIPTIVE NAME = CPSM Sample API Program 2                     */
/*                    (Sample C Version)                            */
/*                                                                  */
/*    5695-081                                                      */
/*    COPYRIGHT = NONE                                              */
/*                                                                  */
/* STATUS = %CP00                                                   */
/*                                                                  */
/* FUNCTION =                                                       */
/*                                                                  */
/*  To provide an example of the use of the following EXEC CPSM     */
/*  commands: CONNECT, SPECIFY FILTER, GET, MARK, COPY, DELETE,     */
/*  LOCATE, FETCH, TRANSLATE, TERMINATE.                            */
/*                                                                  */
/*  When invoked, the program depends upon the values held in the   */
/*  W_CONTEXT and W_SCOPE declarations when establishing a          */
/*  connection with CICSPlex SM. They must take the following       */
/*  values:                                                         */
/*                                                                  */
/*  W_CONTEXT  = The name of a CMAS or CICSplex. Refer to the       */
/*               description of the EXEC CPSM CONNECT command       */
/*               for further information regarding the CONTEXT      */
/*               option.                                            */
/*                                                                  */
/*  W_SCOPE    = The name of a CICSplex, CICS system, or CICS       */
/*               system group within the CICSplex. Refer to the     */
/*               description of the EXEC CPSM CONNECT command       */
/*               for further information regarding the SCOPE        */
/*               option.                                            */
/*                                                                  */
/*  This sample requires no parameters at invocation time.          */
/*                                                                  */
/*  The sample establishes an API connection and issues a SPECIFY   */
/*  FILTER command to create a filter which will match only         */
/*  specific program resource table records. The filter is used     */
/*  later in the program by the MARK command.                       */
/*                                                                  */
/*  A GET command is issued to create a result set containing       */
/*  program resource table records which match the criteria. The    */
/*  result set is then used by the MARK command to flag records     */
/*  meeting the previous filter specification. The marked records   */
/*  are then COPYed to a new result set, and then DELETEd from      */
/*  the original result set. After this sequence of commands we     */
/*  have two results sets; one containing records which did not     */
/*  meet the filter specification (that is, records where the       */
/*  LANGUAGE is not ASSEMBLER), and one containing records          */
```

```
 /*  which did match the filter (that is, records where the         */
 /*  LANGUAGE is ASSEMBLER).                                         */
 /*                                                                  */
 /*  Taking each of the two results sets in turn a LOCATE command    */
 /*  is used to ensure we start at the top of the result set         */
 /*  before a FETCH command is used to retrieve each record in       */
 /*  the result set. Once retrieved the TRANSLATE command is used    */
 /*  to convert those attributes of each record which are EYUDA      */
 /*  or CVDA values into meaningful character representations. A     */
 /*  record is then displayed on the terminal showing the program    */
 /*  name, language, program status, and CEDF status.                */
 /*                                                                  */
 /*  Finally, the API connection is terminated.                      */
 /*                                                                  */
```

```
* --------------------------------------------------------------------*
 /*NOTES :                                                           */
 /*  DEPENDENCIES = S/390, CICS                                      */
 /*  RESTRICTIONS = None                                             */
 /*  REGISTER CONVENTIONS =                                          */
 /*  MODULE TYPE  = Executable                                       */
 /*  PROCESSOR    = C                                                */
 /*  ATTRIBUTES   = Read only, Serially Reusable                     */
 /*                                                                  */
 /*------------------------------------------------------------------*/
 /*                                                                  */
 /*ENTRY POINT = EYUCAPI2                                            */
 /*                                                                  */
 /*   PURPOSE = All Functions                                        */
 /*                                                                  */
 /*   LINKAGE = From CICS either with EXEC CICS LINK or as a CICS    */
 /*             transaction.                                         */
 /*                                                                  */
 /*   INPUT   = None.                                                */
 /*                                                                  */
 /*------------------------------------------------------------------*/
 /*                                                                  */
#include <PROGRAM>
void main()
{
/*------------------------------------------------------------------*/
/*    CHANGE W_CONTEXT AND W_SCOPE TO MATCH YOUR INSTALLATION       */
/*------------------------------------------------------------------*/
char    *W_CONTEXT       = "RTGA    ";
char    *W_SCOPE         = "RTGA    ";
int      W_RESPONSE;
int      W_REASON;
int      W_THREAD;
char    *W_CRITERIA;
int      W_CRITERIALEN;
int      W_FILTER_TOKEN;
int      W_RESULT        = 0;
int      W_COUNT;
int      W_RESULT2       = 0;
int      W_COUNT2;
int      III;
int      JJJ;
int      W_RESULT_TOK;
int      W_RECCNT;
PROGRAM  W_INTO_OBJECT;
int      W_INTO_OBJECTLEN;
char     W_TRANSCVDA??(12??);
char     W_TEXT??(81??);
char     W_MSG_TEXT??(81??);
W_TEXT??(80??) = 0x13;
W_MSG_TEXT??(80??) = 0x13;
/*------------------------------------------------------------------*/
/*    OBTAIN A CPSM API CONNECTION.                                 */
/*                                                                  */
/*    THE API WILL RETURN A TOKEN IDENTIFYING THE THREAD IN         */
/*    VARIABLE W_THREAD.                                            */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Establishing connection...");
/* printf("Establishing connection...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) ERASE;
EXEC CPSM CONNECT
          CONTEXT(W_CONTEXT)
          SCOPE(W_SCOPE)
          VERSION("0310")
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_CONNECT; }
```

```
/*------------------------------------------------------------------*/
/*    CREATE A FILTER.                                              */
/*                                                                  */
/*    CREATE A FILTER WHICH WILL MATCH ONLY THOSE PROGRAMS WITH     */
/*    A LANGUAGE OF ASSEMBLER.                                      */
/*    THE FILTER WILL BE USED IN A SUBSEQUENT MARK COMMAND.         */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Create a filter...        ");
/* printf("Create a filter...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
W_CRITERIA = "LANGUAGE=ASSEMBLER.";
W_CRITERIALEN = strlen(W_CRITERIA);
EXEC CPSM SPECIFY FILTER(W_FILTER_TOKEN)
          CRITERIA(W_CRITERIA)
          LENGTH(W_CRITERIALEN)
          OBJECT("PROGRAM ")
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_FILTER; }
```

```
/*------------------------------------------------------------------*/
/*    GET THE PROGRAM RESOURCE TABLE.                               */
/*                                                                  */
/*    CREATE A RESULT SET CONTAINING ENTRIES FOR ALL PROGRAMS       */
/*    WITH NAMES NOT BEGINNING DFH, EYU OR IBM.                     */
/*    THE NUMBER OF ENTRIES MEETING THE CRITERIA IS RETURNED IN     */
/*    VARIABLE W_COUNT.                                             */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Get the PROGRAM resource table...");
/* printf("Get the PROGRAM resource table...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
W_CRITERIA = "NOT (PROGRAM=DFH* OR PROGRAM=EYU* OR PROGRAM=IBM*).";
W_CRITERIALEN = strlen(W_CRITERIA);
EXEC CPSM GET OBJECT("PROGRAM ")
          CRITERIA(W_CRITERIA)
          LENGTH(W_CRITERIALEN)
          COUNT(W_COUNT)
          RESULT(W_RESULT)
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_GET; }
sprintf(W_TEXT,"Total number of entries: %d", W_COUNT);
/* printf(W_TEXT); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
/*------------------------------------------------------------------*/
/*    MARK SELECTED PROGRAM ENTRIES.                                */
/*                                                                  */
/*    USING THE FILTER WE MARK THOSE ENTRIES IN THE RESULT SET      */
/*    WHICH MEET THE FILTER SPECIFICATION IE. THOSE ENTRIES WITH    */
/*    A LANGUAGE OF ASSEMBLER.                                      */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Mark LANGUAGE=ASSEMBLER entries...");
/* printf("Mark LANGUAGE=ASSEMBLER entries...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
EXEC CPSM MARK FILTER(W_FILTER_TOKEN)
          RESULT(W_RESULT)
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_MARK; }
```

```
/*------------------------------------------------------------------*/
/*    COPY MARKED ENTRIES TO ANOTHER RESULT SET.                    */
/*                                                                  */
/*    HAVING MARKED ENTRIES IN THE RESULT SET WE CAN COPY THEM      */
/*    TO A NEW RESULT SET.                                          */
/*    AFTER THIS COMMAND WE WILL HAVE TWO RESULT SETS. ONE          */
/*    CONTAINING ALL THE PROGRAM ENTRIES, AND THE OTHER CONTAINING  */
/*    JUST THOSE ENTRIES WITH A LANGUAGE OF ASSEMBLER.              */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Copy marked entries...       ");
/* printf("Copy marked entries...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
EXEC CPSM COPY FROM(W_RESULT)
```

```
          TO(W_RESULT2)
          MARKED
          COUNT(W_COUNT2)
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_COPY; }
sprintf(W_TEXT,"Number of entries copied: %d", W_COUNT2);
/* printf(W_TEXT); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
/*------------------------------------------------------------------*/
/*    DELETE MARKED ENTRIES FROM RESULT SET.                        */
/*                                                                  */
/*    WE CAN NOW DELETE THE MARKED ENTRIES FROM THE ORIGINAL        */
/*    RESULT SET.                                                   */
/*    AFTER THIS COMMAND WE HAVE TWO RESULT SETS. ONE RESULT SET    */
/*    CONTAINING ENTRIES WITH LANGUAGE NOT ASSEMBLER, AND THE       */
/*    OTHER CONTAINING ENTRIES WITH A LANGUAGE OF ASSEMBLER.        */
/*------------------------------------------------------------------*/
strcpy(W_TEXT,"Delete marked entries...     ");
/* printf("Delete marked entries...\n"); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
EXEC CPSM DELETE MARKED
          COUNT(W_COUNT)
          RESULT(W_RESULT)
          THREAD(W_THREAD)
          RESPONSE(W_RESPONSE)
          REASON(W_REASON) ;
if (W_RESPONSE != EYUVALUE(OK)) { goto NO_DELETE; }
sprintf(W_TEXT,"Number of entries remaining: %d", W_COUNT);
/* printf(W_TEXT); */
EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
```

