

CICS Transaction Server for z/  
OSバージョン 5 リリース 6

*CICSplex SM* ワークロード管理



## 注記

本書および本書で紹介する製品をご使用になる前に、[製品の特記事項](#)に記載されている情報をお読みください。

本書は、IBM® CICS® Transaction Server for z/OS®, バージョン 5 リリース 6 (製品番号 5655-Y305655-BTA)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

### 原典：

CICS Transaction Server for z/OS  
Version 5 Release 5  
CICSplex SM Managing Workloads

### 発行：

日本アイ・ビー・エム株式会社

### 担当：

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation 1974, 2020.

# 目次

この PDF について .....	v
<b>第 1 章 ワークロード管理 .....</b>	<b>1</b>
仕組み: CICSplex SM ワークロード管理 (WLM) .....	1
ワークロードの考慮事項 .....	3
ルーティングに関する考慮事項 .....	5
CICSplex SM でのワークロード管理のコンポーネント .....	6
CICSplex SM WLM が処理のルーティング先を選択する方法 .....	8
最も単純な CICSplex SM WLM 構成 .....	11
仕組み: z/OS ワークロード・マネージャー (WLM) .....	13
<b>第 2 章 ワークロード管理の構成 .....</b>	<b>17</b>
ワークロードの要件 .....	18
ワークロードの確立 .....	18
ワークロード内の作業の管理 .....	19
ワークロード内の作業の分離 .....	22
親和性の関係の考慮 .....	29
異常終了の確率およびワークロード管理 .....	32
ワークロード管理のアクティブ化 .....	33
<b>第 3 章 動的ルーティングの構成 .....</b>	<b>35</b>
CICSplex SM を使用した動的ルーティング .....	35
動的ルーティング用の CICS リリース要件 .....	37
サンプルのソース・プログラムとコピーブック .....	37
最適化動的ワークロード・ルーティングの実施 .....	38
動的ルーティングの変更 .....	44
CICSplex SM データ域 .....	52
EYU9WRAM のユーザー置換モジュールの作成 .....	52
追加動的ルーティング・サポートの要求 .....	52
CICSplex SM ワークロード管理の開始方法 .....	53
処理に関する考慮事項 .....	54
サンプルの呼び出しシーケンス .....	56
<b>第 4 章 CICSplex SM を使用したワークロードの管理 .....</b>	<b>59</b>
ワークロード・ルーティング .....	59
キュー・アルゴリズム .....	61
リンクに依存しないキュー・アルゴリズム .....	62
ゴール・アルゴリズム .....	62
リンクに依存しないゴール・アルゴリズム .....	63
CICSplex SM ワークロードのルーティングに対する z/OS WLM ヘルプ・サービスの効果 .....	63
ワークロード分離 .....	64
トランザクション間の類縁性 .....	64
CICSplex SM ワークロード管理によるワークロードの管理 .....	65
WLM の計画 .....	65
WLM の実施 .....	66
ワークロード管理リソース .....	70
ワークロード管理定義とそれぞれの関連ビュー .....	72
ワークロード管理定義の作成 .....	76
ワークロード仕様の作成 .....	76
トランザクション・グループの作成 .....	79

ワークロード定義の作成.....	80
ワークロード・グループの作成.....	82
ワークロード・ビューのルート・フィールドについて.....	82
タスクの例.....	84
ワークロードの管理.....	84
既存のターゲット領域スコープへの領域の追加.....	86
ターゲット領域スコープからの領域の除去.....	87
アクティブ・ワークロードへのルーティング領域の追加.....	88
アクティブなワークロードのターゲット領域を静止.....	88
特定のターゲット領域への特定のトランザクションのルーティング.....	89
指定されたユーザーから特定のターゲット領域への特定のトランザクションのルーティング.....	91
疑似会話型トランザクションの受け入れ.....	92
ワークロード定義の非アクティブ化.....	93
アクティブ・ワークロード定義の更新.....	94
ワークロードからのアクティブ・トランザクションの破棄.....	95
ワークロード仕様の更新.....	95
ワークロード・ルーティングのターゲット領域を選択するためのリアルタイム分析の使用.....	96
EXEC CICS START TERMIID を使用した動的ルーティング.....	100
インバウンド・クライアント DPL 要求の動的ルーティング.....	102
ピアツーピア DPL 要求の動的ルーティング.....	103
CICS BTS アクティビティのルーティング.....	104
CICS BTS アクティビティの分離.....	106
Link3270 ブリッジ・ワークロードの管理.....	107
<b>第 5 章アクティブ・ワークロード・ビュー.....</b>	<b>113</b>
アクティブ・ワークロード - WLMWORK.....	113
アクティブなルーティング領域 - WLMAWTOR.....	117
アクティブ・ワークロードのターゲット配布係数 - WLMAWAOR.....	119
アクティブなターゲット領域 - WLMATARG.....	125
定義 - WLMAWDEF.....	134
トランザクション・グループ - WLMATGRP.....	135
動的トランザクション - WLMATRAN.....	138
トランザクション・グループの親和性 - WLMATAFF.....	139
完全なアクティブ・ルーティング領域の詳細 - WLMAROUT.....	141
<b>第 6 章 CICSplex SM API サンプル・プログラム.....</b>	<b>145</b>
EYU#API1.....	145
EYUCAPI2.....	147
EYUAAPI3.....	153
EYULAPI4.....	165
<b>特記事項.....</b>	<b>175</b>
<b>索引.....</b>	<b>181</b>

## この PDF について

---

この PDF では、CICS Transaction Server for z/OS の CICSplex® SM コンポーネントの管理情報を示します。ここでは、ワークロード管理を実行するために必要となる、CICSplex SM 定義を確立および保守する方法について説明します。

使用されている用語と記法について詳しくは、IBM Knowledge Center の [CICS 資料で使用されている表記規則および用語](#)を参照してください。

### この PDF の日付

この PDF は、2020 年 5 月 28 日に作成されました。



# 第1章 ワークロード管理

ワークロード管理は、通常、処理が最も効率的な方法で実行されるように、システム内で効率的に処理を分散することを意味します。CICS ワークロードの枠組みの中では、効率的に処理を実行できるように CICS 領域に処理をルーティングするプロセスのことです。処理が CICS に到達する前に、ワークロード管理を行うことができます。例えば、トランザクションの開始点として CICS 端末専有領域 (TOR) を選択できます。または、処理が CICS に到達した後にすることもできます。例えば、トランザクションのルーティング先となる CICS 内の適切な領域 (通常はアプリケーション専有領域つまり AOR) を選択できます。CICS には、ワークロード管理のためのいくつかのオプションがあります。

## ワークロード管理のオプション

### CICSplex SM ワークロード管理 (WLM)

CICSplex SM WLM は、ワークロード管理の CICS 実装です。z/OS ワークロード・マネージャーとシスプレックス・ディストリビューターは、シスプレックスの効率性、高可用性、および TCP/IP の効率性に関係しています。これらの処理は、通常は CICS に到達する前に行われます。CICSplex SM WLM は、処理が CICS に到達した後の処理の効率性に関係しています。CICSplex SM は、処理の送信先として最も効率的な領域を選択します。

3 つまたは 4 つを超える AOR を実行する場合は、CICSplex SM WLM の利点を活用できます。CICSplex SM は、CICS でのワークロード実行を最もきめ細かく制御できます。したがって、最も広く採用されているワークロード管理ソリューションです。詳しくは、[仕組み: CICSplex SM ワークロード管理 \(WLM\)](#) を参照してください。

### z/OS ワークロード・マネージャー (WLM)

z/OS のワークロード・マネージャー・コンポーネントは、シスプレックスをモニターし、ユーザー定義の目標を満たすためにシスプレックス内の各処理項目にどの程度のリソースを割り当てる必要があるかを決定します。詳しくは、[仕組み: z/OS ワークロード・マネージャー \(WLM\)](#) を参照してください。

### シスプレックス・ディストリビューター

シスプレックス・ディストリビューターは、IBM z/OS Communications Server のコンポーネントであり、シスプレックス (つまり複数の LPAR) 全体の TCP/IP ロード・バランシングを行います。これは、TCP/IP ポート共用と組み合わせることができます。シスプレックス・ディストリビューターは、分散された DVIPA の高可用性機能と WLM のワークロード最適化機能を組み合わせたものです。TCP/IP スタックは、WLM からのワークロード情報を要求するように構成でき、これにより、分散スタックは、各ターゲット・スタックのワークロードに基づいた接続を転送することができます。WLM ワークロード情報は、ターゲット・システムごとに使用可能な汎用 CPU 容量の比較に基づいています。詳しくは、[「z/OS Communications Server IP 構成ガイド」の『シスプレックス・ディストリビューター』](#)を参照してください。

### 独自のソリューションを作成します。

IBM 提供ソリューションほど一般に使用されていませんが、独自のワークロード分散手段を作成することもできます。CICS では、ユーザー作成の動的ルーティング・プログラムを介してこれを行います。詳しくは、[動的ルーティング・プログラムの作成](#)を参照してください。

オプションは相互に排他的なものではありません。z/OS ワークロード・マネージャー、シスプレックス・ディストリビューター、CICSplex SM を任意に組み合わせて使用できます。例えば、高可用性のコンテキストでシスプレックス・ディストリビューターを使用する TCP/IP ワークロードで、着信処理を CICS TOR にルーティングすることができます。次に、CICSplex SM WLM を使用して、処理を実行するうえで最も効率的な領域を判別できます。または、CICSplex SM だけを使用して CICS ワークロードを制御することもできます。

## 仕組み: CICSplex SM ワークロード管理 (WLM)

通常、3 つまたは 4 つを超える AOR を実行する場合は、ワークロードの管理に CICSplex SM WLM を使用するメリットがあります。CICSplex SM におけるワークロード管理 (WLM) とは、トランザクションが CICS 領域に到達したりプログラムが CICS 領域からリンクされたりしたときに、最適な CICS 領域を動的に選択し、その領域に処理をルーティングしてそこで実行させることです。

このセクションでは、CICSplex SM WLM の主要な概念と、CICSplex SM WLM のセットアップに關与するものの概略を説明します。

### CICSplex SM WLM のセットアップ

CICSplex SM WLM のセットアップに誰が關与するかを見てみましょう。2 ページの図 1 には、基本トポロジーが示されています。

- ソフトウェア・アーキテクトが、ワークロード管理の目標を決定し、特定のワークロードに適用する何らかの制約 (例えば、特定のタイプの処理の分離またはリンク) を決定します。
- ソフトウェア・アーキテクトが、CICSplex SM WLM を使用した動的ルーティングに適格となる処理要求を特定します。
- システム管理者が、CICS Explorer® または CICSplex SM WUI のいずれかを使用して、CICSplex SM のコンポーネントを構成します。システム管理者は、CICSplex SM ワークロード管理に参加する CICS 領域のグループを作成し、各ワークロードがどのように管理されるかを指定します。

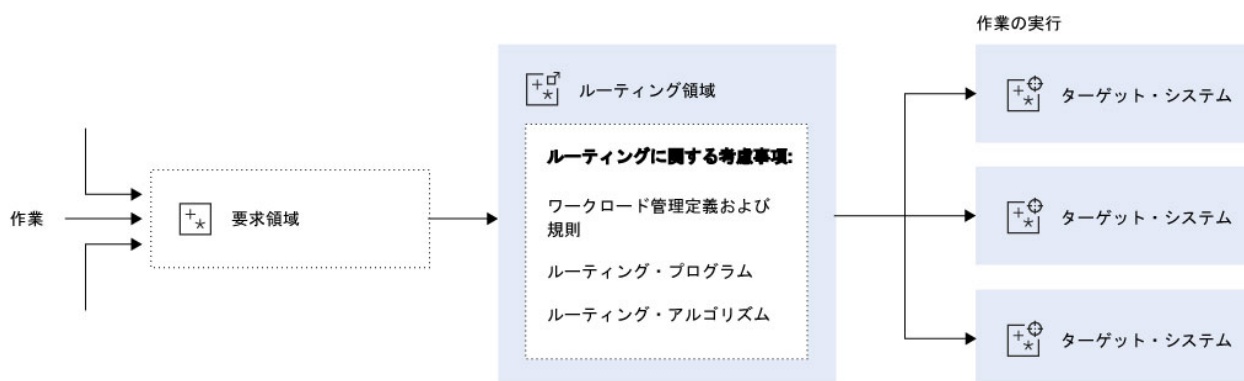


図 1. CICSplex SM WLM の基本トポロジー

CICSplex SM WLM で処理を管理する要求が、CICSplex 内の 1 つの領域から出されます。

各ワークロードの仕様では、ワークロードの処理でルーティング領域として扱われる CICS システムと、ターゲット領域として扱われる CICS システムを定義します。特別なワークロード条件を扱うために必要な細分性をワークロード管理ルールで定めます。例えば、特定のタイプの処理を相互に分離させたり、特定のタイプの処理間の関係 (親和性) を維持したりします。

CICSplex SM は動的ルーティングを使用するので、1 つの処理をどこで実行するかを決定するルーティング・プログラムが必要です。CICSplex SM では、ユーザー置換可能な動的ルーティング・プログラム EYU9XLOP がこの機能を備えています。

CICSplex SM WLM によって提供されるルーティング・アルゴリズムでは、着信処理要求の処理に最適な領域を選択するために、数々の複合的な要因が考慮されます。これらの要因には、以下が含まれます。

- 使用可能な CICS 領域の負荷の状況。
- 使用可能な CICS 領域の正常性の状況。
- 使用可能な CICS 領域の応答の状況。
- 使用可能な CICS システムの場所とその接続方法。

処理が 1 つ以上のターゲット領域に渡されて実行されます。

### CICSplex SM ワークロード管理: 必要な情報の参照先

CICSplex SM WLM を使用するには、IBM Knowledge Center で以下の項目を参照する必要があります。



表 1. CICSplex SM WLM に関する情報の参照先	
トピック	参照先
CICSplex SM WLM でワークロードを管理し始める前に、ワークロードに関してどんなことを考慮すべきかを調べる	<a href="#">ワークロードの考慮事項</a>
領域間の処理のルーティングに関するいくつかのオプションを調べる	<a href="#">ルーティングに関する考慮事項</a>
CICSplex SM WLM の仕組みを調べる	<a href="#">CICSplex SM WLM のコンポーネント</a> および <a href="#">CICSplex SM WLM で の作業のルーティング方法</a>
CICSplex SM WLM をセットアップする	<a href="#">ワークロード管理定義の作成</a> および <a href="#">ワークロード管理タスク</a>
CICS Explorer でワークロードを管理する	<a href="#">CICS Explorer 製品資料内の『ワークロード管理』</a>

## ワークロードの考慮事項

CICSplex SM WLM をセットアップする前に、ワークロードによって管理が何らかの制約を受けるかどうかを理解する必要があります。ワークロード管理において、理想的な状態とは、各ワークロードがいずれかの CICS 領域で実行可能であることです。CICSplex SM WLM は処理を管理し、領域の負荷や正常性などの要因を考慮して、最も適切なターゲット領域に処理をルーティングします。しかしワークロードによっては、アプリケーションの設計や、ワークロード処理に使用できる CICS 環境によって決まる、より複雑なニーズが存在する場合があります。例えば、特定のタイプのワークロードを特定の領域でのみ実行する必要がある場合があります。または、いくつかの特定のトランザクション間の関係(親和性と呼ばれる)が定義されているために、これらのトランザクションをすべて同じ領域で処理する必要がある場合もあります。

### ワークロード・バランシング

ワークロード・バランシングは、CICSplex SM WLM の制御下にあるワークロードを表しています。理解すべき重要な点は、ワークロード・バランシングは使用可能なそれぞれの領域間で処理のバランスを均等に取りわけではないということです。処理は、それを最適に実行できる領域にルーティングされます。これは、単純なラウンドロビン・ルーティングよりも高度です。

例えば、環境内に 9 つの AOR があるとします。着信処理(例えば DEMO というトランザクション)は、使用可能な 9 つの AOR のいずれかにルーティングされて実行されます。ラウンドロビン・ルーティングでは、最初の DEMO トランザクションは AOR 1 にルーティングされます。2 番目の DEMO トランザクションは AOR 2 にルーティングされ、3 番目は AOR 3 にルーティングされ、以降同様に続きます。各領域は、着信の処理の同等部分を受け取ります。

同じ例で、CICSplex SM WLM は、9 つの領域すべての状態、つまり正常性と可用性、および接続方法を考慮します。この情報を使用して、処理を実行する領域を決定します。DEMO 処理を実行するために AOR 6 を選択する可能性があります。次のトランザクションが到着すると、AOR 4 を選択する可能性があります。あるいは、最初の 20 トランザクションをすべて AOR 1 にルーティングできると決定する可能性もあります。要点は、CICSplex SM がルーティングを制御するということです。ルーティングの決定は、処理を実行する領域の全体的な適合性に基づいて行われ、すべての使用可能な領域間で処理を同等に分配するわけではありません。したがって、ラウンドロビン・ルーティングとは大きく異なります。

ラウンドロビン・ルーティングは、領域の分散と利用に関して、より均等な(つまりバランスの取れた)パターンを割り当てるように最初は見えることがあります。しかしこのアプローチは、複雑なワークロード用には単純化されすぎています。例えば、領域の 1 つが突然正常でなくなった場合、ラウンドロビン・ルーティングはどのように対処するのでしょうか。このような理由で、CICSplex SM WLM はラウンドロビン・ルーティングを提供していません。

## ワークロード分離

ワークロード分離は、特定のタイプの処理を特定のターゲット領域に限定するために使用されます。指定された一連のトランザクションに対して使用可能なターゲットは、指定された1つの領域、または指定された一連の領域(こちらの方がより典型的)です。例えば、給与計算トランザクションを特定の領域でのみ処理する必要がある場合があります。

CICSplex SM WLMはワークロード分離をサポートしています。トランザクションとプログラムは、トランザクションによって、トランザクションまたはプログラムのオカレンスに関連付けられている端末 ID およびユーザー ID によって、または CICS BTS アクティビティに関連付けられているプロセス・タイプによって、分離できます。複数の異なるターゲット領域セットに処理が送られ、セット内の領域間でアクティビティのバランスが取られます。

## ワークロードの親和性(トランザクション間の親和性)

ワークロードの親和性では、複数のトランザクション間の関係が定義され、特定の期間にわたってその関係が存在するようになります。トランザクション間に親和性の関係が存在するとき、これらのトランザクションは、同じターゲット領域によって処理される必要があります。要求領域からターゲット領域への処理のルーティングは、トランザクション間の親和性の関係の特定の組み合わせに適用されるルールと、そのルールが適用される期間に基づきます。

例えば、トランザクション AAAA は、一時記憶域キューに書き込みを行います。次のトランザクション AAAB は、そのキューからデータを読み取る必要があります。トランザクション AAAA と AAAB には親和性があり、これらは、処理のために同じ CICS 領域にルーティングされる必要があります。これらの親和性がワークロード・マネージャーによって考慮されない場合、トランザクション AAAA はある AOR にルーティングされ、トランザクション AAAB は別の AOR にルーティングされる可能性があり、この場合、TS キューからのデータの読み取りを試行しても、データがその領域に存在しないため、異常終了します。

運用上、親和性を最小限にすることが常に適切です。親和性が少ないということは、ワークロードの潜在的な障害点が少なく、ワークロード・ルーティングの柔軟性が高まることを意味します。親和性の検出および親和性に影響を与えるプログラミング手法について詳しくは、[類縁性](#)を参照してください。

CICSplex SM WLM では、CICS で定義できる 5 タイプの親和性の関係すべてを処理できます。

- グローバルな類似性の関係: 任意の端末から任意の START コマンドまたは任意の CICS BTS プロセスによって開始されるすべてのトランザクションのすべてのインスタンス間の関係です。
- LU 名(端末)の親和性の関係: 同じ端末に関連付けられているグループ内のすべてのトランザクションのすべてのインスタンス間の関係です。
- ユーザー ID の親和性の関係: ある端末から START コマンドまたは CICS BTS アクティビティによって開始されるトランザクションのすべてのインスタンス間の関係であり、同じユーザー ID のために実行されます。
- BAPPL の親和性の関係: 同じ BTS プロセスに関連付けられているトランザクションのすべてのインスタンス間の関係です。
- LOCKED の親和性の関係: 同じ作業単位を持つ動的にリンクされたプログラムに関連付けられているグループ内のトランザクションのすべてのインスタンス間の関係です。

親和性の期間は以下のとおりです。

- アクティビティ: 親和性は、関連付けられているアクティビティが存在する限り存続します。
- 区切り: 親和性は、疑似会話モードによるトランザクションの END が検出されるまで存続します。
- ログオン: 親和性は、端末が CICS にログオンしている限り存続します。
- 疑似会話 (PCONV): 親和性は、疑似会話全体の間存続します。
- 永続: 親和性は、すべての CICS 再始動をまたいで拡張されます。
- 処理: 親和性は、関連付けられている処理が存在する限り存続します。
- サインオン: 親和性は、ユーザーがサインオンしている限り存続します。
- システム: 親和性は、ターゲット領域が存在する限り存続し、ターゲット領域が終了すると終了します。
- 作業単位 (UOW): 親和性は、作業単位がアクティブである限り存続します。

## CICSplex SM WLM で処理できる処理タイプ

CICSplex SM WLM は、以下の処理タイプをルーティングできます。

- 端末で起動されたトランザクション
- CICS Business Transaction Services (BTS) のアクティビティーに関連付けられたトランザクション
- 関連付けられている端末の有無に関係なく、**EXEC CICS START** コマンドを使用して呼び出される適切なトランザクション
- 以下を含む、分散プログラム・リンク (DPL) 要求。
  - EXCI 呼び出し
  - CICS Transaction Gateway ECI 呼び出し
  - 分散コンピューティング環境 (DCE) リモート手続き呼び出し (RPC)
  - オープン・ネットワーク・コンピューティング (ONC) リモート・プロシージャ・コール (RPC)
  - **EXEC CICS LINK PROGRAM** 要求を発行する関数
  - Link3270 要求

## ルーティングに関する考慮事項

指定した 1 つの領域で常に処理が実行されるようにすることもできます。これは、静的ルーティングと呼ばれます。CICSplex SM WLM では、動的ルーティングを使用して処理要求の実行場所を制御します。ハブ・モデルまたは分散モデルで動的ルーティングを実装できます。

### 動的ルーティングの静的ルーティングとの比較

静的ルーティングでは、処理は常に指定された領域で実行されます。CICSplex または BTS セットでは、ある領域で必要とされるトランザクションおよびプログラムなどのリソースが別の領域によって所有されている場合があります。例えば、端末専有領域 (TOR) があるものの、その TOR ではアプリケーション専有領域 (AOR) が所有するトランザクションへのアクセスが必要とされる場合があります。システムの設計時に (例えば、インストール・リソース定義で) リソースのロケーションを指定すると、そのリソースの要求は常に同じ領域にルーティングされます。

CICS 領域が比較的少ない場合は、静的ルーティングが適切な場合があります。ただし、ルーティングのために正確な SYSID を指定するには考慮事項があります。

- 指定された SYSID に対してのみルーティング可能です。その領域が、使用不可であるか、応答しないか、正常ではない場合、これが原因で問題が発生する可能性があります。
- SYSID が変更された場合、それを指定するコードも変更する必要があります。

動的ルーティングの場合、処理を実行する場所は動的ルーティング・プログラムによって決定されます。CICSplex SM では、プログラムは EYU9XLOP というユーザー置換可能な動的ルーティング・プログラムです。

CICSplex SM WLM 動的ルーティングはアプリケーションに依存しません。値はアプリケーションにコーディングされません。異なる状況を処理するために、スペシャリストのアプリケーション・コードは必要ありません。環境間の移動に柔軟性がもたらされ、環境に応じて名前が変更されます。例えば、領域数の変更を扱うには、CICSplex SM WLM を素早く変更するだけで済みます。静的ルーティングの場合のようにアプリケーション・ソース・コードを変更する必要はありません。

### 動的ルーティングにおける CICS 領域の役割

動的ルーティングに関与する CICS 領域は、以下のうちの 1 つ以上のものとして機能することがあります。

#### 要求側領域

処理要求が開始される CICS 領域。端末開始トランザクションの場合や、インバウンド DPL クライアント要求の場合、要求領域は通常、端末専有領域 (TOR) です。端末に関連付けられている **EXEC CICS START** コマンドの場合、ピアツーピア DPL 要求の場合、非端末関連の **EXEC CICS START** コマンドの場合、CICS BTS プロセスとアクティビティーの場合、および Link3270 ブリッジ要求の場合、要求領域は通常 AOR になります。

### ルーティング領域

処理要求をどのようにルーティングするかを決定する CICS 領域。端末開始トランザクションおよび端末関連 **EXEC CICS START** コマンドの場合や、CICS CICS BTS プロセスとアクティビティーの場合、および Link3270 ブリッジ要求の場合、ルーティング領域は通常 AOR になります。

### ターゲット領域

要求が実行される CICS 領域。すべての動的ルーティング・トランザクション、プログラム、および BTS プロセスとアクティビティーの場合、ターゲット領域は通常 AOR になります。

1 つの領域がルーティング領域とターゲット領域の両方になることがあります。

### EYU9XLOP: CICSplex SM 動的ルーティング・プログラム

CICSplex SM WLM は、EYU9XLOP というユーザー置換可能な動的ルーティング・プログラムを使用することで、動的ルーティングに必要な環境を作成し、CICSplex SM ランタイム環境をセットアップします。

ほとんどの場合、提供されているワークロード管理機能で十分です。ただし、必要に応じて、CICSplex SM ワークロード管理処理を行うためのモジュールをカスタマイズできます。詳しくは、[EYU9WRAM のユーザー置換モジュールの作成](#)を参照してください。

## CICSplex SM でのワークロード管理のコンポーネント

CICSplex SM WLM では、ワークロードを定義して制御するためにオブジェクトの階層が使用されます。

CICSplex SM WUI では、各コンポーネントに対応するビューがあります。これらのビューは、図では括弧内に示しています。CICS Explorer では、一部のコンポーネントは結合されます。その場合、ビューには複数のコンポーネントからの情報が表示されます。例えば、CICS Explorer ではワークロード (ルーティング) ルールという用語を使って、ワークロード定義 (WLMDEF) とトランザクション・グループ (TRANGRP) のペアからの情報を表します。

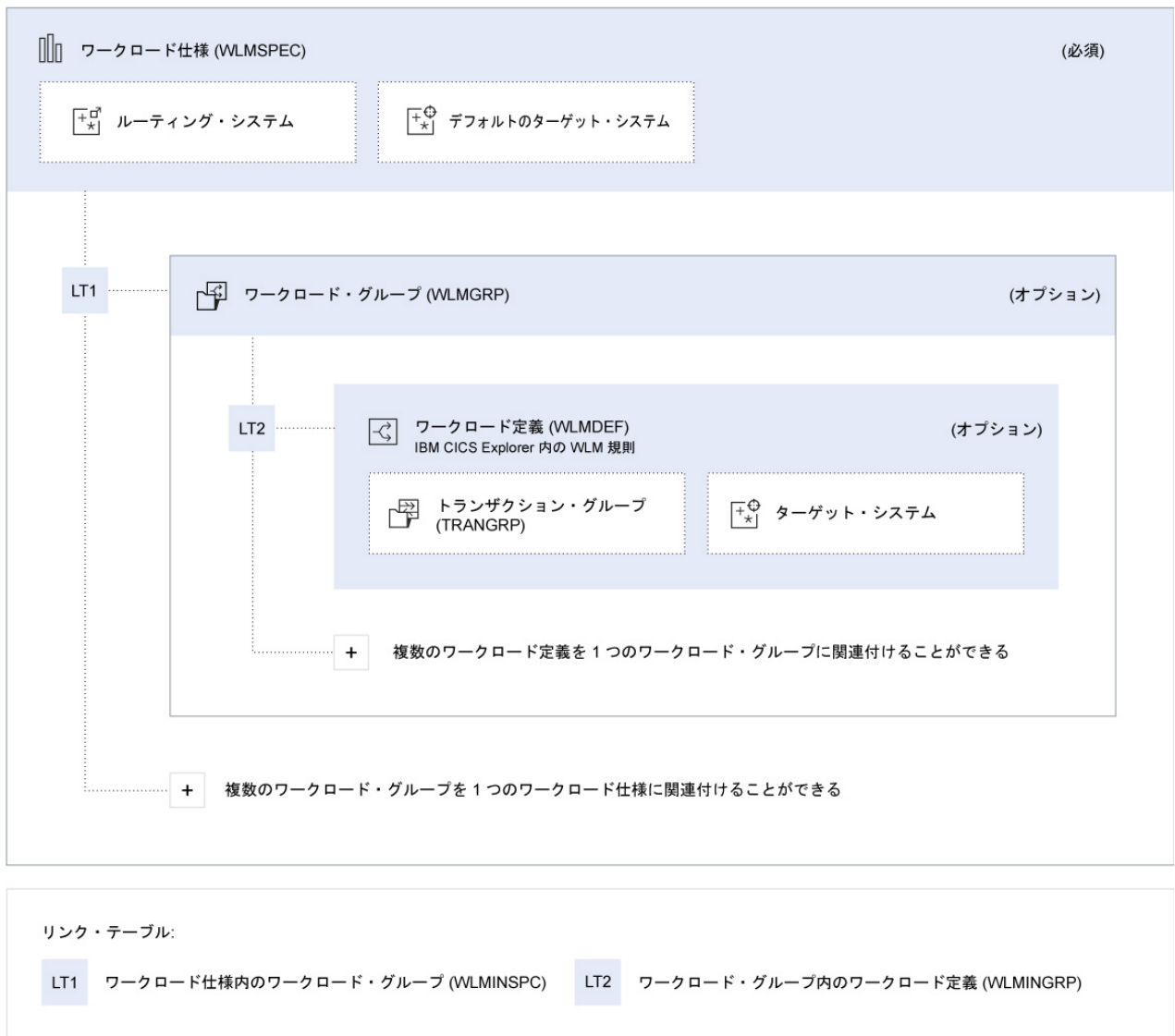


図 2. ワークロード管理オブジェクトと関連付け

## ワークロード管理仕様 (WLMSPEC)

各ワークロードには、ワークロード管理仕様があります。WLM 仕様の内容は以下のとおりです。

- ワークロードの名前
- ルーティング領域として扱われる一連の CICS システムの定義。CICS 領域を WLMSPEC に直接関連付けることも、CICS システム・グループで一連の CICS 領域を定義することもできます。
- ワークロードのデフォルトのターゲット領域の定義
- (オプション) デフォルトの親和性関係および期間の定義 (ワークロードの親和性が存在する場合)

ワークロード管理仕様は、CICSplex SM で WLMSPEC として定義されます。WMSPEC およびその属性について詳しくは、[WLMSPEC Resource Table](#) を参照してください。

## ワークロード管理定義 (WLMDEF)

ワークロードの処理を分離する必要がある場合、またはワークロードに親和性がある場合は、ワークロード管理定義 (CICS Explorer では WLM 規則 と呼ばれる) が使用されます。

ワークロードに対してワークロード管理定義が定義もインストールもされていない場合は、すべてのトランザクションとプログラムは、デフォルトのターゲット・スコープで識別されるターゲット領域にルーティングされます。デフォルトのターゲット領域は、ワークロード管理仕様 (WLMSPEC) で指定されます。



ワークロードに対して 1 つ以上のワークロード管理定義が定義されてインストールされている場合は、各 WLMDEF で定義された基準に一致するすべてのトランザクションとプログラムは、その WLMDEF で定義されているターゲット・スコープにルーティングされます。

ワークロード管理定義は、CICSplex SM で WLMDEF として定義されます。WLMDEF およびその属性について詳しくは、[WLMDEF Resource Table](#) を参照してください。

### ワークロード管理グループ (WLMGROUP)

ワークロード管理グループは、複数のワークロード管理定義 (WLMDEF) が定義されている場合にのみ必要です。これによって、複数のワークロード管理定義をグループ化します。

ワークロード管理定義が 1 つのみの場合は、その定義をワークロード管理グループにリンクするか、ワークロード管理グループを除外して、唯一のワークロード管理定義を直接ワークロード管理仕様にリンクすることができます。

ワークロード管理グループは、CICSplex SM で WLMGROUP として定義されます。WLMGROUP およびその属性について詳しくは、[WLMGROUP Resource Table](#) を参照してください。

### トランザクション・グループ (TRNGRP)

トランザクション・グループ (TRNGRP) は、論理的に類似したトランザクションを関連付けたものです。類似性は、親和性の要件、アプリケーションの共通共用処理の要件、またはユーザーが決定したその他の特性に基づいて判断されます。トランザクション・グループは、ワークロード管理定義 (WLMDEF) の一部として組み込まれます。次に、WLMDEF はターゲット領域として使用される CICS システム・グループを定義します。

具体的に 1 つ以上のトランザクション・グループに関連付けられていないトランザクションは、デフォルトのトランザクション・グループに割り当てられます。

トランザクション・グループは、CICSplex SM で TRNGRP として定義されます。TRNGRP およびその属性について詳しくは、[TRNGRP Resource Table](#) を参照してください。

### リンク・テーブル

リンク・テーブルは、CICSplex SM WLM のコンポーネント間の相互関連を保管するために CICS によって内部で使用されます。この図ではリンク・テーブルを示しますが、これは、説明の完全さを期するためと、WLM コンポーネントを関連付けるときに WUI にリンク・テーブルの用語が出てくるためです。WLM のコンポーネントを理解するうえでリンク・テーブルを気にする必要はありません。リンク・テーブルの用語は CICS Explorer に表示されません。

## CICSplex SM WLM が処理のルーティング先を選択する方法

トランザクションが開始されると、CICSplex SM では、ルーティングの決定に寄与する要因を判別するアルゴリズムを使用して、パフォーマンスが最大になると考えられるターゲット領域が選択されます。アルゴリズムは 4 つあります。ワークロード仕様でデフォルトのルーティング・アルゴリズムを指定します。このデフォルトは、トランザクション・グループの場合、トランザクション・グループ定義で異なるルーティング・アルゴリズムを指定することによってオーバーライドできます。

### 4 つのルーティング・アルゴリズム

CICSplex SM は、以下の 4 つのアルゴリズムの 1 つを使用して、作業を処理するターゲット領域を決定します。

- [キュー](#)
- [ゴール](#)
- [リンクに依存しないキュー](#)
- [リンクに依存しないゴール](#)

ルーティングされるトランザクションについてトランザクション 親和性が顕著である場合、使用されるアルゴリズムとは無関係に、親和性ターゲット領域が選択されます。

ワークロード仕様 (WLMSPEC) でデフォルトのルーティング・アルゴリズムを指定します。オプションにより、トランザクション・グループ (TRANGRP) レベルでルーティング・アルゴリズムを指定することができます。トランザクション・グループ・レベルで指定されるアルゴリズムは、ワークロード仕様に関連付けられているデフォルトのアルゴリズムをオーバーライドします。

実行中のワークロードに対して、ワークロード仕様のデフォルトのルーティング・アルゴリズムを変更するには、そのワークロードに関連付けられているすべての CICS 領域をシャットダウンする必要があります。これにより、ワークロードがリフレッシュされます。トランザクション・グループではルーティング・アルゴリズムを動的に変更できます。CICS 領域をシャットダウンする必要はありません。

#### 注:

特定のアルゴリズムを使用することをルーティング・モード と呼ぶ場合もあります。例えば、ゴール・ルーティング・モード (さらに簡単には「ゴール・モード」) は、ゴール・アルゴリズムを使用する構成を表します。

#### キュー・アルゴリズム

これは最も一般的に使用されるアルゴリズムです。ルーティングされているトランザクションにアクティブな親和性がない場合、このアルゴリズムによってワークロードはターゲット領域の集合全体に分散されます。選択基準には、現行のタスク・ロード、正常性の状態、およびルーターとターゲット領域間の接続のタイプが含まれます。

ルーティングされるトランザクションにアクティブな親和性がない場合、キュー・アルゴリズムを使用すると、CICSplex SM は以下の一連の基準に合致するターゲット領域を選択します。

- ・領域は、ターゲット領域で許可されるタスクの最大数と比較して、処理を待機する作業キューが最短になる。この作業キュー (ロード・カウントとも呼ばれる) は、キューに入れられたすべてのアクティブ・ユーザー・タスクの数です。デフォルトでは、MAXTASKS および TRANCLASS 両方の属性に関してキューに入れられたタスクがロード・カウントに含まれます。CSYSDEF リソース・テーブルで「タスク・ロード・キュー・モード」属性を使用すると、TRANCLASS 属性に関してキューに入れられたタスクをサイトでロード・カウントから除外することができます。CSYSDEF リソース・テーブルの「タスク・ロード・キュー・モード」属性の詳細については、CICS システム定義 - CSYSDEF を参照してください。
- ・領域は、ストレージ不足、SYSDUMP、および TRANDUMP などの条件の影響が一番小さい。
- ・領域は、トランザクションが停止する可能性が一番低い。
- ・領域は、領域間通信のパスが一番短い。
- ・領域の z/OS WLM 正常性値が 1 から 100 までの範囲に入っている ([11 ページの『領域の z/OS WLM 正常性の状態 \(z/OS WLM ヘルス・サービスがオンの場合のみ\)』](#)を参照)。

#### ゴール・アルゴリズム

このアルゴリズムでは、応答時間について事前定義されているゴールに基づいてターゲット領域が選択されます。ゴール・アルゴリズムがターゲット領域を識別しない場合、キュー・アルゴリズムが残りのターゲット領域に適用されます。このアルゴリズムが成功するかどうかは、適切かつ達成可能な応答時間目標の設定に依存します。適切な応答時間目標を定義しない場合、結果として予期しないルーティング・パターンが生じる可能性があります。

ゴール・モードがより適しているユーザーは、CICSplex SM WLM の経験があり、単一 LPAR を使用し、応答時間目標に関連してかなり具体的な要件があるユーザーです。

#### リンクに依存しないキュー・アルゴリズム

リンクに依存しないキュー・アルゴリズムは、ルーティング領域とターゲット領域の間の接続タイプが考慮されない点を除き、キュー・アルゴリズムと同様に機能します。接続タイプの重み付けは無視されます。

キュー・アルゴリズムと同様に、このアルゴリズムでは、プロセッサ能力の違い、ターゲット領域の最大タスク値の違いに対応できます。ただし、このアルゴリズムでは、ルーターとターゲット間の通信リンク速度を考慮に入れません。キュー・アルゴリズムと比較して、このアルゴリズムはターゲット・スコープ全体でワークロードをより均等に分散できるかもしれませんが、ワークロードはそれほど迅速に完了しません。

ワークロード仕様でデフォルトのルーティング・アルゴリズムを、リンクに依存しないアルゴリズムの 1 つに設定する場合は、これがすべての動的トランザクションに適用されるので、注意が必要です。ト

ランザクションが最速にローカルで継続して処理されるようにする場合に、MVS™ サブシステムからのサービスを要求する動的トランザクションが任意の場所で処理されるようにする場合は、ワークロード仕様のデフォルトのルーティング・アルゴリズムとしてキュー・アルゴリズムを考慮してください。次に、トランザクション・グループ・レベルで、トランザクションのサブセットに、リンクに依存しないアルゴリズムを適用します。

#### リンクに依存しないゴール・アルゴリズム

リンクに依存しないゴール・アルゴリズムは、ルーティング領域とターゲット領域の間の接続タイプが考慮されない点を除き、ゴール・アルゴリズムと同様に機能します。接続タイプの重み付けは無視されます。