```
/*------------------------------------------------------------------*/
/*    RETRIEVE INFORMATION ABOUT EACH PROGRAM.                      */
/*                                                                  */
/*    FETCH EACH ENTRY, USE INCLUDED STRUCTURE TO OBTAIN EACH       */
/*    ATTRIBUTE AND USE TRANSLATE TO CONVERT CICS CVDAS.            */
/*    DISPLAY DETAILS OF EACH PROGRAM TO THE USER.                  */
/*------------------------------------------------------------------*/
W_INTO_OBJECTLEN = PROGRAM_TBL_LEN;
for (JJJ = 1; JJJ <= 2; JJJ++)
 {
   if (JJJ == 1)
    {
      sprintf(W_TEXT,"Fetching %d non-ASSEMBLER PROGRAM entries...\n",
              W_COUNT);
      W_RESULT_TOK = W_RESULT;
      W_RECCNT = W_COUNT;
    }
   else
     {
       sprintf(W_TEXT,"Fetching %d ASSEMBLER PROGRAM entries...\n",
               W_COUNT2);
       W_RESULT_TOK = W_RESULT2;
       W_RECCNT = W_COUNT2;
     }
   /* printf(W_TEXT); */
   EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
   EXEC CPSM LOCATE TOP
             RESULT(W_RESULT_TOK)
             THREAD(W_THREAD)
             RESPONSE(W_RESPONSE)
             REASON(W_REASON) ;
   if (W_RESPONSE != EYUVALUE(OK)) { goto NO_LOCATE; }
   strcpy(W_TEXT,"Program  Language     Status       CEDF Status");
   /* printf("Program  Language     Status       CEDF Status\n"); */
   EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
   for (III = 1; III <= W_RECCNT; III++)
    {
      EXEC CPSM FETCH INTO(&W_INTO_OBJECT)
                LENGTH(W_INTO_OBJECTLEN)
                RESULT(W_RESULT_TOK)
                THREAD(W_THREAD)
                RESPONSE(W_RESPONSE)
                REASON(W_REASON) ;
      if (W_RESPONSE != EYUVALUE(OK)) { goto NO_FETCH; }
      memcpy(W_TEXT,W_INTO_OBJECT.PROGRAM,8);
      EXEC CPSM TRANSLATE OBJECT("PROGRAM ")
                ATTRIBUTE("LANGUAGE    ")
                FROMCV(W_INTO_OBJECT.LANGUAGE)
                TOCHAR(W_TRANSCVDA)
                THREAD(W_THREAD)
                RESPONSE(W_RESPONSE)
                REASON(W_REASON) ;
      if (W_RESPONSE != EYUVALUE(OK)) { goto NO_TRANSLATE; }
      memcpy(W_TEXT+9,W_TRANSCVDA,12);
      EXEC CPSM TRANSLATE OBJECT("PROGRAM ")
                ATTRIBUTE("STATUS      ")
                FROMCV(W_INTO_OBJECT.STATUS)
                TOCHAR(W_TRANSCVDA)
                THREAD(W_THREAD)
                RESPONSE(W_RESPONSE)
                REASON(W_REASON) ;
```

```
      if (W_RESPONSE != EYUVALUE(OK)) { goto NO_TRANSLATE; }
      memcpy(W_TEXT+22,W_TRANSCVDA,12);
      EXEC CPSM TRANSLATE OBJECT("PROGRAM ")
                ATTRIBUTE("CEDFSTATUS  ")
                FROMCV(W_INTO_OBJECT.CEDFSTATUS)
                TOCHAR(W_TRANSCVDA)
                THREAD(W_THREAD)
                RESPONSE(W_RESPONSE)
                REASON(W_REASON) ;
      if (W_RESPONSE != EYUVALUE(OK)) { goto NO_TRANSLATE; }
      memcpy(W_TEXT+35,W_TRANSCVDA,12);
      /* printf("%s\n",W_TEXT); */
      EXEC CICS SEND FROM(W_TEXT) LENGTH(81) WAIT;
    }
 }
goto ENDIT;
/*------------------------------------------------------------------*/
/*    PROCESSING FOR API FAILURES.                                  */
/*------------------------------------------------------------------*/
NO_CONNECT:
   strcpy(W_MSG_TEXT,"ERROR CONNECTING TO API.\n");
   goto SCRNLOG;
NO_FILTER:
   strcpy(W_MSG_TEXT,"ERROR CREATING FILTER.\n");
   goto SCRNLOG;
NO_GET:
   strcpy(W_MSG_TEXT,"ERROR GETTING RESOURCE TABLE.\n");
   goto SCRNLOG;
NO_MARK:
   strcpy(W_MSG_TEXT,"ERROR MARKING RESULT SET.\n");
   goto SCRNLOG;
NO_COPY:
   strcpy(W_MSG_TEXT,"ERROR COPYING RESULT SET.\n");
   goto SCRNLOG;
NO_DELETE:
   strcpy(W_MSG_TEXT,"ERROR DELETING FROM RESULT SET.\n");
   goto SCRNLOG;
NO_LOCATE:
   strcpy(W_MSG_TEXT,"ERROR LOCATING TO TOP OF RESULT SET.\n");
   goto SCRNLOG;
NO_FETCH:
   strcpy(W_MSG_TEXT,"ERROR FETCHING RESULT SET.\n");
   goto SCRNLOG;
NO_TRANSLATE:
   strcpy(W_MSG_TEXT,"ERROR TRANSLATING ATTRIBUTE\n");
   goto SCRNLOG;
SCRNLOG:
   /* printf(W_MSG_TEXT); */
   EXEC CICS SEND FROM(W_MSG_TEXT) LENGTH(81) WAIT;
   sprintf(W_MSG_TEXT,"RESPONSE=%d REASON=%d\n",W_RESPONSE,W_REASON);
   /* printf(W_MSG_TEXT); */
   EXEC CICS SEND FROM(W_MSG_TEXT) LENGTH(81) WAIT;
ENDIT:
/*------------------------------------------------------------------*/
/*    TERMINATE API CONNECTION.                                     */
/*------------------------------------------------------------------*/
EXEC CPSM TERMINATE RESPONSE(W_RESPONSE) REASON(W_REASON);
EXEC CICS RETURN;
}
```

The C, C++, COBOL, and PL/I versions of EYUxAPI2 are written for the CICS environment and can be converted to run in the MVS batch environment by commenting the **EXEC CICS SEND** commands and uncommenting the preceding language specific output statements.

EYUAAPI3

Program EYUAAPI3 is written in Assembler for the MVS batch environment.

About EYUAAPI3

This program does the following processing:

- Establishes a connection to the API with the context set to an existing CICSplex.
- Verifies that a proposed new CICSplex name is not already defined to CICSPlex SM as a CICSplex, CMAS, CICS system, or CICS system group.
- Creates a result set containing the CPLEXDEF resource table record for the existing CICSplex definition and retrieves that record.
- Creates a new CPLEXDEF resource table record using the existing record as a model.
- Creates a result set containing the CICSPLEX resource table records associated with the existing CICSplex and retrieves those records.
- Creates new CICSPLEX resource table records using the existing records as models.
- Sequentially retrieves all the resource table records associated with the existing CICSplex, including CICS systems, CICS system groups, workload management definitions, real-time analysis definitions, and resource monitoring definitions.
- Creates all the necessary resource table records for the new CICSplex using the existing records as models.
- If an error occurs before all the necessary resource table records are created, removes the new CICSplex definition.
- Disconnects the API processing thread.

Commands used: CONNECT, CREATE, DISCARD, DISCONNECT, FETCH, GET, PERFORM OBJECT, QUALIFY, QUERY, REMOVE

```
*
EYUAAPI3 TITLE 'EYUAAPI3 - CPSM SAMPLE API PROGRAM 3 - ASSEMBLER'
***********************************************************************
*                                                                     *
* MODULE NAME = EYUAAPI3                                              *
*                                                                     *
* DESCRIPTIVE NAME = API sample program 3 ASSEMBLER Version           *
*                                                                     *
*        5695-081                                                     *
*        COPYRIGHT = NONE                                             *
*                                                                     *
* STATUS = %CP00                                                      *
*                                                                     *
* FUNCTION =                                                          *
*                                                                     *
*    To mirror an existing PLEX to a new PLEX.                        *
*                                                                     *
*    When invoked, the program depends upon the values held in the    *
*    OLDPLEX, NEWPLEX, and MPCMAS variables.  They must be set to     *
*    the following values:                                            *
*                                                                     *
*    OLDPLEX    = The name of an existing PLEX that will be mirrored. *
*                                                                     *
*    NEWPLEX    = The name that will be given to the new PLEX.        *
*                                                                     *
*    MPCMAS     = The maintenance point CMAS of the OLDPLEX.  This    *
*                 will also be the MP for the NEWPLEX.                *
*                                                                     *
*    This sample requires no parameters at invocation time.           *
*                                                                     *
*    The sample processes as follows:                                 *
*                                                                     *
*    -  a CONNECTion is established to CPSM, with the CONTEXT and     *
*       SCOPE of the OLDPLEX.                                         *
*                                                                     *
*    -  since a PLEX can be either a CONTEXT or SCOPE, we verify      *
*       that the NEWPLEX is not already a valid CONTEXT (i.e, an      *
*       existing CICSplex or CMAS) or SCOPE in the OLDPLEX (i.e,      *
*       an existing CICS system or CICS system group).                *
*                                                                     *
*    -  we GET the CPLEXDEF record for the OLDPLEX, and use this as   *
*       a module to CREATE the NEWPLEX.                               *
*                                                                     *
*    -  we GET the CICSPLEX records for the OLDPLEX, and use these    *
*       to add the CMASs in the OLDPLEX to the NEWPLEX.               *
*                                                                     *
*    -  using a list that contains CICSplex definitions including     *
*      CICS systems, CICS system groups, workload management          *
*      definitions, real-time analysis definitions and resource       *
*      monitoring definitions, we GET and FETCH the records from      *
*      the OrigPlex, and CREATE them in the NewPlex.                  *
*                                                                     *
*    -  we then DISCONNECT from CPSM.                                 *
*                                                                     *
```

```
* --------------------------------------------------------------------*
*                                                                     *
* NOTES :                                                             *
*    DEPENDENCIES = S/370                                             *
*    RESTRICTIONS = None                                              *
*    REGISTER CONVENTIONS =                                           *
*        R0              Workarea / external call parameter pointer   *
*        R1              Workarea / external call parameter pointer   *
*        R2              Resource Table record pointer                *
*        R3              Loop counter                                 *
*        R4              List pointer                                 *
*        R5              Loop counter                                 *
*        R6              Unused                                       *
*        R7              Unused                                       *
*        R8              Unused                                       *
*        R9              Subroutine linkage                           *
*        R10             Subroutine linkage                           *
*        R11             Base register                                *
*        R12             Base register                                *
*        R13             Workarea pointer                             *
*        R14             External call linkage                        *
*        R15             External call linkage                        *
*                                                                     *
*    MODULE TYPE  = Executable                                        *
*    PROCESSOR    = Assembler                                         *
*    ATTRIBUTES   = Read only, Serially Reusable                      *
*                   AMODE(31), RMODE(ANY)                             *
*                                                                     *
* --------------------------------------------------------------------*
*                                                                     *
* ENTRY POINT = EYUAAPI3                                              *
*                                                                     *
*     PURPOSE = All Functions                                         *
*                                                                     *
*     LINKAGE = Executed as a batch program.                          *
*                                                                     *
*     INPUT   = None                                                  *
*                                                                     *
*     OUTPUT  = File for messages.                                    *
*               DDNAME  = SYSPRINT                                    *
*               DSORG   = PS                                          *
*               RECFM   = FB                                          *
*               LRECL   = 80                                          *
*               BLKSIZE = a multiple of 80                            *
*                                                                     *
* --------------------------------------------------------------------*
         EJECT
EYUAAPI3 CSECT
         STM   R14,R12,12(R13)
         LR    R12,R15
         USING EYUAAPI3,R12
*---------------------------------------------------------------------*
*        GETMAIN working storage and set up SA chain.                 *
*---------------------------------------------------------------------*
         GETMAIN R,LV=WORKLEN
         ST    R13,4(,1)
         ST    R1,8(,R13)
         L     R1,24(,R13)
         L     R13,8(,R13)
         USING SAVEAREA,R13
```

```
*---------------------------------------------------------------------*
*        Preset return code to error - will change to 0 if all ok.    *
```

```
*---------------------------------------------------------------------*
         MVC   RETCODE,=F'8'
*---------------------------------------------------------------------*
*        OPEN file for error messages.                                *
*---------------------------------------------------------------------*
         OPEN  (SYSPRINT,OUTPUT)
*---------------------------------------------------------------------*
*        Specify variables: OLDPLEX, NEWPLEX, MPCMAS                  *
*                                                                     *
*        Ensure that the values specified are valid NAME type (i.e,   *
*        valid member name) or following code will fail.              *
*---------------------------------------------------------------------*
         MVC   OLDPLEX,=CL8'plexold'    *** SPECIFY AS DESIRED ***
         MVC   NEWPLEX,=CL8'plexnew'    *** SPECIFY AS DESIRED ***
         MVC   MPCMAS,=CL8'mpcmas'      *** SPECIFY AS DESIRED ***
*---------------------------------------------------------------------*
*        Connect to CPSM API via OLDPLEX.                             *
*---------------------------------------------------------------------*
         MVC   CONTEXT,OLDPLEX
         EXEC  CPSM CONNECT                                            X
                    CONTEXT(CONTEXT)                                   X
                    VERSION(=CL4'0130')                                X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BNE   ERRCON                   No - msgs and out
*---------------------------------------------------------------------*
*        Verify that the desired NEWPLEX name is not already a        *
*        PLEX or CMAS.  We do this by trying to set the CONTEXT       *
*        to the NEWPLEX name.  If successful (NEWPLEX already exists  *
*        as a CONTEXT) issue messages and get out.                    *
*---------------------------------------------------------------------*
         EXEC  CPSM QUALIFY                                            X
                    CONTEXT(NEWPLEX)                                   X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BE    ERRNISPC                 Yes - already a CONTEXT
```

```
*---------------------------------------------------------------------*
*        Verify that the NEWPLEX name is not already a                *
*        CSYSDEF or CSYSGRP in the old, soon to be new, CICSplex.     *
*                                                                     *
*        Here we will start issuing EXEC CPSM GET requests, to        *
*        get result sets of different Resource Tables.  We make       *
*        the call through the GETOBJ subroutine.  Variable OBJECT     *
*        must be set with the Resource Table name.  If we only want   *
*        a subset of the records for a given Resource Table, we also  *
*        set variable CRITERIA with a selection criteria string.      *
*        This string can contain references to any fields in the     *
*        Resource Table, connected by logical operators, and must     *
*        end with a period - . -.  Variable CRITLEN must be loaded    *
*        with the length of the criteria string.                      *
*                                                                     *
*        We will check the RESPONSE from GET calls inline, instead    *
*        of in the subroutine.  The reason for this is that sometimes *
*        a RESPONSE of OK will mean that we have a problem (e.g.,     *
*        the NEWPLEX name already exists as a CICS System name).      *
*---------------------------------------------------------------------*
*
*        Ask for a CSYSSYS record equal to the NEWPLEX name.
*
         MVC   OBJECT,=CL8'CSYSDEF'
         MVC   CRITERIA(5),=CL5'NAME='
         MVC   CRITERIA+5(8),NEWPLEX
         MVI   CRITERIA+13,C'.'
         MVC   CRITLEN,=F'14'
         BAS   R10,GETOBJ               Build result set
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BE    ERRNISC                  Yes - already a CICS system
         CLC   RESPONSE,EYUVALUE(NODATA) No CSYSDEF with NEWPLEX name?
         BE    NOTCSYS                  Yes - continue
         B     ERRGETO                  No - some error - msgs and out
NOTCSYS  DS    0H
*
*        Ask for a CSYSGRP record equal to the NEWPLEX name.
*
         MVC   OBJECT,=CL8'CSYSGRP'
         MVC   CRITERIA(6),=CL6'GROUP='
         MVC   CRITERIA+6(8),NEWPLEX
         MVI   CRITERIA+14,C'.'
         MVC   CRITLEN,=F'15'
         BAS   R10,GETOBJ               Build the result set
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BE    ERRNISS                  Yes - already a system group
         CLC   RESPONSE,EYUVALUE(NODATA) No CSYSGRP with NEWPLEX name?
         BE    NOTCGRP                  Yes - continue
         B     ERRGETO                  No - some error - msgs and out
NOTCGRP  DS    0H
*---------------------------------------------------------------------*
*        If we have gotten this far, we know that NEWPLEX is not      *
*        already the name of a CICSplex, CMAS, CICS System, or        *
*        CICS System group - so we can start building the NEWPLEX.    *
*                                                                     *
*        Switch CONTEXT to MPCMAS to build NEWPLEX and add CMASs.     *
*---------------------------------------------------------------------*
         MVC   CONTEXT,MPCMAS
```

```
*---------------------------------------------------------------------*
*        Build new plex using OLDPLEX as a model.                     *
*                                                                     *
*        The record that defines a CICSplex is the CPLEXDEF Resource  *
*        Table.  We will GET the OLDPLEX CPLEXDEF record, modify      *
*        it as needed, and then CREATE the NEWPLEX CPLEXDEF records.  *
*        This creates the NEWPLEX.                                    *
*---------------------------------------------------------------------*
         MVI   PLEXBLT,C'N'             Indicate NEWPLEX not built yet
*
*        First GET CPLEXDEF record for the OLDPLEX.
*
         MVC   OBJECT,=CL8'CPLEXDEF'
         MVC   CRITERIA(9),=CL9'CICSPLEX='
         MVC   CRITERIA+9(8),OLDPLEX
         MVI   CRITERIA+17,C'.'
         MVC   CRITLEN,=F'18'
         BAS   R10,GETOBJ               Build result set
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BNE   ERRGETO                  No - msgs and out
*
*        Here we start using the GETBUF subroutine.  This subroutine
*        GETMAINs a buffer into which we can FETCH the records of the
*        result set that we last issued a GET for.
*
         BAS   R10,GETBUF               Get storage to receive recs
*
*        Here we start using the FETCH subroutine.  This subroutine
*        reads all the records from the result set into the buffer.
*        On return to mainline, R2 points to the first record in
*        the buffer.
*
         BAS   R10,FETCH                Sets R2 to fetched record
*
*        Change the OLDPLEX CPLEXDEF record into the NEWPLEX
*        CPLEXDEF record.
*
         USING CPLEXDEF,R2              Map the record
         MVC   CPLEXDEF_CICSPLEX,NEWPLEX                                X
                                        Set CICSplex name to NEWPLEX
         MVC   CPLEXDEF_DESC,=CL30'API cloned from'                     X
                                        Modify CICSPlex ....
         MVC   CPLEXDEF_DESC+16(8),OLDPLEX                              X
                                        .... description
         MVC   NEWPLXD(CPLEXDEF_TBL_LEN),0(R2)                          X
                                        Save NEWPLEX def and len ....
         MVC   NEWPLXDL,=A(CPLEXDEF_TBL_LEN)                            X
                                        .... for possible later REMOVE
```

```
*
*        Here we start using the CREATE subroutine.  This subroutine
*        will cause a CPSM Resource Table record to be built.  Variable
*        OBJECT needs to be preset to the Resource Table name, the
*        Resource Table record to be built must be pointed to by R2
*        and must be filled out before called CREATE.
*
         BAS   R10,CREATE               CREATE NEWPLEX
         MVI   PLEXBLT,C'Y'             Indicate NEWPLEX now built
*
*        Here we start using the FREEBUF subroutine.  This subroutine
*        FREEMAINs the buffer into which we FETCHed the records.
```

```
*
         BAS   R10,FREEBUF              Free record storage
*
*        When a result set is built (in our program by either GET or
*        PERFORM) an id is associated with the result set and placed
*        into the variable pointed to by keyword RESULT (for GET we
*        are using variable RESULT - for PERFORM, RESULT2).  This is
*        done so that subsequent calls can reference the result set
*        built (e.g, FETCH can retrieve records for GET).  When we
*        are done using a result set, we must DISCARD it, so that
*        CPSM frees us resources allocated for the result set.
*        Note that we have not done this with the 2 previous GETs
*        we did since the object of them was to NOT get a result set.
*        If any of the previous GETs caused a result set to get built,
*        we DISCONNECT from CPSM - which causes all our resources to
*        be released - and exit.
*
         MVC   RESULTD,RESULT           Copy GET result set id for     X
                                        DISCARD
         BAS   R10,DISCARD              Discard the GET result set
         DROP  R2                       Drop mapping to CPLEXDEF rec
*---------------------------------------------------------------------*
*        Add CMASs in OLDPLEX to NEWPLEX.                             *
*                                                                     *
*        There is a CICSPLEX Resource Table record for each CMAS      *
*        that participates in the management of a plex.  We first     *
*        ask for all the CICSPLEX records for OLDPLEX, and use        *
*        this info to add the CMASs to the NEWPLEX.                   *
*---------------------------------------------------------------------*
*        Ask for the CICSPLEX records from the OLDPLEX.
*
         MVC   OBJECT,=CL8'CICSPLEX'
         MVC   CRITERIA(9),=CL9'PLEXNAME='
         MVC   CRITERIA+9(8),OLDPLEX
         MVI   CRITERIA+17,C'.'
         MVC   CRITLEN,=F'18'
         BAS   R10,GETOBJ               Build result set
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BNE   ERRGETO                  no - msgs and out
         BAS   R10,GETBUF               Get storage for records
         BAS   R10,FETCH                Points R2 to first record
         USING CICSPLEX,R2              Map the Resource Table
         L     R5,COUNT                 Will loop the number of        X
                                        returned CMASs
*        The MP CMAS is added to the CICSplex when the CPLEXDEF
*        record was CREATEd.  To add any other CMASs to the CICSplex
*        we issue a PERFORM against the CPLEXDEF record for NEWPLEX,
*        with a parm = CICSPLEX(newplex) CMAS(cmasname).
*
         MVC   ADDCPARM(ADDCLEN),ADDC   Build most of parm
         MVC   PARMLEN,=A(ADDCLEN)      Set its length
         MVC   ADDCPLEX,NEWPLEX         Add CICSplex name to parm
         MVC   OBJECT,=CL8'CPLEXDEF'    PERFORM against CPLEXDEF
ADDCMAS  DS    0H
         CLC   CICSPLEX_CMASNAME,MPCMAS CMAS = MPCMAS?
         BE    NOADDMP                  Yes - don't add it then
         MVC   ADDCCMAS,CICSPLEX_CMASNAME                              X
                                        Add CMAS name to PARM          X
                                        This comes from the CICSPLEX   X
                                        records.
```

```
*
*        Note that we already have the CICSPLEX result set active,
*        with the id in RESULT.  So here we will use RESULT2 for
*        result set that is built for each PERFORM.
*
         MVC   RESULT2,=F'0'            Always build new result set
         EXEC  CPSM PERFORM                                            X
                    OBJECT(OBJECT)                                     X
                    ACTION(=CL12'ASSIGN')                              X
                    PARM(ADDCPARM)                                     X
                    PARMLEN(PARMLEN)                                   X
                    RESULT(RESULT2)                                    X
                    CONTEXT(CONTEXT)                                   X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BNE   ERRPERF                  no - msgs and out
         MVC   RESULTD,RESULT2          Copy PERFORM result set id for X
                                        DISCARD
         BAS   R10,DISCARD              Discard the PERFORM result set
NOADDMP  DS    0H
*
*        We need to get to the next CICSPLEX record for the next CMAS.
*        The GETBUF subroutine places into variable RECLEN the length
*        of the Resource Table record.  We now add this to the address
*        of the current record to point to the next record.
*
         A     R2,RECLEN
         BCT   R5,ADDCMAS               Add the next CMAS
*
*        No more CICSPLEX records - discard the CICSPLEX result set
*        and continue on.
*
         BAS   R10,FREEBUF              Free FETCHed record storage
         MVC   OBJECT,=CL8'CICSPLEX'    For possible DISCARD error msg
         MVC   RESULTD,RESULT           Copy GET result set id for     X
                                        DISCARD
         BAS   R10,DISCARD              Discard the GET result set
         DROP  R2                       Drop mapping to CICSPLEX rec
```