ワークロード仕様でデフォルトのルーティング・アルゴリズムを、リンクに依存しないアルゴリズムの1つに設定する場合は、これがすべての動的トランザクションに適用されるので、注意が必要です。トランザクションが最速にローカルで継続して処理されるようにする場合に、MVS サブシステムからのサービスを要求する動的トランザクションが任意の場所で処理されるようにする場合は、ワークロード仕様のデフォルトのルーティング・アルゴリズムとしてキュー・アルゴリズムを考慮してください。次に、トランザクション・グループ・レベルで、トランザクションのサブセットに、リンクに依存しないアルゴリズムを適用します。

#### CICSplex SM 動的ワークロード・ルーティングの決定に寄与する要因

ワークロード・ルーティングの決定に寄与する各要因に対して、CICSplex SM によって重み付けが適用されます。通常、ルーティングのためにすべての要因が考慮された後で重み付けが最も低いターゲット領域が選択されます。すべての使用可能なターゲット領域の重み付けが等しい場合、つまり、処理の実行能力が等しいと評価された場合、ターゲット領域はグループからランダムに選択されます。したがって、負荷の軽いシステムでは、同等の能力を持つターゲット領域への処理割り振り順序は事前に決まっています。

#### 領域内のタスクの数

パーセンタイル値として因数化され、ターゲット領域の MAXTASKS 設定を現行タスク・カウントで除算して計算されます。

#### 領域の正常性状況

トランザクション・ダンプを取り、システム・ダンプを取り、MAXTASKS 限度または非応答状態や CICSplex SM 停止状態で実行し、領域がストレージ不足であるかどうかに応じて、算術的な重みを割り当てて因数化されます。

注：正常性はシステム可用性モニター (SAM) とは無関係に評価され、SAM の設定から影響を受けることはありません。

#### ルーティング領域とターゲット領域の間の接続タイプ (リンク)

ターゲット領域が MRO 接続、XCF 接続、LU6.2 接続、IPIC 接続によってターゲットにリンクされているかどうかに応じて算術的な重みを割り当てて、またはターゲット領域がルーター自体である場合に、因数化されます。接続タイプは、リンクに依存しない2つのアルゴリズムでは使用されません。

重みは、以下の優先順位で各接続に割り振られます。

1. ローカル
2. MRO/IRC および MRO/XM (ローカル LPAR)
3. MRO/XCF (リモート LPAR)
4. IPIC (ローカル LPAR)
5. IPIC (リモート LPAR)
6. LU6.2
7. 間接

例えば、他のすべての係数が等しい場合、MRO/XCF を使ってその要求領域に接続されているターゲット領域は、IPIC を使って接続されているターゲット領域よりも優先されます (つまり前者のターゲット領域に最も低い重み付けが割り振られます)。同じ LPAR 内のルーティング領域とターゲット領域間の IPIC 接続の重み付けは、異なる LPAR 内にあるターゲット領域への IPIC 接続の重み付けより低くなるため、他の係数が同じであれば、ローカル IPIC 接続はリモート IPIC 接続より優先されます。



## 領域の z/OS WLM 正常性の状態 (z/OS WLM ヘルス・サービスがオンの場合のみ)

z/OS WLM ヘルス・サービスによって提供される領域の z/OS WLM 正常性値に応じて、ペナルティーの重みが割り当てられて因数化されます。領域の z/OS WLM 正常性の状態は、0 から 100 の範囲の z/OS WLM 正常性値によって反映されます。

- z/OS WLM ヘルス値がゼロの領域は、作業を受け取るのに適格ではありません。CICSplex SM WLM は、この領域に処理をルーティングしません。この領域に関連付けられているワークロード親和性は、維持され、考慮されますが、これらの親和性に基づくワークロード・ルーティングは、正常性値がゼロである場合は失敗し、メッセージ EYUWR0003W または SYSIDERR 条件が発生します。メッセージ EYUWR0003W は、EYU9WRAM プログラムが有効であり、親和性 AOR が使用不可であったことを示します。このようなルーティングの障害を解決するには、ヘルス値がゼロ以外になるように領域のウォームアップを開始してから、ワークロードを再実行します。[CICS システム・ウォームアップの開始](#)を参照してください。
- z/OS WLM 正常性値が 1 から 99 の範囲の領域の場合、ペナルティーの重みはルーティング・アルゴリズムで割り当てられます。ヘルス値が高いほど、ペナルティーの重みは低くなります。したがって、ルーティングの決定において、正常性値がより高いターゲット領域が、より望ましいとされます。高いペナルティーの重みが割り当てられた領域もアクティブに保たれ、同じスコープ内で、使用可能な、正常性値がより高い領域が他にない場合は、依然として、処理を受け取る可能性があります。
- z/OS WLM 正常性値が 100 の領域は、ペナルティーの重みは割り当てられず、処理を受け取ることが十分に可能です。

CICS のウォームアップ・プロセスまたはクールダウン・プロセス中、CICS は、領域の z/OS WLM ヘルス値を調整します。CICSplex SM WLM のルーティングの決定に影響し、そのプロセス中の領域へのワークロードが制限されます。一般的に、領域がウォームアップ中の場合は、正常性値は定期的に増加し、領域は徐々により大規模なワークロードを受け取るようになります。領域がクールダウン中の場合は、正常性値は定期的に減少し、領域が受け取る処理が徐々に少なくなり、最後にターゲットとして不適格となります。CICS のウォームアップまたはクールダウン中に行われることについて詳しくは、[z/OS ワークロード・マネージャー正常性サービスの使用による CICS のウォームアップおよび冷却](#)を参照してください。

CICSplex SM のルーティングの決定で、ワークロード・ルーティングのターゲットとして領域を意図的に不適格にすることができます。これを行うには、「**MVS ワークロード管理**」(MVSWLM) ビューでその領域の z/OS WLM ヘルスのオープン状況を CLOSE に設定します。これにより、領域の望ましが徐々に低下し、領域の z/OS WLM 正常性値が 0 にまで下がると、領域が不適格になったことを示します。また、z/OS WLM ヘルスのオープン状況を IMMCLOSE に設定することによって、領域の z/OS WLM ヘルス値を即時にゼロに変更できます。

## ワークロードに関連付けられている未解決の CICSplex SM リアルタイム分析 (RTA) イベント (ある場合)

未解決のイベントの重大度に応じた算術的な重みを割り当てることによって因数化されます。これらのイベントは、ワークロードまたはそれに関連付けられているトランザクション・グループ定義の WLM 仕様でイベント名が指定されている場合にだけ、考慮されます。

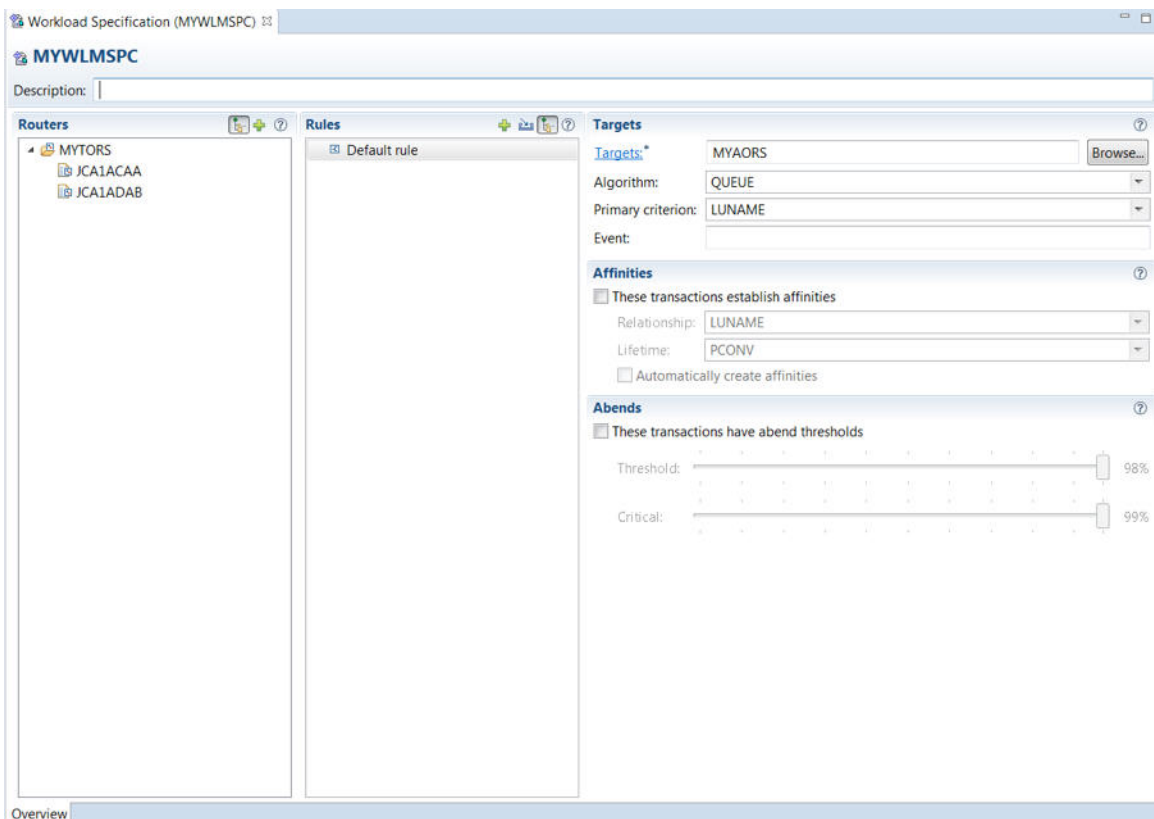
## 動的ルーティングの決定をオーバーライドするために未解決になっているトランザクション親和性 (存在する場合)

他のどのような要因にも関係なく、ルーティング要求に未解決の親和性に関連付けられている場合、その親和性は常にルート決定をオーバーライドします。

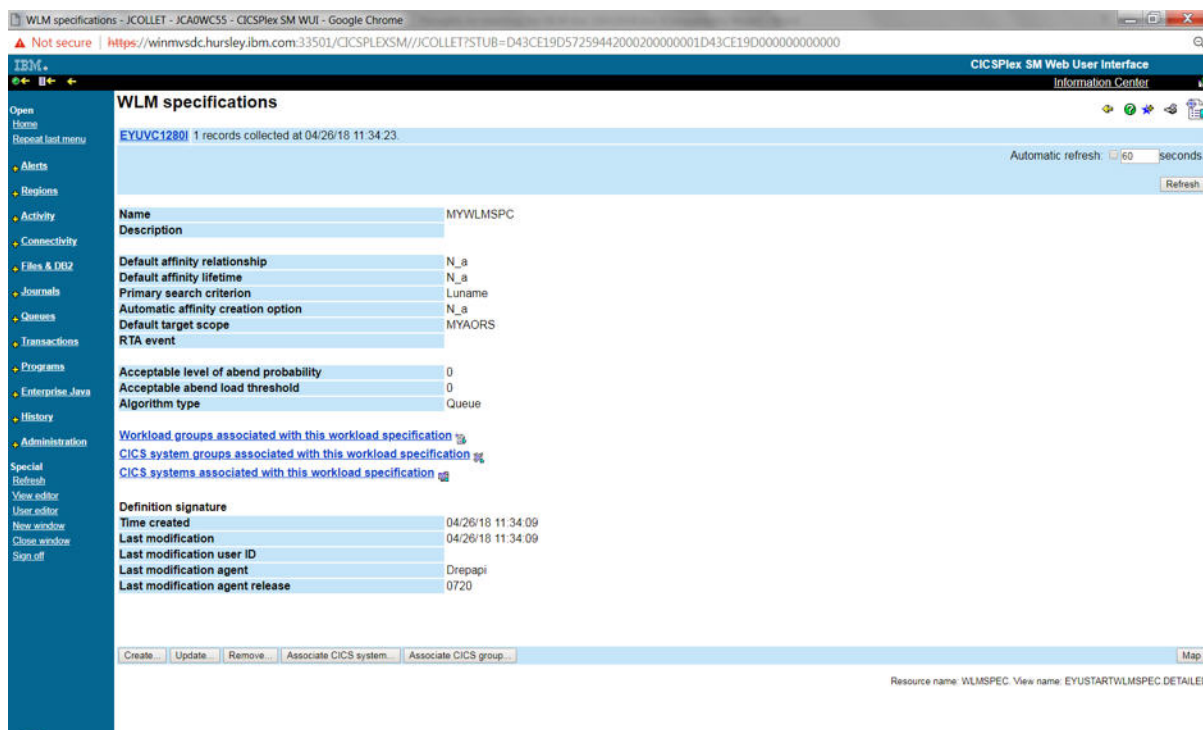
## 最も単純な CICSplex SM WLM 構成

最も基本的な WLM 構成では、ターゲット・スコープ内の CICS システム中のすべての動的トランザクションおよびプログラム・リンクに対するワークロード・バランシングが提供され、オプションとして、デフォルトの親和性タイプおよび存続時間が受け入れられます。対象のワークロードを指定し、処理のルーティング先となるデフォルト領域セットを示す、1 つのワークロード仕様 (WLMSPEC) だけを含む構成を使って、CICSplex SM のワークロード管理を使用することができます。

以下の例では、ターゲット・スコープ MYAORS が指定された、MYWLMSPC という WLMSPEC を示します。ターゲット・スコープは、処理が実行される可能性があるいくつかの領域を含む CICS システム・グループです。次の図は、CICS Explorer で WLMSPEC がどのように表示されるかを示しています。



次の図は、CICSplex SM WUI でどのように表示されるかを示しています。



ほとんどの状況では、上記の例で処理できるよりも複雑な処理が伴います。例えば、すべてのトランザクションをどのターゲット領域にルーティングしてもよいとは限りません。ルーティングのために特定のトランザクションを指定し、ある形式のワークロード分離とワークロード親和性を設定する必要がある場合があります。このような拡張のためには、ワークロード管理定義 (WLMDEF)、ワークロード管理グループ (WLMGROUP)、トランザクション・グループ (TRNGRP) など、いくつかの CICSplex SM WLM 成果物を追加的に構成する必要がありますが、これらは徐々に追加していくことができます。

## 仕組み: z/OS ワークロード・マネージャー (WLM)

z/OS システムのワークロード・マネージャー・コンポーネント (WLM と呼ばれる) は、シスプレックスをモニターし、ユーザーが定義した目標を満たすために、シスプレックス内の各処理項目にどの程度のリソースを割り当てる必要があるかを決定します。また、処理に関するデータもレポートします。

z/OS ワークロード・マネージャーに指定するパラメーターのセットをサービス定義と呼びます。サービス定義には、z/OS ワークロード・マネージャーにシスプレックス内の処理を管理する方法を指示するためにセットアップするポリシー、グループ設定、ルール、およびクラスの組み合わせが含まれています。サービス定義を使用して行える主な項目を以下に示します。

- z/OS ワークロード・マネージャーで管理できるシスプレックス内の各処理項目 (CICS 領域や CICS トランザクションなど) を特定する。
- 要件が類似する処理項目をまとめてグループ化する。
- 処理タイプごとに実装する必要のあるパフォーマンス目標を設定する。
- 各処理タイプのレポート方法を指定する。

ISPF の WLM 管理アプリケーションを使用して、ワークロード管理用のパラメーターをセットアップします。

CICS 領域は、2 つの方法で z/OS ワークロード・マネージャーに識別されます。

- JES バッチ・ジョブまたは開始タスク (STC) のいずれかである、開始サブシステムに基づくアドレス・スペースとして識別する。JES または STC 種別ルールは、JCL から取得されたジョブ名を使用してセットアップされます。これは、CICS 処理を領域またはトランザクションのどちらで管理するように選択した場合でも、常に実行する必要があります。
- CICS サブシステム 種別ルールを使用して、CICS サブシステムとして識別する。CICS サブシステム 種別ルールは、CICS 領域のアプリケーション ID を使用してセットアップされます。個々のトランザクション、または類似する特徴を持つトランザクション・グループに対するサブルールを、CICS サブシステム 種別ルールの下にセットアップできます。トランザクション別に CICS の処理を管理する場合にのみ、これを行う必要があります。

### z/OS ワークロード・マネージャー: サービス・クラスおよびレポート・クラス

z/OS ワークロード・マネージャーの管理対象にはすべてサービス・クラスが設定されており、オプションでレポート・クラスを設定できます。サービス・クラスは処理の管理プロセスに関連し、レポート・クラスは処理に関するデータ収集のプロセスに関連します。

- サービス・クラスは、パフォーマンス目標、リソース要件、および業務上の重要度が類似している処理のグループです。z/OS ワークロード・マネージャーは、サービス・クラスに割り当てられたパフォーマンス目標と、そのパフォーマンス目標に割り当てられた業務上の重要度に従って処理の各グループを管理します。
- レポート・クラスは、z/OS ワークロード・マネージャーがデータをレポートする処理のグループまたは項目です。デフォルトで、データはサービス・クラスごとにまとめてレポートされるため、1 つのサービス・クラス内の異なる項目を識別する必要がある場合は、レポート・クラスが必要になります。

z/OS ワークロード・マネージャーでは、CICS 領域に対して、またはトランザクションに対して、サービス・クラスとレポート・クラスを割り当てることができます。各項目のサービス・クラスとレポート・クラスを選択して調整できます。例えば、2 つの CICS 領域が同じサービス・クラスで定義されているため、それらの領域のパフォーマンス目標とシステム・リソースに対するアクセス・レベルは同じになりますが、異なるレポート・クラスを設定すれば、システム・リソースの使用状況を別個にレポートすることができます。最善の結果を得るために、目標と重要度が同じである同種の処理は可能な限り同じサービス・クラスに入れる必要がありますが、レポートに必要な細分度に応じて、必要な数のレポート・クラスを使用する必要があります。

シスプレックスのサービス定義をセットアップする手順の一部として、サービス・クラスとレポート・クラスをセットアップします。

## CICS 領域およびトランザクションのサービス・クラス

z/OS ワークロード・マネージャーに定義するすべての CICS 領域とトランザクションには、サービス・クラスが必要です。同じサービス・クラスの処理は 1 つのエンティティとして管理されます。

各サービス・クラスはパフォーマンス目標に関連付けられます。この目標によって、z/OS ワークロード・マネージャーがサービス・クラス内の処理を管理する際の目標が指定されます。例えば、目標によって、サービス・クラス内のトランザクションに関する平均応答時間を指定できます。また、パフォーマンス目標には重要度レベルも割り当てます。これは、目標の達成に問題がある場合に適用されます。重要度レベルによって、シスプレックス内の他の処理に設定された目標を満たすことと比較して、その目標を満たすことがどの程度重要であるかが z/OS ワークロード・マネージャーに指示されます。

CICS アドレス・スペースに適用されるサービス・クラスの場合 (JES または STC 種別ルールに従って定義される)、実行速度の目標のみを割り当てることができます。このタイプの目標は、処理の実行準備が整ったときのその処理に対する遅延の許容量を指定します。

CICS 領域のアプリケーション ID またはトランザクションに適用されるサービス・クラスの場合 (CICS サブシステム 種別ルールによって定義)、応答時間の目標のみを割り当てることができます。このタイプの目標は、サービス・クラス内のトランザクションに対する平均応答時間、または指定された割合のトランザクションによって満たす必要のある応答時間 (百分位数で示される応答時間) のいずれかを指定します。百分位数で示される応答時間は、長期にわたって実行されるトランザクションがある場合に便利です。

CICS ワークロードの管理に使用する目標を選択できます。学習パスの次のトピックでは、これについて説明します。

CICS モニター・ドメイン統計は、アドレス・スペースのパフォーマンス目標のタイプや重要度など、CICS アドレス・スペースに関する z/OS ワークロード・マネージャーの設定に関する情報 (JES または STC 種別ルールによって定義) を示します。

## CICS 領域およびトランザクションのレポート・クラス

複数の項目に同じレポート・クラスがある場合、そのレポート・クラスのすべての項目に関するデータは合計としてまとめてレポートされます。異なるサービス・クラスの項目を同じレポート・クラスに入れると、異種レポート・クラスと呼ばれるクラスになります。(同じサービス・クラスの項目が含まれるレポート・クラスは同種レポート・クラスと呼ばれます)。異種レポート・クラスでは、サービス・クラスが異なることにより、異なる目標、重要度、または期間に基づいてデータが収集されるため、誤ったパフォーマンス・データが生成される場合があります。CICS 領域のリソース使用に関して正確な測定値が必要な場合、またはトランザクションの応答時間に関して有意義なデータが必要な場合は、独自のレポート・クラスを個別に設定して、領域またはトランザクションを定義する必要があります。

CICS アドレス・スペースに割り当てられたレポート・クラスに関して生成されたデータ (JES または STC 種別ルールに従って定義) では、CICS 領域によって使用されるプロセッサ時間は得られますが、領域によって実行されるトランザクションの数やそれらの応答時間は得られません。逆に、CICS 領域のアプリケーション ID またはトランザクションに関するレポート・クラス用に生成されたデータ (CICS サブシステム 種別ルールに従って定義) では、実行されたトランザクション数や応答時間は得られますが、CICS 領域によって使用されたプロセッサ時間は得られません。

CICS 領域のワークロードの全体像を得るには、実行する CICS アドレス・スペースとトランザクションの両方を、サービス・クラスとレポート・クラスで定義する必要があります。サービス・クラスは z/OS ワークロード・マネージャーで必要なため、トランザクションにサービス・クラスを使用する予定ではない場合でも、レポート・クラスをセットアップするためにサービス・クラスを定義する必要があります。

## z/OS ワークロード・マネージャー: 領域およびトランザクションの目標

z/OS ワークロード・マネージャーは、領域の目標またはトランザクション応答時間の目標に近づくように、CICS の処理を管理できます。使用する目標を選択できます。

領域の目標を対象にして管理するように選択した場合、z/OS ワークロード・マネージャーは、JES または STC 種別ルールに基づいて CICS アドレス・スペースに割り当てられているサービス・クラスの目標を使用します。トランザクションの目標を対象にして管理するように選択した場合、z/OS ワークロード・マネージャーは、CICS サブシステム 種別ルールに基づいてトランザクション、またはトランザクションのグルー

プに割り当てられているサービス・クラスの目標を使用します。JES または STC 種別ルールを処理する際に、使用するモードを選択できます。

トランザクションの目標を対象にして管理する場合、z/OS ワークロード・マネージャーはトランザクション用のリソース割り振りを直接的には管理しません。代わりに、トランザクションを実行できる CICS 領域 (複数可) に適切なリソースを割り当てるための計算を行います。これは、領域にさまざまなトランザクションと応答時間の目標が混在している場合、うまく機能しない場合があります。このような場合は、領域の目標を対象にして管理するとうまくいく可能性があります。

場合によっては、単一の処理要求を処理するために、CICS 領域の複数のトランザクションが必要になることがあります。例えば、CICS プロバイダー領域でインバウンド SOAP 要求を処理するために、異なるトランザクション ID を持つ、最大で 4 つのトランザクションが必要になる場合があります。トランザクション目標と領域目標のどちらを使用するかを決定する際には、この点を考慮してください。

## 次に行うこと

z/OS ワークロード・マネージャーについてさらに詳しく知りたい場合や、クラスおよび目標のセットアップ方法を学習したい場合は、以下のリソースを参照してください。

- [「z/OS MVS 計画: ワークロード管理」](#)には、z/OS ワークロード・マネージャーの完全な技術的説明と、WLM ISPF アプリケーションの使用に関する説明が記載されています。
- [IBM Redbooks: システム・プログラマーの手引き: ワークロード・マネージャー](#)には、z/OS ワークロード・マネージャーの機能が説明されており、CICS トランザクションや領域に対して効率的に分類やレポート作成を行うための具体的なヒントなど、これらの機能を実装するための情報とベスト・プラクティス・サンプルが提供されています。
- [IBM Workload Manager for z/OS Web ページ](#)には、コンポーネントに追加された機能の記事、プレゼンテーション、履歴など、z/OS ワークロード・マネージャーのリソースがさらに用意されています。





## 第2章 ワークロード管理の構成

CICSplex SM ワークロード管理は、社内のプロセッサ・キャパシティーを最適化するために、存在するトランザクション親和性を考慮に入れたうえで、トランザクションおよびプログラムをその時点で最も適切な CICS 領域に動的にルーティングします。

### 始める前に

1. ワークロード管理に関係する概念、およびトランザクションとプログラムの動的ルーティングを良く理解しておきます。詳しくは、[ワークロード管理](#)を参照してください。
2. 企業において CICSplex SM ワークロード管理を自分がどの程度使用できるかを判別しておきます。これには、企業システムで処理されるワークロードおよびトランザクション間の類縁性を識別することが含まれます。[WLM の計画](#)の説明に従ってください。
3. [WLM の実施](#)で説明されている実装プロセスに従っていることを確認します。

### このタスクについて

ワークロードを設定する場合、作業自体と CICS システム (要求側領域、ルーティング領域、およびターゲット領域として機能する) を関連付けて、単一の動的エンティティーを形成します。このエンティティー内で、作業を以下の宛先へルーティングすることができます。

- 可用性に基づいて選択されたターゲット領域。このタイプのルーティングは ワークロード・バランシングとも呼ばれ、1つのワークロードに関連付けられたすべてのターゲット領域にわたって、作業アクティビティーのバランスを取ることができます。追加情報については、[19 ページの『ワークロード内の作業の管理』](#)を参照してください。
- 特定の基準に基づいたターゲット領域のサブセット。このタイプのルーティングは ワークロード分離と呼ばれ、トランザクションとプログラムのオカレンスを分離して、それらをさまざまなターゲット領域サブセットへ送ることができます。アクティビティーは、サブセット内のすべてのターゲット領域にわたってバランスが取られます。

以下の基準に基づいて、トランザクションやプログラムを分離することができます。

- トランザクションまたはプログラムのオカレンスに関連付けられている端末 ID およびユーザー ID。
- CICS BTS アクティビティーに関連付けられているプロセス・タイプ
- トランザクション。

追加情報については、[22 ページの『ワークロード内の作業の分離』](#)を参照してください。

- 親和性の関係および存続時間に基づいて選択されたターゲット領域。このタイプのルーティングは、ターゲット領域のトランザクションの親和性に基づいており、特定のトランザクション・オカレンスを、指定した期間、同じターゲット領域へルーティングすることができます。詳しくは、[29 ページの『親和性の関係の考慮』](#)を参照してください。

CICSplex に関連付けられた同じワークロード内または異なるワークロード内で、ワークロード・ルーティングとワークロード分離を同時にアクティブにすることができます。

### 注:

1. [この例](#)に示されているように、CICSplex SM のリアルタイム分析機能を使用して、ワークロード管理時にターゲット領域の選択に役立つデータを生成できます。ただし、CICSplex SM ワークロード・マネージャー (WLM) ヘルスは独立して評価されるため、RTA がアクティブである必要はありません。CICSplex SM WLM ヘルスは SAM 設定の影響も受けません。
2. CICSplex SM での動的ルーティングの処理方法、およびこの機能をカスタマイズする方法について詳しくは、[CICSplex SM を使用した動的ルーティング](#)を参照してください。

## ワークロードの要件

すべての CICS システムは、特定の条件の下で、要求側領域、ルーティング領域、またはターゲット領域として機能することができます。

- ワークロードに関連付けられているすべての CICS システムは、同じ CICSplex の一部であるか、または、BTS のプロセスおよびアクティビティーの場合は、同じ BTS セットの一部分です。それらのシステムが同じ MVS イメージ内に存在する必要はありません。
- 端末関連と非端末関連のどちらの **EXEC CICS START** コマンドの動的ルーティングでも、要求側領域、ルーティング領域、およびターゲット領域は、サポートされているリリースの CICS Transaction Server でなければなりません。
- DPL の動的ルーティングの場合、ルーティング領域は、サポートされているリリースの CICS Transaction Server でなければなりませんが、ターゲット領域は任意のレベルの CICS でかまいません。
- CICS BTS アクティビティーの場合、ルーティング領域 (これは要求側領域でもあります) およびターゲット領域は、サポートされているリリースの CICS Transaction Server でなければなりません。また、同じシスプレックスに含まれている必要があります。
- トランザクションの動的ルーティングおよび静的ルーティングの場合、ルーティング領域として機能する CICS システムは、CICS Transaction Server を実行している必要があります。ターゲット領域として機能する CICS システムは、CICSplex SM によってサポートされている任意の CICS プラットフォームの任意のバージョンを実行できます。
- エンタープライズ Bean 関連のトランザクションの動的ルーティングの場合、ルーティング領域およびターゲット領域として機能する CICS システムは、サポートされているリリースの CICS Transaction Server を実行している必要があります。
- ルーティング領域は、CICSplex の管理に関与する CMAS に対してローカルとして定義される必要があります。その CMAS と通信するために、CICSplex SM 機能を使用する必要があります。
- Link3270 ブリッジ要求の場合、ターゲット領域は、サポートされているリリースの CICS Transaction Server でなければなりません。
- ワークロード管理に関与する領域は、互いの間にサービス中の ACQUIRED 接続を備えている必要があります。

CICS システムは、1 つ以上のワークロードでターゲット領域として機能することができます。ただし、ルーティング領域としては、1 つのワークロードでのみ機能することができます。

**注:** 1 つの CICS システムが、同じワークロード内でルーティング領域とターゲット領域のどちらとしても機能することができます。

ターゲット領域として機能する CICS システムが他のターゲット領域と結合されて CICS システム・グループを形成する場合、それぞれのターゲット領域は、その CICS システム・グループヘルディングされる可能性があるすべてのトランザクションに必要なすべてのリソースにアクセスできなければなりません。

CICS BTS ワークロードの場合、それらのターゲット領域はすべて、同じシスプレックスに含まれている必要があります。

## ワークロードの確立

ワークロードの識別に使用される基準は、ワークロード仕様によって提供される情報、1 つ以上のワークロード定義とそのオプションのトランザクション・グループ、および 1 つ以上のワークロード・グループに基づきます。

- ワークロード仕様では、ルーティング領域およびデフォルトのターゲット領域として扱う CICSplex 内の CICS システムまたは CICS システム・グループを指定します。また、ワークロード仕様を使用して、デフォルトの親和性関係および存続期間を指定することもできます。

1 つの CICS 領域に、ワークロード仕様 (WLMSPEC) を 1 つも関連付けないか、または 1 つ関連付けることができます。CICSplex SM ルーティング領域にする CICS 領域には、ワークロードを関連付ける必要があります。CICSplex SM ターゲット領域にする CICS 領域の場合は、使用するルーティングのタイプに応じて以下のように異なります。

- 動的ルーティングの場合、ターゲット領域はワークロード (WLMSPEC) に関連付けられません。ターゲット領域は 1 つ以上のワークロードのターゲットにすることができます。



- 分散ルーティングの場合、ターゲット領域はワークロード (WLMSPEC) に関連付ける必要があります。ターゲット領域は1つのワークロードの1つのターゲットにすることができます。

現在1つのみの動的ルーティングを使用している場合は、ターゲット領域をその唯一のワークロードのターゲットとすることをお勧めします。これにより、後で、分散ルーティングを実装することを決定した場合に、状況が簡単になります。

ワークロード仕様の名前が、ワークロード自体の名前になります。

- ワークロード定義には、固有の、または一般的な端末名とユーザー名や、特定のプロセス・タイプ、または総称プロセス・タイプが含まれ、これらは、トランザクション・オカレンスとワークロード定義の突き合わせを試行するときに使用されます。また、これにより、基準に一致したトランザクション・オカレンスをルーティングするターゲット領域が識別されます。
- トランザクション・グループによって1つ以上の関連トランザクション ID が識別され、ワークロード定義に示された一連のターゲット領域に、どのトランザクションをルーティングするかを決定する1次選択基準として、ワークロード定義に指定された端末 ID やユーザー ID が使用されるかどうかを示されます。トランザクション・グループは親和性の定義にも使用されます。
- ワークロード・グループは、単一のエンティティとして扱われるワークロード定義の集合となります。

**注:** ワークロード定義がワークロードにインストールされていない場合、すべてのトランザクションとプログラムは、ワークロード仕様によって示されたデフォルト・ターゲット・スコープのターゲット領域にルーティングされます。ただし、ワークロード定義がワークロードにインストールされている場合、そのワークロード定義の基準に一致するすべてのトランザクションとプログラムは、定義自体によって示されたターゲット・スコープのターゲット領域にルーティングされます。ワークロード定義に関連付けられていない動的ルーティング要求は、ワークロード仕様のデフォルト・ターゲット・スコープのターゲット領域にルーティングされます。

## ワークロード内の作業の管理

一連の要求領域から開始されたすべての動的トランザクションとプログラムは、ルーティング領域によって、同じ CICSplex 内の特定の連続のターゲット領域にルーティングできます。

各要求のルーティング先となる特定のターゲット領域は、そのセット内のすべてのターゲット領域のアクティビティと可用性によって決定されます。

ワークロードのルーティングを確立するには、ワークロード仕様を定義することのみが必要となります。

20 ページの図 3 を使用して動的ルーティング・プロセスについて説明します。この図は、スターター・セットの構成を示しています。動的トランザクション・ルーティングの場合、要求側領域 EYUMAS1A (TOR) に関連付けられた端末から開始されたトランザクションが、CICS システム・グループ EYUCSG01 内の最も適切なターゲット領域 (AOR) にルーティングされます。

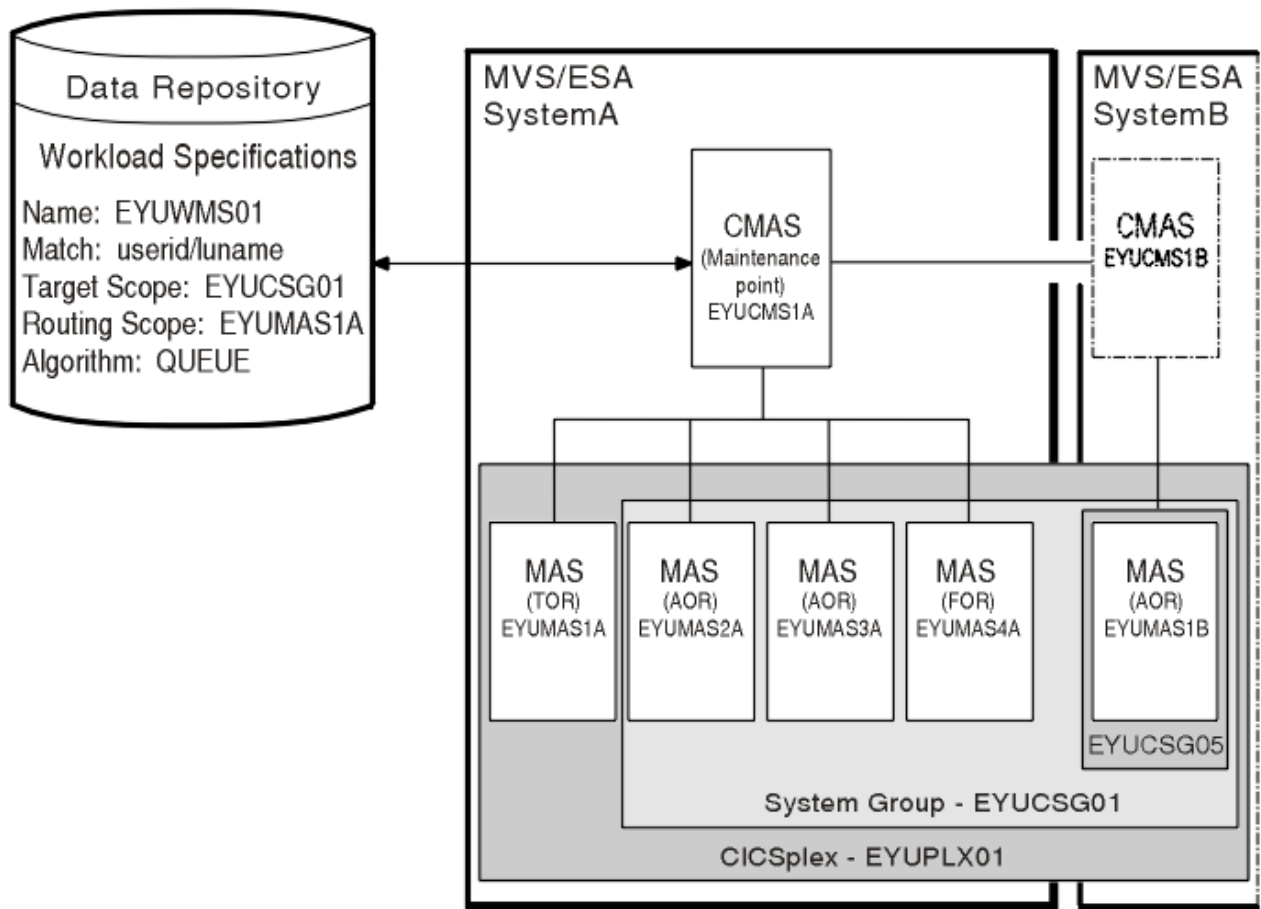


図 3. 動的ルーティングのためのワークロード定義のサンプル

**EXEC CICS START TRANSID TERMID** コマンドの動的ルーティングでは、要求領域 EYUMAS2A (AOR) で開始されたすべてのトランザクションは、EYUMAS1A (TOR)、つまり、START コマンドの TERMID オプションで示された端末に関連付けられたルーティング領域に送信されます。ルーティング領域は、CICS システム・グループ EYUCSG01 内の最も適切なターゲット領域 (AOR) にトランザクションを送信します。

動的プログラム・リンクの場合は、可能なシナリオが 2 つあります。インバウンド・クライアント要求で、要求が TOR EYUMAS1A で受信され、これが要求領域およびルーティング領域として動作します。ターゲット領域は、CICS システム・グループ EYUCSG01 内のいずれかの AOR になります。ピアツーピア要求の場合、要求は、EYUMAS2A (AOR) で実行されるトランザクションによって開始されます。EYUMAS2A はルーティング領域として機能し、ターゲット領域は CICS システム・グループ EYUCSG01 内のいずれかの AOR になります。

### キュー・アルゴリズムの使用

キュー・アルゴリズムを使用するワークロード処理の間、CICSplex SM は、要求領域から開始されたすべてのトランザクションおよびプログラムを、ターゲット領域の指定セット内で最も適切なターゲット領域にルーティングします。[キュー・アルゴリズム](#)を参照してください。

### リンクに依存しないキュー・アルゴリズムの使用

リンクに依存しないキュー (LNQUEUE) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、キュー・アルゴリズムと同等です。[リンクに依存しないキュー・アルゴリズム](#)を参照してください。

## ゴール・アルゴリズムの使用

CICSplex SM は、z/OS ゴール・アルゴリズムをサポートしています。ゴール・アルゴリズムでは、1つのワークロード内のすべての作業に対して、または、定義された、平均、または百分位数応答時間の各ゴールを最も満たすことができるターゲット領域を選択します。

ゴール (目標) を定義するには、z/OS ワークロード・マネージャーを使用して、トランザクションにサービス・クラスを関連付けます。サービス・クラスは、トランザクション、LU 名、およびユーザー ID に基づいて割り当てられます。サービス・クラスでは、複数のタイプの応答時間のゴールを定義することができます。ただし、CICSplex SM が認識するのは、平均とパーセンタイルの応答時間目標のみです。トランザクションに速度または任意の各ゴールが与えられた場合、それらのゴールは満たされると想定されます。CICSplex SM は、サービス・クラス・レベルで管理します。つまり、トランザクション特性の内部的な情報は持ちません。一貫してサービス・クラスをターゲット領域のセットに割り振ることで、z/OS ワークロード・マネージャーが再割り振りするリソース量を最小化できます。