```
*---------------------------------------------------------------------*
*        Take all defs in OLDPLEX and put into NEWPLEX.               *
*                                                                     *
*        We have a list of all CICSplex Resource Table names.  We     *
*        loop through this list, getting all the records for a        *
*        specific Resource Table from the OLDPLEX and adding them     *
*        to the NEWPLEX.                                              *
*---------------------------------------------------------------------*
         MVC   CRITLEN,=F'0'            Want all records from each     X
                                        Resource Table - so we don't   X
                                        want a CRITERIA for GET.
         LA    R3,DEFNUM                Get number of Resource Tables
         LA    R4,DEFLIST               Point R4 to first Resource     X
                                        Table in list
BLDLOOP  DS    0H
         MVC   OBJECT,0(R4)             Move in Resource Table name
*
*        Get old data -  set CONTEXT to OLDPLEX.
*
         MVC   CONTEXT,OLDPLEX
         MVC   SCOPE,OLDPLEX
         BAS   R10,GETOBJ               Build result set
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BE    GOTDEFS                  Yes - FETCH and add
         CLC   RESPONSE,EYUVALUE(NODATA) No records returned?
         BE    NODATA                   Yes - on to next Resource Tab
         B     ERRGETO                  GET error - msgs and out
GOTDEFS  DS    0H
         BAS   R10,GETBUF               Get storage for records
         BAS   R10,FETCH                Point R2 to first record
         L     R5,COUNT                 Load number of records for loop
*
*        Add new data - set CONTEXT to NEWPLEX.
*
         MVC   CONTEXT,NEWPLEX
CRELOOP  DS    0H
*
*        We need to check if the object being created is a RTAINAPS
*        table.  If it is, we need to check if the SCOPE is the
*        OLDPLEX name - and if so, change it to the NEWPLEX name.
*        The RTAINAPS table is the only resource table in our list
*        that may have the OLDPLEX specified as a SCOPE.
*
         CLC   OBJECT,=CL8'RTAINAPS'    Creating an RTAINAPS?
         BNE   CRELOOP2                 No, just CREATE it
         USING RTAINAPS,R2              May to the record
         CLC   RTAINAPS_SCOPE,OLDPLEX   Is SCOPE equal to OLDPLEX?
         BNE   CRELOOP2                 No, don't change record
         MVC   RTAINAPS_SCOPE,NEWPLEX   Alter SCOPE to NEWPLEX
         DROP  R2                       Drop mapping to RTAINAPS rec
CRELOOP2 DS    0H
         BAS   R10,CREATE               CREATE record in NEWPLEX
         A     R2,RECLEN                Point to next record
         BCT   R5,CRELOOP               Loop
         BAS   R10,FREEBUF              Release record storage
         MVC   RESULTD,RESULT           Copy GET result set id for     X
                                        DISCARD
         BAS   R10,DISCARD              Discard the GET result set
NODATA   DS    0H
         LA    R4,8(,R4)                Point to next Resource Table
         BCT   R3,BLDLOOP               Do next Resource Table
*
*        We have gone through all the Resource Tables ok.  Set
*        the return code to 0.
*
         MVC   RETCODE,=F'0'
```

```
*---------------------------------------------------------------------*
*        Disconnect the connection and exit the program.              *
*---------------------------------------------------------------------*
EXITDISC DS    0H
         EXEC  CPSM DISCONNECT                                         X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
EXIT     DS    0H
         CLOSE (SYSPRINT)
*---------------------------------------------------------------------*
*        Unchain save area, FREEMAIN working storage, and restore     *
*        registers.                                                   *
*---------------------------------------------------------------------*
         L     R2,RETCODE               Retrieve return code
         L     R13,4(,R13)
         L     R1,8(,R13)
         FREEMAIN R,A=(R1),LV=WORKLEN
         L     R14,12(,R13)
         LR    R15,R2
         LM    R0,R12,20(R13)
         LA    R15,0
         BR    R14
```

```
*---------------------------------------------------------------------*
*        Error routines.                                              *
*---------------------------------------------------------------------*
ERRCON   DS    0H
         MVC   OUTLINE,=CL80'Error: Connecting to the API'
         BAS   R9,PUTMSG
         BAS   R10,DORR                 Format and msg RESPONSE/REASON
         B     EXIT                     Exit
ERRNISPC DS    0H
         MVC   OUTLINE,=CL80'Error: NEWPLEX is already defined as a CICX
               Splex or CMAS'
         BAS   R9,PUTMSG
         B     EXITDISC                 DISCONNECT and exit
ERRNISC  DS    0H
         MVC   OUTLINE,=CL80'Error: NEWPLEX is already defined as a CICX
               S system in the OLDPLEX'
         BAS   R9,PUTMSG
         B     EXITDISC                 DISCONNECT and exit
ERRNISS  DS    0H
         MVC   OUTLINE,=CL80'Error: NEWPLEX is already defined as a CICX
               S system group in the OLDPLEX'
         BAS   R9,PUTMSG
         B     EXITDISC                 DISCONNECT and exit
ERRPERF  DS    0H
         MVC   OUTLINE,=CL80'Error: Adding a CMAS to the NEWPLEX'
         BAS   R9,PUTMSG
         MVC   OUTLINE,=CL80' '
         MVC   OUTTXT1,=CL10'CMASNAME:'
         MVC   OUTDAT1,ADDCCMAS
         BAS   R9,PUTMSG
         BAS   R10,DORR                 Format and msg RESPONSE/REASON
         B     EXITERR
ERRGETO  DS    0H
         MVC   OUTLINE,=CL80'Error: GETting an object'
         BAS   R9,PUTMSG
         B     DOOBJMSG
ERRQUERY DS    0H
         MVC   OUTLINE,=CL80'Error: QUERYing a record size.'
         BAS   R9,PUTMSG
         B     DOOBJMSG
ERRFETCH DS    0H
         MVC   OUTLINE,=CL80'Error: FETCHing an object.'
         BAS   R9,PUTMSG
         B     DOOBJMSG
ERRCREAT DS    0H
         MVC   OUTLINE,=CL80'Error: CREATEing an object.'
         BAS   R9,PUTMSG
         B     DOOBJMSG
ERRDISCA DS    0H
```

```
         MVC   OUTLINE,=CL80'Error: DISCARDing object.'
         BAS   R9,PUTMSG
DOOBJMSG DS    0H
         MVC   OUTLINE,=CL80' '
         MVC   OUTTXT1,=CL10'OBJECT:'
         MVC   OUTDAT1,OBJECT
         BAS   R9,PUTMSG
         BAS   R10,DORR
EXITERR  DS    0H
         CLI   PLEXBLT,C'Y'             Did we CREATE the NEWPLEX?
         BNE   EXITDISC                 No - just DISCONNECT and exit
```

```
*
*        We had already CREATEd the NEWPLEX when an error occurred
*        so we want to delete the NEWPLEX before ending our program.
*
         EXEC  CPSM REMOVE                                             X
                    OBJECT(=CL8'CPLEXDEF')                             X
                    FROM(NEWPLXD)                                      X
                    LENGTH(NEWPLXDL)                                   X
                    CONTEXT(MPCMAS)                                    X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BE    EXITDISC                 Yes - DISCONNECT and exit
         MVC   OUTLINE,=CL80'Error: REMOVEing NEWPLEX.'
         BAS   R9,PUTMSG
         BAS   R10,DORR
         B     EXITDISC                 DISCONNECT and exit
*---------------------------------------------------------------------*
*        End of error routines.                                       *
*---------------------------------------------------------------------*
*        Subroutines.                                                 *
*---------------------------------------------------------------------*
PUTMSG   DS    0H
         PUT   SYSPRINT,OUTLINE
         BR    R9
DORR     DS    0H
*---------------------------------------------------------------------*
*        Subroutine: DORR                                             *
*        Entry:      Via BAS R10,DORR                                 *
*        Function:   Put out error messages indicating what function  *
*                    failed and the RESPONSE and REASON from that     *
*                    function.                                        *
*        Processing: -  Format the EXEC CPSM RESPONSE and move to the *
*                       OUTLINE.                                      *
*                    -  Format the EXEC CPSM REASON and move to the   *
*                       OUTLINE.                                      *
*                    -  Call the PUTMSG subroutine to send the        *
*                       RESPONSE/REASON data to SYSPRINT.             *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         MVC   OUTLINE,=CL80' '          clear format area
         MVC   OUTTXT1,=CL10'RESPONSE:'  move in ....
         L     R3,RESPONSE               load up the RESPONSE
         CVD   R3,DOUBLE                 convert to decimal
         MVC   OUTDAT1(6),=XL6'402020202120' move in EDIT pattern
         ED    OUTDAT1(6),DOUBLE+5       EDIT RESPONSE to format area
         MVC   OUTTXT2,=CL10'REASON:'    .... constant data
         L     R3,REASON                 load up the REASON
         CVD   R3,DOUBLE                 convert to decimal
         MVC   OUTDAT2(6),=XL6'402020202120' move in EDIT pattern
         ED    OUTDAT2(6),DOUBLE+5       EDIT REASON to format area
         BAS   R9,PUTMSG                 SEND it
         MVC   OUTLINE,=CL80' '          clear out OUTLINE again
         BAS   R9,PUTMSG                 put out blank line
         BR    R10                       return to caller
```

```
GETOBJ   DS    0H
*---------------------------------------------------------------------*
*        Subroutine: GETOBJ                                           *
*        Entry:      Via BAS R10,GETOBJ                               *
*        Function:   Issue the EXEC CPSM GET command to create a      *
*                    result set for a specific object.  Note that     *
*                    all operands for GET must be preset in           *
*                    mainline code - except for RESULT.               *
*        Processing: -  Clear out the result set id - RESULT - so     *
*                       that a new result set is always built.  It    *
*                       is the responsibility of mainline to DISCARD  *
*                       any previous result set for GET.              *
*                    -  Determine if the GET request has a CRITERIA   *
*                       and use the proper EXEC CPSM GET call.        *
*                    -  Note that GETOBJ does not check the RESPONSE  *
*                       from CPSM - this is done in mainline.         *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         MVC   RESULT,=F'0'             Always get new result set
         CLC   CRITLEN,=F'0'
         BE    GETNOCRT
         EXEC  CPSM GET                                                X
                    OBJECT(OBJECT)                                     X
                    CRITERIA(CRITERIA)                                 X
                    LENGTH(CRITLEN)                                    X
                    COUNT(COUNT)                                       X
                    RESULT(RESULT)                                     X
                    THREAD(THREAD)                                     X
                    CONTEXT(CONTEXT)                                   X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         BR    R10
GETNOCRT DS    0H
         EXEC  CPSM GET                                                X
                    OBJECT(OBJECT)                                     X
                    COUNT(COUNT)                                       X
                    RESULT(RESULT)                                     X
                    THREAD(THREAD)                                     X
                    CONTEXT(CONTEXT)                                   X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         BR    R10
```