ルーターとターゲットが同じ CMAS で管理されていれば、以下のようなシナリオでは目標モードを使用し、効率的なルーティング決定処理を実行できます。

- 動的トランザクションのために DTRPGM を使用して動的ルーティングを実行する場合
- APPC 接続または MRO 接続による EXEC CICS START TERMID のために DTRPGM を使用して動的ルーティングを実行する場合
- ビジネス・トランザクション・サービス・ルーティングのために DSRTPGM を使用して分散ルーティングを実行する場合

ゴール・アルゴリズムの詳細については、[ゴール・アルゴリズム](#)および [z/OS MVS 計画: ワークロード管理](#)を参照してください。

サービス・レベル管理者は、ターゲット・システムの基礎的な処理能力に対して現実的な目標を定義する必要があります。属性が類似しているトランザクション (例えば、リソースの消費量が似ているトランザクションや疑似会話型トランザクション) を別々のサービス・クラスに割り当てなければなりません。複数のサービス・クラスで応答時間目標を同じにすることができます。CICS 統計を使用すると、これらのトランザクション集合を定義するのに役立ちます。CICS 統計については、[パフォーマンスの改善](#)を参照してください。

## リンクに依存しないゴール・アルゴリズムの使用

リンクに依存しないゴール (LNGOAL) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、ゴール・アルゴリズムと同等です。[リンクに依存しないゴール・アルゴリズム](#)を参照してください。

## ワークロード・ルーティングの制御レベル

ワークロード・ルーティングを使用するには、ワークロード仕様 (WLMSPEC) レベルでそのワークロードのデフォルトのルーティング・アルゴリズムを指定する必要があります。オプションにより、トランザクション・グループ (TRANGRP) レベルでルーティング・アルゴリズムを指定することができます。トランザクション・グループ内で指定されたアルゴリズムは、ワークロード仕様と関係付けられたデフォルトのアルゴリズムを指定変更します。

デフォルトのルーティング・アルゴリズムは、ワークロード内でルーティングされる各動的トランザクションに適用されます。ただし、ルーティング・アルゴリズムが指定されているトランザクション・グループと関連付けられているトランザクションは除きます。下記のルーティング・アルゴリズムから 1つを指定できます。

- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

ワークロード仕様レベルで指定されたルーティング・アルゴリズムを変更するには、ワークロードが新しいアルゴリズム仕様で最新表示されるように、ワークロードに参加するすべての領域を終了する必要があります。

トランザクション・グループ・レベルでは、ルーティング・アルゴリズムを動的に指定できます。指定された動的ルーティング・アルゴリズムは、トランザクション・グループと関連付けられたルーティングされるすべての動的トランザクションに適用されます。したがって、同じワークロード内の特定のトランザクション・コードに代わりのルーティング・アルゴリズムを適用できます。

トランザクション・グループ・レベルで代わりのルーティング・アルゴリズムを指定する場合、ご使用のルーティング領域を停止させることなく特定のターゲット領域のワークロード・ルーティング特性を動的に変更できます。インストールされたトランザクション・グループを変更する場合、WLM 定義によって指定されたトランザクション・グループも最新表示されるように、トランザクション・グループに関連付けられた WLM 定義 (WLMDEF) を破棄してから、再インストールする必要があります。関連付けられた WLMDEF を破棄したり再インストールしたりせずに、ルーティング・アルゴリズムを直ちに変更するには、「アクティブなワークロード・トランザクション・グループ (WLMATGRP)」ビューおよび **SET** コマンドを使用して、ALGTYPE 属性を変更することができます。

下記のルーティング・アルゴリズムから 1 つを指定できます。

- INHERIT
- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

INHERIT は、トランザクション・グループがそのワークロードに対してワークロード仕様と関連付けられたルーティング・アルゴリズムを使用することを意味します。

## ワークロード内の作業の分離

トランザクションまたはプログラムに関連付けられたユーザーや端末 (あるいはその両方) の名前、トランザクション自体、またはトランザクションに関連付けられた BTS プロセス・タイプを使用して、1 つのワークロード内の作業を分離することができます。

LU 名とユーザー ID、またはプロセス・タイプを使用して、ワークロードを分離することができます。総称プロセス・タイプとユーザー ID などを使用してワークロードを分離することはできません。エンタープライズ Bean ワークロードの分離は、ユーザー ID を介してのみ実装できます。

### 端末またはユーザー名による分離

トランザクションの各オカレンスに関連付けられた端末またはユーザー、あるいはその両方の名前に基づいて、要求領域の単一のセットから、ターゲット領域のさまざまなサブセットに要求をルーティングするワークロードを作成することができます。

例えば、特定の個人が特定の端末から開始したすべての要求を特別なターゲット領域のサブセットにルーティングすることが必要になる場合があります。

23 ページの図 4 は、そのようなワークロードを示しています。このケースでは、トランザクションと関連付けられているユーザー名および端末名が、それぞれ、SM および NET で始まる場合、トランザクションは EYUCSG05 として指定されたターゲット領域のセットにルーティングされます。ユーザー名または端末名が他の文字で始まる場合、トランザクションは、ワークロード仕様に EYUCSG01 として指定されている、ターゲット領域のデフォルトのセットにルーティングされます。

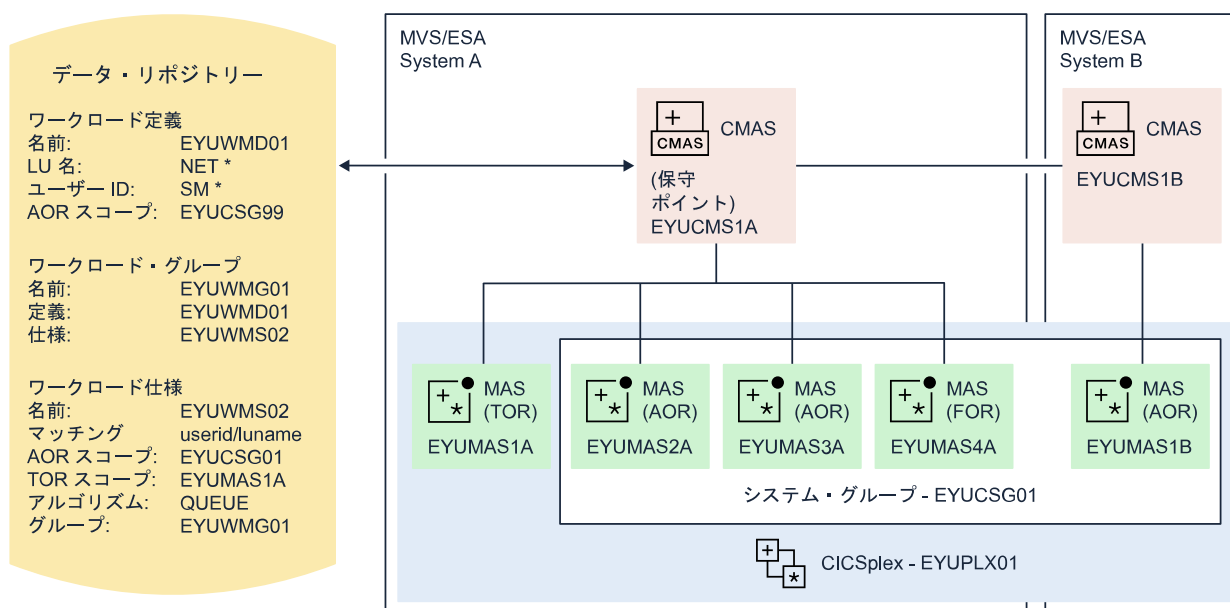


図 4. 端末名およびユーザー名によってワークロードを分離する定義のサンプル

ワークロード処理の過程で、CICSplex SM は要求のルーティング先を決定するために、要求の各オカレンスに関連付けられた端末名とユーザー名を評価します。

- トランザクションに関連付けられた端末名とユーザー名が、インストールされたワークロード定義に指定された選択基準に一致した場合、要求はその定義に指定されたターゲット領域にルーティングされます。
- 端末名やユーザー名が選択基準に一致しない場合、要求は、ワークロード仕様に指定されている、ターゲット領域のデフォルトのセットにルーティングされます。

ターゲット領域の適切なセットを決定した後、CICSplex SM は、そのセットの中のアクティブなターゲット領域の状況に基づいて、ターゲット領域を 1 つ選択します。

### プロセス・タイプによる分離

特定のプロセス・タイプに関連付けられた要求を、特定の 1 つのターゲット領域または特定のセットのターゲット領域にルーティングする CICS BTS ワークロードを作成できます。例えば、STOCK プロセス・タイプに関連付けられたすべての要求を特別なターゲット領域のサブセットにルーティングすることが必要になる場合があります。

25 ページの図 5 は、そのようなワークロードの例を示しています。CICS BTS トランザクションに関連付けられたプロセス・タイプが STOCK の場合、トランザクションは EYUCSG05 として指定されたターゲット領域のセットにルーティングされます。プロセス・タイプが STOCK 以外の場合、トランザクションはワークロード仕様で EYUCSG01 として指定されたデフォルトのターゲット領域のセットにルーティングされます。

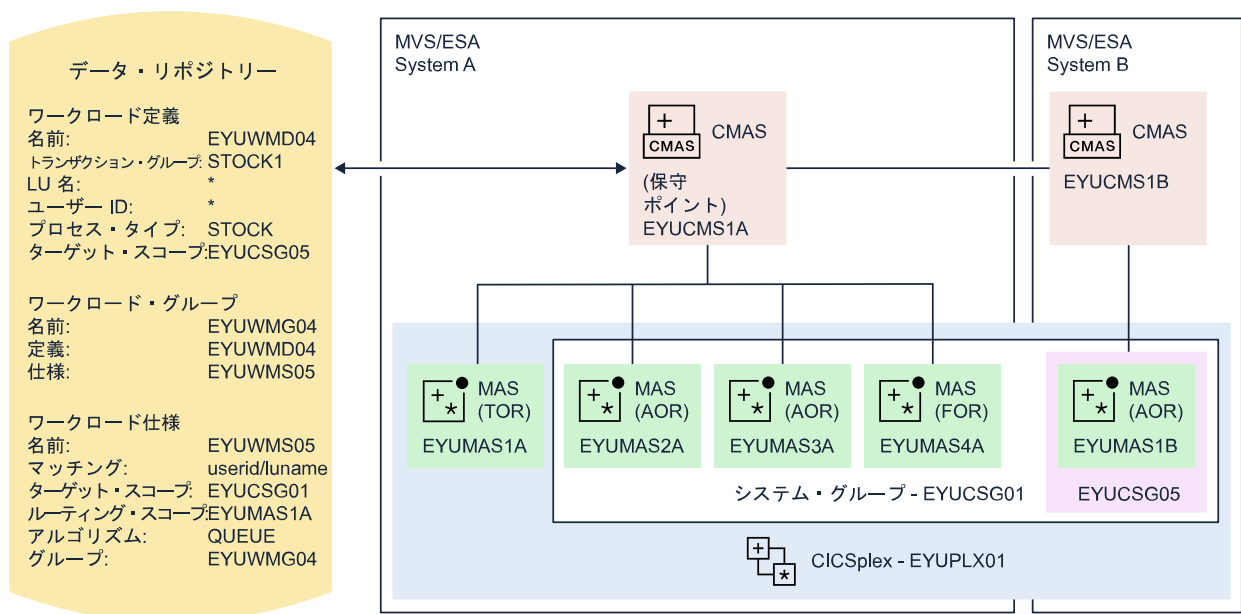


図 5. プロセス・タイプによってワークロードを分離する定義のサンプル



プロセス・タイプでワークロードを分離するには、LU 名およびユーザー ID の各フィールドを \* に設定する必要があります。LU 名とユーザー ID でワークロードを分離する場合は、プロセス・タイプのフィールドを \* に設定する必要があります。エンタープライズ Bean ワークロードを分離する場合は、LU 名とプロセス・タイプの各フィールドを \* に設定する必要があります。ワークロードは、プロセス・タイプで、または LU 名とユーザー ID でのみ分離することができます。

特定のプロセス・タイプまたは総称プロセス・タイプを指定できます。ワークロード分離処理の過程で、CICSplex SM は CICS から提供されたプロセス・タイプを評価して、トランザクションのルーティング先を決定します。

- プロセス・タイプが、インストールされた定義に指定された選択基準に一致した場合、要求はその定義に指定されたターゲット領域にルーティングされます。
- プロセス・タイプが選択基準に一致しない場合は、トランザクションに関連付けられた LU 名とユーザー ID に基づいて一致が見つかります。これらの基準に基づいて一致が見つかった場合、要求は関連付けられた定義で示されたターゲット領域にルーティングされます。
- プロセス・タイプや LU 名とユーザー ID を使用して一致が見つからなかった場合は、要求は、仕様に指定されている、ターゲット領域のデフォルトのセットにルーティングされます。

注：プロセス・タイプによる分離は、LU 名とユーザー ID による分離よりも優先されます。このため、トランザクションに関連付けられたプロセス・タイプ、LU 名およびユーザー ID によって、2 つのワークロード定義、つまり、プロセス・タイプによる分離を指定するものと、LU 名とユーザー ID による分離を指定するものに指定された選択基準が満たされる場合、トランザクションは、プロセス・タイプによる分離を指定しているワークロード定義に指定されたターゲット・スコープの領域にルーティングされます。

### トランザクションによる分離

トランザクション自体に基づいてワークロード内の作業を分離することもできます。例えば、会計部門の端末から開始された、給与関連のトランザクションのすべてのオカレンスを、処理のために、ターゲット領域の特定のセットにルーティングすることが必要になる場合があります。

28 ページの図 6 は、ワークロード内の作業をトランザクション ID に基づいて分離する例を示しています。このケースでは、トランザクション・グループ EYUWMT01 で指定されているすべてのトランザクションと関連付けられているユーザー名および端末名が、それぞれ、SM および NET で始まる場合、トランザクションは EYUCSG05 として指定されたターゲット領域にルーティングされます。トランザクション ID、ユーザー名または端末名が基準に一致しない場合、トランザクションは、EYUCSG01 として指定されている、デフォルトのターゲット領域にルーティングされます。

ワークロード処理の間、CICSplex SM は CICS によって提供されたトランザクション ID を評価して、使用するトランザクション・グループを決定します。

- トランザクションがトランザクション・グループに定義されている場合、CICSplex SM は、そのグループのマッチング・キーが USERID と LUNAME のどちらであるかを認識します。
- トランザクションがトランザクション・グループに含まれていない場合、CICSplex SM は、仕様のデフォルトのトランザクション・グループのマッチング・キーを使用します。突き合わせキーは、関連付けられた仕様で指定された値となります。詳しくは、[トランザクション・グループ定義 - TRANGRP](#) を参照してください。

CICSplex SM は、マッチング・キー値を使用して、トランザクションに関連付けられた端末名とユーザー名を評価する順序を確立します。評価は、トランザクションの送信先を決定するために使用されます。

- トランザクションに関連付けられた端末名とユーザー名が、インストールされたワークロード定義に指定された選択基準に一致した場合、要求はその定義に指定されたターゲット領域にルーティングされます。
- 端末名やユーザー名が選択基準に一致しない場合、要求は、ワークロード仕様に指定されている、ターゲット領域のデフォルトのセットにルーティングされます。

ターゲット領域の適切なセットを決定した後、そのセット内のアクティブなターゲット領域の状況に基づいて 1 つが選択されます。

注：トランザクション・グループに新しいトランザクション・コードを追加したり、トランザクション・グループからトランザクション・コードを削除したりするとき、そのトランザクション・グループの他の属性を変更しない場合は、そのトランザクション・グループを参照する WLM 定義を破棄する必要はありません。そのトランザクション・グループを参照する WLM 定義を再インストールすることによって、アクティ



ブなワークロード内で、トランザクション・グループに追加された新しいトランザクション・コードを動的にアクティブ化できます。

1つのトランザクション (DTRINGRP オブジェクト) は、1つの親トランザクション・グループ (TRANGRP) にのみ関連付けることができます。このことは、親ワークロードの関連付けには関係なく、トランザクション・グループがアクティブに使用されているかどうかにも関係ありません。トランザクション・グループ・レベル、および WLM 定義 (WLMDEF) レベルでの変更は、アクティブなワークロードを終了することなく、動的にアクティブ化できます。WLM 仕様 (WLMSPEC) レベルでの変更についてのみ、通常、アクティブなワークロードを終了することが必要になります。

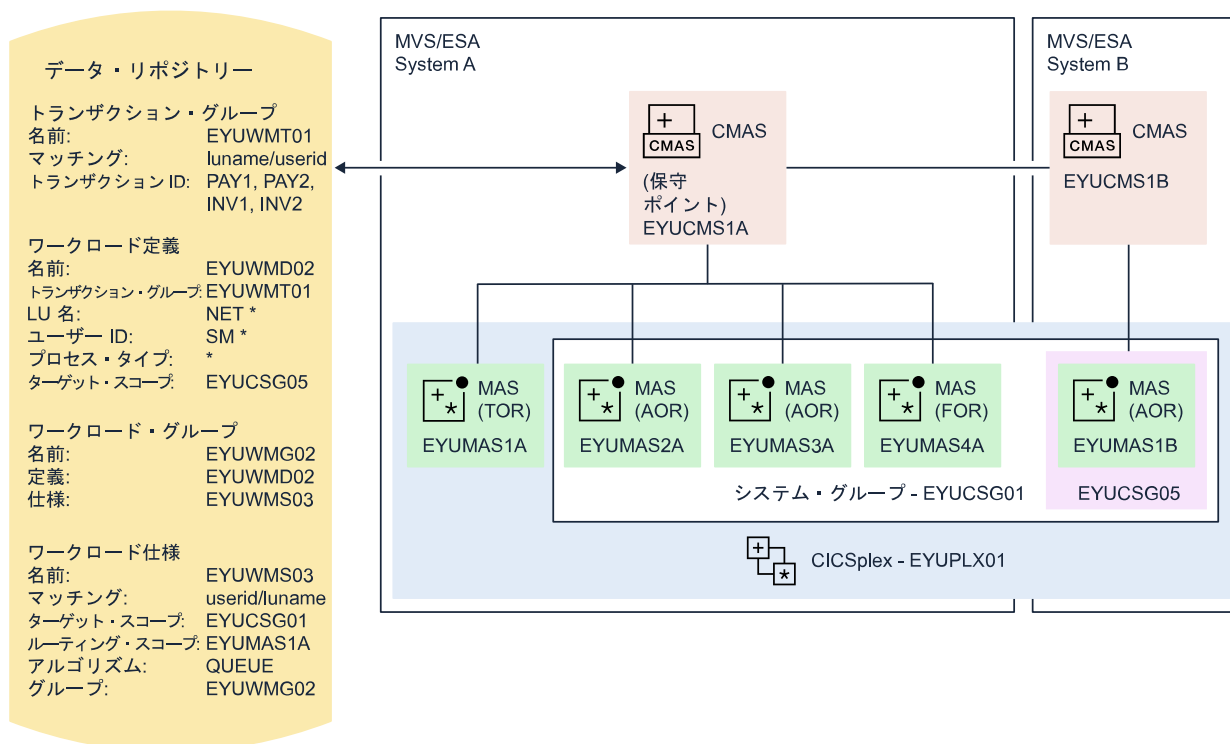


図 6. トランザクションによってワークロードを分離する定義のサンプル

## Link3270 ブリッジのワークロードの分離

Link3270 ブリッジのワークロードは、LU 名、ユーザー ID およびトランザクション・グループで分離できます。

Link3270 ブリッジのワークロードの LU 名は、以下の 3 つの異なる方法で生成されます。

1. ユーザーが Link3270 呼び出しの BRIH-NETNAME パラメーターで指定できます。
2. Link3270 ブリッジ機能によってランダムに生成される場合があります。
3. ユーザー置換が可能な CICS 自動インストール・プログラムと他の 2 つの方法のいずれかを使用して、指定または生成した NETNAME を受け入れる、拒否する、あるいは変更する。

Link3270 ブリッジのワークロードを LU 名によって分離するには、リストされたうち、あらかじめ LU 名が分かる、1 または 3 の方法を使用する必要があります。方法 2 を使用する場合は、LU 名が前もって分からないため、これをワークロードの分離に使用することはできません。

Link3270 ブリッジを開始したクライアント・プログラムに関連付けられた名前ではなく、ブリッジ機能 NETNAME で分離するには、EYU9WRAM モジュールを変更する必要があります。CICS API コマンドの **EXEC CICS ASSIGN USERID()** および **EXEC CICS ASSIGN BRFACILITY NETNAME()** を使用して、ユーザー ID と LU 名を割り当てることができます。DFHDYPDS commarea パラメーター DYRNETNM を経由して返される NETNAME ではなく、**INQUIRE BRFACILITY()** コマンドから返された NETNAME を使用して、ワークロードを分離できます。

Link3270 ブリッジ機能定義について詳しくは、[CICS 外部インターフェースの概要](#)を参照してください。

## 親和性の関係の考慮

親和性とは 2 つ以上のトランザクション間に定義する関係で、その関係の期間または存続時間において有効です。

トランザクション間に親和性の関係が存在するとき、それらのトランザクションは、同じターゲット領域によって処理される必要があります。親和性を使用すると、親和性の関係と存続時間の特定の組み合わせに適用される規則に基づいて、1 つ以上の要求領域から特定のターゲット領域にトランザクションをルーティングすることができます。親和性の関係および存続時間の値のリストについては、[類縁性の関係および類縁性の存続期間](#)を参照してください。

CICSplex の複数の CMAS によってワークロードの親和性が管理され、かつ、これらの CMAS のうちの 1 つが停止し、ローカル MAS が稼働している場合は、ワークロードは凍結されます。ワークロードが凍結されると、変更を加えることができなくなりますが、現在のワークロードはアクティブなままです。

CMAS が停止しているとき、以下のいずれかの親和性存続時間と親和性の関係がある場合は、新しい親和性インスタンスを作成することができず、親和性に関連付けられたターゲット MAS にトランザクションをルーティングすることができません。これは、ワークロードが凍結されている間は、ローカル TOR にワークロードの変更を伝達できないためです。

親和性の関係	親和性存続時間
BAPPL	<ul style="list-style-type: none"><li>• SYSTEM</li><li>• PERMANENT</li><li>• ACTIVITY</li><li>• PROCESS</li></ul>
GLOBAL	<ul style="list-style-type: none"><li>• SYSTEM</li><li>• PERMANENT</li></ul>
LOCKED	<ul style="list-style-type: none"><li>• UOW</li></ul>
LUNAME	<ul style="list-style-type: none"><li>• SYSTEM</li><li>• PERMANENT</li></ul>

親和性の関係	親和性存続時間
USERID	<ul style="list-style-type: none"> <li>• SYSTEM</li> <li>• PERMANENT</li> </ul>

CMAS が再び稼働するようになり、MAS に再接続すると、ワークロードは凍結が解除され、変更できるようになります。

IBM CICS Interdependency Analyzer for z/OS を使用すると、既存のトランザクション間の親和性および BTS プロセスとアクティビティの間の親和性を検出できます。このユーティリティのレポーター・コンポーネントからの出力を、CICSplex SM バッチ・リポジトリ更新機能への入力として使用できます。詳しくは、[CICS Interdependency Analyzer for z/OS の概要](#)を参照してください。

注：CICS Interdependency Analyzer を使用して、非端末関連の EXEC CICS START コマンド間の親和性や、ユーザー ID も端末 ID も関連付けられていない DPL 間の親和性を検出することはできません。このような要求タイプの場合は、すべての親和性を削除または回避するようにして、残った親和性をアプリケーションで適用できるようにする必要があります。

### CICS BTS に関する考慮事項

BTS 自体は親和性をもたらすわけではなく、親和性をもたらすプログラミング手法を抑制するのですが、親和性をもたらす可能性のある既存コードをサポートしています。

ワークロード管理に対しては、このような類縁性を定義しなければなりません。個々の類縁性の存続期間を指定することが、特に重要です。これを指定しないと、ワークロード管理のルーティング・オプションを不必要に制限する可能性があります。

与えられたアクティビティが、同期にも非同期にも実行できることに注意してください。ワークロード管理は、非同期に行われる呼び出しを受け入れることができるのみです。さらに、これらの類縁性、特にアクティビティと処理の類縁性はできる限り発生させないようにしてください。なぜなら、これらの類縁性は、BTS セット上で同期化されるためです。このことによって、システムに重大なパフォーマンス上の影響が生じる可能性があります。

また、CICSplex SM では、類縁性を維持できる最長期間は、ワークロードに使用される CMAS がアクティブである期間に相当することに注意してください。つまり、類縁性として PERMANENT を指定した場合にはなります。システム全体に障害が発生したり、計画シャットダウンが行われたりすると、類縁性は失われますが、CICS のアクティビティは BTS の RLS データ・セットからリカバリーされます。

### Link3270 ブリッジに関する考慮事項

Link3270 ブリッジ・トランザクションの場合、親和性は CICSplex SM ではなく CICS によって管理されます。ワークロード管理ビューの、トランザクション親和性の関係と存続時間の各フィールドはブランクのままにしてください。

### ワークロードへの親和性の追加

非 Link3270 ブリッジ・トランザクションの場合、親和性はワークロード管理ビューの CICSplex SM **トランザクションの親和性の関係**および**存続時間**の各フィールドによって管理されます。これらのフィールドは、特定のシステムにトランザクションを切り分けたり、親和性タイプと存続時間を定義したりするために指定します。

31 ページの図 7 は、ワークロード内の作業をトランザクション ID に基づいて分離し、それらのトランザクションに親和性の関係および存続時間を関連付ける例を示しています。この例では、PAY1 という名前のトランザクションの最初のオカレンスは、NET1 と SMITH という端末名とユーザー名にそれぞれ関連付けられ、EYUCSG05 として指定されたターゲット領域のセット内の、適切なターゲット領域に送信されます。トランザクションを受信する特定のターゲット領域、および、PAY1 が属するトランザクション・グループに関連付けられた親和性の関係と存続時間が記録されます。そのトランザクション・グループのトランザクションの、後続のすべてのオカレンスのうち、端末とユーザー名の基準を満たすものは、指定された期間、同じターゲット領域に送信されます。

注：デフォルトの親和性の関係と存続時間を指定しなかった場合は、ワークロードのルーティングと分離に、同じワークロード仕様を使用できます。

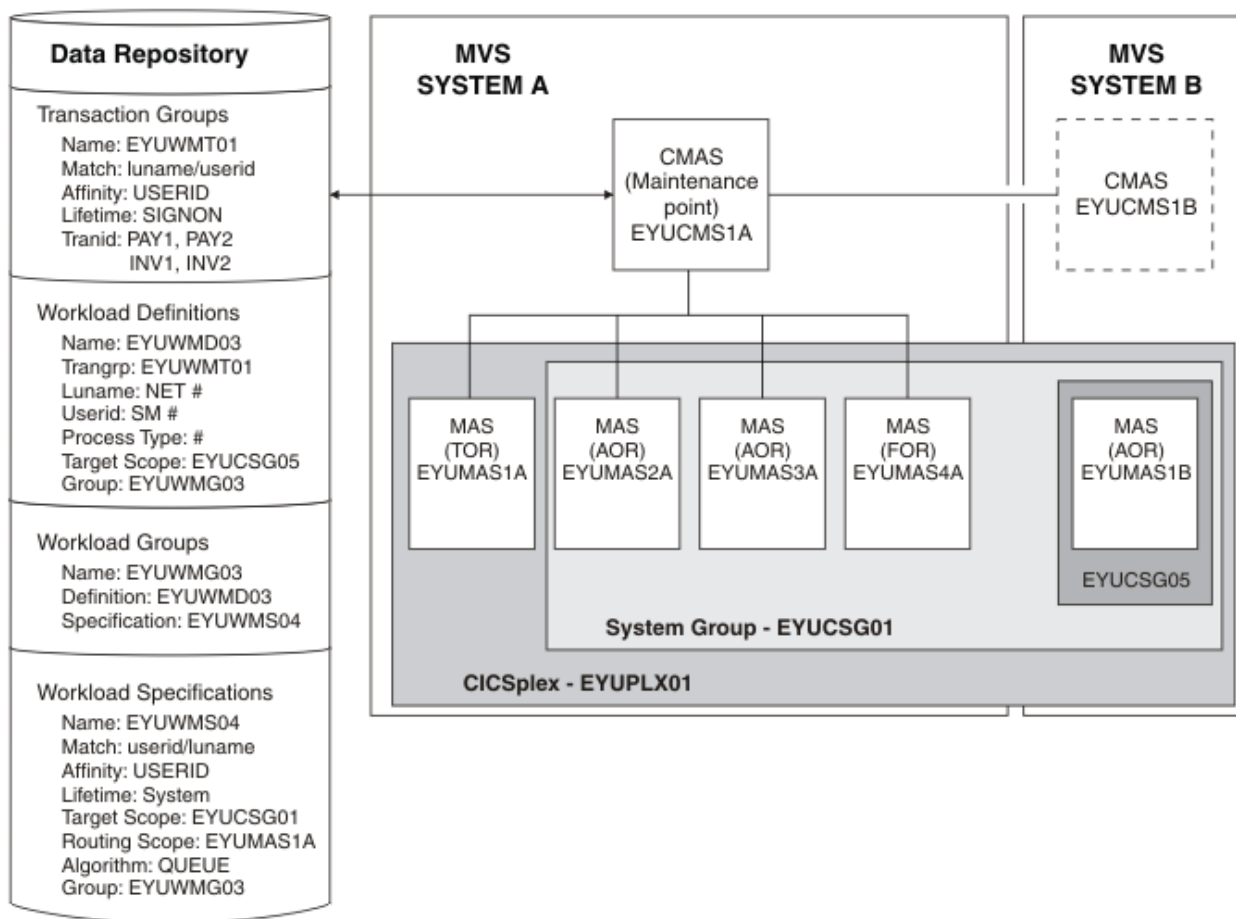


図 7. ワークロードにトランザクション親和性を追加する定義のサンプル

注：ワークロード処理の間、CICSplex SM は CICS によって提供されたトランザクション ID を評価して、使用するトランザクション・グループを決定します。

- トランザクションがトランザクション・グループに定義されている場合、CICSplex SM は、そのグループのマッチング・キーを認識します。
- トランザクションがトランザクション・グループに含まれていない場合、CICSplex SM は、ワークロード仕様のデフォルトのトランザクション・グループのマッチング・キーを使用します。

CICSplex SM は、この値を使用して、トランザクションに関連付けられた端末名とユーザー名を評価する順序を確立します。この評価の目的は、トランザクションの送り先を決定することです。

- トランザクションに関連付けられた端末名とユーザー名が、インストール済みのワークロード定義で指定されている選択基準と一致する場合、CICSplex SM は、親和性の関係と存続時間がトランザクション・グループに関連付けられているかどうかを確認します。
  - 親和性の関係と存続時間がトランザクション・グループに関連付けられている場合：
    - それが最初のおカレンスであれば、CICSplex SM はその親和性の関係および存続時間を保管します。CICSplex SM は次に、ターゲット領域の可用性に基づいてターゲット領域を選択し、そこにトランザクションを送ります。
    - それが最初のおカレンスでなければ、CICSplex SM は、その前に選択したターゲット領域にトランザクションをルーティングします。親和性の関係と存続時間が適用される限り、そのトランザクション・グループのトランザクションの後続のおカレンスは、同じターゲット領域に送信されます。
  - 親和性の関係と存続時間がトランザクション・グループに関連付けられていない場合、トランザクションは、ターゲット領域の指定セット内の最適のターゲット領域にルーティングされます。

- 端末名やユーザー名が選択基準に一致しない場合、トランザクションは、ワークロード仕様に指定されている、ターゲット領域のデフォルトのセットにルーティングされます。そのセット内のアクティブなターゲット領域の状況に基づいて1つが選択されます。

注: 一連の 端末開始トランザクションに含まれているトランザクションが、別々のトランザクション・グループに関連付けられている場合は、各トランザクション・グループの親和性は、そのトランザクション・グループに対して定義された規則に従います。

## 異常終了の確率およびワークロード管理

CICSplex SM は、ターゲット・スコープで識別される各領域の正常性およびアクティビティー (またはロード) を評価することができます。CICSplex SM はこの情報と、CICSplex SM にトランザクション・グループを定義するときに指定する許容できる異常終了ロードしきい値 (ABENDTHRESH) および異常終了する確率の許容レベル (ABENDCRIT) を併用して、トランザクションをルーティングするターゲット 領域を決定します。

CICSplex SM では、トランザクション・グループに関連付けられているトランザクションが異常終了する確率は、トランザクションごとに個別に計算されるか、グループ全体で計算されます。

- 親和性の関係と存続時間が定義されていない、またはワークロード仕様に定義されたデフォルト値を使用するトランザクション・グループにトランザクションが関連付けられている場合、異常終了の確率は、そのトランザクションに対して個別に計算されます。
- 親和性の関係と存続時間が定義されているトランザクション・グループに関連付けられているトランザクションの場合、そのトランザクションの異常終了の確率は、トランザクションのグループ全体の異常終了の確率として計算されます。

トランザクション・グループに関連付けられていないトランザクションの異常終了の確率は、ワークロード仕様に定義された、デフォルトの異常終了の確率の値から導出されます。

許容できる異常終了ロードしきい値 (ABENDTHRESH) および異常終了する確率の許容レベル (ABENDCRIT) を指定すると、ターゲットになる可能性のある領域の正常性とロードの両方に適用されます。これらの値は、以下の表に示すように、WUI または CICS® TS Explorer を使用して、ワークロード仕様 (デフォルト規則) またはトランザクション・グループ定義 (特定の規則に関連付けられる) のいずれかに指定できます。

表 2. WUI および CICS TS Explorer で ABENDTHRESH および ABENDCRIT に相当するフィールド		
パラメーター	WUI	CICS Explorer
ABENDTHRESH	許容できる異常終了ロードしきい値	しきい値
ABENDCRIT	異常終了する確率の許容レベル	クリティカル

ターゲット領域で未処理の異常終了が発生した場合、次のようになります。

- 異常終了したトランザクション、または異常終了したトランザクションに関連するトランザクションは、ターゲット領域における異常終了の確率のパーセンテージが 100% に設定されます。その結果、ターゲット領域は即座に正常でないと見なされます。
- それから CICSplex SM は、異常終了する確率のパーセンテージを徐々に減らしていきます。異常終了する確率のパーセンテージが、異常終了する確率の許容レベル (ABENDCRIT) の指定を下回った場合、まだ許容できる異常終了ロードしきい値 (ABENDTHRESH) の指定を上回っていても、ターゲット領域は非正常と見なされなくなります。しかし、その領域の認識されるロードが 2 倍にされるので、同量をロードされた他の領域と比較して、ルーティングのターゲットとしての魅力が低くなります。
- 引き続き CICSplex SM は異常終了する確率のパーセンテージを減らします。異常終了する確率のパーセンテージが、許容できる異常終了ロードしきい値を下回ると、ターゲット領域のロードが正常と設定され、ルーティングのターゲットとしての魅力は、同量をロードされた他の領域と等しくなります。
- いずれかの時点で未処理の異常終了がさらに発生する場合、処理はリセットされ、異常終了の確率は最初の開始点から再計算されて、領域は正常でないと見なされます。



- ターゲット領域が非正常と見なされると、CICSplex SM はトランザクションをそのターゲット領域にルーティングしないようにします。ただし、より正常なターゲット領域がない場合は、トランザクションがそのターゲット領域にルーティングされることがあります。

注：

- 異常終了の確率は、動的ルーティング出口 DTRPGM を使用している場合にのみ考慮されます。分散ルーティング出口 DSRTPGM を使用している場合、異常終了の確率は考慮されません。
- 親和性がアクティブである場合、すべての後続トランザクションは、親和性がアクティブである限り、異常終了の確率に関係なく、同じターゲット領域にルーティングされます。
- EXEC CICS HANDLE ABEND** コマンドがアクティブなターゲット領域でトランザクションが異常終了した場合、CICSplex SM は異常終了が発生したという情報を受け取りません。このため、CICSplex SM では、そのターゲット領域におけるトランザクションの異常終了の確率を計算するときに、そのような異常終了を考慮することができません。この状況においては、異常終了が発生しても、異常終了の確率は低いままになる場合があります。
- アプリケーションが、リソース・マネージャーへの接続がアクティブでないことを認識してリソース・マネージャーへの呼び出しを行わない場合、または接続できなかった結果としてエラー戻りコードを処理し、異常終了するのではなく、エラー・メッセージを出して正常に戻る場合、ワークロード・マネージャーが、状況を正しく認識できず、その CICS 領域により多くの作業をルーティングする可能性があります。この状況を「ストーム・ドレーン作用」といいます。[ストーム・ドレーン作用の回避](#)を参照してください。

## ワークロード管理のアクティブ化

ワークロード管理をアクティブにするためには、ワークロード仕様を CICS システムに関連付け、動的ルーティング・プログラムまたは分散ルーティング・プログラムを使用するように CICS システムを更新する必要があります。

### 始める前に

ワークロードの動的ルーティングで使用する CICS システムを決定する必要があります。要求側領域は、要求が開始される場所 (例えば、端末専有領域) です。ルーティング領域は、トランザクションまたはプログラムをどこにルーティングするかを決定します。ターゲット領域は、トランザクションまたはプログラムが実行される場所 (例えば、アプリケーション専有領域) です。要求側領域、ルーティング領域、ターゲット領域で構成される単一のエンティティーの中で、作業をルーティングできます。

### このタスクについて

動的ルーティングについて詳しくは、[CICSplex SM を使用した動的ルーティング](#)を参照してください。

### 手順

- WLMSPEC ビューを使用して、ワークロード仕様を CICS システムに関連付けます ([ワークロード仕様への、CICS システムまたはシステム・グループの関連付け](#)を参照)。
- 要求側領域およびルーティング領域として機能する CICS システムに対して、動的ルーティング・プログラム EYU9XLOP を指定します。
  - CICS システムがアクティブでない場合は、次回 CICS システムが始動したときにワークロード管理処理を有効にすることができます。要求領域、およびルーティング領域として動作するすべてのターゲット領域で [DTRPGM](#) システム初期設定パラメーターを設定します。

```
DTRPGM=EYU9XLOP
```

非端末関連の **EXEC CICS START** コマンド、BTS アクティビティー、および論理サーバー内のすべての領域については、要求を開始する領域、および可能性のあるすべてのターゲット領域の [DSRTPGM](#) システム初期設定パラメーターを以下のように設定します。

```
DSRTPGM=EYU9XLOP
```

- CICS システムがアクティブな場合は、「**CICS 領域**」の詳細ビュー (CICS 領域 - CICS RGN を参照) を使用して、動的ルーティング・プログラムと分散ルーティング・プログラムの名前を設定できます。
3. CICS システムのワークロード管理をアクティブにします。
- このことは、以下のビューを使用して実行できます。
- 「**CICS システム定義**」ビューを使用して、データ・リポジトリ内の CICS システム定義を変更します。
  - 「**CICSplex 認知の MAS**」ビューを使用して、アクティブな CICS システムでワークロード管理を一時的にアクティブにします。

## タスクの結果

CICSplex に関連付けられた最初のルーティング領域が開始したとき、または「**CICSplex 認知の MAS**」詳細ビューを使用して、CICSplex の最初のルーティング領域のワークロード管理をアクティブ化したとき、適切なワークロード仕様が自動的にインストールされます。CICSplex の管理に関与するすべての CMAS に対しても通知されます。ワークロード・グループによって仕様に関連付けられたすべてのワークロード定義とトランザクション・グループも、自動的にインストールされます。CICS システムがアクティブである限り、追加のワークロード定義を手動で CICS システムにインストールできます。

注：CICSplex に関連付けられ、DTRPGM=EYU9XLOP か DSRTPGM=EYU9XLOP のいずれかを使用して定義された要求領域が開始すると、その領域はそのターゲット CMAS に接続し、ワークロード管理がアクティブ化されます。ただし、ルーティング領域が開始するときにターゲット CMAS がアクティブになっていない場合、そのルーティング領域から開始され、EYU9XLOP プログラムによって処理されるすべてのトランザクションは、CMAS がアクティブになって要求領域が完全に接続するまで、無期限に待機します。要求領域が CMAS に接続し、ワークロード管理がアクティブ化された後、CMAS が非アクティブになり、ワークロード管理がアクティブなままになることがあります。

ワークロードがアクティブになっているとき、ワークロード定義およびトランザクション・グループに対する後続の変更は、データ・リポジトリに記録されます。これらの変更をアクティブなワークロードに組み込むには、インストールまたは破棄のためのアクション・コマンドを使用する必要があります。

インストールされたワークロード定義は即時にアクティブになり、ワークロードがアクティブである間、または破棄されるまで、アクティブに維持されます。定義がワークロードにインストールされたこと、およびそのワークロードを使用する CICSplex の管理に関与するすべての CMAS がその定義について認識していることを確認するには、「**アクティブ・ワークロード定義**」ビューを使用します。

ワークロードがどれだけの時間アクティブに維持されるかは、以下のように、ワークロードに関連付けられた親和性存続時間によって異なります。

- PERMANENT の親和性存続時間が有効である場合、ワークロードは、そのワークロードの管理に関与するいずれかの CMAS がアクティブである限り、アクティブに維持されます。
- ワークロードにその他の親和性存続時間が関連付けられている場合、または親和性存続時間が関連付けられていない場合、そのワークロードに関連付けられた CICS システムが、その CICS システムが属する CICSplex を管理する CMAS に接続されている限り、ワークロードはアクティブなままです。

## 次のタスク

ワークロード管理がアクティブになっているとき、アクティブなワークロードがある間に非アクティブ化しようとししないでください。ワークロード内の作業を CICSplex SM がルーティングまたは分離しているときにワークロード管理を非アクティブにしようすると、予測できない結果になる恐れがあります。アクティブなワークロードに親和性の関係が関連付けられているときにこのアクションを試行すると、受け入れられない結果が発生します。詳しくは、[ワークロードからのアクティブ・トランザクションの破棄](#)を参照してください。

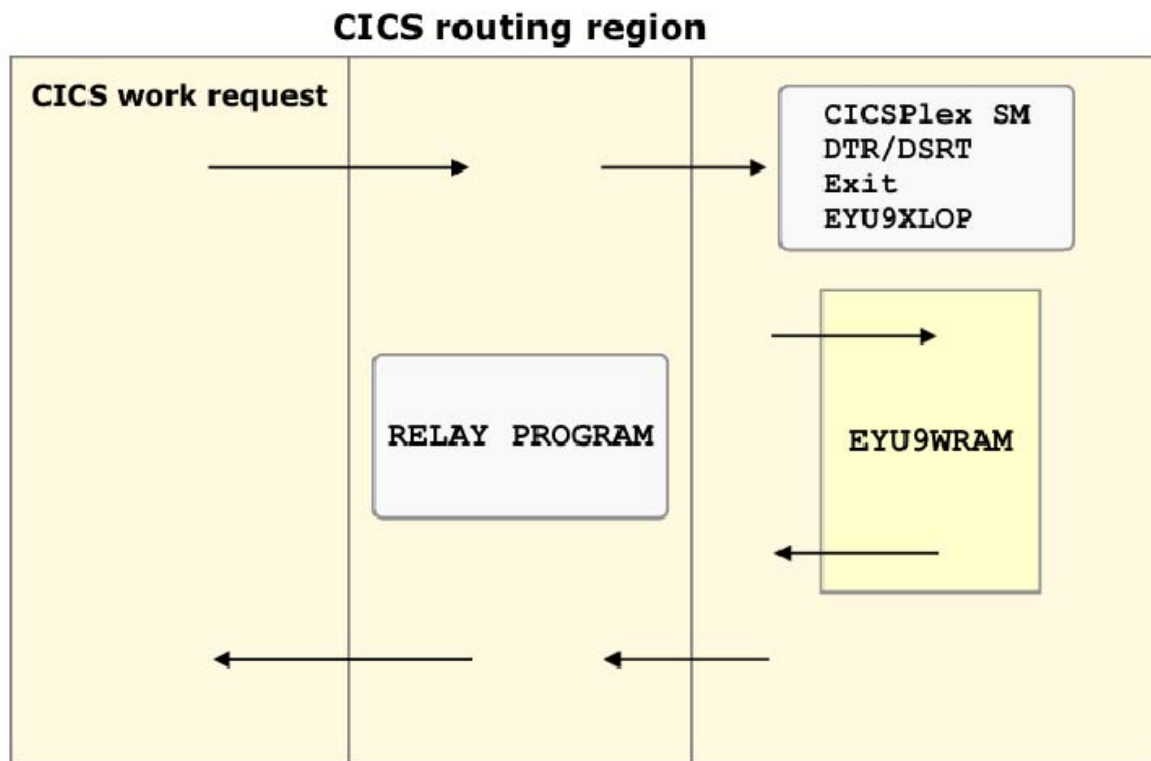


## 第3章 動的ルーティングの構成

CICS にトランザクションをローカルまたはリモートとして定義できます。ローカル・トランザクションは常に要求領域で実行されます。一方、リモート・トランザクションはルーティング領域に接続されたいずれかの CICS システムに経路指定される可能性があります。リモート・トランザクションのルーティングは、動的、静的、または ATI 制御のいずれかにすることができます。

### CICSplex SM を使用した動的ルーティング

リモート・トランザクションが開始されると、CICS 中継プログラムが呼び出されます。CICS 中継プログラムは、動的ルーティングプログラム EYU9XLOP にリンクします。EYU9XLOP は、CICSplex SM ベースの動的ルーティングに必要な環境を作成し、CICSplex SM ランタイム環境をセットアップします。



ルーティング領域の場合、システム初期設定テーブル (SIT) パラメーターで EYU9XLOP を指定します。

#### DTRPGM

静的ルーティング、ATI、および動的ルーティングでは、トランザクションを開始する要求側領域に関連付けられている DTRPGM システム初期設定パラメーター に EXU9XLOP を設定します。ルーティング領域として機能するすべてのターゲット領域も、SIT で DTRPGM を指定する必要があり、ワークロード仕様への、CICS システムまたはシステム・グループの関連付けで説明されているように、ルーティング領域としてセットアップされている必要があります。

#### DSRTPGM

非端末関連の **EXEC CICS START** コマンド、BTS アクティビティ、またはエンタープライズ Bean の場合、以下の DSRTPGM システム初期設定パラメーター に EYU9XLOP を設定します。

- 要求を開始する要求側領域。
- (DFHDYPDS コピーブックによってマップされる) ルーティング・プログラムの連絡域またはコンテナの DYROPTER フィールドを Y に設定する予定がある場合は、ターゲット領域になる 可能性のあるすべての領域。

CICSplex SM を使用して非端末関連 **EXEC CICS START** コマンドをルーティングする場合、DYROPTER フィールドは常に Y に設定されます。したがって、DSRTPGM=EYU9XLOP を常に指定して、ターゲット領域がワークロードの一部になるようにする必要があります。ターゲット領域をワークロードに対して定義しなかった場合は、ワークロードを待っている開始済みトランザクションが使用可能になり、トランザクションはハングします。

注：ターゲット領域をルーティング領域としてセットアップしない場合は、DTRPGM および DSRTPGM の SIT パラメーターで EYU9XLOP を指定しないでください。このパラメーターを指定すると、決して到着しないワークロードを待つ EYU9XLOP で、ルーティングされたトランザクションのエンドレス・ループが発生します。

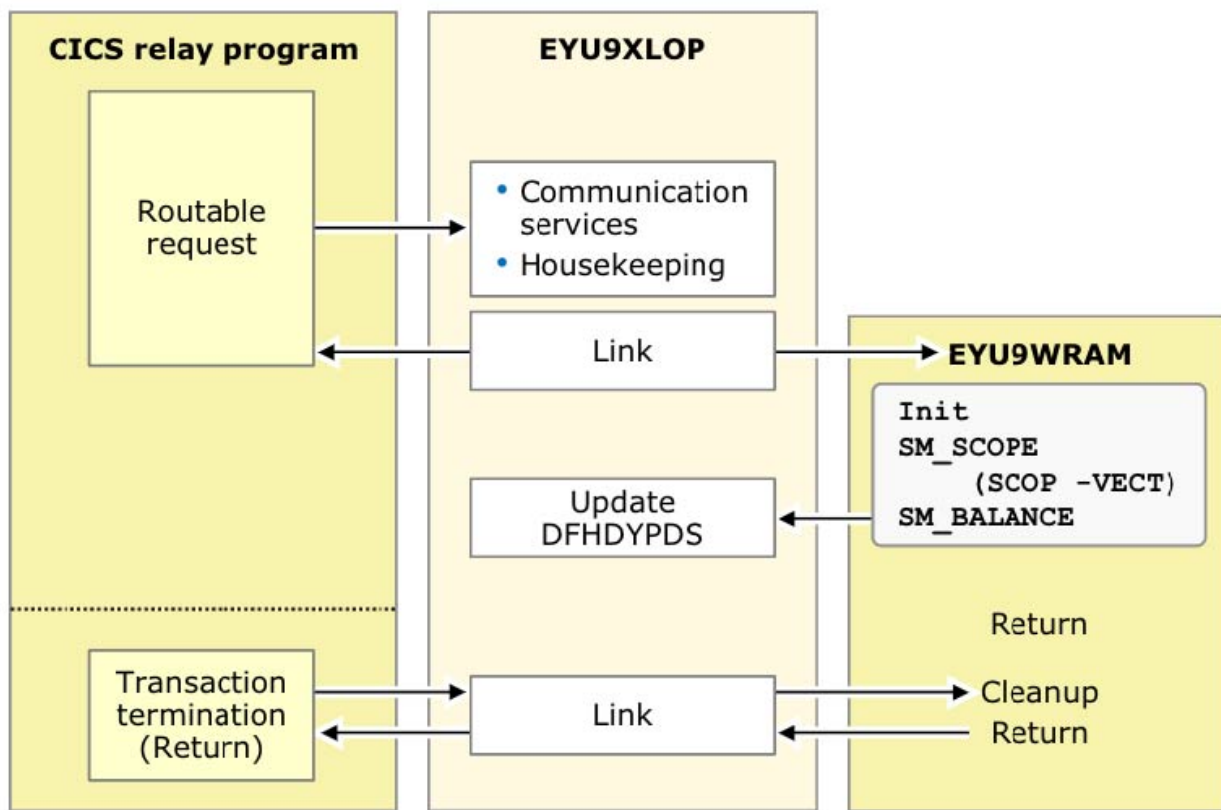
CICS は、すべてのルーティング要求を EYU9XLOP に通知します。これらの要求は、以下のとおりです。

- ルート選択、ルート選択エラー、およびトランザクション終了。
- BTS (サポートされるすべてのリリースの CICS) では、トランザクション開始、トランザクション異常終了、およびルーティング試行完了。
- 非端末 **EXEC CICS START** 要求 (サポートされるすべてのリリースの CICS) では、トランザクション開始、トランザクション異常終了、およびルーティング試行完了。

CICS は、EYU9XLOP にリンクすると、CICS 連絡域 DFHDYPDS を EYU9XLOP に渡します。

その後、制御は CICSplex SM ワークロード管理機能に渡ります。CICSplex SM はワークロード管理 MAS エージェント・コードを初期設定して、そのルーティング・アクション・プロセスを稼働させます。このプロセスは、以下のいずれかの条件が真である場合は、内部で呼び出されます。

- EYU9WRAM ユーザー置換可能モジュールが CICS に対して定義されていない
- EYU9WRAM が、CICS に対して定義されているが使用可能でない
- EYU9WRAM が CICS に対して定義済みで、使用可能であり、ロード・モジュールが CICSplex SM で配布されたとおりのモジュールのアセンブリ言語バージョンである



内部ルーティング・アクション・プロセスは、CICSplex SM で配布されたとおりの EYU9WRAM のアセンブリ言語バージョンを実行したのと同じ結果を生成します。エンタープライズで追加の動的ルーティング制御が必要でない場合は、この内部プロセスの方が、より高いパフォーマンスを提供します。

EYU9WRAM モジュールまたはそれと同等の内部プロセスは、CICSplex SM ベースの連絡域 EYURWCOM を受け取ります。インストール時に、CICS システムごとに、EYU9WRAM のエントリーが CICS システム定義 ファイル DFHCSD に追加されます。CICSplex SM で配布される EYU9WRAM は、CICSplex SM ワークロード管理処理を行います。EYU9WRAM は、これを行うために、最初にトランザクション・グループと、端末 ID、LU 名、ユーザー ID、またはプロセス・タイプに基づいて、該当するターゲット領域候補のリストを取得します。次に、EYU9WRAM は候補のリストからターゲット領域を選択します。

ほとんどの場合、提供される EYU9WRAM モジュールにはワークロード管理機能が用意されています。ただし、必要な場合には、EYU9WRAM を変更して CICSplex SM ワークロード管理の処理をカスタマイズすることもできます。詳しくは、[EYU9WRAM のユーザー置換モジュールの作成](#)を参照してください。

## 動的ルーティング用の CICS リリース要件

ここでは、動的ルーティングに関与する領域の CICS リリース要件を示します。

表 3. 動的ルーティングに関与する領域のリリース要件			
リリース要件の適用対象	ルーティング領域	ターゲット領域	要求側領域
動的トランザクション・ルーティング	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース	-
EXEC CICS START コマンド	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース
CICS ビジネス・トランザクション・サービス (BTS)	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース
分散プログラム・リンク (DPL)	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース	-
エンタープライズ Bean	CICS Transaction Server for z/OS、バージョン 4 リリース 2 以前	CICS Transaction Server for z/OS、バージョン 4 リリース 2 以前	-
Link3270 ブリッジ	サポートされている任意の CICS リリース	サポートされている任意の CICS リリース	-

## サンプルのソース・プログラムとコピーブック

インストールの直後に、モジュール EYU9WRAM が CICSplex SM にロードされます。

これはアセンブラ言語のコマンド・レベル・プログラムです。対応するコピーブックは、以下のとおりです。

### EYURWCOM

連絡域を定義します。

### EYURWCOD

EYURWCOM のリテラルを定義します。

### EYURWSVE

ターゲット領域スコープ・リストの各エレメントを定義します。

### EYURWSVD

EYURWSVE のリテラルを定義します。

カスタマイズ作業を支援するために、アセンブラー、C、COBOL、および PL/I のサンプルのソース・プログラムとコピーブックが CICSplex SM と共に配布されます。コピーブックは、言語固有のライブラリーに入っています。すべてのサンプル・プログラムは、SEYUSAMP ライブラリーに入っています。サンプル・プログラムとコピーブック、およびそれらを見つけることができる CICSplex SM ライブラリーの名前は、[38 ページの表 4](#) にリストされています。

表 4. 表 1. サンプル・プログラムとコピーブック

言語	メンバー名	別名	ライブラリー
アセンブラー:			
プログラム	EYUAWRAM	EYU9WRAM	SEYUSAMP
コピーブック 1	EYUAWCOM	EYURWCOM	SEYUMAC
コピーブック 2	EYUAWCOD	EYURWCOD	SEYUMAC
コピーブック 3	EYUAWSVE	EYURWSVE	SEYUMAC
コピーブック 4	EYUAWSVD	EYURWSVD	SEYUMAC
COBOL:			
プログラム	EYULWRAM	EYU9WRAM	SEYUSAMP
コピーブック 1	EYULWCOM	EYURWCOM	SEYUCOB
コピーブック 2	EYULWCOD	EYURWCOD	SEYUCOB
コピーブック 3	EYULWSVE	EYURWSVE	SEYUCOB
コピーブック 4	EYULWSVD	EYURWSVD	SEYUCOB
PL/I:			
プログラム	EYUPWRAM	EYU9WRAM	SEYUSAMP
コピーブック 1	EYUPWCOM	EYURWCOM	SEYUPL1
コピーブック 2	EYUPWCOD	EYURWCOD	SEYUPL1
コピーブック 3	EYUPWSVE	EYURWSVE	SEYUPL1
コピーブック 4	EYUPWSVD	EYURWSVD	SEYUPL1
C:			
プログラム	EYUCWRAM	EYU9WRAM	SEYUSAMP
コピーブック 1	EYUCWCOM	EYURWCOM	SEYUC370
コピーブック 2	EYUCWCOD	EYURWCOD	SEYUC370
コピーブック 3	EYUCWSVE	EYURWSVE	SEYUC370
コピーブック 4	EYUCWSVD	EYURWSVD	SEYUC370

## 最適化動的ワークロード・ルーティングの実施

CICSplex SM は、CICS に動的ワークロード管理を提供します。CICSplex SM は、領域状況 (RS) サーバーを使用して CICS から直接ポストされる現在の状況に関する情報を使用して、ワークロード管理の決定を拡張することができます。

### 始める前に

完全なワークロード最適化は、シスプレックスの最適化ワークロード・ルーティングを使用して行うことができます。ただし、この機能を使用するには、すべてのワークロード領域が CICS TS for z/OS バージョン 4.1 以降にマイグレーションされている必要があります。

シスプレックス最適化ワークロード・ルーティングは、すべてのワークロード領域が CICS TS 4.1 以降で実行されていて、かつ領域状況 (RS) サーバーが CICSplex のワークロードの各領域と同じ z/OS イメージ内で始動すると、自動的に実行されます。

ワークロード内の CICS TS for z/OS バージョン 4.1 以降の領域と、CICS TS 4.1 より前のレベルの領域を混用できます。ただし、ワークロードは非最適化状態で実行されます。

シスプレックス最適化ワークロード・ルーティングについて詳しくは、[シスプレックスの最適化ワークロード・ルーティング概要](#)を参照してください。

## このタスクについて

シスプレックス内のワークロード・ルーティングを最適化するには、カップリング・ファシリティ・データ・テーブルの一環として、領域状況 (RS) サーバーを構成し、モニターする必要があります。説明は、[領域状況サーバーのセットアップと稼働および Security for coupling facility data tables](#) を参照してください。

## 手順

CICSplex 定義で、領域状況 (RS) サーバーのカップリング・ファシリティ (CF) チューニング・パラメーターを設定できます。これにより、シスプレックスの最適化ワークロード・ルーティングが提供されます。**EYUSTARTCPLEXDEF** ビュー・セットを使用して、CICSplex を定義する操作や変更する操作を実行できます。カップリング・ファシリティ (CF) チューニング・パラメーターは、**CPLEXDEF** 詳細ビューにあります。

## ワークロード・ルーティングのための構成

以下の構成を設定することができます。

- シスプレックスの最適化ワークロード・ルーティングのためにリフレッシュを要求する前に、領域状況データがルーティング領域によってキャッシュに入れられる長さ。  
[領域状況サーバー、読み取り間隔パラメーター \(READRS\)](#)を参照してください。
- シスプレックスの最適化ワークロード・ルーティングのために、カップリング・ファシリティをタスク・スループット・データで更新する頻度。  
[領域状況サーバー、更新頻度パラメーター \(UPDATERS\)](#)を参照してください。
- 低スループット期間中にカップリング・ファシリティ (CF) に対してワークロード分散更新をトリガーするタスク・ロード範囲。  
[領域状況サーバー、最下位層パラメーター \(BOTRSUPD\)](#)を参照してください。
- 高スループットの期間中に領域が MAXTASK (MXT) 設定に近づいたときに、ワークロードを制御するタスク・ロード最上位層。  
[領域状況サーバー、最上位層パラメーター \(TOPRSUPD\)](#)を参照してください。

## 最適化の状況の表示

- ワークロードの場合、CICSplex SM Web ユーザー・インターフェース (WUI) の「アクティブ・ワークロード」リスト・ビューを表示します。
- CICS** 領域の場合、「アクティブ・ワークロード」メニューにある「ルーティング領域」または「ターゲット領域」ビューを表示します。

詳しくは、[42 ページの『最適化の状況』](#)を参照してください。

## 領域状況サーバー、読み取り間隔パラメーター (READRS)

CPLEXDEF、CSYSDEF、および MAS ビューで領域状況サーバー読み取り間隔パラメーターを使用して、ルーティング領域がどれほどの期間に渡り領域状況データをキャッシュに入れてから、シスプレックスの最適化ワークロード・ルーティングのためのリフレッシュを要求するかを制御します。

## READRS={200|number}

最適化モードで実行されているワークロードの場合、**READRS** 値は、CICS 領域状況 (RS) サーバーからのターゲット領域状況のリフレッシュと次のリフレッシュの間の最小間隔をミリ秒で指定します。こうしたリフレッシュ要求は、動的ルーティング要求に関してターゲット領域を評価しているルーティング領域によって発行されます。

**注：****READRS** 値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。

この値の範囲は 0 から 2000 です。

- 値 0 は、ターゲット領域の状況を調べるたびに、ターゲット領域の状況更新をルーティング領域が要求することを意味します。
- 値 1 から 2000 の場合、ターゲット領域の状況をリフレッシュできるようになる前に経過する必要がある最小時間間隔を示します。

間隔値が小さければ、状況更新のために RS サーバーがポーリングされる回数がそれだけ多くなります。QUEUE モードのワークロードの場合、この値が小さいと、ワークロード・ターゲットの有効範囲内にある複数の CICS 領域にタスク・ロードが平滑化されて広がることになります。この場合、他の正常性やリンクに関する要因はすべて等しいと想定されます。ただし、RS サーバーの使用率はそれ相応に増えるので、z/OS カップリング・ファシリティの使用率が高くなる可能性があります。

**READRS** パラメーターのデフォルト値は、200 ミリ秒です。

EYUSTARTCPLEXDEF ビュー・セットで指定する値により、CICSplex レベルの読み取り間隔が設定されます。ただし、この間隔を CICS 定義レベルでオーバーライドすると、個別の CICS ターゲット領域ごとに値を適切に調整できます。

CICSplex 定義 (CPLEXDEF) レベルで UPDATERS パラメーターと READRS パラメーターの両方を指定して、CICSplex 内のすべての領域のデフォルト値を設定します。こうした値は、CICS システム定義 (CSYSDEF) レベルまたは MAS エージェント・ランタイム (MAS) レベルでオーバーライドできます。

CICS 定義レベルの場合、**UPDATERS** と **READRS** の値は INHERIT なので、CICS システム定義からこれらの値は継承されます。こうした値を数値に明示的に変更すると、CICS システムはこれ以降再始動するたびに、指定された値を採用します。実行中の CICS 領域に変更をインプリメントするには、MAS 基本テーブルまたはビュー・セットを使用してこれらの値を適用する必要があります。ターゲット領域を再始動すると、CSYSDEF の指定を再び採用します。

#### 領域状況サーバー、更新頻度パラメーター (UPDATERS)

CPLEXDEF、CSYSDEF、および MAS ビューで領域状況サーバー更新頻度パラメーターを使用して、シスプレックスの最適化ワークロード・ルーティングのためにタスク・スループット・データでカップリング・ファシリティをどれほどの頻度で更新するかを制御します。

#### UPDATERS={15|number}

最適化モードで実行されているワークロードの場合、**UPDATERS** 値は、ターゲット CICS 領域内のタスク・ロードの値を変更するために CICS 領域状況 (RS) サーバーが呼び出される頻度を示します。この値は、現在の CICSplex 定義におけるすべてのターゲット CICS 領域のデフォルト頻度値になります。

**注：** **UPDATERS** 値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。

この値の範囲は 0 から 25 です。

- 値 0 は、タスク・ロード・カウントの変更が RS サーバーには全く通知されないことを意味します。この CICSplex 内のターゲット領域の最適化ワークロード機能が使用不可であるためです。

**注：** 値 0 は、ワークロード・ルーティング機能の最適化を抑止するわけではありません。値 0 によって抑止されるのは、ターゲット領域における RS ドメインのブロードキャスト・プロセスのみです。

- 1 から 25 までの値は、ターゲットの MAXTASKS 設定に対する算術パーセンテージとして適用されます。その結果のタスク・カウント値は、RS サーバーに対して更新呼び出しを駆動する数値しきい値として使用されます。

値 0 を指定すると、ターゲット領域の最適化ルーティング機能は事実上使用不可になります。最適化ルーティングを実行する場合、ルーターとターゲットの両方が最適化モードでなければなりません。

例えば、MAXTASKS 設定が 120 で、UPDATERS を 20 に設定する場合、ターゲット領域のタスク・カウントが以下のタスク数の間で変化すると、WLM ロード・カウントを更新するために RS サーバーが呼び出されます。

- 23、24 タスク (120 の 20%)
- 47、48 タスク (120 の 40%)
- 71、72 タスク (120 の 60%)



- 95、96 タスク (120 の 80%)
- 119、120 タスク (120 の 100%)

こうした境界をまたいでターゲット領域のタスク・ロードが増加または減少すると、RS サーバーが更新されます。

**UPDATERS** パラメーターを小さな値に設定すると、タスク・ロード範囲内で、RS サーバーに対する更新頻度が増加します。QUEUE モードのワークロードの場合、この値が小さいと、ワークロードの有効範囲内にある複数のターゲット CICS 領域にタスク・ロードが平滑化されて広がることになります。この場合、他の正常性やリンクに関する要因はすべて等しいと想定されます。ただし、RS サーバーの使用率はそれ相応に増えるので、z/OS カップリング・ファシリティの使用率が高くなる可能性があります。

**UPDATERS** パラメーターのデフォルト値は 15 です。

EYUSTARTCPLEXDEF ビュー・セットで指定する値により、CICSplex レベルの更新間隔が設定されます。ただし、この間隔を CICS 定義レベルでオーバーライドすると、個別の CICS ターゲット領域ごとに値を適切に調整できます。

CICSplex 定義 (CPLEXDEF) レベルで **UPDATERS** パラメーターと **READRS** パラメーターの両方を指定して、CICSplex 内のすべてのターゲット領域のデフォルト値を設定します。こうした値は、CICS システム定義 (CSYSDEF) レベルまたは MAS エージェント・ランタイム (MAS) レベルでオーバーライドできます。

CICS 定義レベルの場合、**UPDATERS** と **READRS** の値は INHERIT なので、CICS システム定義からこれらの値は継承されます。こうした値を数値に明示的に変更すると、CICS システムはこれ以降再始動するたびに、指定された値を採用します。実行中の CICS 領域に変更をインプリメントするには、MAS 基本テーブルまたはビュー・セットを使用してこれらの値を適用する必要があります。ターゲット領域を再始動すると、CSYSDEF の指定を使用します。

#### 領域状況サーバー、最下位層パラメーター (BOTRSUPD)

CPLEXDEF、CSYSDEF、および MAS のビューで領域状況サーバー最下位層チューニング・パラメーターを使用して、低スループット期間中のカップリング・ファシリティ (CF) に対するワークロード分散更新を制御します。

##### **BOTRSUPD={1|number}**

シスプレックスの最適化されたワークロードの場合、**BOTRSUPD** 値はタスク・ロード比率から実際のタスク・カウントに変換されます。そのカウントは、最下位層タスク・ロードをゼロからこの値までの範囲で定義するために使用されます。

**注 :** **BOTRSUPD** 値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。

値の範囲は 1 から 25 までとなります。領域のタスク・ロードがこの範囲内に入る場合は、タスク・ロードの変更ごとに、タスク・ロードがカップリング・ファシリティ (CF) にブロードキャストされます。ロードがこの値に達すると、RS サーバー更新頻度のタスク規則が活動化されます。

**BOTRSUPD** パラメーターのデフォルト値は 1% です。

この値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。**BOTRSUPD** パラメーターは、低スループットの期間中にワークロードをより円滑に分散できるようにします。CICSplex が多数のターゲット領域から構成されている場合、この値を大きくしすぎると、更新要求で CF が過負荷になるリスクを冒すことになり、その結果、WLM および一般的な z/OS サブシステムのパフォーマンスが低下することがあります。

**BOTRSUPD** 値を変更する必要がある場合は、必ず、カップリング・ファシリティ機能と WLM スループット機能のパフォーマンスを少なくとも変更後数日間はモニターしてください。

#### 領域状況サーバー、最上位層パラメーター (TOPRSUPD)

CPLEXDEF、CSYSDEF、および MAS のビューで、領域状況サーバー最上位層チューニング・パラメーターを使用します。このパラメーターは、ターゲット領域のワークロード管理 (WLM) 最大タスク・ヘルス・インディケーターがオフに切り替えられている場合、高スループットの期間中に、その領域が MAXTASK (MXT) 設定に近づいたときに、カップリング・ファシリティ (CF) に対する更新を制限するためにワークロードを制御します。

## TOPRSUPD={5|number}

シスプレックスの最適化されたワークロードの場合、**TOPRSUPD** 値はタスク・ロード比率から実際のタスク・カウントに変換されます。そのカウントは、領域の **MAXTASKS** 値から減算されて、最上位層のタスク・ロード範囲を決定します。

**注：** **TOPRSUPD** 値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。

値の範囲は 1 から 25 までです。この値は、演算比率として領域の **MAXTASKS** 設定に適用されます。その後、タスク・カウント値は領域の **MAXTASKS** 設定から減算され、タスク・ロード最上位層が確立されます。領域内のタスク・ロードがその **MAXTASKS** 限度まで上がった場合、領域の **MAXTASKS** 状態がオフに切り替えられてカップリング・ファシリティにブロードキャストされる前に、タスク・ロードがこの値を下回る必要があります。

**TOPRSUPD** パラメーターのデフォルト値は 5% です。

**TOPRSUPD** 値を変更する場合は、変更がワークロードおよびカップリング・ファシリティのスループットに与える影響を考慮した上で行ってください。この値を大きくしすぎると、ワークロードの上位ロード範囲でワークロードのバッチ処理の影響が見られます。この値を小さくしすぎると、最上位層のバッチ処理の影響は減少しますが、カップリング・ファシリティに対する更新が大幅に増える可能性があります。

**TOPRSUPD** 値を変更する必要がある場合は、必ず、カップリング・ファシリティおよび WLM スループット機能のパフォーマンスを少なくとも変更後数日間はモニターしてください。

## 最適化の状況

CICSplex SM Web ユーザー・インターフェース (WUI) のアクティブ・ワークロード・リスト・ビューを使用して、ワークロードの状況を表示できます。

## ワークロードの最適化状況

CICSplex SM Web ユーザー・インターフェース (WUI) のアクティブ・ワークロード・リスト・ビューを使用して、CICSplex 内でアクティブな各ワークロードの状況を表示できます。

**最適化状況**属性は、現在のワークロードのシスプレックス最適化状況を報告します。値は、すべてのワークロード・ルーター最適化状況と、すべてのワークロード・ターゲット最適化状況を組み合わせたものです。

以下の値があります。

- **ACTIVE:** ワークロードのすべてのターゲットとルーターは最適化されたワークロード状態で実行されています。
- **PARTIAL:** 少なくとも 1 つのターゲットと 1 つルーターが最適化されたワークロード・モードで実行されています。「**アクティブなルーティング領域**」と「**アクティブなターゲット領域**」のハイパーリンクを使用して、最適化された状態で実行されない領域を判別します。
- **INACTIVE:** ワークロードは、以下の 1 つ以上の理由で最適化された状態で実行されていません。
  - 最適化された状態で実行されているルーティング領域がワークロードにない。
  - 最適化された状態で実行されているターゲット領域がワークロードにない。
  - 最適化された状態で実行されている領域がワークロードにない。
  - このワークロードについて、CICSplex 定義またはすべてのターゲット CICS システム定義で RS サーバーの更新頻度として 0 の値が指定され、ワークロードが非最適化として指定された。

## CICS 領域の最適化状況

「アクティブ・ワークロード」メニューにある「ルーティング領域」ビューまたは「ターゲット領域」ビューを使用して、領域の最適化の状況を表示できます。

**最適化状況**属性は、最適化されたワークロード・ルーティング機能の現在のターゲット領域の状況を報告します。

以下の値があります。

- **ACTIVE:** このターゲット領域は、最適化されたワークロード状態で実行されています。
- **INACTIVE:** この領域は、最適化されたワークロード状態で実行できます。しかし、以下の 1 つ以上の理由で現在は最適化されていません。
  - 領域に RS サーバーへの接続がない。
  - 領域は RS サーバーに接続されている。しかし、サーバーが z/OS カップリング・ファシリティに接続できない。
  - 領域に対する最適化の使用可能化の設定が無効に設定されている。最適化ルーティング機能を活動化するには、前もって領域の使用可能化設定を有効に設定しておく必要があります。

**注:** 「CICS システム定義」ビューで最適化の使用可能化設定をリセットし、次の領域の開始に備えて値を変更することができます。使用可能化設定は、MAS ビューを使用してアクティブ領域で変更することもできますが、領域が再始動されると、CICS システム定義内の設定に戻されます。

  - この領域の RS サーバーの更新頻度値が 0 である。これは、この領域の最適化機能が、ルーティング・ターゲットとして機能している場合は使用可能でないことを意味します。

**注:** *UPDATERS* 値を 0 に設定できるのは、動的ルーターとして使用されている領域の場合だけです。この値を 0 に設定すると、領域は、領域状況 (RS) サーバーへの不要な領域状況ブロードキャストをしなくなります。
- **N/A:** ターゲット領域は、領域状況の記録をサポートしている CICS リリース上にありません。最適化されていない WLM ルーティング決定しか行えません。

### ワークロード・ルーティングの最適化

CICS TS for z/OS バージョン 4.1 より前のリリースでは、非最適化ワークロード管理の場合で、カップリング・ファシリティが使用できないと、ワークロード・ルーティングは CICSplex SM ワークロード・マネージャーによって管理されます。このワークロード・マネージャーは CMAS が所有するデータ・スペースを使用して、領域間のロードと状況データを共有します。

各 CMAS は、その直接の管理対象であるすべてのユーザー CICS 領域 (MAS) と共有している、単一の WLM データ・スペースを管理します。CMAS は初期化時に、管理対象である CICS 領域に関連付けられているすべてのワークロードに必要な構造を使用して、データ・スペースの検証とフォーマット設定を行います。ユーザー CICS 領域が動的トラフィックのルーティングを開始すると、それらの CICS 領域の状態が、このデータ・スペースに記録されます。ユーザー CICS 領域内の CICSplex SM エージェントは、15 秒ごとに、その時点でのタスク・カウントを判別して、その領域の所有 CMAS に報告します。CMAS は、WLM データ・スペースのターゲット領域記述子内のロード・カウントを更新し、その値を、そのユーザー CICS 領域に関連付けられているワークロードに参加している他の CMAS にブロードキャストします。

すべてのユーザー CICS 領域が同じ CMAS によって管理されている環境では、すべてのルーティング領域とターゲット領域は WLM データ・スペース内の同じ物理構造を参照します。動的ルーティングの決定は、ルーティング・ターゲット領域候補の最新のロード・データに基づいています。ルーティングの決定は、要因の組み合わせに基づいています。詳しくは、[CICSplex SM WLM が処理のルーティング先を選択する方法](#)を参照してください。

ワークロードは複数の z/OS イメージ間に分散されるため、異種の LPAR 上のユーザー CICS 領域を管理するために、追加の CMAS が構成されます。各 WLM データ・スペースは、各 CMAS が責任を負う CICS 領域だけでなく、ワークロード内のあらゆる CICS 領域を記述するために、完全な構造セットを保守する必要があります。各 CMAS が所有する WLM データ・スペースは、ワークロードに参加している他の CMAS が所有する WLM データ・スペースと定期的に同期を取る必要があります。この同期化は、MAS からその CMAS へ 15 秒ごとに発生し、その後、ワークロード内の他のすべての CMAS に波及します。

CICSplex SM の動作は、DTRPGM 要求と DSRTPGM 要求で以下のように異なります。

- **DTRPGM 要求の場合、**ルーティング領域は CICS から呼び出しを行い、各ターゲット領域が選択されたターゲットでの要求の実行と同期しているかどうかを判断します。その後、動的要求の完了時に CICS からのコールバックが行われます。この呼び出しにより、ルーターは CICS にターゲット領域システム ID を通知する前にタスク・ロード・カウントを増加することができ、要求の完了時にカウントを減少させることもできます。
- **DSRTPGM 要求の場合、**ルーティング領域は CICS から呼び出しを行い、各ターゲットが選択されたターゲットと同期していないかどうかを判断します。一般に、これらの動的要求は、非同期の CICS の始動です。ルーターには、ルーティングされたトランザクションがいつ開始され、終了したかの通知がありません。

ん。通知がないために、CICSplex SM は DSRTPGM ターゲット領域がそれに関連付けられているワークロードも持つ必要があることを要求し、ターゲットは論理ルーティング領域に変換されます。この呼び出しを使用して、CICSplex SM ルーティング・プロセスは、DSRTPGM ターゲットで呼び出されていることを判別でき、したがって、トランザクションの開始時と終了時にタスク・ロード・カウントを調整することができます。

CICSplex SM ルーティング領域は CICSplex 内の動的トランザクション・スループットをカウントするため、ターゲット領域上でローカルに開始されたトランザクションは、ハートビートが発生するまでルーティング領域によって説明できないままになるという追加の問題が発生します。ルーター・トランザクション・カウントは、2 回のハートビートが (最初はカウントを増加させるために、2 回目はカウントを再び減少させるために) 発生するまで、正しく同期化されません。しかし、この不一致はルーターとターゲットが異なる CMAS によって管理されている場合ほど重大であるとは見なされません。

複数の CMAS の場合、ルーター領域はターゲット領域の状況データを、ローカル WLM データ・スペース内に記述されているとおりに評価します。そのターゲット領域が、ルーターの所有するものと異なる CMAS によって管理されている場合、そのターゲット領域を記述している状況データは、最大 15 秒古いことがあります。DTRPGM 要求の場合、この遅延による重大な影響はありません。しかし、DSRTPGM 要求では、特にワークロード・スループットが増加する場合、大きな影響が生じる可能性があります。この作用は、ワークロードのバッチ処理と呼ばれます。

詳しくは、[44 ページの『ワークロードのバッチ処理』](#)を参照してください。

### ワークロードのバッチ処理

ワークロードのバッチ処理は、動的分散 (DSRTPGM) ルーティング要求が処理されている複数 CMAS 環境の、使用頻度が高いワークロードで発生します。

ターゲット領域とそのルーティング領域を、異なる CMAS によって管理することができます。これは、ルーターがターゲットと異なる LPAR で実行されているときは、常に当てはまります。このシナリオでは、ルーターはターゲット自体が採用している実の記述子構造と異なる記述子構造を使用して、ターゲットの状況を評価します。

ルーターによって検討されるターゲット記述子は、CICSplex SM ハートビートによる 15 秒のインターバルで実際の記述子と同期化されます。ルーターから見たターゲット領域のタスク・カウントは、それと同時にリフレッシュされます。15 秒のハートビート・インターバル中に、ルーターはワークロード内の他の潜在的ターゲット領域と比較して、ターゲットが比較的にビジー状態である (または比較的に静止状態である) と見なします。したがって、その 15 秒の期間中、ルーターはターゲットがどれだけビジーに見えるかに応じて、作業をそのターゲットに向けたり、そのターゲットから方向転換したりする作業を続行します。ターゲット内の現行タスク・カウントの変更が、次のハートビートまで認識されないためです。このように、以前の状況に基づいたルーティングが行われるため、ターゲットの使用の度合いが過大または過小になる可能性があります。ルーターは、この状況に反応するために、ターゲットに向けての、またはターゲットから方向転換するルーティングをより活発に行うので、バッチ化サイクルが続行します。この状態はワークロード・スループット内のアクティビティーが減少するまで続き、減少するとバッチ化サイクルは再びスループット・アクティビティーが増加するまで沈静化します。