```
GETBUF   DS    0H
*---------------------------------------------------------------------*
*        Subroutine: GETBUF                                           *
*        Entry:      Via BAS R10,GETBUF                               *
*        Function:   Get a buffer to hold all the records contained   *
*                    in the last result set we build though GET.      *
*        Processing: -  Issue EXEC CPSM QUERY to get the length of    *
*                       the Resource Table record.  We use the same   *
*                       OBJECT and RESULT from the GET.  Variable     *
*                       RECLEN gets the record length.                *
*                    -  Check the RESPONSE from QUERY and issue msgs  *
*                       and EXIT if not OK.                           *
*                    -  Multiple the RECLEN times the COUNT (returned *
*                       from last GET) to determine the buffer size   *
*                       required and GETMAIN it.                      *
*                    -  Save the buffer length (BUFLEN) and buffer    *
*                       address (BUFFER) for the FREEMAIN call in     *
*                       the FREEBUF subroutine.                       *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         EXEC  CPSM QUERY                                              X
                    OBJECT(OBJECT)                                     X
                    DATALENGTH(RECLEN)                                 X
                    RESULT(RESULT)                                     X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)    RESPONSE OK?
         BNE   ERRQUERY                 No - msgs and out
         L     R0,RECLEN
         L     R1,COUNT
         MR    R0,R0
         GETMAIN R,LV=(R1)
         ST    R0,BUFLEN
         ST    R1,BUFFER
         BR    R10
FREEBUF  DS    0H
*---------------------------------------------------------------------*
*        Subroutine: FREEBUF                                          *
*        Entry:      Via BAS R10,FREEBUF                              *
*        Function:   To FREEMAIN the buffer created to hold the       *
*                    records from the last result set we built .      *
*                    through GET.                                     *
*        Processing: -  Use BUFLEN and BUFFER from GETBUF, FREEMAIN   *
*                       the buffer area.                              *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         L     R0,BUFLEN
         L     R1,BUFFER
         FREEMAIN R,A=(R1),LV=(R0)
         BR    R10
```

```
FETCH    DS    0H
*---------------------------------------------------------------------*
*        Subroutine: FETCH                                            *
*        Entry:      Via BAS R10,FETCH                                *
*        Function:   Issue the EXEC CPSM FETCH command to retrieve    *
*                    the result set created by the last GET.          *
*                    mainline code - except for RESULT.               *
*        Processing: -  For FETCH we must provide a receiving area    *
*                       and length.  We put in the area length into   *
*                       R2 and the area length in variable LENGTH.    *
*                       Note that we got both the area and length     *
*                       in the GETBUF routine.                        *
*                    -  Issue the FETCH request using the result set  *
*                       id - RESULT - from the last GET.              *
*                    -  Check the RESPONSE - if not OK, issue msgs    *
*                       and exit.                                     *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         L     R2,BUFFER
         MVC   LENGTH,BUFLEN
         EXEC  CPSM FETCH                                              X
                    ALL                                                X
                    INTO(0(,R2))                                       X
                    LENGTH(LENGTH)                                     X
                    COUNT(COUNT)                                       X
                    RESULT(RESULT)                                     X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)
         BNE   ERRFETCH
         BR    R10
```

```
CREATE   DS    0H
*---------------------------------------------------------------------*
*        Subroutine: CREATE                                           *
*        Entry:      Via BAS R10,CREATE                               *
*        Function:   Issue the EXEC CPSM CREATE to build a Resource   *
*                    Table record.                                    *
*        Processing: -  Place the length of the record to be build    *
*                       (RECLEN from GETBUF) into variable LENGTH.    *
*                       R2 should have been set by mainline to point  *
*                       to the record itself.                         *
*                    -  When CREATEing a LNKxxCG record (spec to      *
*                       group link) we need to specify a parm -       *
*                       NONE. - to indicate that we only want the     *
*                       CREATE to associate the spec to the group.    *
*                       Any systems in the group that need to be      *
*                       added to the spec have already been done      *
*                       by CREATE of LNKxxCS records (spec to         *
*                       system link).  If this is a LNKxxCG record,   *
*                       set the PARM and PARMLENgth.                  *
*                    -  Issue the proper format of EXEC CPSM CREATE   *
*                       (either with PARM/PARMLEN or without).        *
*                    -  Check the RESPONSE - if not OK, issue msgs    *
*                       and exit.                                     *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         MVC   LENGTH,RECLEN
         CLC   OBJECT(4),=CL4'LNKS'
         BNE   CRENOPRM
         CLC   OBJECT+6(2),=CL2'CG'
         BNE   CRENOPRM
         MVC   PARM,=CL5'NONE.'
         MVC   PARMLEN,=F'5'
         EXEC  CPSM CREATE                                             X
                    OBJECT(OBJECT)                                     X
                    FROM(0(,R2))                                       X
                    LENGTH(LENGTH)                                     X
                    PARM(PARM)                                         X
                    PARMLEN(PARMLEN)                                   X
                    THREAD(THREAD)                                     X
                    CONTEXT(CONTEXT)                                   X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         B     CRECHKRR
CRENOPRM DS    0H
```

```
         EXEC  CPSM CREATE                                             X
                    OBJECT(OBJECT)                                     X
                    FROM(0(,R2))                                       X
                    LENGTH(LENGTH)                                     X
                    THREAD(THREAD)                                     X
                    CONTEXT(CONTEXT)                                   X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
CRECHKRR DS    0H
         CLC   RESPONSE,EYUVALUE(OK)
         BNE   ERRCREAT
         BR    R10
```

```
DISCARD  DS    0H
*---------------------------------------------------------------------*
*        Subroutine: DISCARD                                          *
*        Entry:      Via BAS R10,DISCARD                              *
*        Function:   Issue the EXEC CPSM DISCARD to discard a result  *
*                    set built by CPSM.  In our program, both GET     *
*                    and PERFORM build result sets.                   *
*        Processing: -  Issue EXEC CPSM DISCARD for the result set.   *
*                       The result set id must be placed into         *
*                       RESULTD by mainline.                          *
*                    -  Check the RESPONSE - if not OK, issue msgs    *
*                       and exit.                                     *
*                    -  Return to caller.                             *
*---------------------------------------------------------------------*
         EXEC  CPSM DISCARD                                            X
                    RESULT(RESULTD)                                    X
                    THREAD(THREAD)                                     X
                    RESPONSE(RESPONSE)                                 X
                    REASON(REASON)
         CLC   RESPONSE,EYUVALUE(OK)
         BNE   ERRDISCA
         BR    R10
*---------------------------------------------------------------------*
*        End of subroutines.                                          *
*---------------------------------------------------------------------*
```

```
*---------------------------------------------------------------------*
* Copy the CPSM definitions from OrigPlex to NewPlex                  *
*                                                                     *
* Following is a list of all CPSM Resource Tables that will           *
* be copied into NewPlex. The order that they are in (which           *
* is the order they will be built in our program) is                  *
* important, since some Resource Tables will reference other          *
* Resource Tables previously built. The order of the following        *
* list is OK for the current release of CPSM.                         *
*                                                                     *
*---------------------------------------------------------------------*
DEFLIST  DS    0C
         DC    CL8'PERIODEF'    Time period definitions
         DC    CL8'ACTION  '    RTA action definitions
         DC    CL8'CSYSDEF '    CICS system definitions
         DC    CL8'CSYSGRP '    CICS system group definitions
         DC    CL8'CSGLCGCS'    CICS systems in groups links
         DC    CL8'CSGLCGCG'    CICS groups in groups links
         DC    CL8'MONDEF  '    Monitor definitions
         DC    CL8'MONGROUP'    MON group definitions
         DC    CL8'MONSPEC '    MON specification definitions
         DC    CL8'MONINGRP'    MON def in MON group links
         DC    CL8'MONINSPC'    MON spec to MON group links
         DC    CL8'LNKSMSCS'    MON spec to CICS system links
         DC    CL8'LNKSMSCG'    MON spec to CICS group links
         DC    CL8'EVALDEF '    RTA evaluation definitions
         DC    CL8'RTADEF  '    Real time analysis definitions
         DC    CL8'STATDEF '    User status probe definitions
         DC    CL8'RTAGROUP'    RTA group definitions
         DC    CL8'RTASPEC '    RTA specification definitions
         DC    CL8'RTAINGRP'    RTADEF in RTA group links
         DC    CL8'STAINGRP'    STATDEF in RTA group links
         DC    CL8'RTAINSPC'    RTA spec to RTA group links
         DC    CL8'LNKSRSCS'    RTA spec to CICS group links
         DC    CL8'LNKSRSCG'    RTA spec to CICS system links
         DC    CL8'APSPEC  '    RTA/APM specification defs
         DC    CL8'RTAINAPS'    RTA/APM spec to RTA group links
         DC    CL8'CMDMPAPS'    RTA spec to primary CMAS links
         DC    CL8'CMDMSAPS'    RTA spec to secondary CMAS links
         DC    CL8'TRANGRP '    transaction group definitions
         DC    CL8'WLMDEF  '    Workload definitions
         DC    CL8'WLMGROUP'    WLM group definitions
         DC    CL8'WLMSPEC '    WLM specification definitions
         DC    CL8'DTRINGRP'    Transactions in trangrp links
         DC    CL8'WLMINGRP'    WLM def in WLM group links
         DC    CL8'WLMINSPC'    WLM spec to WLM group links
         DC    CL8'LNKSWSCS'    WLM spec to CICS group links
         DC    CL8'LNKSWSCG'    WLM spec to CICS system links
DEFNUM   EQU   (*-DEFLIST)/8
ADDC     DS    0X
         DC    CL09'CICSPLEX('
         DC    CL08' '
         DC    CL07') CMAS('
         DC    CL08' '
         DC    CL02').'
ADDCLEN  EQU   *-ADDC
SYSPRINT DCB   DDNAME=SYSPRINT,DSORG=PS,MACRF=PM
WORKSTOR DSECT
SAVEAREA DS    18F
DFHEIPL  DS    13F
         DS    51F
```

```
DOUBLE   DS    D
RETCODE  DS    F
RESPONSE DS    F
REASON   DS    F
THREAD   DS    F
RESULT   DS    F
RESULT2  DS    F
RESULTD  DS    F
COUNT    DS    F
LENGTH   DS    F
PARMLEN  DS    F
BUFLEN   DS    F
BUFFER   DS    F
RECLEN   DS    F
NEWPLXDL DS    F
CRITLEN  DS    F
CRITERIA DS    CL80
CONTEXT  DS    CL8
SCOPE    DS    CL8
OBJECT   DS    CL8
OLDPLEX  DS    CL8
NEWPLEX  DS    CL8
MPCMAS   DS    CL8
OUTLINE  DS    0CL80
OUTTXT1  DS    CL10
OUTDAT1  DS    CL8
         DS    CL2
OUTTXT2  DS    CL10
OUTDAT2  DS    CL8
         DS    CL42
PARM     DS    CL5
PLEXBLT  DS    CL1
ADDCPARM DS    0XL(ADDCLEN)
         DS    CL09
ADDCPLEX DS    CL08
         DS    CL07
ADDCCMAS DS    CL08
         DS    CL02
         DS    D
NEWPLXD  DS    XL(CPLEXDEF_TBL_LEN)
WORKLEN  EQU   *-WORKSTOR
         COPY  CPLEXDEF
         COPY  CICSPLEX
         COPY  RTAINAPS
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
```