CICSplex 全体のタスク・ロードをモニターしていると、一部の領域がその MAXTASKS 限度で稼働していて、動的にルーティングされたトラフィックを継続的に与えられている一方、他の領域が未使用のままになっているのが見られます。15 秒後に取られたスナップショットでは、使用状況の逆転が示されている場合もあります。ビジー領域がアイドル状態になっていたり、アイドル領域が現在では MAXTASKS 限度に達していたりします。

領域状況サーバーを始動してワークロードを最適化すると、ワークロードのバッチ処理による影響が除去されます。

## 動的ルーティングの変更

モジュール EYU9WRAM を変更することにより、CICSplex SM ワークロード管理処理をカスタマイズできます。

EYU9WRAM は、CICSplex SM 動的ルーティング・アプリケーション・プログラミング・インターフェース (API) を使用します。この API は、ワークロード管理アクションを要求するために必要なメカニズムを提供する、特殊目的の CALL レベルのインターフェースです。すべての呼び出しが、標準 CALL ステートメン

トを使用して構成されます。CALL ステートメントは、EYU9WRAM モジュールと CICSplex SM ワークロード・マネージャー・コンポーネントとの間のリンケージを生成します。CALL ステートメントの形式は次のとおりです。

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

ここで、

#### **DA\_TOKEN**

EYURWCOM 連絡域を経由して介して提供される動的ルーティング API トークンを指定します。このトークンは EYU9WAPI によって使用されるので、変更してはいけません。

#### **function**

実行する機能です。以下のように指定します。

##### **SM\_SCOPE**

適格ターゲット領域のリストを返します。

##### **SM\_BALANCE**

適格ターゲット領域のリストからターゲット領域を選択します。

##### **SM\_ROUTE**

特定のターゲット領域にトランザクションをルーティングします。

##### **SM\_CREAFF**

親和性を作成します。

##### **SM\_DELAFF**

アクティブな親和性を削除します。

SM\_SCOPE と SM\_BALANCE を続けて使用すると、ターゲット領域のリストを取得してから、トランザクションのルーティング先のターゲット領域を選択できます。

常に特定ターゲット領域にトランザクションが送信されることが分かっている場合は、SM\_ROUTE を使用するだけで済みます。

配布されている EYU9WRAM は、SM\_SCOPE 呼び出しと SM\_BALANCE 呼び出しを発行します。

SM\_ROUTE 呼び出しは含まれません。実行されない SM\_CREAFF 呼び出しと SM\_DELAFF 呼び出しが含まれます。これらの呼び出しを発行するようにプログラムを変更することができます。

すべての例でアセンブラ言語形式の動的ルーティング API verb が使用されていますが、これらの verb は C、COBOL、および PL/I で作成されたプログラムでも使用できます。CICSplex SM と共に配布されるサンプル・プログラムには、これらの言語のそれぞれについて CALL ステートメントの例が含まれています。

**注：**動的ルーティングを変更する場合は、以下の点に注意してください。

- アプリケーション・コンテキストの値が設定されていないと、それらの値はゼロ (バージョンの場合) または NULL/低値(プラットフォーム、アプリケーション、オペレーション) のいずれかになります。
- スtring・フィールドの埋め込みには、スペースではなく NULL が使用されます。

**重要：**配布されている EYU9WRAM は、ワークロード管理定義を通じて設定されたワークロードのルーティングと分離を、API verb を使用して処理します。そのため、EYU9WRAM を変更すると、CICSplex SM ワークロード管理機能に悪影響を与える可能性があります。例えば、SM\_SCOPE、SM\_BALANCE、SM\_ROUTE のいずれも使用しない場合、各トランザクション・オカレンスは、そのトランザクションが CICS に対して定義されたときに識別されたデフォルト・ターゲット領域にルーティングされます。その結果、すべての CICSplex SM ワークロード管理定義がバイパスされます。

#### **非動的トランザクションに関する考慮事項**

トランザクションが静的にルーティングされるか、ATI によって開始される場合は、API を介した呼び出しを組み込む必要はありません。そのような場合、ターゲット領域を変更することはできません。ルーティング機能はルート通知です。ルート通知機能は、そのようなトランザクションがルーティングされることを EYU9WRAM プログラムに通知することを意図したものです。

これらの機能のいずれかを組み込んだ場合、以下のことが起きます。



- SM\_SCOPE 呼び出しは、SCOP\_VECT に単一のエントリーを返します。そのエントリーは、定義時に静的定義トランザクションに関連付けられたターゲット領域であるか、EXEC CICS START で指定された宛先に関連付けられているターゲット領域です。ATI トランザクション用のコマンド
- SM\_BALANCE 呼び出しは、CICS に対する定義時に、そのトランザクションに関連付けられたターゲット領域を選択します。CICSplex SM ワークロード・マネージャーは、トランザクションの親和性を一切作成しません。
- SM\_ROUTE 呼び出しは、例外応答を返します。

EYU9WRAM が呼び出される場合:

- 通知。既存の親和性関係は、どれも無視され、何も作成されません。
- ルート選択エラー。既存の親和性関係は、どれも無視され、何も作成されません。さらに、配布された EYU9WRAM プログラムは、端末メッセージを書き込んで終了します。
- ルーティング試行完了 (CICS BTS トランザクションの場合のみ)。既存の親和性は、どれも無視されます。動的ルーティング API 関数呼び出しは、どれも誤った応答で終わります。[46 ページの『非端末関連の START および CICS BTS に関する考慮事項』](#)を参照してください。
- トランザクション開始 (CICS BTS トランザクションの場合のみ)。既存の親和性は、どれも無視されます。動的ルーティング API 関数呼び出しは、どれも誤った応答で終わります。[46 ページの『非端末関連の START および CICS BTS に関する考慮事項』](#)を参照してください。
- トランザクション終了。既存の親和性は、どれも無視されます。動的ルーティング API 関数呼び出しは、どれもエラー応答で終わります。
- トランザクション異常終了。既存の親和性は、どれも無視されます。動的ルーティング API 関数呼び出しは、どれもエラー応答で終わります。

#### 非端末関連の START および CICS BTS に関する考慮事項

CICS BTS アクティビティーまたは非端末関連 EXEC CICS START コマンドのいずれかに関連付けられているトランザクションをルーティングすると、静的ルーティングと動的ルーティングのどちらの場合も、DSRTPGM SIT パラメーターで識別されたルーティング・プログラムが呼び出されます。

静的ルーティングされるトランザクションの場合、EYU9WRAM プログラムはターゲット領域を変更できません。[45 ページの『非動的トランザクションに関する考慮事項』](#)を参照してください。動的ルーティングされるトランザクションの場合、EYU9WRAM プログラムはターゲット領域を変更することができます。[46 ページの『一連のターゲット領域からのターゲット領域の選択』](#)を参照してください。ただし、どちらの場合も、EYU9WRAM プログラムは以下に対してのみ、呼び出されます。

- 要求側領域内:
  - 通知
  - ルート選択
  - ルート選択エラー
  - ルーティング試行完了
- ターゲット領域内:
  - トランザクション開始
  - トランザクション終了
  - トランザクション異常終了

ルート開始の機能について詳しくは、[DPL 要求の動的ルーティング](#)を参照してください。

#### 一連のターゲット領域からのターゲット領域の選択

配布されている EYU9WRAM は、SM\_SCOPE および SM\_BALANCE を使用して、トランザクション用に定義されているターゲット領域のリストからターゲット領域を選択します。ユーザーが設定したオプションは、ターゲット領域の選択方法に影響します。

- SM\_SCOPE は、現行のトランザクション・オカレンス用のターゲット領域リストを返します。これらのターゲット領域は、ワークロード仕様によって要求側の領域に明示的または暗黙的に関連付けられているターゲット領域です。



- SM\_BALANCE は、SM\_SCOPE が返すターゲット領域候補リストから、ターゲット領域を選択します。

SM\_SCOPE または SM\_BALANCE の呼び出しの後、操作の結果が EYURWCOM 連絡域の応答フィールドおよび理由フィールドに示されます。

## ルート選択時のアクション

トランザクションまたはプログラムのルーティングがスケジュールされると、動的ルート選択が発生します。

- SM\_SCOPE が発行されると、EYURWCOM 連絡域のスコープ・ベクトルのポインターは、ターゲット領域スコープ・リストの記述域 SCOP\_VECT のアドレスで更新されます。また、エレメント・カウント・フィールドが更新され、スコープ・ベクトル内のエレメントのカウントが格納されます。スコープ・ベクトル内の各エレメントは、ターゲット領域候補を識別し、その現在の状況を示します。

SM\_BALANCE の処理時には、スコープ・ベクトル・エレメント EYURWSVE に無視のマークが付いている場合を除き、番号付きリストの最初にあるターゲット領域がルーティング用に選択されます。

- 現行のトランザクション・オカレンスに関連付けられている親和性がない場合、SM\_SCOPE 呼び出しは SCOP\_VECT エレメントをソートして、最適のターゲット領域候補がリストの最初に配置されるようにします。

SM\_BALANCE の処理時には、最適のターゲット領域の APPLID と SYSID が、WCOM\_SEL\_AOR と WCOM\_SEL\_SYSID のそれぞれのフィールドに配置されます。一般に、それはスコープ・ベクトルに表されている最初のターゲット領域になります。ただし、EYU9WRAM プログラムによって一部の SCOP\_VECT エレメントの WSVE-IGNORE フィールドに、その領域を無視する必要があることを示すマークが付けられている場合は、無視するようにマーク付けされていないスコープ・ベクトル内の最初のターゲット領域が選択されます。

注：

1. トランザクション ID およびそのトランザクションに関連付けられている端末とユーザーの ID が、親和性が定義されているトランザクション・グループに一致する場合、SM\_BALANCE の処理時に親和性がアクティブになります。
  2. DPL を使用して、SM\_SCOPE 呼び出しの前にトランザクション ID を更新できます。
- 現行のトランザクション・オカレンスに親和性が関連付けられている場合、通常の SM\_SCOPE 呼び出しでは、親和性が存在するターゲット領域のみが SCOP\_VECT 内に示されます。その場合、SM\_BALANCE 呼び出しはそのターゲット領域を選択します。そのターゲット領域をルーティングに使用できない場合、SM\_BALANCE 機能は警告標識を設定します。別のターゲット領域の選択を試みることはしません。
  - トランザクションが DTRTRAN として定義されている場合、EYURWCOM 連絡域には、そのトランザクションが DTRTRAN であり、ワークロード・マネージャーによって拒否されないことを示す標識が含まれます。処理は、一般に、通常の非 DTRTRAN の場合と同じです。SM\_SCOPE を発行すると、SCOP\_VECT が返されます。SM\_BALANCE を発行すると、ターゲット領域が選択されます。

EYU9WRAM モジュールは、SM\_SCOPE 呼び出しを発行する前に、トランザクションを拒否することを選択する場合があります。その場合、ワークロード・マネージャーは CICS に戻り、DFHDYPDS 連絡域内の拒否標識を Y に設定します。サンプルの EYU9WRAM プログラムでは、トランザクション ID がすべてブランクである場合にのみ、DTRTRAN トランザクションが拒否されます。

## 通知時のアクション

通知は、静的トランザクションまたは ATI トランザクションのルーティングがスケジュールされているときに発生します。

- 静的にルーティング可能なトランザクション、ATI トランザクション、および BTS 静的ルーティング要求の場合、SM\_SCOPE を発行すると、スコープ・ベクトルにターゲット領域が 1 つだけ入ります。それは、そのトランザクションに関連付けられているターゲット領域です。

## ルーティング試行完了時のアクション

ありません。ただし、このステージでリソースを整理し、解放してもかまいません。

## ルート選択エラー時のアクション

ルート選択エラー呼び出しは、要求側領域とターゲット領域との間の CICS リンクが使用できない場合や、定義されていない場合に発生します。

- アクティブな親和性がない場合: SM\_SCOPE を再発行します。返される SCOP\_VECT によって示されるターゲット領域候補には、エラーの原因となったターゲット領域は含まれません。その場合、SM\_BALANCE を使用して新規のターゲット領域を選択できます。
- 前のルート選択の結果として親和性がアクティブになっている場合:
  - その親和性に関連付けられている存続時間が PERMANENT、SYSTEM、ACTIVITY、または PROCESS である場合、エラーの原因となっているターゲット領域は SM\_SCOPE が返すスコープ・リストに組み込まれます。SM\_BALANCE は、親和性の規則により、そのターゲット領域を選択する必要があります。また、EYU9WRAM に警告を返します。その場合、EYU9WRAM はエラーが発生したことをユーザーに通知する必要があります。
  - 親和性に関連付けられている存続時間が SIGNON、LOGON、DELIMIT、または PCONV である場合、アクティブな親和性状況は、EYU9WRAM に制御が戻される前に除去されます。SM\_SCOPE を再発行すると、エラーの原因となっているターゲット領域はスコープ・リストに組み込まれません。ワークロード仕様が、そのトランザクション・グループに対して Create Affinity YES で定義されている場合、EYU9WRAM には、親和性が定義されていることと、SM\_BALANCE を発行するとそれが活動化されることが通知されます。
- 前のトランザクション・インスタンスによって親和性が活動化されている場合、SM\_SCOPE を再発行すると、返されるスコープ・リストには前に選択されたターゲット領域が含まれています。SM\_BALANCE は、親和性の規則により、そのターゲット領域を選択する必要がありますので、EYU9WRAM に警告が返されます。その場合、EYU9WRAM はエラーが発生したことをユーザーに通知する必要があります。

## トランザクション終了時のアクション

トランザクション終了は、トランザクションが正常に終了したときに発生します。

- EYU9WRAM は、獲得している可能性があるすべてのリソースを解放する必要があります。
- SM\_SCOPE、または SM\_BALANCE、または SM\_ROUTE を発行すると、例外応答が返されます。

## トランザクション異常終了時のアクション

トランザクション異常終了は、トランザクションが異常な状態で終了したときに発生します。

- EYU9WRAM は、獲得している可能性があるすべてのリソースを解放する必要があります。
- SM\_SCOPE、または SM\_BALANCE、または SM\_ROUTE を発行すると、例外応答が返されます。

## トランザクション開始時のアクション

トランザクション開始は、CICS BTS 関連トランザクションまたはエンタープライズ Bean 関連トランザクションが、ターゲット領域ヘルペティングされたときに発生します。実行する特定のアクションはありません。この呼び出しは、通知の目的でのみ発行されます。

## 特定のターゲット領域の選択

SM\_ROUTE は、特定のターゲット領域をルーティング用に選択することを要求します。SM\_ROUTE は、明示的なルーティングを必要とするアプリケーションやデータがあるときに使用します。例えば、特定のユーザー ID に関連付けられているトランザクションを、特定のターゲット領域にルーティングすることができます。

SM\_ROUTE 呼び出しの後、操作の結果が EYURWCOM 連絡域の応答フィールドおよび理由フィールドに格納されます。

サンプルの SM\_ROUTE 呼び出しを示します。

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

DA\_TOKEN は、EYURWCOM 連絡域を介して提供される動的ルーティング API トークンを識別します。このトークンは EYU9WAPI によって使用されるので、変更してはいけません。

## ルート選択時のアクション

ルート選択時のアクションは、以下のとおりです。

- 要求側領域とターゲット領域との間に 1 つだけ接続が存在する場合は、そのターゲット領域の SYSID または APPLID のいずれかを指定できます (CICSplex SM は、該当する適切な ID を判別します)。APPLID は、EYURWCOM 連絡域の WCOM\_SEL\_AOR フィールドに入れてください。SYSID は、WCOM\_SEL\_SYSID フィールドに入れてください。

要求側領域とターゲット領域との間に複数の接続が存在する場合は、前に説明したように SYSID と APPLID の両方を指定して、正しいターゲット領域が確実に選択されるようにしてください。SYSID と APPLID の両方を指定した場合、それらの妥当性の検査は行われないうちに注意してください。

ターゲット領域が CICSplex SM に対して定義されている必要はありません。親和性状況は検査されません。このため、この呼び出しの結果として親和性が確立されることはなく、有効だった親和性も無視されます。

- ルート通知処理中に SM\_ROUTE を発行すると、例外応答が返されます。

## ルーティング試行完了時のアクション

ありません。ただし、このステージでリソースを整理し、解放してもかまいません。

## ルート選択エラー時のアクション

EYU9WRAM プログラムがメッセージを発行して終了する場合があります。その場合は、別のターゲット領域を指定して SM\_ROUTE を再発行するか、SM\_SCOPE と SM\_BALANCE を発行することができます。

## トランザクション終了時のアクション

- EYU9WRAM は、獲得している可能性があるすべてのリソースを終了する必要があります。
- いずれかの API 関数を呼び出すと、例外応答が返されます。

## トランザクション異常終了時のアクション

- いずれかの API 関数を呼び出すと、例外応答が返されます。

## トランザクション開始時のアクション

トランザクション開始時には、以下のとおりです。

- EYU9WRAM は、獲得している可能性があるすべてのリソースを終了する必要があります。
- いずれかの API 関数を呼び出すと、例外応答が返されます。

## 親和性の作成

トランザクション用に確立されたトランザクション・グループ内にまだ親和性が存在しない場合は、SM\_CREAFF を使用して親和性を作成できます。

トランザクション・グループは、親和性タイプと存続時間を使用して定義する必要があります。作成された親和性は、トランザクション・グループで定義されているのと同じ親和性タイプと存続時間を持ちます。

SM\_CREAFF を使用する前に、50 ページの『[親和性に関する考慮事項](#)』を検討してください。

SM\_CREAFF を呼び出す前に、まず SM\_SCOPE を呼び出して、スコープ・リストを取得する必要があります。SM\_CREAFF は、スコープ・リスト内にないターゲット領域に対しては親和性を作成しません。また、EYUWRWCOM 連絡域の WCOM\_SEL\_AOR フィールドと WCOM\_SEL\_SYSID フィールドに、親和性を作成するターゲット領域の APPLID と SYSID をそれぞれ設定してください。

EYU9WRAM は実行されないコードのフラグメントを含んでおり、これを、SM\_CREAFF 機能を実装するためのテンプレートとして使用できます。

以下のときに SM\_CREAFF 機能呼び出しはできません。

- ルート終了

- ルート異常終了
- ルート通知
- ルート開始
- ルート完了

### 親和性の削除

SM\_DELAFF を使用して、アクティブな親和性を削除することができます。

SM\_DELAFF を使用する前に、[50 ページの『親和性に関する考慮事項』](#)を検討してください。

SM\_DELAFF を呼び出す前に、まず SM\_SCOPE を呼び出して、スコープ・リストを取得する必要があります。EYURWCOM 連絡域の WCOM\_AFF\_STAT フィールドには、親和性がアクティブであるかコミット済みであることを示す値が入っています。アクティブな親和性は、SM\_DELAFF を使用して削除できます。コミット済みの親和性には SYSTEM または PERMANENT の存続時間があり、SM\_DELAFF を使用しても削除できません。

EYU9WRAM は実行されないコードのフラグメントを含んでおり、これを、SM\_DELAFF API 関数を実装するためのテンプレートとして使用できます。

以下のときに SM\_DELAFF 機能呼び出すことはできません。

- ルート通知
- ルート開始
- ルート完了

### 親和性に関する考慮事項

親和性を定義するときは、その存続時間を指定する必要があります。通常、親和性は指定された存続時間が満了するまで存続します。

以下に示す、いずれかのタイプの存続時間を定義できます。

#### アクティビティー

CICS BTS アクティビティーが終了した時点で満了します。

#### 限界指定型

トランザクションの PCONV モードが END になった時点で満了します。

#### ログオン型

端末ユーザーがログオフした時点で満了します。

#### Pconv

トランザクションが NEXTTRANSID を指定しない EXEC CICS RETURN を使用したか、トランザクションの PCONV モードが END になった時点で満了します。

CICS は、APPC (LUTYPE6.2) 装置の疑似会話をサポートしません。

#### 永続型

そのターゲット領域を含んでいるワークロードが終了した時点で満了します。

#### 処理

CICS BTS プロセスが終了した時点で満了します。

#### サインオン

端末ユーザーがサインオフした時点で満了します。

#### システム

ターゲット領域が終了した時点で満了します。

#### UOW

そのトランザクションに関連付けられている作業単位が終了した時点で満了します。作業単位は、CICS SYNCPOINT または ROLLBACK 要求が実行されたとき、または親タスクが終了したときのいずれかに終了します。

親和性ターゲット領域が 1 つだけ含まれているスコープ・リストを SM\_SCOPE が生成するが、そのターゲット領域がルーティングに使用可能でないという環境も存在します。これは、以下の場合に発生します。

- ターゲット領域がダウンしている。
- ターゲット領域への CICS リンクがダウンしている。
- ターゲット領域は現時点ではアクティブだが、かつてシャットダウンされ、親和性が作成された後に再始動された。

上記の場合、EYU9WRAM のデフォルト処理では、親和性ターゲット領域が使用可能でないことを示す端末メッセージが発行され、トランザクションは終了します。親和性の存続時間が PCONV (疑似会話) の場合、CICSplex SM は自動的にその親和性を削除します。その親和性は満了しているからです (NEXTTRANSID が存在しない)。しかし、デフォルトの EYU9WRAM 処理では、それ以外のどの親和性も、ターゲット領域に対する実際の親和性の特性が不明であるため、削除されません。例えば、LOGON 親和性が、ターゲット領域に情報を渡すための TCTUA の使用に関与している場合があります。その親和性が、ターゲット領域が使用可能でないときに削除されると、そのトランザクション・グループについての次のトランザクションによって、新規のターゲット領域が選択されます。そのトランザクションは、新しいターゲット領域へルーティングされたとき、TCTUA の内容を使用すると障害を起こす可能性があります。

デフォルトの処理を実装するサンプルの EYU9WRAM プログラムには、SM\_SCOPE を呼び出した後に親和性の状況を検査するサブルーチンが含まれています。ターゲット領域に対してアクティブである親和性がコミットされておらず、親和性ターゲット領域状況が OK でない場合、サブルーチンは EYU9WRAM プログラムが終了できるよう、端末ユーザーにメッセージを送信してから終了します。このサブルーチンには、実行されないコード・フラグメントが含まれており、これを使用すると、SM\_DELAFF 呼び出しを使用して親和性を削除することができます。実行されないコード・フラグメントの前に、アクティブな親和性の親和性存続時間についての一連のテストがあります。1 つ以上の分岐を変更して、そのコード・フラグメントにジャンプすることができます。そのコード・フラグメント自体は、親和性を削除し、メッセージを発行した後に戻りを行います。これによって EYU9WRAM プログラムは終了することができ、その結果、トランザクションは終了します。この処理を変更して、親和性が削除され、メッセージが発行されず、サブルーチンが終了するようにし、その結果としてメインライン・プロセスに SM\_SCOPE 呼び出しを再実行させることができます。その場合、SM\_BALANCE 用に新規のターゲット領域セットが受信されます。

同じサブルーチンに、使用可能にすると親和性を作成することができる、実行されないコード・フラグメントも含まれています。この例では、サブルーチンは、親和性が定義されているがアクティブでないことを判別しました。次に、サブルーチンは WCOM\_AFF\_AUTO 標識を検査して、CICSplex SM で SM\_BALANCE 中に自動的に親和性を作成する必要があるかどうかを判別します。サブルーチンは、その答えが何であろうと、正常に終了します。ユーザーは、親和性が作成されるようにするために、SM\_CREAFF フラグメントを使用可能にすることができます。SM\_CREAFF 呼び出しは、WCOM\_AFF\_AUTO が SM\_BALANCE 中の親和性の自動作成を指示しているかどうかに関係なく、使用することができます。したがって、SM\_CREAFF 呼び出しを使用して、以下のことができます。

- 親和性を、CICSplex SM が作成しないときに作成する。
- 通常なら CICSplex SM が選択しないターゲット領域に対して、親和性を作成する。

SM\_CREAFF コード・フラグメントは、SM\_SCOPE が返すスコープ・リストの最初にあるターゲット領域に対して親和性を作成します。

EYURWCOM 連絡域の以下のフィールドは、CICSplex SM の親和性処理に関する情報を提供します。

- WCOM\_AFF\_STAT
- WCOM\_AFFAOR\_STAT
- WCOM\_AFF\_TYPE
- WCOM\_AFF\_LIFE
- WCOM\_AFF\_AUTO
- WCOM\_WORK\_NAME
- WCOM\_TGRP\_NAME

## CICSplex SM データ域

EYU9WRAM プログラムは 2 つのデータ域を使用します。連絡域 EYURWCOM とスコープ・ベクトル・エレメント EYURWSVE です。

EYURWCOM のリテラルは EYURWCOD で定義されており、EYURWSVE のリテラルは EYURWSVD で定義されています。

これらのデータ域をマップするため使用できるコピーブックの名前は、[サンプル・プログラムとコピーブック](#)で示されています。

スコープ・ベクトル・エレメント EYURWSVE には、SM\_SCOPE が返したターゲット領域候補のリストに関連付けられている個々のターゲット領域に関する情報が入っています。

## EYU9WRAM のユーザー置換モジュールの作成

CICSTS56.CPSM.SEYUPROC で提供されているサンプル・プロシージャーを使用して、EYU9WRAM の置換モジュールを作成できます。

プロシージャー・メンバーは以下のとおりです。

アセンブラー	EYUEITAL
C	EYUEITDL
PL/I	EYUEITPL
COBOL	EYUEITVL

1. これらのプロシージャーを、カタログされたシステム・プロシージャー・ライブラリーにコピーします。[サンプル・ユーザー置換モジュール](#)にあるサンプルを使用できます。
2. JCL サンプルを使用してプロシージャーを実行し、EYU9WRAM の置換モジュールを作成します。サンプル JCL に示されている小文字の値を、ご使用のサイトの適切な値に置き換えます。また、EYU9WRAM 内に CICSplex SM API を使用することはできません。

## 追加動的ルーティング・サポートの要求

このセクションには、[プロダクト・センシティブ・プログラミング・インターフェース](#)の情報が記載されています。

アプリケーション・プログラムから、CICSplex SM ワークロード管理機能に直接アクセスできます。

**注：**CICS Transaction Server for OS/390® バージョン 1 リリース 3 以降を実行している場合は、すべての動的ルーティング要件に CICSplex SM 提供プログラムの EYU9XLOP を使用することをお勧めします。このセクションの情報が必要になるのは、この機能を CICS Transaction Server for OS/390 バージョン 1 リリース 3 へのマイグレーション中も引き続き使用する場合だけです。

使用するルーティング領域が CICS Transaction Server for OS/390 バージョン 1 リリース 3 以降である場合、または CICS BTS またはエンタープライズ Bean を使用している場合は、このセクションの情報は必要ありません。[CICSplex SM を使用した動的ルーティング](#)で説明されている分散ルーティング機能を使用してください。

そのようなプログラムを作成するには、以下について理解しておく必要があります。

- CICS 中継プログラムと定義済みの動的ルーティング・プログラムとの間のインターフェース。
- [ワークロード管理](#)で説明されている、CICSplex SM のワークロード管理処理。
- 35 ページの『[CICSplex SM を使用した動的ルーティング](#)』で説明されている、CICSplex SM ワークロード管理のルーティング・アクション・モジュール EYU9WRAM。

CICSplex SM ワークロード管理を呼び出す基本的な方法は、モジュール EYU9XLOP を CICS に対して動的ルーティング・プログラムとして定義することです。これにより、CICS 中継プログラムによって EYU9XLOP が呼び出されるようになり、CICSplex SM ワークロード管理は、別の CICS システムへ送信できる各作業要求のルーティングを決定できるようになります。



必要であれば、CICS 中継プログラムを介さずに、アプリケーション・プログラムから、CICSplex SM ワークロード管理を直接呼び出すことができます。CICS Transaction Server for OS/390 バージョン 1 リリース 3 より前の CICS リリースでは、CICSplex SM ワークロード管理を使用して、要求側領域からの分散プログラム・リンク (DPL) に最適のターゲット領域を判別できます。ユーザーが作成するアプリケーション・プログラムは、以下のガイドラインに従っている必要があります。

- プログラムは、ルーティング領域として定義済みであり実行中である CICS システム内で実行される必要があります。ターゲット領域から CICSplex SM ワークロード管理にアクセスすることはできません。
- プログラムは、同じタスクから、そのプログラムの INIT 呼び出しと TERM 呼び出しを発行する必要があります。CICSplex SM は、プログラムが CICS 中継プログラムによって実行される処理と同様な動作をすることを予期します (CICS 中継プログラムは、常にルート開始とルート終了で呼び出しを行います)。

**注:** EXEC CICS START コマンドを発行するプログラムを介したワークロード管理へのアクセスは、非同期トランザクション開始を使用する場合には推奨されません。さらに、端末またはユーザーのいずれかに関連していない EXEC CICS START コマンドを、親和性関係があるトランザクションに使用しないでください。このインターフェースではそれらの関係を正しく解決できないからです。

動的プログラム・ロードを使用するプログラムから、ワークロード管理にアクセスすることは、サポートされていません。CICSplex SM ワークロード管理は、CICS アプリケーション・プログラミング・インターフェースを使用して、そのワークロード管理を実行している現在の環境に関する情報を収集します。その情報の一部を DPL の制約事項のために入手できず、予測不能な結果が発生する場合があります。

、CICSplex SM ワークロード管理は、ルーティングの決定を行うために特定の情報を必要とし、通常、その情報は CICS 中継プログラムによって提供されます。CICS 中継プログラムは、中継されるトランザクションのインスタンスに関して、CICS が入手できるデータから情報を導出します。CICSplex SM ワークロード管理を直接呼び出すときは、その情報を、端末向けデータやルーティングされる作業を識別するためのトランザクション名に相当するデータも含め、ユーザーが提供する必要があります。

## CICSplex SM ワークロード管理の開始方法

CICSplex SM ワークロード管理機能を開始するには、EXEC CICS LINK コマンドを使用し、プログラムとして EYU9XLOP を指定し、連絡域として EYURWTRA を識別します。

コマンドの形式は次のとおりです。

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

CICS 中継プログラムから DFHDYPDS 連絡域の動的ルーティング・プログラムに通常渡されるデータの多くは、ユーザーのプログラムで生成して、CICSplex SM ワークロード管理に渡す必要があります。それ以外の、通常では CICSplex SM ワークロード管理によってトランザクション向け環境および端末向け環境から生成されるデータも、ユーザーのプログラムが提供する必要があります。アセンブラーのコピーブック EYURWTRA は、CICSplex SM ワークロード管理に渡される連絡域のマップを提供します。

対応するコピーブックと、それらを見つけることができる CICSplex SM ライブラリーは、以下のとおりです。

言語	メンバー名	ライブラリー
アセンブラー	EYUAWTRA	SEYUMAC
COBOL	EYULWTRA	SEYUCOB
PL/I	EYUPWTRA	SEYUPL1
C	EYUCWTRA	SEYUC370

CICSplex SM ワークロード管理を呼び出す前に、EYURWTRA 連絡域を初期設定する必要があります。

WTRA\_API\_RESP および WTRA\_API\_REAS で返される可能性がある、応答コードと理由コードの値のリストについては、アセンブラーのコピーブック・メンバー EYURWCOD を参照してください。

## 処理に関する考慮事項

CICSplex SM ワークロード管理の処理の後、WTRA\_RESPONSE フィールドにゼロ (0) または 8 のいずれかの戻りコードが入ります。

戻りコードが 0 の場合、WTRA\_SYSID に選択されたターゲット領域の SYSID が入っており、WTRA\_APPLID にその APPLID が入っています。この情報を使用して、ターゲット領域への作業単位のルーティングを続行できます。

戻りコードが 8 の場合は、WTRA\_API\_RESP フィールドと WTRA\_API\_REAS フィールドで詳細を調べる必要があります。これらのフィールドには、CICSplex SM ワークロード管理のルーティング・アクション・モジュール (EYU9WRAM) に返された最後の応答コードと理由コードが入っています。ほとんどの場合、応答コードと理由コードは CICSplex SM ワークロード管理が戻りコード 8 を生成した理由を記述しています。この情報に基づいて、続行する方法を決定できます。

ただし、場合によっては、WTRA\_RESPONSE フィールドに戻りコード 8 が入っている可能性があるのに、WTRA\_API\_RESP フィールドと WTRA\_API\_REAS フィールドが CICSplex SM ワークロード管理によって設定されていないことがあります。そのような場合は、ルーティング・アクション・プロセスを呼び出す前にエラーが検出されています。それらのタイプのエラーでは、CICS トレース・データ・セットに CICSplex SM ユーザー・トレース・レコードが書き込まれます。その場合は、そのプログラムの処理を終了してください。

戻りコードが 8 の場合は、WTRA\_OPTER フィールドを調べる必要もあります。その値が WTRA\_CALLYES の場合は、以下のようにします。

1. ルート終了のために、WTRA\_FUNC フィールドを WTRA\_FUNCTRM に設定します。
2. CICSplex SM ワークロード管理を再度呼び出します。
3. 提示された作業単位に関する処理を終了します。

WTRA\_OPTER フィールドに WTRA\_CALLNO の値が入っている場合は、提示された作業単位について、それ以上の処理を終了します。

その作業単位がターゲット領域内で完了した後、作業単位がどのように完了したかに応じて、WTRA\_FUNC フィールドを設定します。作業単位が正常に完了した場合は、以下のようにします。

1. ルート終了のために、WTRA\_FUNC フィールドを WTRA\_FUNCTRM に設定します。
2. オプションとして、WTRA\_NEXTTRAN フィールドに、そのプロセスで次に使用されるトランザクションの ID を設定します。次のトランザクションがない場合は、フィールドをすべてブランクに設定します。
3. CICSplex SM ワークロード管理を再度呼び出して、トランザクションの終了処理を要求します。

作業単位が異常終了するか、異常と考えられるエラーを生成した場合は、以下のようにします。

1. WTRA\_FUNC フィールドを WTRA\_FUNCABD に設定して、ルーティングされたトランザクションが異常終了したことを CICSplex SM ワークロード管理に通知します。
2. CICSplex SM ワークロード管理を再度呼び出して、トランザクションの異常終了処理を要求します。

### ルート・エラーに関する考慮事項

ターゲット領域の SYSID および APPLID を受け取った後、そのターゲット領域への作業単位のルーティングを試みることができます。

ただし、システムが使用不能の場合は、ルーティング試行がエラー状態で終わることもあります。その場合は、以下のようにしてください。

1. WTRA\_FUNC フィールドを WTRA\_FUNCERR に設定して、ルーティング・エラーを示します。
2. WTRA\_ERR フィールドでエラーの理由を指定します。
3. CICSplex SM ワークロード管理を再度呼び出します。

CICSplex SM ワークロード管理は、可能な場合、スコープ・リストから別のターゲットを提供します。他に使用可能なシステムがない場合は、WTRA\_RESPONSE フィールドが 8 に設定され、WTRA\_API\_RESP フィールドと WTRA\_API\_REAS フィールドに失敗の理由が記述されます。有効なターゲット領域を受け取

るまで、ルーティング・エラーのために CICSplex SM ワークロード管理を複数回呼び出さなければならない場合もあります。

#### トランザクション親和性に関する考慮事項

WTRA\_TRANID、WTRA\_USERID、および WTRA\_LUNAME の各フィールドによって、親和性を含んでいるトランザクション・グループが CICSplex SM ワークロード管理によって選択された場合、トランザクション親和性の規則が順守されます。

その場合は、ルート終了機能呼び出す前に必ず、WTRA\_NEXTTRAN フィールドを次のトランザクションの ID によって初期設定してください。

作業単位または作業単位のサブセットに親和性関係が定義されている場合は、その親和性の定義方法に注意してください。プログラムで SIGNOFF や LOGOFF の処理を提供できないため、SIGNON または LOGON の親和性存続時間によって、まだ有効であるのに親和性関係が構築され、要求側領域またはターゲット領域のいずれかが終了するまで除去されなくなる場合があります。

WTRA\_NEXTTRAN フィールドは、疑似会話親和性に最も役立ちます。この親和性存続時間の場合、CICSplex SM ワークロード管理は通常、EXEC CICS ASSIGN NEXTTRANSID コマンドを発行して、機能のための次のトランザクション ID があれば、それを取得します。このコマンドは、CICSplex SM ワークロード管理がユーザーのプログラムによって呼び出された場合は使用できないので、次のトランザクションの ID を提供するのにはユーザーの責任です。

CICSplex SM ワークロード管理の親和性処理については、[親和性の関係の考慮](#)を参照してください。親和性関係と存続時間の値の指定について詳しくは、[トランザクション・グループの作成](#)を参照してください。

#### 異常終了の補正に関する考慮事項

ターゲット領域が CICS TS を実行している場合、プログラムによって開始されたトランザクションは、異常終了の補正処理に参加できます。ただし、アクティブな CICSplex SM ワークロードがそれを指定している必要があります。

作業要求が非端末関連 EXEC CICS START コマンドまたは BTS アクティビティのいずれかである場合、ルーティング領域が異常終了の発生を検出するためには、ルーティング領域とターゲット領域が同じ MVS イメージ内になければならないことに注意してください。

プログラムはフィールドに WTRA\_FUNCABD を指定することにより、CICSplex SM ワークロード管理に特定の作業単位が異常終了したことを通知する必要があります。その後、CICSplex SM ワークロード管理は、可能な場合、同じターゲット領域を後続のルート選択機能でトランザクション (または、親和性が定義されている場合はトランザクションのセット) 用に選択しない傾向を持つようになります。

CICSplex SM ワークロード管理の異常終了補正処理については、[異常終了の確率およびワークロード管理](#)を参照してください。異常終了の補正の要求について詳しくは、[ワークロード仕様の作成](#)を参照してください。

#### CMAS の可用性に関する考慮事項

ユーザーのプログラムで、EYU9XLOP プログラムに対する EXEC CICS LINK を発行した場合、ユーザーのプログラムを実行している要求側領域がまだターゲット CMAS に完全に接続していないと、ESSS 接続が進行中であることを示すメッセージ EYUXL0020I が表示されます。

ユーザーのプログラムは CMAS が使用可能になるまで無限に待機し、要求側領域はワークロードに結合されます。CMAS と MAS への接続が正常に完了していても、CMAS が MAS に対するワークロード定義のインストールを完了していない場合は、要求側領域がワークロードを待っていることを示すメッセージが表示されます。この状態を修正するには、ワークロードのセットアップとインストールが必要になる場合があります。

**注：**要求側領域が CMAS への接続を正常に完了すれば、CMAS は非アクティブになることができ、ワークロード管理はアクティブのまま残ります。

## サンプルの呼び出しシーケンス

このコード例では、CICSplex SM ワークロード管理機能にアクセスするプログラムのサンプルの呼び出しシーケンスを示します。これはサンプル・プログラムとして使用されることを意図しているものではありません。したがって、すべての詳細について完全であるとは限りません。

```

*-----*
          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_ERRROUT Say out of service.
          B       ROUTE_SELECT      Go Get another system.

*-----*
* Invoke the WLM MAS Agent for Route Abnd.
*-----*
ROUTE_ABND   DS      0H
          MVI     WTRA_FUNC,WTRA_FUNCABD Set the Abnd 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

BAS    R5,LINK_WLM                Set the next TRANID.
CLC    WTRA_RESPONSE,=F'0'        Go terminate.
BE     EXIT_TERMOK                OK?
                                           ..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.