```
R15      EQU   15
         END   EYUAAPI3
```

EYULAPI4

Program EYULAPI4 is written in COBOL for the CICS environment.

About EYULAPI4

This program does the following processing:

- Establishes a connect to the API.
- Creates a BAS definition for a TS Model (TSMDEF) specifying a version of 1.
- Creates a result set containing the previously defined TSMDEF.
- Issues a PERFORM OBJECT command to INSTALL the TSMDEF into the target scope.
- Terminates the API connection.
- BAS errors are processed using BINCONRS, BINCONSC, and BINSTERR resource table records.

Commands used: CONNECT, CREATE, GET, PERFORM OBJECT, FEEDBACK, FETCH, TERMINATE, TRANSLATE

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. EYULAPI4
      ******************************************************************
      *                                                                *
      * MODULE NAME = EYULAPI4                                         *
      *                                                                *
      * DESCRIPTIVE NAME = CPSM SAMPLE API PROGRAM 4                   *
      *                    (SAMPLE COBOL VERSION)                      *
      *                                                                *
      *  COPYRIGHT = Licensed Materials - Property of IBM              *
      *              5695-081                                          *
      *              (C) Copyright IBM Corp. 1995, 1997                *
      *              All Rights Reserved                               *
      *                                                                *
      *              US Government Users Restricted Rights - Use,      *
      *              duplication or disclosure restricted by GSA ADP   *
      *              Schedule Contract with IBM Corp.                  *
      *                                                                *
      * STATUS = %CP00                                                 *
      *                                                                *
      * FUNCTION =                                                     *
      *                                                                *
      * TO PROVIDE AN EXAMPLE OF THE USE OF THE FOLLOWING EXEC CPSM    *
      * COMMANDS: CONNECT, CREATE, FEEDBACK, FETCH, GET,               *
      *           PERFORM OBJECT, TERMINATE.                           *
      *                                                                *
      * WHEN INVOKED, THE PROGRAM DEPENDS UPON THE VALUES HELD IN THE  *
      * W-CONTEXT AND W-SCOPE DECLARATIONS WHEN ESTABLISHING A         *
      * CONNECTION WITH CICSPLEX SM. THEY MUST TAKE THE FOLLOWING      *
      * VALUES:                                                        *
      *                                                                *
      * W-CONTEXT  = THE NAME OF A CMAS OR CICSPLEX. REFER TO THE      *
      *              DESCRIPTION OF THE EXEC CPSM CONNECT COMMAND      *
      *              FOR FURTHER INFORMATION REGARDING THE CONTEXT     *
      *              OPTION.                                           *
      *                                                                *
      * W-SCOPE    = THE NAME OF A CICSPLEX, CICS SYSTEM, OR CICS      *
      *              SYSTEM GROUP WITHIN THE CICSPLEX. REFER TO THE    *
      *              DESCRIPTION OF THE EXEC CPSM CONNECT COMMAND      *
      *              FOR FURTHER INFORMATION REGARDING THE SCOPE       *
      *              OPTION.                                           *
      *                                                                *
      * THIS SAMPLE REQUIRES NO PARAMETERS AT INVOCATION TIME.         *
      *                                                                *
      * WHEN CREATING THE BAS DEFINITION THE PROGRAM DEPENDS UPON THE  *
      * VALUES HELD IN THE W-DEFNAME AND W-DEFPREFIX DECLARATIONS.     *
      * THEY MUST TAKE THE FOLLOWING VALUES:                           *
      *                                                                *
      * W-DEFNAME  = THE NAME OF THE CREATED BAS DEFINITION. A         *
      *              1 TO 8 CHARACTER VALUE.                           *
      *                                                                *
      * W-DEFPFIX  = THE MODEL PREFIX OF THE CREATED BAS DEFINITION.  *
      *              A 1 TO 16 CHARACTER VALUE.                       *
      *                                                               *
      *                                                               *
```

```
      * WHEN INSTALLING THE BAS DEFINITION THE PROGRAM USES THE       *
      * VALUE HELD IN THE W-TSCOPE DECLARATION AS THE TARGET FOR      *
      * THE INSTALL OPERATION. IT MUST TAKE THE FOLLOWING VALUE :     *
      *                                                               *
      * W-TSCOPE   = THE NAME OF A CICS SYSTEM, OR CICS               *
      *              SYSTEM GROUP WITHIN THE CICSPLEX. REFER TO THE   *
      *              DESCRIPTION OF THE TARGET PARAMETER OF AN        *
      *              INSTALL ACTION IN THE RESOURCE TABLE REFERENCE   *
      *              FOR FURTHER INFORMATION REGARDING THE TARGET     *
      *              SCOPE VALUE.                                     *
      *                                                               *
      *                                                               *
      * THE SAMPLE ESTABLISHES AN API CONNECTION AND ISSUES A CREATE  *
      * COMMAND TO CREATE A BAS DEFINITION. A GET COMMAND IS ISSUED   *
      * TO OBTAIN A RESULT SET CONTAINING THE CREATED BAS DEFINITION. *
      *                                                               *
      * USING THE PERFORM OBJECT ACTION(INSTALL) COMMAND EACH RECORD  *
      * IN THE RESULT SET IS INSTALLED INTO THE TARGET SCOPE          *
      * IDENTIFIED BY THE W-SCOPE DECLARATION.                        *
      *                                                               *
      * FINALLY, THE API CONNECTION IS TERMINATED.                    *
      *                                                               *
      * ANY BAS ERRORS ARE REPORTED USING THE BINCONRS, BINCONSC, AND *
      * BINSTERR RESOURCE TABLES.                                     *
      *                                                               *
```

```
      * NOTES :                                                       *
      *   DEPENDENCIES = S/390, CICS                                  *
      *   RESTRICTIONS = NONE                                         *
      *   REGISTER CONVENTIONS =                                      *
      *   MODULE TYPE  = EXECUTABLE                                   *
      *   PROCESSOR    = COBOL                                        *
      *   ATTRIBUTES   = READ ONLY, SERIALLY REUSABLE                 *
      *                                                               *
      * ------------------------------------------------------------- *
      * ENTRY POINT = EYULAPI4                                        *
      *                                                               *
      * PURPOSE = ALL FUNCTIONS.                                      *
      *                                                               *
      * LINKAGE = FROM CICS EITHER WITH EXEC CICS LINK OR AS A CICS   *
      *           TRANSACTION.                                        *
      *                                                               *
      * INPUT   = NONE.                                               *
      *                                                               *
      * ------------------------------------------------------------- *
       ENVIRONMENT DIVISION.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
      *-------------------------------------------------------------*
      *    CHANGE W-CONTEXT AND W-SCOPE TO MATCH YOUR INSTALLATION  *
      *    CHANGE W-DEFNAME AND W-DEFPFIX FOR THE CREATE COMMAND.   *
      *    CHANGE W-TSCOPE FOR THE PERFORM OBJECT COMMAND.          *
      *-------------------------------------------------------------*
       01 W-CONTEXT         PIC X(8) VALUE 'RTGA    '.
       01 W-SCOPE           PIC X(8) VALUE 'RTGA    '.
       01 W-DEFNAME         PIC X(8) VALUE 'EYULAPI4'.
       01 W-DEFPFIX         PIC X(16) VALUE 'EYUL*           '.
       01 W-TSCOPE          PIC X(8) VALUE 'RTGF    '.
      *-------------------------------------------------------------*
```

```
       01 W-RESPONSE        PIC S9(8) USAGE BINARY.
       01 W-REASON          PIC S9(8) USAGE BINARY.
       01 W-BUFFER          PIC X(32767).
       01 W-BUFFERLEN       PIC S9(8) COMP.
       01 W-FBBUFF          PIC X(248).
       01 W-FBTTKN          PIC S9(8) COMP.
       01 W-THREAD          PIC S9(8) USAGE BINARY.
       01 W-RESULT          PIC S9(8) USAGE BINARY.
       01 W-RECCNT          PIC S9(8) USAGE BINARY.
       01 W-CRITERIA        PIC X(80) VALUE SPACES.
       01 W-CRITERIALEN     PIC S9(8) USAGE BINARY.
       01 W-PARM            PIC X(80) VALUE SPACES.
       01 W-PARMLEN         PIC S9(8) USAGE BINARY.
       01 W-MSG-TEXT.
```

```
   02 W-TEXT          PIC X(80) VALUE SPACES.
   02 W-LINECTL       PIC X(1) VALUE X'13'.
 01 ARRAYS.
   02 CH8ARR          OCCURS 20 TIMES PIC X(8).
   02 FULLARR         OCCURS 60 TIMES PIC S9(8) COMP.
 01 III               PIC S9(8) VALUE ZERO.
 01 CODEV             PIC S9(8) COMP.
 01 CHARV             PIC X(12).
 01 LASTCMD           PIC X(20).
 01 LASTTHR           PIC S9(8) COMP.
 01 LASTRES           PIC S9(8) COMP VALUE 0.
 01 BINZERO           PIC X(1)  VALUE X'00'.
 01 BLNKPAD           PIC X(40)
     VALUE '                                        '.
 01 FBCHAR2           PIC X(2).
 01 FBHALF4           REDEFINES FBCHAR2.
   02 FBHALF          PIC S9(4) COMP.
 01 PICZZZ9A          PIC ZZZ9.
 01 PICZZZ9B          PIC ZZZ9.
 01 PICZZZ9           PIC ZZZ9.
 01 PYCZZZ9           PIC ZZZ9.
 01 PIKZZZ9           PIC ZZZ9.
 01 PYKZZZ9           PIC ZZZ9.
 01 PICZZZZZZZ9       PIC ZZZZZZZ9.
 01 CHR8              PIC X(8).
 01 CHR12             PIC X(12).
 01 CHAR6             PIC X(6).
 01 CHAR12            PIC X(12).
* Include the resource table copybooks...
 COPY TSMDEF.
 COPY FEEDBACK.
 COPY BINCONRS.
 COPY BINCONSC.
 COPY BINSTERR.
```

```
****************************
* Start of LINKAGE section *
****************************
 LINKAGE SECTION.

 PROCEDURE DIVISION.
 EYULAPI4-START SECTION.
 EYULAPI4-00.

*-------------------------------------------------------------*
*    OBTAIN A CPSM API CONNECTION.                            *
*                                                             *
*    THE API WILL RETURN A TOKEN IDENTIFYING THE THREAD IN    *
*    VARIABLE W-THREAD.                                       *
*-------------------------------------------------------------*
     MOVE 'Establishing Connection...' TO W-TEXT.
*    DISPLAY W-TEXT.
     EXEC CICS SEND FROM(W-TEXT) LENGTH(81) ERASE END-EXEC.
     EXEC CPSM CONNECT
               CONTEXT(W-CONTEXT)
               SCOPE(W-SCOPE)
               VERSION('0140')
               THREAD(W-THREAD)
               RESPONSE(W-RESPONSE)
               REASON(W-REASON)
     END-EXEC.
     IF W-RESPONSE NOT = EYUVALUE(OK) GO TO NO-CONNECT.
```

```
*-------------------------------------------------------------*
*    CREATE A TS MODEL DEFINITION (TSMDEF)                    *
*                                                             *
*    A TSMDEF is created with a version of 1.                 *
*-------------------------------------------------------------*
     INITIALIZE TSMDEF.
     MOVE X'01' TO DEFVER OF TSMDEF.
     MOVE W-DEFNAME TO NAME-R OF TSMDEF.
     MOVE W-DEFPFIX TO PREFIX OF TSMDEF.
     MOVE DFHVALUE(AUXILIARY) TO LOCATION OF TSMDEF.
     MOVE EYUVALUE(NO) TO RECOVERY OF TSMDEF.
     MOVE EYUVALUE(NO) TO SECURITY-R OF TSMDEF.
     MOVE 'Sample TSMDEF definition' TO DESCRIPTION OF TSMDEF.
* Copy the definition into our buffer...
     MOVE TSMDEF TO W-BUFFER.
```

```
           MOVE TSMDEF-TBL-LEN TO W-BUFFERLEN.
           MOVE 'Creating TSMDEF...' TO W-TEXT.
      *    DISPLAY W-TEXT.
           EXEC CICS SEND FROM(W-TEXT) LENGTH(81) WAIT END-EXEC.
           EXEC CPSM CREATE
                     OBJECT('TSMDEF')
                     FROM(W-BUFFER)
                     LENGTH(W-BUFFERLEN)
                     THREAD(W-THREAD)
                     RESPONSE(W-RESPONSE)
                     REASON(W-REASON)
           END-EXEC.
           MOVE 'CREATE' TO LASTCMD.
           MOVE W-THREAD TO LASTTHR.
           MOVE 0 TO LASTRES.
           IF W-RESPONSE NOT = EYUVALUE(OK) GO TO UNEXPECTED.
```

```
      *--------------------------------------------------------------*
      *    GET THE TSMDEF RESOURCE TABLE.                            *
      *                                                              *
      *    CREATE A RESULT SET CONTAINING ENTRIES FOR ALL TSMDEFS   *
      *    WITH NAMES EQUAL TO THE VALUE OF W-DEFNAME. .            *
      *    THE NUMBER OF ENTRIES MEETING THE CRITERIA IS RETURNED   *
      *    IN VARIABLE W-RECCNT.                                     *
      *--------------------------------------------------------------*
           MOVE 'Get the created TSMDEF Resource Table...' TO W-TEXT.
      *    DISPLAY W-TEXT.
           EXEC CICS SEND FROM(W-TEXT) LENGTH(81) WAIT END-EXEC.
           STRING 'NAME=' DELIMITED BY SIZE
                  W-DEFNAME DELIMITED BY SIZE
                  '.' DELIMITED BY SIZE
                  INTO W-CRITERIA.
           MOVE LENGTH OF W-CRITERIA TO W-CRITERIALEN.
           MOVE BINZERO TO W-RESULT.
           EXEC CPSM GET OBJECT('TSMDEF')
                         CRITERIA(W-CRITERIA)
                         LENGTH(W-CRITERIALEN)
                         COUNT(W-RECCNT)
                         RESULT(W-RESULT)
                         THREAD(W-THREAD)
                         RESPONSE(W-RESPONSE)
                         REASON(W-REASON)
           END-EXEC.
           IF W-RESPONSE NOT = EYUVALUE(OK) GO TO NO-GET.
```

```
      *--------------------------------------------------------------*