```





## 第 4 章 CICSplex SM を使用したワークロードの管理

CICSplex SM ワークロード管理 (WLM) 機能によって、パフォーマンスとワークロード・スループットを最適化するために処理要求を実行する場所を制御することができるようになります。WLM は、動的ルーティングによってこれを実現します。WLM は、CICSplex SM 動的ルーティング・プログラム EYU9XL0P を使用して、処理要求を、事前定義された一群のターゲット領域の中から最適なターゲット領域に経路指定します。

CICSplex SM ワークロード管理の概念と体系の概要については、[仕組み: CICSplex SM ワークロード管理 \(WLM\)](#)を参照してください。

### ワークロード・ルーティング

ワークロード・ルーティングとは、ターゲット領域の可用性、正常性、およびアクティビティー・レベルに従って、それらのターゲット領域グループ内でトランザクションまたはプログラムの方向付けをすることです。ワークロード分離に加えて、またはその代わりに、ワークロード・ルーティングを使用できます。

CICSplex SM は、各トランザクションの開始時に、パフォーマンスが最大になると考えられるターゲット領域を選択することにより、定義済みのターゲット領域グループ内でワークロードをルーティングすることができます。

ワークロード・ルーティングは統計に基づいて実施されます。CICSplex SM は、以下の 4 つのアルゴリズムの 1 つを使用して、作業を処理するターゲット領域を決定します。

#### キュー・アルゴリズム (QUEUE)

CICSplex SM は、要求領域で開始された処理要求を、ターゲット領域の指定セット内で最も適切なターゲット領域にルーティングします。

#### リンク・ニュートラル・キュー・アルゴリズム (LNQUEUE)

リンク・ニュートラル・キュー・アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、キュー・アルゴリズムと同等です。

#### ゴール・アルゴリズム (GOAL)

CICSplex SM は、z/OS ワークロード・マネージャーを使用して事前定義されたゴールに最適なターゲット領域に処理要求を送付します。

#### リンク・ニュートラル・ゴール・アルゴリズム (LNGOAL)

リンク・ニュートラル・ゴール・アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、ゴール・アルゴリズムと同等です。

ルーティングされるトランザクションについてトランザクション 親和性が顕著である場合、使用されるアルゴリズムとは無関係に、親和性ターゲット領域が選択されます。

ゴール・アルゴリズムとリンクに依存しないゴール・アルゴリズムでは、予期される応答時間を達成するターゲット領域の能力に基づいて、適切なターゲット領域が選択されます。キュー・アルゴリズムとリンクに依存しないキュー・アルゴリズムでは、ターゲット領域のセット全体にロードが分散されるように、適切なターゲット領域が選択されます。

ゴール・アルゴリズムまたはリンクに依存しないゴール・アルゴリズムが特定のターゲット領域を識別しない場合、残りのターゲット領域のセットにキュー・アルゴリズムまたはリンクに依存しないキュー・アルゴリズムがそれぞれ適用されます。

使用可能なすべてのターゲット領域の作業処理能力が同等な場合、ターゲット領域はそのターゲット領域グループからランダムに選択されます。そのため、ロードの軽いシステムでは、同等の能力を持つターゲット領域への作業割り振り順序は事前に決まっています。

キュー・アルゴリズムとゴール・アルゴリズムでは、ターゲット領域とそのルーティング領域間の接続のタイプが考慮されます。リンクは以下の順番の優先順位になるように、各タイプの接続に重み付け係数が割り振られます。

- ローカル

- MRO/IRC および MRO/XM (ローカル LPAR)
- MRO/XCF (リモート LPAR)
- IPIC (ローカル LPAR)
- IPIC (リモート LPAR)
- LU6.2
- 間接

CICSplex SM は、タスク・ロードに対する乗数としてのリンクの重み付け係数およびその他の係数を使用して、全体のルーティングの重みを決定します。計算が終了すると、重みの最も軽い領域がターゲット領域として選択されます。

例えば、他のすべての係数が等しいとすると、MRO/XCF を使用してその要求領域に接続されているターゲット領域が IPIC を使用して接続されているターゲット領域より優先されます。同じ LPAR 内のルーティング領域とターゲット領域間の IPIC 接続の重み付けは、異なる LPAR 内にあるターゲット領域への IPIC 接続の重み付けより低くなるため、他の係数が同じであれば、ローカル IPIC 接続はリモート IPIC 接続より優先されます。

### ワークロード・ルーティングの制御レベル

ワークロード・ルーティングを使用するには、ワークロード仕様 (WLMSPEC) レベルでそのワークロードのデフォルトのルーティング・アルゴリズムを指定する必要があります。オプションにより、トランザクション・グループ (TRANGRP) レベルでルーティング・アルゴリズムを指定することができます。トランザクション・グループ内で指定されたアルゴリズムは、ワークロード仕様と関係付けられたデフォルトのアルゴリズムを指定変更します。

デフォルトのルーティング・アルゴリズムは、ワークロード内でルーティングされる各動的トランザクションに適用されます。ただし、ルーティング・アルゴリズムが指定されているトランザクション・グループと関連付けられているトランザクションは除きます。下記のルーティング・アルゴリズムから 1 つを指定できます。

- QUEUE
- LNQUEUE
- GOAL
- LNGOAL

ワークロード仕様レベルで指定されたルーティング・アルゴリズムを変更するには、ワークロードが新しいアルゴリズム仕様で最新表示されるように、ワークロードに参加するすべての領域を終了する必要があります。

トランザクション・グループ・レベルでは、ルーティング・アルゴリズムを動的に指定できます。指定された動的ルーティング・アルゴリズムは、トランザクション・グループと関連付けられたルーティングされるすべての動的トランザクションに適用されます。したがって、同じワークロード内の特定のトランザクション・コードに代わりのルーティング・アルゴリズムを適用できます。

トランザクション・グループ・レベルで代わりのルーティング・アルゴリズムを指定する場合、ご使用のルーティング領域を停止させることなく特定のターゲット領域のワークロード・ルーティング特性を動的に変更できます。インストールされたトランザクション・グループを変更する場合、WLM 定義によって指定されたトランザクション・グループも最新表示されるように、トランザクション・グループに関連付けられた WLM 定義 (WLMDEF) を破棄してから、再インストールする必要があります。関連付けられた WLMDEF を破棄したり再インストールしたりせずに、ルーティング・アルゴリズムを直ちに変更するには、「アクティブなワークロード・トランザクション・グループ (WLMATGRP)」ビューおよび **SET** コマンドを使用して、ALGTYPE 属性を変更することができます。

下記のルーティング・アルゴリズムから 1 つを指定できます。

- INHERIT
- QUEUE
- LNQUEUE
- GOAL

## • LINGOAL

INHERIT は、トランザクション・グループがそのワークロードに対してワークロード仕様と関連付けられたルーティング・アルゴリズムを使用することを意味します。

### リンクに依存しないワークロード・ルーティング

状況によっては、リンク重み付けはルーティング動作に重大な影響を及ぼす可能性があり、単一サイトのパラレル・シスプレックス環境でシステム境界を越えた CICS トランザクション・ルーティングを妨げる可能性があります。リンクの重み付けなしにワークロード・ルーティングを使用するには、リンクに依存しないゴール・アルゴリズムとリンクに依存しないキュー・アルゴリズムを使用できます。これらのアルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、それぞれゴール・アルゴリズムおよびキュー・アルゴリズムと同等です。

リンクに依存しないアルゴリズムは、動的トランザクション (例えば、MVS サブシステムからのサービスが必要とする可能性のある動的トランザクション) をルーティングするのに役立ちます。キュー・アルゴリズムとゴール・アルゴリズムでは、ルーターは、システム上で最も速いリンク (恐らく、同じ LPAR 上にある) を持つ動的トラフィックに焦点を合わせています。この動作は、ローカル MVS イメージにあるサブシステムに過負荷を加え、ワークロードに参加するリモート MVS イメージを十分に活用しない可能性があります。これらのトランザクションを、リンクに依存しないアルゴリズムを使用するトランザクション・グループに割り当てて場合、ルーティングされる動的トラフィックはローカル LPAR とリモート LPAR の間でさらに均等にルーティングされ、それらのサブシステムへの負荷は分散されます。

ただし、リンクに依存しないアルゴリズムでは、他の係数が等しい場合、すべてのターゲット領域が等しいプリファレンスを持つことを忘れないでください。最も遅い通信リンクに接続された最もリモートにあるターゲット領域は、ローカルに接続された MRO 領域や、ルーティング領域がルーティング・ターゲット・スコープの一部である場合にはルーティング領域自体と同じプリファレンスを持つ可能性があります。したがって、ワークロード仕様 (WLMSPEC) レベルでリンクに依存しないアルゴリズムを指定するかどうかは慎重に考慮してください。それが動的にルーティングされるすべてのトランザクションに影響を及ぼす可能性があるからです。その結果、ワークロード・マネージャーは、動的ルーティング・トラフィックに最善のターゲット領域を選択しない可能性があり、全体のワークロード・スループットが悪化する可能性があります。

特定のトランザクション集合に、リンクに依存しないアルゴリズムを必要とする場合は、それらのトランザクションを識別するワークロード管理トランザクション・グループ (TRANGRP) にそのアルゴリズムを割り当ててください。

## キュー・アルゴリズム

CICSplex SM がキュー・アルゴリズムを使用するとき、ルーティングされるトランザクションにアクティブな親和性がない場合、ロードはターゲット領域の集合全体に分散されます。選択基準には、現行のタスク・ロード、正常性の状態、およびルーターとターゲット領域間の接続のタイプが含まれます。

ルーティングされるトランザクションにアクティブな親和性がない場合、キュー・アルゴリズムを使用すると、CICSplex SM は以下の一連の基準に合致するターゲット領域を選択します。

- 領域は、ターゲット領域で許可されるタスクの最大数と比較して、処理を待機する作業キューが最短になる。この作業キュー (ロード・カウントとも呼ばれる) は、キューに入れられたすべてのアクティブ・ユーザー・タスクの数です。デフォルトでは、MAXTASKS および TRANCLASS 両方の属性に関してキューに入れられたタスクがロード・カウントに含まれます。CSYSDEF リソース・テーブルで「タスク・ロード・キュー・モード」属性を使用すると、TRANCLASS 属性に関してキューに入れられたタスクをサイトでロード・カウントから除外することができます。CSYSDEF リソース・テーブルの「タスク・ロード・キュー・モード」属性の詳細については、CICS システム定義 - CSYSDEF を参照してください。
- 領域は、ストレージ不足、SYSDUMP、および TRANDUMP などの条件の影響が一番小さい。
- 領域は、トランザクションが停止する可能性が一番低い。
- 領域は、領域間通信のパスが一番短い。
- 領域は、1 から 100 までの範囲の z/OS WLM ヘルス値を持っている。

キュー・アルゴリズムでは、作業スループットが最大化され、CICSplex 全体の応答時間が標準化されます。このアルゴリズムは非常に堅固なものです。プロセッサ能力の違い、ターゲット領域の最大タスク値の違い、ターゲット領域の非対称な構成、および予測不能なワークロードに対応できます。

## リンクに依存しないキュー・アルゴリズム

リンクに依存しないキュー (LNQUEUE) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、キュー・アルゴリズムと同等です。

ルーティングされるトランザクションにアクティブな 親和性がない場合、リンクに依存しないキュー・アルゴリズムを使用すると、CICSplex SM は以下の一連の基準に合致するターゲット領域を選択します。

- 領域は、ターゲット領域で許可されるタスクの最大数と比較して、処理を待機する作業キューが最短になる。この作業キュー (ロード・カウントとも呼ばれる) は、キューに入れられたすべてのアクティブ・ユーザー・タスクの数です。デフォルトでは、MAXTASKS および TRANCLASS 両方の属性に関してキューに入れられたタスクがロード・カウントに含まれます。CSYSDEF リソース・テーブルで「タスク・ロード・キュー・モード」属性を使用すると、TRANCLASS 属性に関してキューに入れられたタスクをサイトでロード・カウントから除外することができます。CSYSDEF リソース・テーブルの「タスク・ロード・キュー・モード」属性の詳細については、CICS システム定義 - CSYSDEF を参照してください。
- 領域は、ストレージ不足、SYSDUMP、および TRANDUMP などの条件の影響が一番小さい。
- 領域は、トランザクションが停止する可能性が一番低い。
- 領域は、1 から 100 までの範囲の z/OS WLM ヘルス値を持っている。

リンクに依存しないキュー・アルゴリズムでは、CICSplex 全体で効率的な作業スループットと応答時間が得られます。キュー・アルゴリズムと同様に、このアルゴリズムでは、プロセッサ能力の違い、ターゲット領域の最大タスク値の違いに対応できます。ただし、このアルゴリズムでは、ルーターとターゲット間の通信リンク速度を考慮に入れません。キュー・アルゴリズムと比較して、このアルゴリズムはターゲット・スコープ全体でワークロードをより均等に分散できるかもしれませんが、ワークロードはそれほど迅速に完了しません。

## ゴール・アルゴリズム

ゴール・アルゴリズムは、動的ルーティングの決定が、主に応答時間ターゲットを達成する領域の機能に関連している時に使用される場合があります。

ゴール・モード・ルーティングは、以下のシナリオでルーターとターゲットが同じ CMAS によって管理されている場合のみ使用します。

- 動的トランザクションのために DTRPGM を使用して動的ルーティングを実行する場合。
- APPC 接続または MRO 接続による EXEC CICS START TERMID のために DTRPGM を使用して動的ルーティングを実行する場合。
- ビジネス・トランザクション・サービス・ルーティングのために DSRTPGM を使用して分散ルーティングを実行する場合。

その他の動的ルーティング・シナリオでは、QUEUE アルゴリズムまたは LNQUEUE アルゴリズムを使用します。

ルーティングされるトランザクションにアクティブな 親和性がない場合、ゴール・アルゴリズムを使用すると、CICSplex SM は以下の一連の基準に合致するターゲット領域を選択します。

- ストレージ不足、SYSDUMP、および TRANDUMP などの条件の影響が最も小さい。
- トランザクションを停止させる原因になる可能性が最も小さい。
- z/OS のワークロード・マネージャー・コンポーネントを使用してこのトランザクションに対して (およびこのクラスのその他のトランザクションに対して) 設定された、トランザクション応答時間の目標 (平均または百分位数のいずれか) を満たす可能性が最も高い。
- 領域間通信のパスが一番短い。
- MAXTASK と比較して、処理を待機するトランザクション・キューが最短となる AOR である。
- 1 から 100 までの範囲の z/OS WLM ヘルス値を持っている。

## リンクに依存しないゴール・アルゴリズム

リンクに依存しないゴール (LNGOAL) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、ゴール・アルゴリズムと同等です。

ルーティングされるトランザクションにアクティブな親和性がない場合、リンクに依存しないゴール・アルゴリズムを使用すると、CICSplex SM は以下の一連の基準に合致するターゲット領域を選択します。

- 領域は、ストレージ不足、SYSDUMP、および TRANDUMP などの条件の影響が一番小さい。
- 領域は、トランザクションが停止する可能性が一番低い。
- 領域は、z/OS のワークロード・マネージャー・コンポーネントを使用してこのトランザクションに対して (およびこのクラスのその他のトランザクションに対して) 設定された、トランザクション応答時間の目標 (平均または百分位数のいずれか) を満たす可能性が最も高い。
- 領域は、MAXTASK と比較して、処理を待機するトランザクション・キューが最短となる AOR である。
- 領域は、1 から 100 までの範囲の z/OS WLM ヘルス値を持っている。

## CICSplex SM ワークロードのルーティングに対する z/OS WLM ヘルス・サービスの効果

CICS 領域で、z/OS ワークロード・マネージャー (z/OS WLM) のヘルス・サービスがアクティブな場合、CICSplex SM WLM は、ルーティングの決定において、その領域の z/OS WLM の正常性の状態を考慮します。

### 前提条件:

CICSplex SM WLM がターゲット領域の z/OS WLM の正常性の状態を考慮するためには、以下の条件が適用されます。

- ルーティング領域とターゲット領域の両方が、少なくとも CICS TS V5.4 (CICS および CICSplex SM ライブラリー) でなければなりません。そうでない場合、z/OS WLM の正常性の状態は無視され、ターゲット領域は作業を受け取る準備ができていると見なされます。
- z/OS WLM ヘルス・サービスが CICS 領域で有効になっている必要があります。これは、デフォルトで使用可能になっています。サービスが無効の場合にも、領域は作業を受け取る準備ができていると見なされます。このサービスをアクティブにするには、**WLMHEALTH** システム初期設定パラメーターを設定する必要があります。

### CICS 領域の z/OS WLM の正常性の状態が CICSplex SM のルーティングの決定にどのように影響するか

領域の z/OS WLM の正常性の状態は、z/OS WLM ヘルス値によって 0 - 100 の範囲で反映されます。作業の経路指定先を決定する際に、CICSplex SM WLM は、ターゲット領域のヘルス値に基づいて、そのターゲット領域のルーティング・アルゴリズム内に追加の重みを割り当てて場合があります。

- ターゲット領域の z/OS WLM のヘルス値がゼロの場合、そのターゲット領域は作業を受け取るのに適格ではありません。CICSplex SM WLM は、その領域に作業を経路指定しません。この領域に関連付けられているワークロード親和性はそのまま残されて受け入れられます。ただし、ヘルス値がゼロであると、これらの親和性に基づくワークロード・ルーティングは失敗し、メッセージ EYUWR0003W または SYSIDERR 条件が生成されます。メッセージ EYUWR0003W は、EYU9WRAM プログラムが有効であり、親和性 AOR が使用不可であったことを示します。このようなルーティングの障害を解決するには、ヘルス値がゼロ以外になるように領域のウォームアップを開始してから、ワークロードを再実行します。[CICS システム・ウォームアップの開始](#)を参照してください。
- ヘルス値が 1 から 99 までのターゲット領域の場合、CICSplex SM WLM は、ルーティング・アルゴリズムにヘルス値に基づいて追加の重みを割り当てます。ヘルス値が大きいほど重みが小さくなり、その領域はルーティング決定で有利になります。同様に、ヘルス値が小さいほど重みが大きくなり、その領域が追加の作業を受け取る可能性が低くなります。同じスコープ内でより正常性の高い他の領域を使用できない場合には、高い値の追加の重みが割り当てられている領域がアクティブのままであり、引き続き作業を受け取る可能性があることに注意してください。
- ターゲット領域のヘルス値が 100 の場合、追加の重みは割り当てられません。

CICS のウォームアップ・プロセスまたはクールダウン・プロセス中、CICS は、領域の z/OS WLM ヘルス値を調整します。その結果、CICSplex SM WLM のルーティングの決定に影響し、そのプロセス中の領域へのワークロードが制限されます。一般に、領域がウォームアップ中のときは、そのヘルス値は定期的に増



加し、その領域は徐々により大きいワークロードを受け取るようになります。一方、領域がクールダウン中のときは、そのヘルス値は定期的に減少し、最終的にその領域がターゲットとして不適格になるまで、徐々により小さい作業を受け取るようになります。CICS のウォームアップまたはクールダウン中に行われることについて詳しくは、[z/OS ワークロード・マネージャー正常性サービスの使用による CICS のウォームアップおよび冷却](#)を参照してください。

### 領域をワークロード・ルーティングのターゲットとして不適格にする

CICSplex SM のルーティングの決定で、ワークロード・ルーティングのターゲットとして領域を意図的に不適格にすることができます。これを行うには、「**MVS ワークロード管理**」(MVSWLM) ビューでその領域の z/OS WLM ヘルスのオープン状況を CLOSE に設定します。これにより、領域の z/OS WLM ヘルス値が 0 に下がり、その領域が不適格になったことが示されるまで、その領域はターゲットとして徐々に不適切な状態になっていきます。また、z/OS WLM ヘルスのオープン状況を IMMCLOSE に設定することによって、領域の z/OS WLM ヘルス値を即時にゼロに変更できます。

## ワークロード分離

ワークロード分離は、要求領域またはルーティング領域からの作業をターゲット領域セットに分散してルーティングしますが、ユーザー、端末、トランザクション自体、およびそれらの BTS プロセス・タイプの要件によって、どのターゲット領域セットが使用されるかに影響がある場合があります。

ターゲット領域の特定のグループへの特定のトランザクションまたはプログラムのルーティングは、ユーザー ID、端末 ID、およびトランザクション・グループのいずれかの組み合わせに基づいて行われます。BTS プロセスでは、ルーティングはトランザクション・グループとプロセス・タイプの組み合わせに基づいて行われます。エンタープライズ Bean に関連したトランザクションでは、ルーティングはトランザクション・グループに基づいて行われます。Link3270 ブリッジ要求では、端末 LU 名ごとの分離はいくつかの制限に従います。詳細については、[Link3270 ブリッジのワークロードの分離](#)を参照してください。

例えば、以下を指定できます。

- ユーザー ID の先頭文字が「PAY」であるユーザーによって開始されるすべてのトランザクションは、CICS システム・グループ AORPAY1 内のターゲット領域にルーティングされる必要があります。
- LU 名 の先頭文字が「NYORK」である端末から開始されるすべてのトランザクションは、CICS システム・グループ AORNYORK 内のターゲット領域にルーティングされる必要があります。
- トランザクション・グループ ACCOUNTS に属するすべてのトランザクションは、CICS システム・グループ AORACCNT 内のターゲット領域にルーティングされる必要があります。
- BTS プロセス・タイプ TRAVEL に関連付けられたすべてのプロセスは 1 つの BTS セットに含まれるターゲット領域にルーティングされ、BTS プロセス・タイプ PAYROLL に関連付けられたすべてのプロセスは別の BTS セットにルーティングされます。
- トランザクション・グループ STOCK に属するすべてのエンタープライズ Bean 関連のトランザクションは、ユーザー ID がワークロード定義内のユーザー ID に一致する場合は、CICS システム・グループ STOCKTG 内の任意のターゲット領域にルーティングされます。

ワークロード分離基準を満たすターゲット領域のグループから選択される領域は、59 ページの『ワークロード・ルーティング』で説明されているルーティング基準と同じ基準に基づいて決定されます。ルーティング決定時に適用されるアルゴリズムは、個別のワークロード分離レベルで指定することもでき、基本ワークロード・ルーティングによって使用されるデフォルトのアルゴリズムのままにしておくこともできます。

## トランザクション間の類縁性

トランザクション間の親和性とは、指定した期間内に複数のトランザクションを同じターゲット領域で処理する必要がある場合の、トランザクション間の関係です。例えば、3 つの別個のトランザクションで構成される疑似会話があり、各トランザクションは一時記憶域キュー (分散モデルで共用される場合がある) を介してデータを順番に次のトランザクションに渡すかもしれません。そのとき、3 つすべてのトランザクションを同じターゲット領域で処理する必要があることを指定し、さらにこの親和性が疑似会話の期間中存続することを指定するかもしれません。



この親和性を CICSplex SM に定義しなかった場合、各トランザクションは別のターゲット領域にルーティングされて、結果として直前のトランザクションで残された一時記憶域データにアクセスできなくなる可能性があります。ターゲット領域自体は、指定したターゲット・スコープから CICSplex SM によって選択されます。

ワークロード管理および IBM CICS Interdependency Analyzer for z/OS は、BTS プロセスとアクティビティの間の親和性を認識します。BTS 自体は親和性をもたらすわけではなく、親和性をもたらすプログラミング手法を抑制しますが、親和性をもたらす可能性のある既存コードの処理を行います。そのようなワークロード管理の親和性を定義して、目的にかなったルーティングを決定できるようにする必要があります。それぞれの親和性の存続期間を指定することが特に重要です。これに失敗すると、ワークロード管理のルーティング・オプションが不必要に制限される場合があります。

ワークロード管理および IBM CICS Interdependency Analyzer for z/OS は、端末に関係していないルーティング可能な EXEC CICS START コマンド間の親和性、またはユーザー ID と端末のいずれにも関連付けられていない DPL 間の親和性を認識しません。アプリケーションからすべての親和性を除去するか、アプリケーションがすべての親和性を必ず受け入れるかの、いずれかの手順を実行する必要があります。

**EXEC CICS RETURN** コマンドで COMMAREA を介してトランザクション間でデータが渡される場合にはそのような親和性は存在しない点に注意してください。COMMAREA は要求領域に戻されるため、次のトランザクションを順番に処理するために選択されたターゲットに渡すことができます。

### トランザクション間の親和性が受け入れられる方法

関連するトランザクションのグループからの最初のトランザクションが開始されるとき、CICSplex SM は、指定したターゲット・スコープから該当する領域を選択します。

ターゲット・スコープに複数の適切な領域がある場合、CICSplex SM は現行のワークロード・ルーティング・アルゴリズムを使用して 1 つを選択します。同じグループ内の、親和性基準を満たす後続のトランザクションは、最初のトランザクションと同じ領域に送信されます。後続のトランザクションが親和性基準を満たさない場合 (例えば、同じ疑似会話が別のユーザー ID から開始された場合など)、適切な領域に関する選択プロセスが再度開始します。

## CICSplex SM ワークロード管理によるワークロードの管理

CICSplex SM ワークロード管理では、エンタープライズ・システムのプロセッサ能力を最適化します。このためにワークロード管理では、トランザクションおよびプログラムをその時点で最も適切な CICS 領域に動的に経路指定します。その際、存在しているトランザクションの親和性が考慮されます。

ワークロード管理の要件を確認したら、[ワークロード管理の構成](#)の説明のように CICSplex SM に対してそれらを定義してください。

### WLM の計画

このセクションでは、企業において CICSplex SM ワークロード管理をどの程度使用できるかを決定する上で役立つ説明を記載します。

#### ワークロード・ルーティングかワークロード分離かの選択

ワークロード・ルーティングを使用すると、使用可能な CICS システムを最大限活用でき、スループットを向上し、パフォーマンスを改善することができるため、可能な限りワークロード・ルーティングを使用してください。ワークロード分離 (プロセス・タイプ別、ユーザー別、端末別、トランザクション別) は、CICSplex SM ワークロード・ルーティング機能を最大限に活用する際の妨げとなるため、どうしても必要である場合のみ実施してください。

複数の CICSplex を定義し、例えば異なるユーザー・グループによる CICS システムの使用を反映するように分割した場合、CICSplex 内でワークロード分離ではなく、単純なワークロード・ルーティングを使用できる可能性があります。例えば、2 つの CICSplex を定義することにより、グループ A によって使用される CICS システムとグループ B によって使用される CICS システムを分離した場合、各 CICSplex 内でワークロード・ルーティングを実施できます。このようなアプローチを取らなかった場合、ワークロード分離を実施することによってそのようなグループを認識することが必要であると考えられるかもしれません。

## ワークロードの識別

WLM の計画を立てるには、企業内で処理されるワークロードを識別することから開始します。ワークロードは、基礎の TOR-AOR-FOR 構成で確実に明らかになります。識別されたワークロードが、CICS システムの現行の CICSplex SM 構成でサポートされることを確認します。

特に以下の点に注意してください。

- 単一のワークロードにおけるルーティング領域およびターゲット領域は、同じ CICSplex に存在する必要があります。つまり、提供された動的ルーティング・プログラムは、CICSplex の境界を越えてトランザクションをルーティングすることはできません。(提供された動的ルーティング・プログラムをカスタマイズすることにより、CICSplex の外側にトランザクションをルーティングすることは可能です。提供された動的ルーティング・プログラムのカスタマイズについては、動的ルーティングの変更に説明されています。)
- ルーティング領域には以下を選択する必要があります。
  - CICS TS 領域。
  - ローカル MAS。つまりルーティング領域は、CMAS が存在しない MVS イメージ上では実行できません。
  - 1 つのワークロードのみ。つまりルーティング領域は、1 つの時点において 1 つのアクティブ・ワークロード仕様としか関連付けができません。
- ターゲット領域には以下を選択できます。
  - ローカル MAS。
  - 複数のワークロード内。
  - CICSplex SM によって管理される任意の CICS システム。

## トランザクション間の親和性の識別

トランザクション間の親和性 (関連したトランザクションを同じターゲット領域で処理する必要がある) があると、最適なワークロード分散が妨げられます。一般的にこの親和性が使用される理由は、1 つのトランザクションが別のトランザクションにデータを渡す方法のため、または 2 つ以上のトランザクションの処理を調整する要件のためのいずれかの理由です。親和性の識別は常に容易にできるわけではありませんが、いくつかの方式を使用できます。

例えば、アプリケーション設計文書またはソース・コードを検討できます。CICS トレースを実行できます。あるいは IBM CICS Interdependency Analyzer for z/OS を使用できます。

ワークロードに含まれる親和性を識別したら、それらを除去するためのあらゆる努力をする必要があります。それらを完全には除去できない場合は、親和性の期間を測定し、それを最小化するようにしてください。トランザクション間の親和性が次の期間存続するように CICSplex SM に対して定義できます。

- ユーザーのセッションがアクティブである間
- 端末セッションの期間
- ターゲット領域がアクティブ状態になっている間
- ワークロードがアクティブである間
- 疑似会話の期間
- BTS アクティビティがアクティブである間
- BTS プロセスがアクティブである間

CICSplex SM がアクティブな親和性を受け入れるようにしてください。親和性がアクティブであってもターゲット領域が使用不可になると、トランザクションはルーティングされません。BTS トランザクションの場合、BTS は領域が開始されるまで待機します。

## WLM の実施

CICSplex SM ワークロード管理機能を使用するために何を行う必要があるかは、ワークロード・ルーティングのみを実施するか、それともワークロード分離も実施した上で同じ CICSplex に 1 つ以上のトランザクション親和性を定義する可能性があるかによって異なります。

以下のプロセスに従って作業すると、最良の結果が得られます。

1. 動的ルーティングの候補を識別する。
2. CICSplex の中で、動的ルーティングが可能なワークロードごとにワークロード・ルーティングを実施する。
3. 必要なワークロード分離要件を追加する。
4. トランザクション間の親和性を定義する。

### 動的ルーティングの候補の識別

作業要求のすべてが動的ルーティングの候補になるわけではありません。

作業要求が適格となるために適用する必要がある条件は、[動的ルーティング](#)にリストされています。

詳細については、以下を参照してください。

- [EXEC CICS LINK](#) コマンドおよび [EXEC CICS START](#) コマンド。
- [EXEC CICS RUN ASYNCHRONOUS](#) コマンド。
- CICS および CICSplex SM リソースについての情報は、[CICS リソース](#)および [BAS の管理](#)を参照してください。
- Java™ に関する詳細は、[JVM サーバーへのアプリケーションのデプロイ](#)を参照してください。
- Link3270 ブリッジに関する詳細は、[CICS 外部インターフェースの概要](#)を参照してください。

### ワークロード・ルーティングの実施

ターゲット領域グループのワークロード・ルーティングを実施するには、ルーティング領域またはルーティング・システム・グループをワークロード仕様に関連付ける必要があります。オプションにより、ワークロード内の指定されたトランザクション・グループに代わりのルーティングを適用できます。

### このタスクについて

ワークロード・ルーティングを使用するには、ワークロード仕様 (WLMSPEC オブジェクト) でルーティング特性を指定します。これらの特性は、ワークロード内のすべてのトランザクションに適用されます。

ただし、1 つ以上のトランザクションに適用される特殊ルーティング評価を指定する、関連付けられた WLM 定義とトランザクション・グループ定義 (WLMDEF オブジェクトと TRANGRP オブジェクト) のペアを提供することによりこれらのルーティングの特性をオーバーライドできます。ワークロード・ルーティングでは、トランザクション・グループ定義に異なるルーティング・アルゴリズムを指定します。トランザクション・グループ・レベルで代わりのルーティング・アルゴリズムを指定する場合、ルーティング領域を停止させることなくワークロード・ルーティング特性を動的に変更できます。

### 手順

1. 単一のワークロードに複数のルーティング領域がある場合、CICS システム・グループ (CSYSGRP) オブジェクトを使用して、それらのルーティング領域の CICS システム・グループを作成します。
2. ルーティング領域がトランザクションのルーティング先として指定できるターゲット領域の CICS システム・グループを (この場合も CSYSGRP を使用して) 作成します。  
このグループには、CICSplex 内のすべての CICS システムを含めることもできます。
3. ワークロード管理仕様 (WLMSPEC) オブジェクトを使用して、CICSplex のワークロードごとにワークロード仕様を 1 つずつ作成します。  
ワークロード仕様では、次のようになっています。
  - a) ターゲット・システム・グループの名前をターゲット・スコープ値として指定します。
  - b) ワークロード・ルーティングにどのルーティング・アルゴリズムを使用するか特定します。  
以下のいずれかのアルゴリズムを使用できます。
    - QUEUE
    - LNQUEUE
    - GOAL
    - LNGOAL

4. ルーティング領域、またはルーティング・システム・グループを、ワークロード仕様に関連付けます。  
単一の CICSplex に複数のワークロードを配置できますが、1つのルーティング領域または1つのルーティング領域グループは、一度に1つのアクティブ・ワークロード仕様にしか属することができません。
5. オプション: トランザクション・グループ定義に異なるルーティング・アルゴリズムを指定することで、ワークロード内の特定のトランザクションのルーティング・アルゴリズムをオーバーライドします。  
トランザクション・グループの作成を参照してください。

インストールされたトランザクション・グループを変更する場合、WLM 定義によって指定されたトランザクション・グループも最新表示されるように、トランザクション・グループに関連付けられた WLM 定義 (WLMDEF) を破棄してから、再インストールする必要があります。関連付けられた WLMDEF を破棄したり再インストールしたりせずに、ルーティング・アルゴリズムを直ちに変更するには、「アクティブなワークロード・トランザクション・グループ (WLMATGRP)」ビューおよび **SET** コマンドを使用して、ALGTYPE 属性を変更することができます。

### ワークロード分離要件の追加

同じワークロード内で、ユーザー ID、端末 ID、およびトランザクション・グループの組み合わせ、またはプロセス・タイプとトランザクション・グループの組み合わせのいずれかによってワークロード分離を実施する場合、以下を行います。

1. ターゲット領域セットごとに、「**システム・グループ定義 (System group definitions)**」ビュー (CSYSGRP オブジェクト) を使用して CICS システム・グループを作成します。例えば次の状況を考えます。

- 端末 NETA\* からユーザー ABC\* によって開始されたトランザクションをターゲット領域 2 から 7 にルーティングする必要がある。

かつ

- 任意の端末からユーザー HIJ\* によって開始されたトランザクションをターゲット領域 5 から 9 にルーティングする必要がある。

この場合、2つのターゲット・システム・グループを定義する必要があります。1つのグループにはターゲット領域 2 から 7 が入り、1つのグループにはターゲット領域 5 から 9 が入ります。

2. トランザクションによるワークロード分離を実施するには、「**トランザクション・グループ (Transaction groups)**」ビュー (TRANGRP オブジェクト) を使用してトランザクション・グループを定義します。例えば、トランザクション EFG1、EFG2、および EFG3 を特定のターゲット領域グループにルーティングする必要がある場合、それらのトランザクションが含まれるトランザクション・グループを定義する必要があります。

3. ユーザー、端末、およびトランザクション・グループの組み合わせごとに、「**ワークロード定義 (Workload definitions)**」ビュー (WLMDEF オブジェクト) を使用して1つのワークロード定義を作成します。例えば、項目 68 ページの『1』(前述) では次のようにします。

- 任意のトランザクション、ユーザー ABC\*、および端末 NETA\* の組み合わせにおいて、1つのワークロード定義が必要になります。
- 任意のトランザクション、ユーザー HIJ\*、および任意の端末の組み合わせにおいて、1つのワークロード定義が必要になります。

項目 68 ページの『2』で説明されている、トランザクションによるワークロード分離をサポートするために、3番目のワークロード定義が必要になります。各ワークロード定義に、適格となるトランザクションがルーティングされる適切なターゲット領域のシステム・グループの名前を含めます。

4. ワークロードの最初の要求領域が開始されるときにワークロード定義が自動的にインストールされる場合、「**ワークロード・グループ (Workload groups)**」ビュー (WLMGROUP オブジェクト) を使用して以下を行う必要があります。
  - a. ワークロード・グループを作成する。
  - b. ワークロード定義をワークロード・グループに追加する。
  - c. ワークロード・グループをワークロード仕様に関連付ける。



通常、ワークロード分離が正規の一貫した要件であるため、ワークロード定義の自動インストールが必要であると考えられます。ただし、すぐに効果を得るため、ワークロード・グループまたは個々のワークロード定義を手動でインストールすることもできます。

### トランザクション間の親和性の定義

同じターゲット領域によって処理されるワークロードに、複数のトランザクションに対する 1 つの要件がある場合、以下を行います。

1. 親和性を共有するワークロード内の一連のトランザクションごとに、「**トランザクション・グループ (Transaction groups)**」ビュー (TRANGRP オブジェクト) を使用してトランザクション・グループを作成します。例えば、トランザクション MNO1、MNO2、および MNO3 を同じターゲット領域で処理する必要がある場合 (それらが順番に実行され、次のトランザクションのためのデータが残るためなどの理由で)、親和性タイプとその期間を指定することにより、トランザクション・グループにこれら 3 つのトランザクションが含まれることを定義します。
2. ワークロード分離を実施する場合、トランザクション・グループごと、およびユーザーまたは端末のパターンごとに、「**ワークロード定義 (Workload definitions)**」ビュー (WLMDEF オブジェクト) を使用して、ワークロード定義を作成します。例えば、項目 69 ページの『1』(前述)で説明されているトランザクション間の親和性が端末 NYORK\* からの任意のユーザーに適用される場合、ワークロード定義にはその情報を含める必要があります。加えて、ワークロード定義は、CICSplex SM がそれを選択できるターゲット領域セットを指定します。これは、CICSplex 内の各ターゲット領域、またはターゲット領域のいくつかのサブセットにできます。

トランザクション間の親和性は、ワークロード分離と一緒に実施しなくても CICSplex SM に定義できる点に注意してください。この場合、ワークロード定義は必要なく、CICSplex SM はグループ内の最初のトランザクションを、ワークロード仕様で指定されるセット内の適切なターゲット領域にルーティングします。トランザクション・グループ内の後続のトランザクションは、親和性がアクティブである間は同じターゲット領域にルーティングされます。

3. ワークロードの最初のルーティング領域が開始されるときにワークロード定義が自動的にインストールされる場合、「**ワークロード・グループ (Workload groups)**」ビュー (WLMGROUP オブジェクト) を使用して以下を行う必要があります。
  - a. ワークロード・グループを作成する。
  - b. ワークロード定義をワークロード・グループに追加する。
  - c. ワークロード・グループをワークロード仕様に関連付ける。

すぐに効果を得るため、ワークロード・グループまたは個々のワークロード定義を手動でインストールすることもできます。

注: グループ単位でルーティング領域とターゲット領域を定義することは重要です。これにより、これらのグループの構成に変更を加えると、CICSplex SM のワークロード管理に自動的に反映されます。

### 推奨されている方法

「**マップ (Map)**」ボタンをクリックすると、データ・リポジトリに含まれる WLM 定義の視覚マップを表示できます。企業システムで WLM 機能を実施することを決定した場合、初めに適切なオブジェクトを定義する必要があります。

以下を実行する必要があります。

- 「**WLM 定義 (WLM definitions)**」ビューを使用し、ワークロード定義 (WLMDEF オブジェクト) を作成します。
- 「**WLM グループ (WLM groups)**」ビューを使用し、ワークロード・グループ (WLMGROUP オブジェクト) を作成します。
- 「**WLM 仕様 (WLM specifications)**」ビューを使用し、ワークロード仕様 (WLMSPEC オブジェクト) を作成します。

その後、これらのオブジェクト間の関連を作成するには以下のようにします。

- 「**WLM 定義 (WLM definitions)**」ビューを使用し、ワークロード定義をワークロード・グループ (WLMINGRP オブジェクト) に追加します。

- 「**ワークロード・グループ (Workload groups)**」ビューを使用し、ワークロード・グループをワークロード仕様 (WLMINSPC オブジェクト) に追加します。
- 「**ワークロード仕様 (Workload specifications)**」ビューを使用し、ワークロード仕様を CICS システム (LNKSWSCS オブジェクト) または CICS システム・グループ (LNKSWSCG オブジェクト) に関連付けます。LNKSxSCG パラメーターについて詳しくは、[LNKSxSCG Records \(LNKSMSCG, LNKSRSRG, LNKSWSCG\)](#) を参照してください。

既に定義したオブジェクトおよび関連のリストを表示する場合、以下のいずれかのビューを使用します。

- 「**グループ内のワークロード定義 (Workload definitions in groups)**」ビュー (WLMINGRP オブジェクト)
- 「**ワークロード仕様内のワークロード・グループ (Workload groups in workload specifications)**」ビュー (WLMINSPC オブジェクト)
- 「**WLM 仕様と CICS システムとの間のリンク (WLM specifications to CICS system links)**」ビュー (LNKSWSCS オブジェクト)
- 「**WLM 仕様と CICS システム・グループとの間のリンク (WLM specifications to CICS system group links)**」ビュー (LNKSWSCG オブジェクト)

## ワークロード管理リソース

ワークロード環境をセットアップおよび管理するには、ワークロード管理 WUI ビュー・セットおよびリソース・オブジェクトを使用します。

WLM 構成要素の図については、[CICSplex SM WLM のコンポーネント](#)を参照してください。

表 5. ワークロードを管理するための WUI ビュー・セットおよびリソース・オブジェクト		
WUI ビュー・セット	オブジェクト名	説明
トランザクション・グループ内のトランザクション	DTRINGRP	このオブジェクトには、1 つ以上のトランザクション・グループに関連したトランザクションに関する情報が表示されます。
WLM 仕様とシステム・グループとの間のリンク	LNKSWSCG	ワークロード仕様に関連付けられた CICS システム・グループ
WLM 仕様と CICS システムとの間のリンク (WLM specifications to CICS system links)	LNKSWSCS	ワークロード仕様に関連付けられた CICS システム
トランザクション・グループ (Transaction groups)	TRANGRP	トランザクション・グループは、論理的に類似したトランザクションを関連付けたものです。類似性は、親和性の要件、共通共用処理の要件、またはその他のユーザー定義の特性に基づいて判断されることがあります。トランザクション・グループはワークロード定義の一部として組み込まれ、次にワークロード定義はターゲット領域として使用される CICS システム・グループを定義します。具体的にトランザクション・グループに関連付けられていないトランザクションは、デフォルトのトランザクション・グループに割り当てられます。
アクティブ・ワークロードのトランザクション・グループの親和性	WLMATAFF	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したトランザクション・グループのアクティブな親和性に関する情報を示します。トランザクション・グループに関連した最初のトランザクションが動的にターゲット領域にルーティングされる際に、親和性がアクティブになります。
アクティブ・ワークロードのトランザクション・グループ	WLMATGRP	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したトランザクション・グループに関する情報を示します。



表 5. ワークロードを管理するための WUI ビュー・セットおよびリソース・オブジェクト (続き)		
WUI ビュー・セット	オブジェクト名	説明
アクティブ・ワークロードの動的トランザクション	WLMATRAN	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したすべてのアクティブ・トランザクションに関する情報を示します。
アクティブ・ワークロードのターゲット配布係数	WLMAWAOR	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したすべてのターゲット領域に関する情報を示します。
アクティブ・ワークロード定義	WLMAWDEF	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したアクティブ・ワークロード定義に関する情報を示します。
アクティブ・ワークロード	WLMAWORK	<p>このオブジェクトは、コンテキストとして識別される CICSplex 内のアクティブ・ワークロードに関する情報を示します。ワークロードは、以下の条件に該当する限り、CICSplex 内でアクティブです。</p> <ul style="list-style-type: none"> <li>ルーティング領域またはターゲット領域として作動していて、ワークロードに関与している CICS システムは、その CICSplex に接続されます。</li> <li>すべてのトランザクションは、PERMANENT のアフィニティー存続時間を確立します。</li> </ul>
アクティブ・ワークロード内のルーティング領域 (Routing regions in an active workload)	WLMAWTOR	このオブジェクトは、コンテキストとして識別される CICSplex 内のワークロードに関連したルーティング領域に関する情報を示します。
ワークロード定義 (Workload definitions)	WLMDEF	このオブジェクトは、どのトランザクションがどの CICS システム・グループにルーティングされるかを指定します。トランザクションは、トランザクション・グループごとに指定し、そのグループ内では LU 名とユーザー ID によって、またはプロセス・タイプによって指定します。
ワークロード・グループ (Workload groups)	WLMGROUP	このオブジェクトは、1 つ以上のワークロード定義を関連付けるために使用されます。
グループ内のワークロード定義 (Workload definitions in groups)	WLMINGRP	このオブジェクトは、ワークロード・グループの名前と、それらに関連付けられたワークロード定義を表示します。
ワークロード仕様内のワークロード・グループ (Workload groups in workload specifications)	WLMINSPC	このオブジェクトは、ワークロード仕様の名前と、それらに関連付けられたワークロード・グループを表示します。
ワークロード仕様 (Workload specifications)	WLMSPEC	ワークロード仕様は、1 つのワークロードと、ターゲット領域として作動する 1 つ以上の CICS システム・グループを指定します。さらにデフォルトのトランザクション・グループの属性も定義します。

## ワークロード管理定義とそれぞれの関連ビュー

Web ユーザー・インターフェース (WUI) ワークロード・マネージャーの管理ビューを使用して、さまざまなワークロード管理属性を定義します。

72 ページの図 8 は、CICSplex SM オブジェクト・モデルの観点からワークロード管理ビューの概要を示しています。これらのビューに加え、アクティブ・ワークロードに関する情報を表示したり、アクティブ・ワークロードを管理したりするためのビューを使用できます。また、「マップ」ボタンを使用することによって、ワークロード管理定義のビジュアル・マップを表示することもできます。

73 ページの図 9 は、CICSplex 内のワークロードのコンポーネントと、ワークロードを確立するために使用されるビューとの関係を示しています。

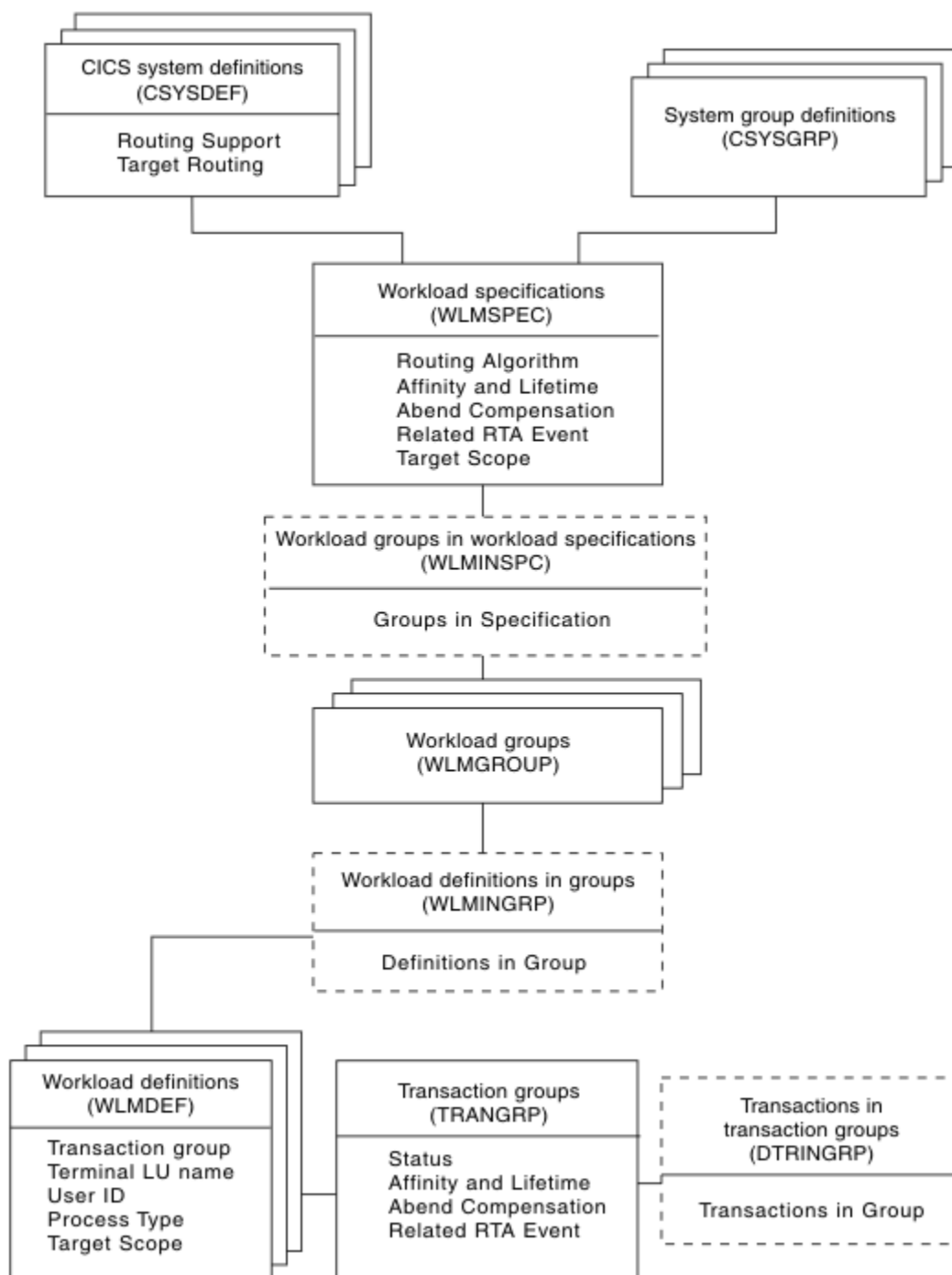


図 8. ワークロード管理オブジェクトと関連付けを作成するためのビュー

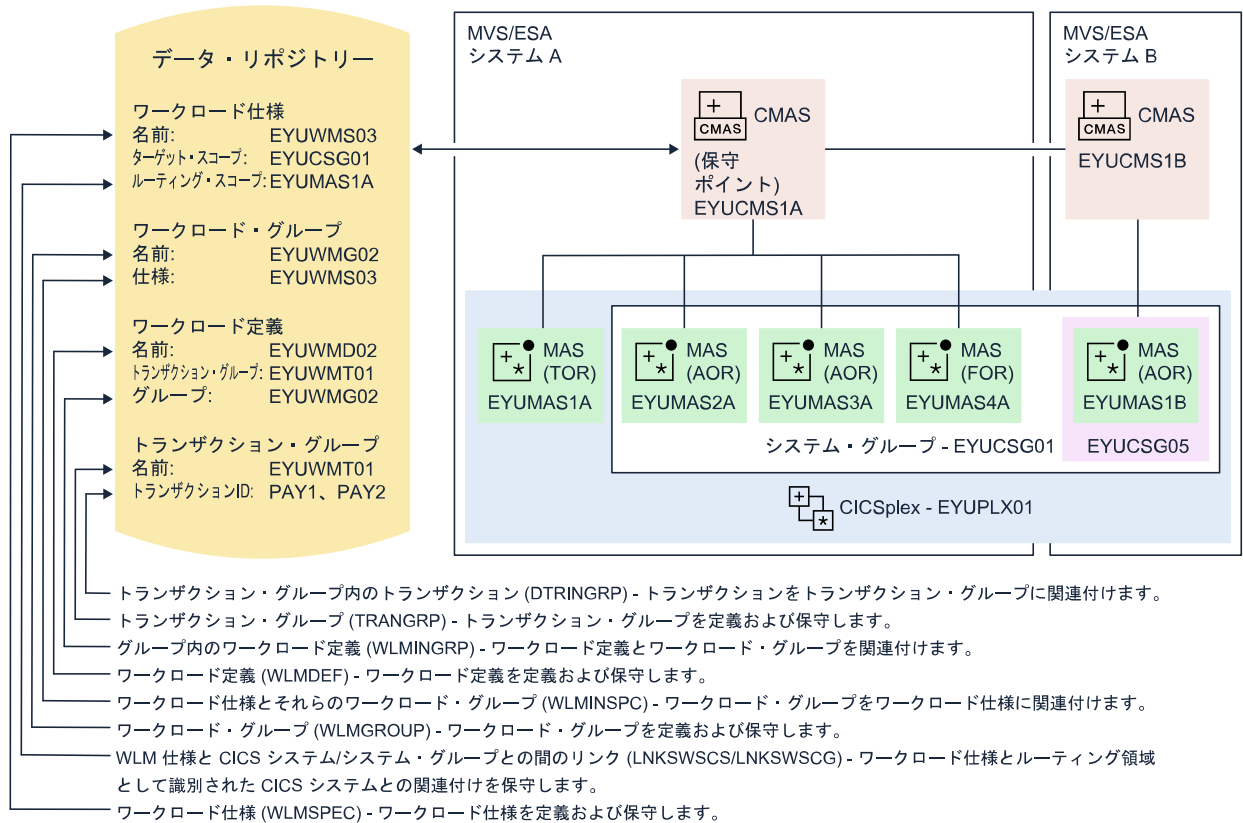


図 9. ワークロードとワークロード管理ビューとの関係

74 ページの表 6 は、ワークロード管理定義の作成に使用できるビューを示しています。また、表示できる情報やこれらのビューを使用して実行できるアクションも示しています。Web ユーザー・インターフェースでこれらのビューにアクセスするには、メインメニューから「管理」>「ワークロード・マネージャー管理」をクリックします。これらのビューの詳細については、[CPSM 管理ビュー](#)を参照してください。

表 6. ワークロード管理定義を作成および保守するためのビュー

WUI ビュー	オブジェクト名	サポートされるタスク
トランザクション・グループ内のトランザクション	DTRINGRP	<ul style="list-style-type: none"> <li>• 現行コンテキスト内のすべてのトランザクション・グループを表示します。</li> <li>• トランザクション・グループからトランザクションを除去します。</li> <li>• 指定されたトランザクション・グループを開始点として使用してワークロード定義のマップを表示します。</li> </ul>
ワークロード仕様に関連付けられた CICS システム・グループ	LNKSWSCG	<ul style="list-style-type: none"> <li>• ワークロード仕様と CICS システム・グループとの間の関連を表示、作成、または削除します。</li> </ul>
ワークロード仕様に関連付けられた CICS システム	LNKSWSCS	<ul style="list-style-type: none"> <li>• ワークロード仕様と CICS システムとの間のリンクを表示、作成、更新、または削除します。</li> <li>• 指定された定義を開始点として使用してワークロード定義のマップを表示します。</li> </ul>
トランザクション・グループ (Transaction groups)	TRANGRP	<ul style="list-style-type: none"> <li>• トランザクション・グループを表示、作成、更新、または削除します。</li> <li>• トランザクションをトランザクション・グループに追加します。</li> <li>• インストール</li> <li>• 指定されたトランザクション・グループを開始点として使用してワークロード定義のマップを表示します。</li> </ul>
ワークロード定義 (Workload definitions)	WLMDEF	<ul style="list-style-type: none"> <li>• ワークロード定義を作成、変更、または削除します。</li> <li>• ワークロード定義とワークロード・グループとの間の関連を追加します。</li> <li>• ワークロード定義をワークロードにインストールします。</li> </ul>

表 6. ワークロード管理定義を作成および保守するためのビュー (続き)

WUI ビュー	オブジェクト名	サポートされるタスク
ワークロード・グループ (Workload groups)	WLMGROUP	<ul style="list-style-type: none"> <li>ワークロード・グループを表示、作成、更新、または削除します。</li> <li>ワークロード・グループとワークロード仕様との間の関連を追加します。</li> <li>ワークロード・グループに関連付けられたワークロード定義をワークロードにインストールします。</li> <li>指定されたワークロード・グループを開始点として使用してワークロード定義のマップを表示します。</li> </ul>
グループ内のワークロード定義 (Workload definitions in groups)	WLMINGRP	<ul style="list-style-type: none"> <li>ワークロード定義とワークロード・グループとの間の関連を追加または削除します。</li> </ul>
ワークロード仕様 (Workload specifications)	WLMSPEC	<ul style="list-style-type: none"> <li>ワークロード仕様を表示、作成、更新、または削除します。</li> <li>ワークロード仕様と CICS システムの間の関連を追加します。</li> <li>ワークロード仕様と CICS システム・グループの間の関連を追加します。</li> <li>指定されたワークロード仕様を開始点として使用してワークロード定義のマップを表示します。</li> </ul>
ワークロード仕様内のワークロード・グループ (Workload groups in workload specifications)	WLMINSPC	<ul style="list-style-type: none"> <li>ワークロード・グループとワークロード仕様との間の関連付けを削除します。</li> <li>指定された定義を開始点として使用してワークロード定義のマップを表示します。</li> </ul>

75 ページの表 7 は、アクティブ・ワークロードの管理に使用できるビューを示しています。また、表示できる情報やこれらのビューを使用して実行できるアクションも示しています。Web ユーザー・インターフェースでこれらのビューにアクセスするには、メインメニューから「**アクティブ・ワークロード (WLM)**」をクリックします。

表 7. アクティブ・ワークロードを管理するためのビュー

WUI ビュー	オブジェクト名	サポートされるタスク
アクティブ・ワークロード	WLMAWORK	<ul style="list-style-type: none"> <li>現行コンテキスト内のアクティブ・ワークロードを表示します。</li> </ul>
アクティブ・ワークロード内のルーティング領域 (Routing regions in an active workload)	WLMAWTOR	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたアクティブ・ルーティング領域を表示します。</li> </ul>

表 7. アクティブ・ワークロードを管理するためのビュー (続き)

WUI ビュー	オブジェクト名	サポートされるタスク
アクティブ・ワークロードのターゲット配布係数	WLMAWAOR	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたターゲット領域を表示します。</li> <li>アクティブ・ワークロードに関連付けられたターゲット領域をアクティブにします。</li> <li>アクティブ・ワークロードに関連付けられたターゲット領域を静止します。</li> </ul>
アクティブ・ワークロード定義	WLMAWDEF	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたアクティブ・ワークロード定義を表示します。</li> <li>アクティブなワークロード定義を破棄します。</li> </ul>
アクティブ・ワークロードのトランザクション・グループ	WLMATGRP	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたアクティブ・トランザクション・グループを表示します。</li> <li>トランザクション・グループの状況をアクティブに変更します。</li> <li>アクティブ・トランザクション・グループの状況を休止に変更します。</li> </ul>
アクティブ・ワークロードの動的トランザクション	WLMATRAN	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたアクティブ・トランザクションを表示します。</li> <li>トランザクション・グループからトランザクションを破棄します。</li> </ul>
アクティブ・ワークロードのトランザクション・グループの親和性	WLMATAFF	<ul style="list-style-type: none"> <li>ワークロードに関連付けられたトランザクション・グループのアクティブな親和性を表示します。</li> <li>親和性のエンティティを破棄します。</li> </ul>

## ワークロード管理定義の作成

このセクションでは、WUI または CICS Explorer を使用してワークロード管理定義を作成および管理する手順について説明します。特に断りのない限り、ワークロード管理定義を作成および保守している時に認識されるのはコンテキストのみです。

## ワークロード仕様の作成

WLM 仕様は、1 つのワークロードと、ターゲット領域として作動する 1 つ以上の CICS システムを指定します。この仕様は、デフォルト・トランザクション・グループの属性も定義します。「**ワークロード仕様**」ビューを使用してワークロード仕様を作成できます。

### このタスクについて

CICSplex SM WUI または CICS Explorer を使用して、ワークロード仕様を作成し、データ・リポジトリに追加します。



### CICS Explorer ユーザーの場合:

CICS Explorer 製品資料内の『ワークロード仕様の作成』の手順に従ってください。

### CICSplex SM WUI ユーザーの場合:

#### 手順

1. WUI メインメニューから「管理」>「ワークロード・マネージャー管理」>「仕様」をクリックして、「WLM 仕様」の表形式ビューを開きます。

このビューに、既存のワークロード仕様のリストが表示されます。このビューにはアクション・ボタンがあり、それらを使用して、ワークロード仕様を作成、更新、および削除し、CICS システムおよび CICS システム・グループをワークロード仕様に関連付けることができます。

2. オプション: 新しい定義を作成する時に、既存の定義の情報をいくつか使用するには、「レコード」列内の隣接するチェック・ボックスを選択して既存の定義を選択します。
3. 「作成 (Create)」をクリックします。
4. 各フィールドに入力し、「はい」をクリックして新しい仕様を作成します。それ以外の場合は、「いいえ」をクリックして処理を中止します。

「ワークロード仕様」ビューが再表示され、新規定義の項目が現れます。

#### ワークロード仕様への、CICS システムまたはシステム・グループの関連付け

各ワークロード仕様には、デフォルトのターゲット・スコープがあり、1 つ以上の CICS システムまたは CICS システム・グループが関連付けられます。

これらの関連付けられた CICS システムまたはシステム・グループは、ルーティング・スコープと呼ばれます。デフォルト・ターゲット・スコープは、仕様を作成する際に識別されます。関連付けられた CICS システムおよびシステム・グループは、ルーティング領域を仕様に追加する際に識別されます。

CICS システム・グループを既存のワークロード仕様に関連付けるには、以下の手順に従ってください。

1. 「管理」>「ワークロード・マネージャー管理」->「仕様」をクリックして、「WLM 仕様」ビューを開きます。

このビューに既存のワークロード仕様のリストが表示されます。

2. ワークロード仕様を選択し、「CICS グループの関連付け...」ボタンをクリックします。これにより、「CICS グループの関連付け...」ビューが開きます。
3. 「CICS システム・グループ」フィールドに、1 つ以上のルーティング領域を表す既存の CICS システム・グループの固有名または総称名を入力します。

**注:** ルーティング領域として機能している CICS システムまたは CICS システム・グループは、一度に 1 つのワークロード仕様にしか関連付けることができません。ただし、仕様は、任意の数の CICS システムおよび CICS システム・グループに関連付けることができます。

4. 以下のいずれかのオプションを選択して、CICS システム・グループを形成する CICS システムが、CICS システム・グループに関連付けられているワークロード仕様をどのように使用するかを示します。

#### FORCE

CICS システム・グループに現在関連付けられているすべての CICS システムが、このワークロード仕様を使用します。(それぞれの CICS システムのワークロード仕様属性は INHERIT に変更されます。INHERIT は、この仕様が CICS システム・グループから獲得されたものであることを示しています)。新しい CICS システムを CICS システム・グループに追加した場合、そのシステムは、自動的に FORCE オプションを継承しません。CICS システムを CICS システム・グループに追加する際に NULL または FORCE を指定する必要があります。

#### NULL

ワークロード仕様に関連付けられていない、CICS システム・グループ内の CICS システムは、このワークロード仕様を使用します。(これらの CICS システムのワークロード仕様属性は INHERIT に変更されます。これは、この仕様が CICS システム・グループから獲得されたものであることを示しています。)

なし

CICS システム・グループのみがワークロード仕様に関連付けられます。CICS システム・グループ内の CICS システムは影響を受けません。つまり、CICS システムにワークロード仕様が関連付けられている場合、何も設定されません。明示的に設定されたものであれ、別の CICS システム・グループから継承されたものであれ、モニター仕様がシステムに関連付けられている場合、その関連付けは変更されません。

注:

- a. CICS システム・グループに他の CICS システム・グループが含まれる場合、従属 CICS システム・グループにあるものを含むすべての CICS システムが、このフィールドで指定した値の影響を受けます。
  - b. ワークロード仕様の作成時にデフォルト・ターゲット・スコープとして指定された CICS システムは、ルーティング・スコープとして識別される CICS システムからアクセス可能でなければなりません。したがって、それぞれのルーティング領域と、ここで識別されるターゲット・スコープ内のすべての CICS システムとの間に、適切な CICS 接続が存在することを確認する必要があります。
5. データ・リポジトリ内のワークロード仕様を更新するには、「はい」をクリックします。それ以外の場合は、「いいえ」をクリックして処理を中止します。

CICS システムまたはシステム・グループをワークロード仕様に関連付けると、関連付けられた CICS システムの始動時にはいつでも、その仕様が自動的にインストールされます。ワークロード・グループを介してその仕様に関連付けられているワークロード定義も自動的にインストールされます。

ただし、既にアクティブな CICS システムにワークロード仕様を関連付けた場合、新しい仕様はすぐには有効になりません。ワークロード管理をオンにするには、以下の手順を実行します。

1. 「管理」 > 「トポロジー管理」 > 「システム定義」をクリックして、アクティブな CICS システムのリストを表示します。
2. 目的の CICS システムを選択し、「更新...」ボタンをクリックします。
3. 「ワークロード・マネージャー状況」フィールドに「YES」を指定し、「はい」ボタンをクリックしてワークロード管理をオンにします。それ以外の場合は、「いいえ」をクリックして処理を中止します。

## WLM 仕様と CICS システムとの間のリンクの更新

### 手順

1. メインメニューで、「管理」 > 「ワークロード・マネージャー管理」をクリックします。
2. 「ワークロード・マネージャー管理ビュー」メニューで、「仕様とシステムとの間のリンク」をクリックします。  
「WLM 仕様と CICS システムとの間のリンク」ビュー (LNKSRSCS オブジェクト) が表示されます。
3. 更新する WLM 仕様のボックスにチェック・マークを付けます。  
複数のボックスにチェック・マークを付けることができます。
4. 「仕様の関連付けの変更」ボタンをクリックして「仕様の関連付けの変更 (EYUSTARTLNKSWSCS.CHGSPEC)」ビューを表示します。
5. 「新規 WLM 仕様名」フィールドに新しい名前を入力し、適切な継承オプションを選択します。
  - ・ モニター仕様と CICS システム・グループの間のリンクを更新する場合は、「はい」をクリックします。
  - ・ 処理を中止する場合は、「いいえ」をクリックします。

## WLM 仕様と CICS システム・グループとの間のリンクの更新

### 手順

1. メインメニューで、「管理」 > 「ワークロード・マネージャー管理」をクリックします。
2. 「ワークロード・マネージャー管理ビュー」メニューで、「仕様とシステム・グループとの間のリンク」をクリックします。

「WLM 仕様と CICS システムとの間のリンク」ビュー (LNKSWSCG オブジェクト) が表示されます。  
LNKSxSCG パラメーターについて詳しくは、[LNKSxSCG Records \(LNKSMSCG, LNKSRSRG, LNKSWSCG\)](#) を参照してください。

3. 更新する WLM 仕様のボックスにチェック・マークを付けます。  
複数のボックスにチェック・マークを付けることができます。
4. 「仕様の関連付けの変更」ボタンをクリックして「仕様の関連付けの変更 (EYUSTARTLNKSWSCG.CHGSPEC)」ビューを表示します。
5. 「新規 WLM 仕様名」フィールドに新しい名前を入力し、適切な継承オプションを選択します。
  - ・ モニター仕様と CICS システム・グループの間のリンクを更新する場合は、「はい」をクリックします。
  - ・ 処理を中止する場合は、「いいえ」をクリックします。

## トランザクション・グループの作成

トランザクション・グループは、論理的に類似したトランザクションを関連付けたものです。この類似性は、ワークロード管理 (WLM) または親和性の要件、共通の共用処理要件、またはユーザーが決定する他の何らかの特性に基づくものとすることができます。「トランザクション・グループ定義」ビューを使用して、トランザクション・グループを作成できます。

### このタスクについて

ワークロード管理の場合、トランザクション・グループで WLM 情報 (ルーティング・アルゴリズム・タイプを含む) を指定すると、関連するワークロード仕様 (WLMSPEC) で提供されるデフォルト値がオーバーライドされます。トランザクション・グループに代替 WLM 属性を指定すると、ルーティング領域を停止することなく、関連するトランザクションのワークロード・ルーティング特性を動的に変更できます。

CICSplex SM WUI または CICS Explorer を使用して、トランザクション・グループを作成できます。

#### CICS Explorer ユーザーの場合:

手順については、[CICS Explorer の製品資料](#)を参照してください。

#### CICSplex SM WUI ユーザーの場合:

### 手順

1. WUI メインメニューから「管理」>「ワークロード・マネージャー管理」>「トランザクション・グループ定義」の順をクリックして、「トランザクション・グループ定義」ビューを開きます。  
このビューに既存のトランザクション・グループ定義のリストが表示されます。トランザクション・グループ定義の作成、更新、および削除と、トランザクション・グループへのトランザクションの追加を行うことができます。
2. オプション: 新しい定義を作成する時に既存の定義の情報を使用するには、必要な既存の定義の横にある「レコード」列内のチェック・ボックスを選択します。
3. 「作成 (Create)」をクリックします。
4. トランザクション・グループ定義を作成するための適切な情報を指定します。  
このビュー内のフィールドの説明については、[トランザクション・グループ定義 - TRANGRP](#) を参照してください。
5. 「はい」をクリックして新しい定義を作成し、データ・リポジトリに追加します。  
新しい定義のエントリーを含む「トランザクション・グループ定義」ビューが再表示されます。

### 次のタスク

インストールされているトランザクション・グループの WLM 情報を変更する場合は、関連付けられている WLM 定義 (WLMDEF) を破棄してから再インストールする必要があります。それにより、WLM 定義によって指定されるトランザクション・グループも更新されます。ただし、ルーティング・アルゴリズム・タイプ (ALGTYPE 属性) を変更する場合は、「アクティブ・ワークロード・トランザクション・グループ (WLMATGRP)」ビューと SET コマンドを使用することによって、関連付けられている WLMDEF を破棄して再インストールすることなく、すぐに変更することができます。

## ワークロード定義の作成

ワークロード定義を使用して、作業要求を特定のターゲット領域セットに経路指定することができます。この経路指定は、該当の作業要求に関連付けられている端末名とユーザー名、またはプロセス・タイプに基づいて行われます。このセクションでは、ワークロード定義を作成し、データ・リポジトリを追加する方法について説明します。

端末名およびユーザー名は、固有名の場合もあれば、総称名の場合もあります。例えば、NET で始まる論理装置名を持つ端末から任意のユーザーによって開始されたトランザクションすべてが EYUCSG01 という名前で識別されるターゲット・スコープに経路指定されるワークロード定義を作成できます。

トランザクションを経路指定すべきかどうかを判断するために CICSplex SM が使用するトランザクション突き合わせ基準は、以下のとおりです。

- トランザクション ID は、ワークロード定義に関連付けられたトランザクション・グループの一部か？
- プロセス・タイプで分離を指定する定義があるか？ 一致はあるか？ ある場合、この定義を使用する。ない場合、端末の論理装置名およびユーザー ID で分離を指定する定義はあるか？
- トランザクションに関連付けられた端末名とユーザー名をどの順序で評価するか？ つまり、どちらの名前を 1 次フィルターとして使用するか？
- トランザクションに関連付けられたユーザー名と端末名は、ワークロード定義によって指定された名前パターンに一致するか？

端末とユーザーの名前パターンを確立するためのルールは、以下のとおりです。

- 複数のワークロード定義がある場合、1 次フィルターとして識別される名前を含むフィールドは、同じ固有または総称のパターンを含むことができます。2 次フィルターとして使用されるフィールドの内容は常に固有でなければなりません。例えば、以下の定義は、USERID が 1 次フィルターで LUNAME が 2 次フィルターの場合は有効です。しかし、LUNAME が 1 次フィルター場合、ユーザー ID が固有でないため、以下の定義は無効です。

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

- 総称名が指定されている場合は、一致する先行文字の数が最も多いパターンが優先されます。例えば、以下のパターンで考えてみます。

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

パターン A37AR\* は、入力が A37AR123 の時に選択されます。パターン A37+R\* は、入力が A37TRAP の時に選択されます。

- 端末 LU 名パターンは、ネットワーク名と論理装置名の連結値で突き合わされます。以下は、有効な端末 LU 名パターンです。

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

ワークロード定義を作成し、データ・リポジトリに追加するには、以下の手順を実行します。

- 「管理」 > 「ワークロード・マネージャー管理」 > 「定義」をクリックして、「ワークロード管理定義」ビューを開きます。

このビューに既存のワークロード管理定義のリストが表示されます。このビューにはアクション・ボタンがあり、それらを使用してワークロード定義の作成、更新、削除、インストール、およびワークロード・グループへの追加を行うことができます。

- 新しい定義を作成する時に既存の定義の情報をいくつか使用する場合は、「レコード」列内の隣接するチェック・ボックスを選択して既存の定義を選択します。
- 「作成」アクション・ボタンをクリックします。

4. 必要に応じて、以下の情報を指定します。

#### ワークロード管理定義名

ワークロード定義の名前を 1 から 8 文字までで指定します。名前には英字、数字、国別文字を含めることができます。ただし、先頭文字は英字でなければなりません。

#### 説明

(オプション) 1 文字から 30 文字で定義の説明を指定します。

#### トランザクション・グループ名

(オプション) トランザクション・グループの固有名または総称名を入力します。総称値を入力すると、有効なトランザクション・グループのリストが表示されます。

トランザクション・グループを特定しない場合は、仕様のデフォルト・トランザクション・グループが想定されます。

#### 端末 LU 名

次のフォーマットを使用して、固有の論理装置名またはパターンを入力します。

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

ここで、network\_qualifier\_name と luname は、両方とも 1 文字から 8 文字の長さです。network\_qualifier\_name と luname はオプションですが、いずれかを指定する必要があります。luname を指定する場合は、ピリオド記号を接頭部として付加する必要があります。

パターンには、文字 + (正符号) または \* (アスタリスク)、あるいはその両方を含めることができます。ここで、正符号は、任意の非ブランク文字を表し、パターン内の任意の場所に指定できます。アスタリスクは、ブランクを含め、ゼロ個以上の文字の範囲を表します。アスタリスクは、パターンの終わりにのみ使用する必要があります。

#### ユーザー ID

固有のユーザー ID またはパターンを入力します。パターンには文字 + (正符号) または \* (アスタリスク)、あるいはその両方を含めることができます。

**注:** 端末 LU 名とユーザー ID の値は、CICSplex SM が、トランザクションをこの定義と突き合わせようとしている時に使用できます。

#### BTS プロセス・タイプ

固有のプロセス・タイプまたはパターンを入力します。パターンには、+ (正符号) または \* (アスタリスク)、あるいはその両方を含めることができます。

**注:** このフィールドに \* 以外のものを指定する場合、LU 名とユーザー ID の各フィールドは \* に設定する必要があります。同様に、LU 名およびユーザー ID の各フィールドに \* 以外のものがある場合、このフィールドは \* に設定する必要があります。プロセス・タイプ、LU 名、およびユーザー ID でワークロードを分離することはできません。

#### ターゲット・システム・セットのスコープ名

動的ルーティング要求のターゲット領域として使用する CICS システムまたは CICS システム・グループの固有名または総称名を入力します。総称値を入力すると、有効な CICS システムと CICS システム・グループのリストが表示されます。

**注:** 指定された CICS システムまたは CICS システム・グループは、この定義が関連付けられているワークロード仕様によってルーティング・スコープと識別される CICS システムからアクセス可能である必要があります。したがって、それぞれのルーティング領域と、ここでターゲット領域として識別されるすべての CICS システムとの間に、適切な CICS 接続が存在していることを確認する必要があります。

5. 「はい」をクリックして新しい新規定義を作成します。

「ワークロード管理定義」ビューが再表示されて、新規定義の項目が現れます。

## ワークロード・グループの作成

ワークロード・グループは、1つ以上の関連したワークロード定義を関連付けるために使用します。

このビューの使用例は、89 ページの『特定のターゲット領域への特定のトランザクションのルーティング』に記載されています。このセクションでは、ワークロード・グループ定義を作成し、データ・リポジトリに追加する方法について説明します。

- 「管理ビュー」 > 「ワークロード・マネージャー管理ビュー」 > 「グループ」をクリックして、「WLM グループ」ビューを開きます。

このビューに既存のワークロード管理グループ定義のリストが表示されます。このビューにはアクション・ボタンがあり、それらを使用してワークロード・グループの作成、更新、削除、インストール、およびワークロード仕様へのグループの追加を行うことができます。

- 新しい定義を作成する時に既存の定義の情報をいくつかを使用する場合は、「レコード」列内の隣接するチェック・ボックスを選択して既存の定義を選択します。
- 「作成」アクション・ボタンをクリックします。
- 必要に応じて、以下の情報を指定します。

### ワークロード管理グループ名

ワークロード・グループの名前を 1 から 8 文字までで指定します。名前には英字、数字、国別文字を含めることができます。ただし、先頭文字は英字でなければなりません。

### 説明

(オプション。) ワークロード・グループの説明を 1 から 30 文字までで指定します。

- 「はい」をクリックして新しい新規定義を作成します。

「ワークロード管理グループ」ビューが再表示されて、新規定義の項目が現れます。

## ワークロード・ビューのルート・フィールドについて

「アクティブ・ワークロードのターゲット配布係数」ビューと「アクティブ・ターゲット領域の配布傾向の集計」ビューを、アクティブ・ワークロード・ビューに表示されるルート・フィールド統計と共に使用して、ワークロード管理で起きていることを理解します。

「アクティブ・ターゲット領域の配布傾向の集計」ビューのルート・フィールド統計情報を使用して、既存のワークロードが正しく実行されており、トランザクションを消失していないことを確認できます。「アクティブ・ターゲット領域の配布傾向の集計」ビューには、ルーティング・プロセス中の WLM 内のイベントに関する新しいパースペクティブも提供されます。

「アクティブ・ワークロード・ビュー」を使用することにより、セットアップの潜在的な問題を強調することができます。例えば、ルーティングが呼び出され、他の使用可能なターゲットの検出を再試行し続けている場合などです。

WLM 内で起きていることのカウントを確認できます。例えば、Route complete の数対 Route select の数などです。この例としては Route notify フィールドがあります。このフィールドは、**DYNAMIC(YES)** として定義されているプログラムにリンクしており、SYSID が指定されている場合に表示されます。

「アクティブ・ターゲット領域の配布傾向の集計」ビューを「アクティブ・ワークロードのターゲット配布係数」ビューと共に使用すると、WLM のルーティングの決定と CICSplex SM の動作を理解するのに役立ちます。例えば、一部の領域が他の領域より多くの作業を受け取っている理由を知りたい場合、Route Selections の分析により、どの領域が他の領域より多くの作業を受け取っているかが示されます。