      *    INSTALL EACH RECORD INTO THE SCOPE IDENTIFIED BY THE      *
      *    VALUE OF W-TSCOPE.                                        *
      *--------------------------------------------------------------*
           MOVE W-RECCNT TO PICZZZZZZZ9.
           STRING 'Installing ' DELIMITED BY SIZE
                  PICZZZZZZZ9 DELIMITED BY SIZE
                  ' TSMDEF Entries...' DELIMITED BY SIZE
                  INTO W-TEXT.
      *    DISPLAY W-TEXT
           EXEC CICS SEND FROM(W-TEXT) LENGTH(81) WAIT END-EXEC.
           STRING '(USAGE(LOCAL) TARGET(' DELIMITED BY SIZE
                  W-TSCOPE DELIMITED BY SIZE
                  ')).' DELIMITED BY SIZE
                  INTO W-PARM.
           MOVE LENGTH OF W-PARM TO W-PARMLEN.

           EXEC CPSM PERFORM OBJECT('TSMDEF')
                     ACTION('INSTALL')
                     PARM(W-PARM)
                     PARMLEN(W-PARMLEN)
                     RESULT(W-RESULT)
                     THREAD(W-THREAD)
                     RESPONSE(W-RESPONSE)
                     REASON(W-REASON)
           END-EXEC.
           MOVE 'PERFORM OBJECT' TO LASTCMD.
           MOVE W-THREAD TO LASTTHR.
           MOVE W-RESULT TO LASTRES.
           IF W-RESPONSE NOT = EYUVALUE(OK) GO TO UNEXPECTED.

           MOVE 'Completed. Remove TSMDEF to re-run.' TO W-TEXT.
```

```
     GO TO SCRNLOG2.
```

```
**************************************************
* Branch here if an unexpected CPSM error occurs *
**************************************************
 UNEXPECTED.
     MOVE W-RESPONSE TO PICZZZ9.
     STRING '*** RESPONSE=' DELIMITED BY SIZE PICZZZ9
     DELIMITED BY SIZE BLNKPAD DELIMITED BY SIZE INTO W-TEXT.
     PERFORM SCRNLOG2.
     MOVE W-REASON TO PICZZZ9.
     STRING '*** REASON=' DELIMITED BY SIZE PICZZZ9
     DELIMITED BY SIZE BLNKPAD DELIMITED BY SIZE INTO W-TEXT.
     PERFORM SCRNLOG2.
     MOVE '*** Unexpected error condition arose' TO W-TEXT.
     PERFORM SCRNLOG2.
* Obtain FEEDBACK information
     IF LASTCMD = 'DISCONNECT' GO TO NOFEED.
     IF LASTCMD = 'FEEDBACK' GO TO NOFEED.
     IF LASTCMD = 'TERMINATE' GO TO NOFEED.
     STRING
     '*** Getting FEEDBACK data for ' DELIMITED BY SIZE
     LASTCMD DELIMITED BY SIZE
     INTO W-TEXT.
     PERFORM SCRNLOG2.
     STRING
     BLNKPAD DELIMITED BY SIZE
     BLNKPAD DELIMITED BY SIZE
     INTO W-TEXT.
* Get the FEEDBACK data
 GETFEED.
* Clear error result set count
     MOVE 0 TO FULLARR(1).
     PERFORM GETFB THROUGH EGETFB
* Display FEEDBACK information
* Display information
     IF W-RESPONSE = EYUVALUE(OK)
       PERFORM DISPFEED
       IF FULLARR(1) NOT = 0 PERFORM GETFERT THROUGH EGETFER END-I
-F
       IF LASTRES NOT = 0 GO TO GETFEED END-IF
       MOVE '*** End of FEEDBACK data' TO W-TEXT
       PERFORM SCRNLOG2
       GO TO NOFEED
     END-IF.
     MOVE W-RESPONSE TO PICZZZ9.
     MOVE W-REASON TO PYCZZZ9.
     STRING '*** FEEDBACK not available (' DELIMITED BY SIZE
     PICZZZ9 DELIMITED BY SIZE ',' DELIMITED BY SIZE
     PYCZZZ9 DELIMITED BY SIZE ')' DELIMITED BY SIZE
     BLNKPAD DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
 NOFEED.
     EXEC CICS DELAY FOR SECONDS(10) END-EXEC.
* Exit from test case
     EXEC CICS RETURN END-EXEC.
     GOBACK.
     EXIT.
```

```
*********************************************
* This subroutine obtains the FEEDBACK data *
*********************************************
 GETFB.
* Use exact buffer size
     MOVE FEEDBACK-TBL-LEN TO W-BUFFERLEN.
     IF LASTRES = 0 GO TO NORESULT.
 RESULT.
     EXEC CPSM FEEDBACK
               INTO(W-FBBUFF) LENGTH(W-BUFFERLEN)
               RESULT(LASTRES)
               THREAD(LASTTHR)
               RESPONSE(W-RESPONSE)
               REASON(W-REASON)
     END-EXEC.

* If command didn't execute, get FEEDBACK no result set
* Command didn't execute?
     IF W-RESPONSE = EYUVALUE(NODATA)
```

```
       MOVE 0 TO LASTRES
       GO TO NORESULT
     END-IF.
     GO TO ENDFBACK.
 NORESULT.
* Use exact buffer size
     MOVE FEEDBACK-TBL-LEN TO W-BUFFERLEN.
     EXEC CPSM FEEDBACK
               INTO(W-FBBUFF) LENGTH(W-BUFFERLEN)
               THREAD(LASTTHR)
               RESPONSE(W-RESPONSE)
               REASON(W-REASON)
     END-EXEC.

 ENDFBACK.
 EGETFB.
     EXIT.
```

```
********************************************************
* Branch here if FEEDBACK Error Result Token available *
********************************************************
 GETFERT.
     MOVE ERR-OBJECT OF FEEDBACK TO CH8ARR(1).
     STRING
     '*** Getting ' DELIMITED BY SIZE
     CH8ARR(1) DELIMITED BY SIZE
     ' error result set data for FEEDBACK' DELIMITED BY SIZE
     INTO W-TEXT.
     PERFORM SCRNLOG2.
 FERTRES.
* Use largest buffer size
     MOVE FEEDBACK-TBL-LEN TO W-BUFFERLEN.
     EXEC CPSM FETCH
               INTO(W-BUFFER) LENGTH(W-BUFFERLEN)
               RESULT(ERR-RESULT OF FEEDBACK)
               THREAD(LASTTHR)
               RESPONSE(W-RESPONSE)
               REASON(W-REASON)
     END-EXEC.

* Display FEEDBACK Error Result Token information
* Display information
     IF W-RESPONSE = EYUVALUE(OK)
       IF CH8ARR(1)= 'FEEDBACK'
         MOVE W-BUFFER TO W-FBBUFF
         PERFORM DISPFEED
       END-IF
       IF CH8ARR(1)= 'BINSTERR'
         PERFORM DISPBIER
       END-IF
       IF CH8ARR(1)= 'BINCONRS'
         PERFORM DISPBIRS
       END-IF
       IF CH8ARR(1)= 'BINCONSC'
         PERFORM DISPBISC
       END-IF
       GO TO FERTRES
     END-IF.
     MOVE W-RESPONSE TO PICZZZ9.
     MOVE W-REASON TO PYCZZZ9.
     STRING '*** FEEDBACK not available (' DELIMITED BY SIZE
     PICZZZ9 DELIMITED BY SIZE ',' DELIMITED BY SIZE
     PYCZZZ9 DELIMITED BY SIZE ')' DELIMITED BY SIZE
     BLNKPAD DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
 EGETFER.
     EXIT.
```

```
*************************************************
* This subroutine displays FEEDBACK information *
*************************************************
 DISPFEED.
     MOVE W-FBBUFF TO FEEDBACK.
     STRING BINZERO COMMAND OF FEEDBACK DELIMITED BY SIZE
     INTO FBCHAR2.
     MOVE FBHALF TO PICZZZ9.
     MOVE RESPONSE OF FEEDBACK TO PYCZZZ9.
     MOVE REASON OF FEEDBACK TO PIKZZZ9.
```

```
           MOVE RSLTRECID OF FEEDBACK TO PYKZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING 'Cmd=' PICZZZ9 ' Attr=' ATTRDATAVAL OF
           FEEDBACK ' Eib=' CEIBDATAVAL OF FEEDBACK ' Err='
           ERRCODEVAL OF FEEDBACK ' Rspn=' PYCZZZ9 ' Reas='
           PIKZZZ9 ' ResId=' PYKZZZ9
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE ERROR-CODE OF FEEDBACK TO PICZZZ9.
           MOVE CEIBRESP OF FEEDBACK TO PYCZZZ9.
           MOVE CEIBRESP1 OF FEEDBACK TO PIKZZZ9.
           MOVE CEIBFN OF FEEDBACK TO PYKZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING '  ECode=' PICZZZ9 ' RESP=' PYCZZZ9
           ' RESP1=' PIKZZZ9 ' EibFn=' PYKZZZ9 ' Obj='
           OBJECT-A OF FEEDBACK ' OAct=' OBJECT-ACT OF FEEDBACK
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE SPACES TO W-TEXT.
           STRING '  Att1=' ATTR-NM1 OF FEEDBACK ' 2='
           ATTR-NM2 OF FEEDBACK ' 3=' ATTR-NM3 OF FEEDBACK
           ' 4=' ATTR-NM4 OF FEEDBACK ' 5=' ATTR-NM5 OF
           FEEDBACK DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE ERR-COUNT OF FEEDBACK TO PICZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING '  FObj=' ERR-OBJECT OF FEEDBACK
           '  FCnt=' PICZZZ9
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE ERR-COUNT OF FEEDBACK TO FULLARR(1).
           EXIT.
```

```
      *************************************************
      * This subroutine displays BINSTERR information *
      *************************************************
       DISPBIER.
           MOVE W-BUFFER TO BINSTERR.
           MOVE SPACES TO W-TEXT.
           STRING 'CMAS=' CMASNAME OF BINSTERR ' Plex='
           PLEXNAME OF BINSTERR ' CSys=' CICSNAME OF BINSTERR
           ' ResName=' RESNAME OF BINSTERR
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE RESVER OF BINSTERR TO PICZZZ9.
           MOVE ERRCODE OF BINSTERR TO PYCZZZ9.
           MOVE CRESP1 OF BINSTERR TO PIKZZZ9.
           MOVE CRESP2 OF BINSTERR TO PYKZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING '  ResVer=' PICZZZ9 ' ECode=' PYCZZZ9
           ' RESP=' PIKZZZ9 ' RESP1=' PYKZZZ9
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE CEIBFN OF BINSTERR TO PICZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING ' EibFn=' PICZZZ9
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           EXIT.
```

```
      *************************************************
      * This subroutine displays BINCONRS information *
      *************************************************
       DISPBIRS.
           MOVE W-BUFFER TO BINCONRS.
           MOVE ERROP OF BINCONRS TO PICZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING 'CMAS=' CMASNAME OF BINCONRS ' Plex='
           PLEXNAME OF BINCONRS ' CSys=' CICSNAME OF BINCONRS
           ' ResType=' RESTYPE OF BINCONRS ' EOp=' PICZZZ9
           DELIMITED BY SIZE INTO W-TEXT END-STRING.
           PERFORM SCRNLOG2.
           MOVE CANDVER OF BINCONRS TO PICZZZ9.
           MOVE SPACES TO W-TEXT.
           STRING '  CandName=' CANDNAME OF BINCONRS
           ' CandVer=' PICZZZ9 ' CResGrp=' CANDRGRP OF BINCONRS
           ' CResAss=' CANDRASG OF BINCONRS ' CResDes='
           CANDRDSC OF BINCONRS
```

```
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     MOVE CANDUSAGE OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'CANDUSAGE' TO CHR12.
     PERFORM XCV2CH
     MOVE CHARV TO CHAR6.
     MOVE CANDTYPE OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'CANDTYPE' TO CHR12.
     PERFORM XCV2CH
     MOVE CHARV TO CHAR12.
     MOVE CANDASGOVR OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'CANDASGOVR' TO CHR12.
     PERFORM XCV2CH
     MOVE SPACES TO W-TEXT.
     STRING '  CandUsa=' CHAR6
     ' CandSGrp=' CANDSGRP OF BINCONRS
     ' CandSTyp=' CHAR12 ' CandAssO=' CHARV
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     MOVE EXISTVER OF BINCONRS TO PICZZZ9.
     MOVE EXISTUSAGE OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'EXISTUSAGE' TO CHR12.
     PERFORM XCV2CH
     MOVE SPACES TO W-TEXT.
     STRING '  ExistName=' EXISTNAME OF BINCONRS
     ' ExistVer=' PICZZZ9 ' EResGrp=' EXISTRGRP OF
     BINCONRS ' EResAss=' EXISTRASG OF BINCONRS
     ' EResDes=' EXISTRDSC OF BINCONRS ' ExistUsa=' CHARV
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     MOVE EXISTTYPE OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'EXISTTYPE' TO CHR12.
     PERFORM XCV2CH
     MOVE CHARV TO CHAR12.
     MOVE EXISTASGOVR OF BINCONRS TO CODEV.
     MOVE 'BINCONRS' TO CHR8.
     MOVE 'EXISTASGOVR' TO CHR12.
     PERFORM XCV2CH
     MOVE SPACES TO W-TEXT.
     STRING '  ExistSGrp=' EXISTSGRP OF BINCONRS
     ' ExistSTyp=' CHAR12 ' ExistAssO=' CHARV
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     EXIT.
```

```
*************************************************
* This subroutine displays BINCONSC information *
*************************************************
 DISPBISC.
     MOVE W-BUFFER TO BINSTERR.
     MOVE ERROP OF BINCONSC TO PICZZZ9.
     MOVE ERRCODE OF BINCONSC TO PYCZZZ9.
     MOVE SPACES TO W-TEXT.
     STRING 'CMAS=' CMASNAME OF BINCONSC ' Plex='
     PLEXNAME OF BINCONSC ' EOp=' PICZZZ9 ' ECode='
     PYCZZZ9 ' TScope=' TARGSCOPE OF BINCONSC
     ' TAssgn=' TARGRASG OF BINCONSC
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     MOVE SPACES TO W-TEXT.
     STRING '  TDesc=' TARGRDSC OF BINCONSC ' RScope='
     RELSCOPE OF BINCONSC ' RAssgn=' RELRASG OF BINCONSC
     ' RDesc=' RELRDSC OF BINCONSC ' CSys=' CICSNAME OF
     BINCONSC
     DELIMITED BY SIZE INTO W-TEXT END-STRING.
     PERFORM SCRNLOG2.
     EXIT.
```

```
************************************************************
* This subroutine converts coded value to character string *
************************************************************
 XCV2CH.
* Use new thread for TRANSLATE
```

```
           EXEC CPSM CONNECT
                     VERSION('0140')
                     THREAD(W-FBTTKN)
                     RESPONSE(W-RESPONSE)
                     REASON(W-REASON)
           END-EXEC.

      * Translate internal coded value to character value
           EXEC CPSM TRANSLATE
                     OBJECT(CHR8)
                     ATTRIBUTE(CHR12)
                     FROMCV(CODEV) TOCHAR(CHARV)
                     THREAD(W-FBTTKN)
                     RESPONSE(W-RESPONSE)
                     REASON(W-REASON)
           END-EXEC.
           EXIT.

      *-------------------------------------------------------------*
```

```
      *    PROCESSING FOR API FAILURES.                             *
      *-------------------------------------------------------------*
       NO-CONNECT.
           MOVE 'ERROR CONNECTING TO API.' TO W-MSG-TEXT.
           GO TO SCRNLOG.
       NO-CREATE.
           MOVE 'ERROR CREATING DEFINITION.' TO W-MSG-TEXT.
           GO TO SCRNLOG.
       NO-GET.
           MOVE 'ERROR GETTING RESOURCE TABLE.' TO W-MSG-TEXT.
           GO TO SCRNLOG.
       NO-INSTALL.
           MOVE 'ERROR INSTALLING RESULT SET.' TO W-MSG-TEXT.
           GO TO SCRNLOG.
       NO-TRANSLATE.
           MOVE 'ERROR TRANSLATING ATTRIBUTE.' TO W-MSG-TEXT.
           GO TO SCRNLOG.
       SCRNLOG.
      *    DISPLAY W-MSG-TEXT.
           EXEC CICS SEND FROM(W-MSG-TEXT) LENGTH(81) WAIT END-EXEC.
           MOVE W-RESPONSE TO PICZZZ9A.
           MOVE W-REASON TO PICZZZ9B.
           STRING 'RESPONSE=' DELIMITED BY SIZE
                  PICZZZ9A DELIMITED BY SIZE
                  ' REASON= ' DELIMITED BY SIZE
                  PICZZZ9B DELIMITED BY SIZE
                  INTO W-MSG-TEXT.
       SCRNLOG2.
      *    DISPLAY W-MSG-TEXT.
           EXEC CICS SEND FROM(W-MSG-TEXT) LENGTH(81) WAIT END-EXEC.

       ENDIT.
      *-------------------------------------------------------------*
      *    TERMINATE API CONNECTION.                                *
      *-------------------------------------------------------------*
           EXEC CPSM TERMINATE RESPONSE(W-RESPONSE) REASON(W-REASON)
           END-EXEC.
           EXEC CICS RETURN END-EXEC.
      *    GOBACK
           EXIT.
       EYULAPI4-END.
```

The COBOL version of EYUxAPI4 is written for the CICS environment and can be converted to run in the MVS/ESA batch environment by commenting the **EXEC CICS SEND** commands and uncommenting the preceding language specific output statement.

Notices

This information was developed for products and services offered in the United States of America. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property rights may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Client Relationship Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Programming interface information

IBM CICS supplies some documentation that can be considered to be Programming Interfaces, and some documentation that cannot be considered to be a Programming Interface.

Programming Interfaces that allow the customer to write programs to obtain the services of CICS Transaction Server for z/OS, Version 5 Release 5 (CICS TS 5.5) are included in the following sections of the online product documentation:

- Developing applications
- Developing system programs
- CICS security
- Developing for external interfaces
- Reference: application development
- Reference: system programming
- Reference: connectivity

Information that is NOT intended to be used as a Programming Interface of CICS TS 5.5, but that might be misconstrued as Programming Interfaces, is included in the following sections of the online product documentation:

- Troubleshooting and support
- Reference: diagnostics

If you access the CICS documentation in manuals in PDF format, Programming Interfaces that allow the customer to write programs to obtain the services of CICS TS 5.5 are included in the following manuals:

- Application Programming Guide and Application Programming Reference
- Business Transaction Services

- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- Installation Guide
- Security Guide
- CICS Transactions
- CICSPlex System Manager (CICSPlex SM) Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM Application Programming Guide and Application Programming Reference
- Java Applications in CICS

If you access the CICS documentation in manuals in PDF format, information that is NOT intended to be used as a Programming Interface of CICS TS 5.5, but that might be misconstrued as Programming Interfaces, is included in the following manuals:

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSPlex SM Problem Determination Guide

Trademarks

UNIX is a registered trademark of The Open Group in the United States and other countries.

Zowe™, the Zowe logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.

The Stack Exchange name and logos are trademarks of Stack Exchange Inc.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

IBM Software products, including software as a service solutions, (*Software Offerings*) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information (PII) is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect PII. If this Software Offering uses cookies to collect PII, specific information about this offering’s use of cookies is set forth below:

For the CICSPlex SM Web User Interface (main interface):

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user’s user name and other PII for purposes of session management, authentication, enhanced user usability, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface (data interface):

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name and other PII for purposes of session management, authentication, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface ("hello world" page):

Depending upon the configurations deployed, this Software Offering may use session cookies that do not collect PII. These cookies cannot be disabled.

For CICS Explorer:

Depending upon the configurations deployed, this Software Offering may use session and persistent preferences that collect each user’s user name and password, for purposes of session management, authentication, and single sign-on configuration. These preferences cannot be disabled, although storing a user's password on disk in encrypted form can only be enabled by the user's explicit action to check a check box during sign-on.

If the configurations deployed for this Software Offering provide you, as customer, the ability to collect PII from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Privacy Policy and IBM Online Privacy Statement, the section entitled *Cookies, Web Beacons and Other Technologies* and the IBM Software Products and Software-as-a-Service Privacy Statement.

Index