83 ページの図 10 は、分散ルーティングに関連するルート・フィールド・プロセスを示しています。



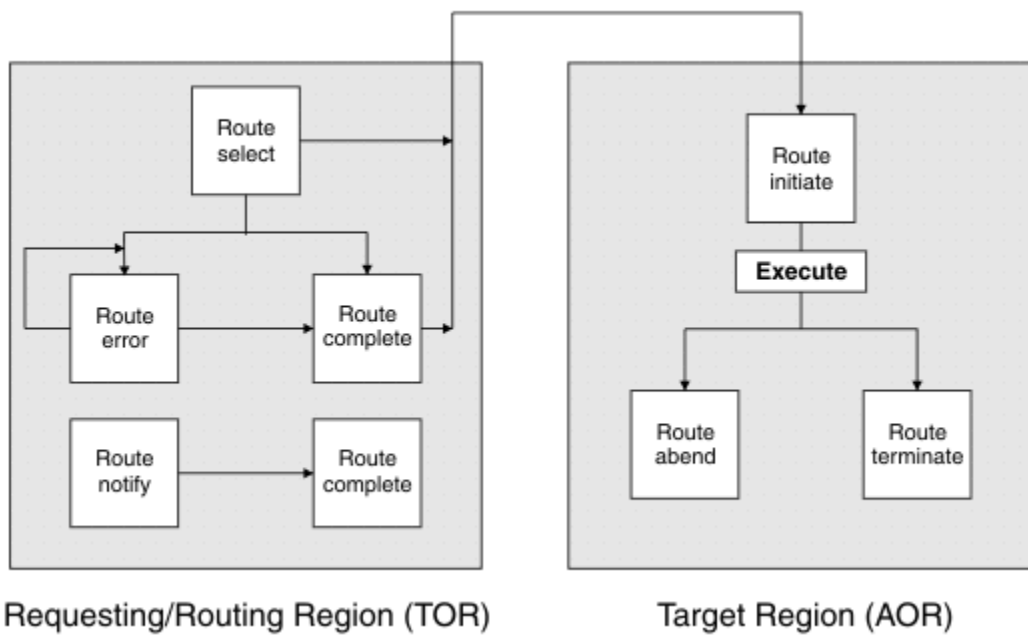


図 10. 分散ルーティングの概念ビュー

83 ページの図 11 は、動的ルーティングに関連するルート・フィールド・プロセスを示しています。

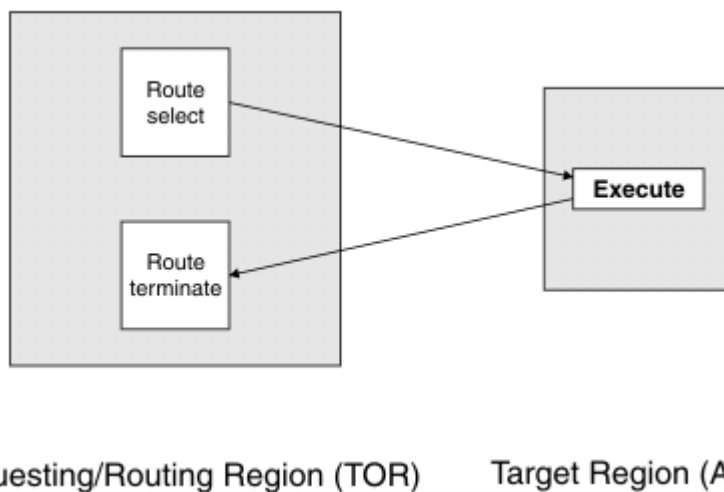


図 11. 動的ルーティングの概念ビュー

これらの定義は、以下のようにルート・フィールドを記述しています。

#### Route select

Route select は、CICSplex SM ワークロード管理によってこの領域がターゲットとして選択された時に更新されます。

#### Route initiate

Route initiate は、分散ワークロード・モデルを使用してルーティングされたタスクがターゲット・システムで開始された時に更新されます。

#### Route error

Route error は、CICSplex SM によって選択された領域が到達不能な時に更新されます。例えば、接続サービスが休止していたり、セッションが使用可能でなかったりする場合があります。この場合、CICSplex SM は別の領域の選択を試行します。この再選択は Route select としてカウントされず、後続の領域の Route select カウントは更新されません。route error の数が多い場合は、ワークロードを

実行するセッションの数が不足していることを示しています。セッションが不足していると、CICSplex SM が再度 (可能性として複数回) ルーティングを試行しなければならないため、応答時間と CPU 時間の増加の原因になる可能性があります。

#### Route complete

Route complete は、分散ワークロード・モデルを使用してルーティングされるタスクがターゲット・システムで開始された時に更新されます。Route complete は、このタスクの責任が完了したことをルーティング・システムに示します。

#### Route notify

Route notify は、LINK コマンドまたは START コマンドの SYSID オプションによって選択されたターゲット領域にタスクがルーティングされた時に更新されます。Route notify は、CICSplex SM にルーティングを通知します。CICSplex SM は領域を選択しません。Route notify は、静的経路指定の変形です。静的経路指定では、プログラムまたはトランザクションは動的と定義されますが、それを呼び出す API は SYSID を指定します。

#### Route abend

Route abend は、タスクがターゲット領域に経路指定され、その後、トランザクションがシステムに定義されていない、またはプログラムが使用不可であるなどの理由により異常終了した時に更新されます。

#### Route terminate

Route terminate は、ターゲット領域でタスクが完了した時に更新されます。分散プログラム・リンク (DPL) 要求または動的トランザクション・ルーティング (DTR) 要求の終了時であるか、分散ワークロード・モデルを使用して経路指定されたタスクの終了時であるかは関係ありません。

## CICSplex SM ワークロード管理タスクの例

ワークロード管理のすべての例は、同じ構成に基づいています。CICSplex、PLXPROD1 は、1 つの TOR (CICSPT01) と 3 つの AOR (CICSPA01、CICSPA02、および CICSPA03) から成ります。これらの CICS 領域が果たす役割は、個々のタスクの性質に応じて異なります。

以下のセクションには、多くの標準的ワークロード管理のタスク例が含まれており、Web ユーザー・インターフェース (WUI) を使用してこれらのタスクをどのように実行できるかが示されています。

どのタスクについても、作業している CICS システムの範囲を認識する必要があります。範囲が単一の CICS システムの場合、CICSplex SM から取得したすべてのデータはその単一システムに関連しています。範囲が CICS システムのグループの場合、データは、そのグループ内のすべてのシステムに関連しています。範囲が CICSplex の場合、データは、その CICSplex 内のすべてのシステムに関連しています。以下のセクション内のすべての例で、初期範囲は CICSplex PLXPROD1 になります。

#### 始める前に

最適化ワークロード管理をサポートするには、CICS 領域状況データを迅速に共有するように CICSplex 内で CICS 領域状況サーバーがセットアップされ、実行されていなければなりません。詳しい説明は、[領域状況サーバーのセットアップと稼働](#)を参照してください。

## ワークロードの管理

この例では、Web ユーザー・インターフェース (WUI) を使用した CICSplex SM でのワークロードの管理方法、つまり、すべての作業を 1 つのルーティング領域から適切なターゲット領域に経路指定する方法について説明します。

このタスクで「適切な」ターゲット領域とは、ターゲット領域で許可されるタスクの最大数と比較して、作業要求のキューが最も短い領域になります。この作業キュー (ロード・カウントとも呼ばれる) は、キューに入れられたすべてのアクティブ・ユーザー・タスクの数です。デフォルトでは、MAXTASKS および TRANCLASS 両方に関してキューに入れられたタスクがロード・カウントに含まれます。CSYSDEF リソース・テーブルで「タスク・ロード・キュー・モード」属性を使用すると、TRANCLASS に関してキューに入れられたタスクをサイトでロード・カウントから除外することができます。CSYSDEF リソース・テーブルの「タスク・ロード・キュー・モード」属性の詳細については、[を参照してください](#)。

TOR (CICSPT01) は、要求側領域およびルーティング領域であり、3 つの AOR (CICSPA01、CICSPA02、および CICSPA03) はすべてターゲット領域です。どの CICS 領域も現在実行されていません。

一部の作業要求を自由に経路指定できない場合 (例えば、特定のユーザーからの作業要求を常に同じターゲット領域に送信したい場合など)、心配ありません。その要件の追加方法については、後のタスク例で説明します (89 ページの『特定のターゲット領域への特定のトランザクションのルーティング』)。

1. 作業要求 (「ワークロード」) を動的に経路指定できるすべてのターゲット領域を含む CICS システム・グループを作成します。

- メインメニューから「管理」>「トポロジー管理」>「システム・グループ」をクリックして、「システム・グループ定義」表形式ビューを開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

- 「作成」をクリックして、「システム・グループ定義」作成ビューを開きます。
- 以下のように入力します。

システム・グループ名	CSGTGTS1
説明	All target regions in CICSplex PLXPROD1

- 「はい」をクリックして、システム・グループを作成します。

「システム・グループ定義」表形式ビューが再表示され、CICS システム・グループ CSGTGTS1 のエントリーが示されます。

この時点では、グループ CSGTGTS1 は存在しますが、中身は空です。

2. CICSplex SM に、このグループにどのターゲット領域が属するかを指定します。CICS システム・グループ CSGTGTS1 にターゲット領域を追加するには、以下の手順を実行します。

- メインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックして、「CICS システム定義」表形式ビューを開きます。
- CICS システムの CICSPA01、CICSPA02 および CICSPA03 のエントリーを選択し、「CICS システム・グループへの追加」をクリックします。
- 「メンバーが加わるグループ」フィールドに CSGTGTS1 と入力し、「はい (残りの 3 個もすべて含む)」をクリックします。これにより、選択した 3 つのシステムが新しいグループに追加されます。

3. ワークロード仕様を作成します。

- メインメニューから「管理」>「ワークロード・マネージャー管理」>「仕様」をクリックして、「WLM 仕様」表形式ビューを開きます。
- 「作成」をクリックして、以下の情報を入力します。

WLM 仕様名	WLSPAY01
説明	Manage payroll workload in PLXPROD1
主検索基準	USERID
デフォルト・ターゲット・スコープ	CSGTGTS1
アルゴリズムのタイプ	QUEUE

残りのフィールドは空のままにするか、デフォルトを受け入れます。

注:

- この例では、主検索基準値 USERID が示されていますが、実際には、主検索基準値は単純なワークロード検索には影響しないため、USERID を指定するか LUNAME を指定するかは問題ではありません。これは、ある種のワークロード分離にのみ使用されます。それについては、後の例で説明します。CICSplex SM は、このワークロード仕様がワークロード分離に使用されないことをこの段階では認識していないため、値を指定する必要があります。
- デフォルト・ターゲット・スコープ値は、作業要求を経路指定できる単一ターゲット領域、またはターゲット領域グループの名前です。

- **アルゴリズム・タイプ**値 **QUEUE** は、作業要求のキューが最も短いターゲット領域を選択するよう CICSplex SM に指示します。
- 「**説明**」はオプションですが、指定することをお勧めします。(指定しておく、仕様名のリストが示されて、見分けがつかない時に役立ちます。)

「**はい (Yes)**」をクリックします。「**WLM 仕様**」表形式ビューが再表示され、今回は新しい WLM 仕様 **WLSPAY01** のエントリーが示されています。

4. グループ **CSGTGTS1** 内のターゲット領域に作業要求をルーティングする領域について、CICSplex SM に指定します。ワークロード仕様をルーティング領域に関連付けるには、以下の手順を実行します。
  - 「**WLM 仕様**」ビューで **WLSPAY01** 仕様のエントリーを選択し、「**CICS システムの関連付け**」をクリックします。
  - 「**CICS システム**」フィールドに、ルーティング領域の名前 (この例では **CICSPT01**) を入力し、「**はい**」をクリックします。

ルーティング領域とワークロード仕様との関連付けが機能しているかどうか確認するには、以下の手順を実行します。

- 「**WLM 仕様**」表形式ビューで名前「**WLSPAY01**」をクリックして、この仕様の詳細を表示します。
- 「**このワークロード仕様に関連する CICS システム**」リンクをクリックして、「**CICS システムから WLM 仕様へのリンク**」表形式ビュー (LNKSWSCS オブジェクト) を表示します。

注: このビューには、「**管理**」>「**ワークロード・マネージャー管理**」>「**仕様とシステムとの間のリンク**」をクリックして、メインメニューからアクセスすることもできます。

5. 以下の手順を実行して、ルーティング領域でワークロード管理をアクティブにします。
  - メインメニューから「**管理**」>「**トポロジー管理**」>「**システム定義**」をクリックして、「**CICS システム定義**」表形式ビューを開きます。
  - **CICSPT01** のエントリーを選択し、「**更新**」をクリックして、**CICSPT01** の詳細ビューを開きます。
  - 「**AOR 動的ルーティング・モード**」フィールドを「**はい**」に変更します。「**はい**」をクリックして、「**CICS システム定義**」表形式ビューに戻ります。

この変更は、ターゲット領域 **CICSPT01** を次回開始した時に有効になります。

6. 以下の手順を実行して、ターゲット領域でワークロード管理をアクティブにします。
  - 「**CICS システム定義**」表形式ビューから **CICSPA01** のエントリーを選択し、「**更新**」をクリックして **CICSPA01** の詳細ビューを開きます。
  - 「**ワークロード・マネージャー状況**」フィールドを「**YES**」に変更します。「**はい**」をクリックして、「**CICS システム定義**」表形式ビューに戻ります。

この変更は、ターゲット領域 **CICSPT01** を次回開始した時に有効になります。

ターゲット領域の **CICSPA02** および **CICSPA03** について、このステップを繰り返します。

ルーティング領域とターゲット領域が開始されると、「**アクティブ・ワークロード (WLM)**」>「**アクティブ・ワークロード**」をクリックして、ワークロードがアクティブであることをメインメニューから確認できます。これにより、「**アクティブ・ワークロード**」表形式ビューが開き、ワークロード仕様 **WLSPAY01** がアクティブであることが示されます。

どのターゲット領域にルーティングされているか確認するには、「**アクティブ・ワークロード (WLM)**」>「**アクティブ・ワークロードのターゲット配布係数**」をクリックします。表示されたビューに、ワークロード **WLSPAY01** によるルーティング先であるすべてのアクティブ・ターゲット領域が示されます。リストされているすべてのターゲット領域は、CICS システム・グループ **CSGTGTS1** に属しています。

## 既存のターゲット領域スコープへの領域の追加

この例では、Web ユーザー・インターフェース (WUI) を使用して、既存のワークロードを中断せずに、既存のターゲット・スコープ内の領域数を増加する方法について説明します。

ワークロード仕様 **WLSPAY01** を介して CICSplex **PLXPROD1** でワークロード・ルーティングを実装しており、作業要求がターゲット領域 **CICSPA01**、**CICSPA02**、および **CICSPA03** に経路指定されていると想定します。これらのターゲット領域はすべて CICS システム・グループ **CSGTGTS1** に属しています。そして今、

4 番目の領域 CICSPA04 をグループ CSGTGTS1 に追加しようとしています。領域 CICSPA04 は CICSplex PLXPROD1 に定義されていて、ルーティング領域 CICSPT01 を実行し、このルーティング領域にリンクされています。

1. 以下の手順を実行して、ターゲット領域 CICSPA04 を CICS システム・グループ CSGTGTS1 に追加します。
  - a. Web ユーザー・インターフェースのメインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックして、「**CICS システム定義**」表形式ビューを開きます。  
現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - b. CICSPA04 のエントリーを選択して「**CICS システム・グループへの追加**」ボタンをクリックします。「**CICS システム・グループへの追加**」ビューが表示されます。
  - c. 「メンバーが加わるグループ」フィールドに名前 CSGTGTS1 を入力し、「はい」をクリックします。
2. 以下の手順を実行して、ターゲット領域 CICSPA04 の CICSplex SM 定義を更新します。
  - a. 「**CICS システム定義**」表形式ビューから CICSPA04 のエントリーを選択し、「更新」をクリックします。
  - b. メニューから、「ワークロード・マネージャー状況」フィールドを「はい」に変更します。「はい」をクリックして、更新を確認します。

この変更は、ターゲット領域 CICSPA04 を次回開始した時に有効になります。

ターゲット領域 CICSPA04 が開始されたら、「アクティブ・ワークロード (WLM)」>「アクティブ・ワークロードのターゲット配布係数」をクリックして、ワークロードがアクティブであることを確認できます。これにより、「アクティブ・ワークロードのターゲット配布係数」ビューが開きます。このビューには、このワークロード内の作業要求を経路指定できるすべてのターゲット領域 (CICSPA04 を含む) が表示されます。

## ターゲット領域スコープからの領域の除去

CICS システム・グループ CSGTGTS1 は、CICSplex PLXPROD1 内に 4 つのターゲット領域 (CICSPA01、CICSPA02、CICSPA03、および CICSPA04) を含んでいます。作業要求はこれらのターゲット領域にルーティングされ、ルーティングはワークロード仕様 WLSPAY01 によって制御されます。

この例では、Web ユーザー・インターフェース (WUI) を使用して、アクティブ・ワークロードを中断せずに、グループ CSGTGTS1 から領域 CICSPA04 を除去する方法について説明します。

1. CICS システム・グループ CSGTGTS1 からターゲット領域 CICSPA04 を除去するには、以下の手順を実行します。
  - Web ユーザー・インターフェースのメインメニューから「管理」>「トポロジー管理」>「システム・グループ」をクリックして、「システム・グループ定義」表形式ビューを開きます。  
現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - 「システム・グループ名」フィールド内の CSGTGTS1 をクリックして詳細ビューを開き、「この CICS システム・グループ内の CICS システム」リンクをクリックします。これにより、「CICS システムと CICS システム・グループとの間のリンク」表形式ビューが開きます。
  - CICSPA04 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「除去...」ボタンをクリックします。これにより、「除去」確認ビューが開きます。
  - 「はい」をクリックして、更新を確認します。
2. CICSPA04 が除去されていることを確認するには、メインメニューから「アクティブ・ワークロード (WLM)」>「アクティブ・ワークロードのターゲット配布係数」をクリックします。これにより、ワークロードの経路指定が可能なすべてのターゲット領域が表示された「アクティブ・ワークロードのターゲット配布係数」ビューが開きます。CICSPA04 はこのリストに表示されないはずです。

## アクティブ・ワークロードへのルーティング領域の追加

Web ユーザー・インターフェース (WUI) を使用して、アクティブ・ワークロードを中断せずに、2 つ目のルーティング領域をそのワークロードに追加します。

ワークロード仕様 WLSPAY01 を介して CICSplex PLXPROD1 内のワークロードをまだルーティングしており、作業要求が CICSPT01 によって CICS システム・グループ CSGTGTS1 内のターゲット領域に経路指定されていると想定します。そして今度は、2 番目のルーティング領域 CICSPT02 をワークロードに追加しようとしています。領域 CICSPT02 は、「**CICS システム定義**」ビューを使用して) CICSplex PLXPROD1 に定義されており、ターゲット領域 CICSPA01、CICSPA02、および CICSPA03 を実行しており、これらの領域にリンクされています。

1. 以下の手順を実行して、CICS システム CICSPT02 の CICSplex SM 定義を更新します。

- Web ユーザー・インターフェースのメインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックして、「**CICS システム定義**」表形式ビューを開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

- CICSPT02 のレコードを選択して「更新...」ボタンをクリックします。
- 「**AOR 動的ルーティング・モード**」フィールドを「はい」に変更します。
- 「はい」をクリックして、更新を確認します。この変更は、ルーティング領域 CICSPT02 を次回開始した時に有効になります。

2. 以下の手順を実行して、CICSPT02 をワークロード仕様 WLSPAY01 に関連付けます。

- メインメニューで、「管理」>「ワークロード・マネージャー管理」>「仕様」をクリックします。
- 「**WLM 仕様**」ビューで、WLSPAY01 のエントリーの横にある「レコード」チェック・ボックスをクリックし、「**CICS システムの関連付け...**」をクリックします。
- 「**CICS システム**」フィールドに CICSPT02 と入力し、「はい」をクリックして更新を確認します。

CICSPT02 が再始動されたら、以下の手順を実行して、WLSPAY01 に追加されたことを確認できます。

- 「**WLM 仕様**」表形式ビューを開き、WLSPAY01 のエントリーをクリックします。これにより詳細ビューが開きます。
- 「このワークロード仕様に関連する **CICS システム**」リンクをクリックします。CICSPT01 と CICSPT02 の両方が、「**WLM 仕様と CICS システムとの間のリンク**」表形式ビューにリストされているはずです。

CICSPT02 がアクティブ・ワークロードの一部であることを確認するには、メインメニューから「**アクティブ・ワークロード (WLM)**」>「**アクティブ・ワークロード内のルーティング領域**」をクリックします。これにより、CICSPT01 と CICSPT02 の両方のエントリーを示す表形式ビューが開きます。

## アクティブなワークロードのターゲット領域を静止

この例では、Web ユーザー・インターフェース (WUI) を使用して、アクティブ・ワークロード WLSPAY01 に属するターゲット領域 CICSPA03 を静止する方法について説明します。

このタスクは、例えば領域に保守を適用できるようにするために実行する必要がある場合があります。このタスクを実行すると、領域 CICSPT01 および CICSPT02 は、作業要求を 3 つのターゲット領域 (CICSPA01、CICSPA02、および CICSPA03) にルーティングします。

領域で作業が既に実行されているかどうかに関係なく、その領域に作業がルーティングされないようにする場合は、その領域の z/OS WLM ヘルスのオープン状況を CLOSE または IMMCLOSE に設定できます。詳しくは、63 ページの『[CICSplex SM ワークロードのルーティングに対する z/OS WLM ヘルス・サービスの効果](#)』を参照してください。

1. ワークロード WLSPAY01 に関連付けられたターゲット領域をリストします。

- Web ユーザー・インターフェースのメインメニューから、「**アクティブ・ワークロード (WLM)**」>「**アクティブ・ワークロードのターゲット配布係数**」をクリックして、「**アクティブ・ワークロードのターゲット配布係数**」ビューを開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。



- ・「ワークロード名」フィールドに WLSPAY01 と入力し、「最新表示」をクリックします。これにより、「アクティブ・ワークロードのターゲット配布係数」ビューが再表示されます。CICSPA01、CICSPA02、および CICSPA03 という 3 つの CICS 領域がリストされます。それらの状況はアクティブです。
2. 以下の手順を実行して、ターゲット領域 CICSPA03 を静止します。
    - ・「アクティブ・ワークロードのターゲット配布係数」ビューで、ターゲット領域 CICSPA03 のエントリーを選択し、「静止...」ボタンをクリックします。これにより、「静止」確認ビューが開きます。
    - ・「はい」をクリックして、アクションを確認します。「アクティブ・ワークロードのターゲット配布係数」ビューが再表示され、CICSPA03 の「ターゲット領域状況」値が QUIESCED に変更されているのが示されます。新しい要求はこのターゲット領域に経路指定されませんが、すでにそこで実行中の作業は完了まで実行することが許可されます。

## 特定のターゲット領域への特定のトランザクションのルーティング

Web ユーザー・インターフェース (WUI) を使用して、いくつかのワークロード分離要件を CICSplex SM に定義します。

この例では、Web ユーザー・インターフェース (WUI) を使用して、トランザクション PAY1 を CICS システム・グループ CSGTGTS1 に属するターゲット領域 CICSPA02 に常に経路指定する方法について説明します。作業は引き続き CICSplex PLXPROD1 で行われ、グループ CSGTGTS1 のターゲット領域内の単一 CICSPT01 からのワークロード・ルーティングが有効になります。

1. 経路指定されるトランザクションが EXEC CICS START を使用して開始される場合は、ROUTABLE として定義する必要があります。
2. トランザクション・グループを作成します。
  - ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「トランザクション・グループ定義」をクリックします。これにより、「トランザクション・グループ定義」ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

  - ・「作成...」ボタンをクリックして、以下の情報を入力します。

名前	TRGPAY01
説明	PAY1 を CICSPA02 に送信
親和性の関係および存続時間チェック状況	ACTIVE
主検索基準	USERID

残りのフィールドは、ブランクのままにしても、デフォルトを受け入れても構いません。

「はい (Yes)」をクリックします。「トランザクション・グループ定義」ビューが再表示され、トランザクション・グループ TRGPAY01 の名前が現在表示されています。

3. 以下の手順を実行して、トランザクション PAY1 をトランザクション・グループ TRGPAY01 に追加します。
  - ・「トランザクション・グループ定義」ビューで TRGPAY01 のエントリーを選択し、「トランザクションの追加...」ボタンをクリックします。
  - ・「トランザクション名」フィールドに PAY1 という名前を入力し、「はい」をクリックして確定します。「トランザクション・グループ定義」ビューが再表示されます。
4. ワークロード定義を作成します。
  - ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックします。これにより、「ワークロード定義」ビューが開き、PLXPROD1 に作成済みのワークロード定義がリストされます。
  - ・「作成...」ボタンをクリックして、以下の情報を入力します。

ワークロード管理定義	WLDPAY01
------------	----------

説明	TRGPAY01 を CICSPA02 に分離
トランザクション・グループ名	TRGPAY01
ターゲット・システム・セットのスコープ名	CICSPA02

- ・「はい」をクリックして確定します。

5. ワークロード・グループを作成する。(トランザクションをルーティングするルーティング領域が開始されたときにワークロード定義を自動的にインストールする場合は、ワークロード・グループが不可欠です。)

- ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「グループ」をクリックします。これにより、「WLM グループ」ビューが開き、PLXPROD1 に作成済みのワークロード・グループがリストされます。
- ・「作成...」ボタンをクリックして、以下の情報を入力します。

ワークロード管理グループ	WLGPAY01
説明	WLDAPAY01 のワークロード・グループ

- ・「はい」をクリックして確定します。「WLM グループ」ビューが再表示されます。

6. 以下の手順を実行して、ワークロード・グループ WLGPAY01 を既存のワークロード仕様 WLSPAY01 に追加します。

- ・「WLM グループ」ビューで WLGPAY01 の項目を選択して「WLM 仕様への追加...」ボタンをクリックします。
- ・「仕様名」フィールドに既存のワークロード仕様の名前 WLSPAY01 を入力し、「はい」をクリックします。「WLM グループ」ビューが再表示されます。

7. 以下のようにして、ワークロード定義をワークロード・グループに追加します。

- ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックして、「WLM 定義」ビューを開きます。
- ・WLDAPAY01 の項目を選択して「WLM グループへの追加...」ボタンをクリックします。
- ・「リソース・グループ名」フィールドに WLGPAY01 と入力して、「はい」をクリックします。「WLM 定義」ビューが再表示されます。

8. ワークロード・グループをアクティブ・ワークロードにインストールします。

ワークロード WLSPAY01 は既にアクティブなため、新しいワークロード・グループ WLGPAY01 を明示的にインストールする必要があります。WLGPAY01 をインストールしなかった場合は、ルーティング領域 CICSP01 と、CICS システム・グループ CSGTGTS1 内のターゲット領域が次回開始されるまで有効になりません。

- ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「グループ」をクリックして、「WLM グループ」表形式ビューを開きます。
- ・WLGPAY01 の項目を選択して、「インストール...」ボタンをクリックします。
- ・「ワークロード名」フィールドに WLSPAY01 と入力し、「ワークロード所有者」フィールドに、ワークロード仕様 WLSPAY01 が作成されたシステムの SYSID を入力します。

「はい」をクリックして確定します。

CICSplex PLXPROD1 で既にアクティブなワークロード仕様を再使用しており、ワークロード・グループはインストール済みのため、この例で定義したワークロード分離は即時に有効になります。

「WLM 定義」ビューを開いて、新しいワークロード定義 WLDAPAY01 がアクティブなことを確認できます。このビューには、ワークロード WLSPAY01 内の WLDAPAY01 のエントリーが含まれているはずです。任意のユーザーによって任意の端末からトランザクション PAY1 が次回開始された時、CICSplex SM はそれをターゲット領域 CICSPA02 に経路指定します。

## 指定されたユーザーから特定のターゲット領域への特定のトランザクションのルーティング

この例では、Web ユーザー・インターフェース (WUI) を使用して、指定されたユーザーから特定のターゲット領域に特定のトランザクションをルーティングする方法について説明します。

このタスク例では、前の例 (89 ページの『特定のターゲット領域への特定のトランザクションのルーティング』) を改良して Web ユーザー・インターフェースの使用方法について再び説明します。今回は、ユーザー ID 値が、トランザクションのルーティング先の判別の追加要因になります。トランザクション PAY6、PAY7、PAY8、および PAY9 は、ユーザー ID USRPAY03 から開始された時、CICSplex PLXPROD1 内のターゲット領域 CICSPA03 にルーティングされる必要があります。

### 1. トランザクション・グループを作成します。

- Web ユーザー・インターフェースのメインメニューから「管理」>「ワークロード・マネージャー管理」>「トランザクション・グループ定義」をクリックします。これにより、「トランザクション・グループ定義」表形式ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

- 「作成...」ボタンをクリックして、以下の情報を入力します。

名前	TRGPAY02
説明	USRPAY03 のトランザクションを CICSPA03 に送信
親和性の関係および存続時間チェック状況	ACTIVE
主検索基準	USERID

残りのフィールドはブランクのままにするか、デフォルトを受け入れることができます。

- 「はい」をクリックして確定します。「トランザクション・グループ定義」表形式ビューが再表示されます。

### 2. 以下の手順を実行して、グループ TRGPAY02 に属するトランザクションを識別します。

- 「トランザクション・グループ定義」表形式ビューで TRGPAY02 のエントリーを選択し、「トランザクションの追加...」ボタンをクリックします。
- 「トランザクション名」フィールドに最初のトランザクションの名前 PAY6 を入力し、「はい」をクリックして確定します。

さらに 3 つのトランザクション PAY7、PAY8、および PAY9 に対してこのステップを繰り返します。

### 3. ワークロード定義を作成します。

- メインメニューから「管理」>「ワークロード・マネージャー」>「定義」をクリックします。これにより、「ワークロード定義」表形式ビューが開き、PLXPROD1 に作成済みのワークロード定義がリストされます。
- 「作成...」ボタンをクリックして、以下の情報を入力します。

ワークロード管理定義	WLDPAY02
説明	TRGPAY02 を CICSPA03 に分離
トランザクション・グループ名	TRGPAY02
ユーザー ID	USRPAY03
ターゲット・システム・セットのスコープ名	CICSPA03

- 「はい」をクリックして確定します。

### 4. 以下のようにして、既存のワークロード定義をワークロード・グループに追加します。

- メインメニューから「管理」>「ワークロード・マネージャー管理」->「定義」をクリックして、「WLM 定義」ビューを開きます。

- WLDAPY0 の項目を選択して「**WLM グループへの追加**」ボタンをクリックします。
  - 「**リソース・グループ名**」フィールドに WLGPAY01 (前の例で作成したワークロード・グループの名前) を入力します。ワークロード・グループ WLGPAY01 は、アクティブ・ワークロード仕様 WLSPAY01 に既に関連付けられています。
  - 「**はい**」をクリックして確定します。
5. 以下の手順を実行して、新しいワークロード定義をアクティブ・ワークロードにインストールします。
- グループ WLGPAY01 は既にアクティブ・ワークロード WLSPAY01 に関連付けられているため、そのグループ対して行った変更は、ルーティング領域 CICSPT01 と CICS システム・グループ CSGTGTS1 内のターゲット領域が次回開始されるまで有効になりません。新しいワークロード定義を即時に有効にするには、以下の手順を実行して、WLSPAY01 に明示的にインストールする必要があります。
- 「**WLM 定義**」ビューを再度開き、WLDAPY02 のエントリーを選択し、「**インストール...**」ボタンをクリックします。
  - 「**ワークロード名**」フィールドに WLSPAY01 と入力し、「**ワークロード所有者**」フィールドに、ワークロード仕様 WLSPAY01 が作成されたシステムの SYSID を入力します。
- ワークロード定義 WLDAPY02 をアクティブ・ワークロード WLSPAY01 に明示的にインストールしたため、この例で定義したワークロード分離要件は即時に有効になります。

## 疑似会話型トランザクションの受け入れ

この例では、Web ユーザー・インターフェース (WUI) を使用して、親和性が存在する複数のトランザクションが確実に同じターゲット領域にルーティングされるようにする方法について説明します。

以前と同様に、作業は CICSplex PLXPROD1 内で行い、CICSPT01 から CICS システム・グループ CSGTGTS1 内のターゲット領域にトランザクションをルーティングします。

- 以下のとおり、トランザクション・グループを作成します。
    - メインメニューから「**管理**」>「**ワークロード・マネージャー管理**」>「**トランザクション・グループ定義**」をクリックします。これにより、「**トランザクション・グループ定義**」表形式ビューが開きます。
- 現行のコンテキストが PLXPROD1 ではない場合は、「**コンテキスト**」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。
- 「**作成...**」ボタンをクリックして、以下の情報を入力します。

名前	TRGPAY03
説明	疑似会話型トランザクション
親和性の関係および存続時間チェック状況	ACTIVE
主検索基準	USERID
親和性の関係	USERID
親和性存続時間	PCONV

「**親和性の関係**」フィールドと「**親和性存続時間**」フィールドに入力する必要がある点に注意してください。これらの値は、このグループ内のトランザクションが疑似会話型トランザクション (PCONV) を構成しており、それらのトランザクションが同じユーザー ID (USERID) から開始されている間はこの親和性が継続することを CICSplex SM に示します。それらのいずれかが別のユーザー ID から開始された場合、CICSplex SM は別のターゲット領域を選択することができます。もちろん、その 2 つ目のターゲット領域で同じタイプの親和性が動作します。ブランクのままになっているフィールドをすべて無視するか、デフォルトを受け入れることができます。

- 「**はい**」をクリックして確定します。「**トランザクション・グループ定義**」表形式ビューが再表示されます。
- 以下の手順を実行して、グループ TRGPAY03 に属するトランザクションを識別します。

- ・「トランザクション・グループ定義」表形式ビューで TRGPAY03 のエントリーを選択し、「トランザクションの追加...」ボタンをクリックします。
- ・「トランザクション名」フィールドに最初のトランザクション PAY2 の名前を入力し、「はい」をクリックして確定します。

さらに3つのトランザクション PAY3、PAY4、および PAY5 に対してこのステップを繰り返します。

PAY2 が疑似会話を開始し、PAY5 がそれを終了するのを確認できます。

### 3. ワークロード定義を作成します。

- ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックします。これにより、「ワークロード定義」ビューが開き、PLXPROD1 に作成済みのワークロード定義がリストされます。
- ・「作成...」ボタンをクリックして、以下の情報を入力します。

ワークロード管理定義	WLDAPAY03
説明	TRGPAY03 を同じターゲット領域へ
トランザクション・グループ名	TRGPAY03
ターゲット・システム・セットのスコープ名	CSGTGTS1

- ・「はい」をクリックして確定します。

### 4. 以下のとおり、ワークロード定義をワークロード・グループに追加します。

- ・メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックして、「WLM 定義」表形式ビューを開きます。
- ・WLDAPAY03 の項目を選択して「WLM グループへの追加...」ボタンをクリックします。
- ・「リソース・グループ名」フィールドに WLGPAY01 と入力します。
- ・「はい」をクリックして確定します。

### 5. グループ WLGPAY01 は既にアクティブ・ワークロード WLSPAY01 に関連付けられているため、そのグループ対して行った変更は、ルーティング領域 CICSPT01 と CICS システム・グループ CSGTGTS1 内のターゲット領域が次回開始されるまで有効になりません。新しいワークロード定義を即時に有効にするには、WLSPAY01 に明示的にインストールする必要があります。新しいワークロード定義をアクティブ・ワークロードにインストールするには、以下の手順を実行します。

- ・「WLM 定義」ビューを再度開き、WLDAPAY03 のエントリーを選択し、「インストール...」ボタンをクリックします。
- ・「ワークロード名」フィールドに WLSPAY01 と入力し、「ワークロード所有者」フィールドに、ワークロード仕様 WLSPAY01 が作成されたシステムの SYSID を入力します。

ワークロード定義 WLDAPAY03 をアクティブ・ワークロード WLSPAY01 に明示的にインストールしたため、CICSplex SM は、この疑似会話型トランザクションを即時受け入れることができます。この単一ワークロード仕様 WLSPAY01 をワークロード・ルーティングとワークロード分離の両方に使用できることに注意してください。その理由は、**親和性の関係**と「**親和性存続時間**」のデフォルト値を WLSPAY01 に指定しなかったためです。これらの値を指定していた場合は、ワークロード・ルーティング用とワークロード分離用に別々のワークロード仕様を作成しなければなりませんでした。

## ワークロード定義の非アクティブ化

この例では、Web ユーザー・インターフェース (WUI) を使用してワークロード定義を非アクティブにする方法について説明します。

この例では、Web ユーザー・インターフェース (WUI) を使用して、例 91 ページの『指定されたユーザーから特定のターゲット領域への特定のトランザクションのルーティング』で作成されたワークロード定義 WLDAPAY02 を非アクティブ化にする方法について説明します。

### 1. 以下の手順を実行して、アクティブ・ワークロード定義を表示します。

- メインメニューで、「**アクティブ・ワークロード (WLM)**」>「**定義**」をクリックします。これにより、「**アクティブ・ワークロード定義**」ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「**コンテキスト**」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。

- 「**アクティブ・ワークロード定義**」ビューで WLSPAY01 と入力し、「**最新表示**」をクリックします。ワークロード仕様 WLSPAY01 に関連付けられたアクティブ・ワークロード定義がリストされます。

2. 以下の手順を実行して、ワークロード定義 WLDPAY02 を破棄します。

- a. WLDPAY02 の項目を選択して「**破棄...**」ボタンをクリックします。これにより、「**破棄**」確認ビューが開きます。
- b. 「**はい**」をクリックしてアクションを確認します。

アクティブ・ワークロード定義を非アクティブにすると、その定義に関連付けられているすべてのトランザクション・グループは、同じワークロード内の別のワークロード定義によって参照されていない場合、同様に非アクティブになることに注意してください。

## アクティブ・ワークロード定義の更新

この例では、Web ユーザー・インターフェース (WUI) を使用して、アクティブ・ワークロード定義を更新し、それをアクティブ・ワークロードに再インストールする方法について説明します。

例 91 ページの『指定されたユーザーから特定のターゲット領域への特定のトランザクションのルーティング』で、トランザクション・グループ TRGPAY02 を作成し、ワークロード定義 WLDPAY02 に指定しました。この例では、TRGPAY02 を削除し、それを、既に作成されている新しいトランザクション・グループ TRGPAY04 に置き換える方法について説明します。

1. 以下の手順を実行して、アクティブ・ワークロード定義を表示します。

- メインメニューで、「**アクティブ・ワークロード (WLM)**」>「**定義**」をクリックします。これにより、「**アクティブ・ワークロード定義**」表形式ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「**コンテキスト**」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。

- 「**アクティブ・ワークロード定義**」ビューで WLSPAY01 と入力し、「**最新表示**」をクリックします。ワークロード仕様 WLSPAY01 に関連付けられたアクティブ・ワークロード定義がリストされます。

WLDPAY02 の「**ワークロード所有者のシステム ID**」フィールド内の値をメモします (この例では CM1B)。

2. ワークロード定義 WLDPAY02 を更新します。

- メインメニューから「**管理**」>「**ワークロード・マネージャー管理**」>「**定義**」をクリックして、「**WLM 定義**」表形式ビューを開きます。
- WLDPAY02 のエントリーを選択して「**更新...**」ボタンをクリックします。「**WLM 定義**」作成ビューが表示されます。
- 「**WLM 定義**」作成ビューで、「**トランザクション・グループ名**」を TRGPAY04 に変更し、「**説明**」テキストを「Separate TRGPAY04 to CICS PA03」に変更します。
- 「**はい (Yes)**」をクリックします。「**WLM 定義**」表形式ビューが再表示されます。

3. 以下の手順を実行して、更新されたワークロード定義を WLSPAY01 にインストールします。

「**WLM 定義**」表形式ビューで WLDPAY02 のエントリーを選択し、「**インストール...**」をクリックします。「**インストール**」ビューが表示されます。「**ワークロード名**」フィールドに WLSPAY01 と入力します。「**ワークロード所有者**」フィールドに、ステップ 94 ページの『1』でメモした、ワークロード所有者の 4 文字の ID を入力します。「**はい (Yes)**」をクリックします。更新されたワークロード定義がワークロード WLSPAY01 にインストールされます。

4. 以下の手順を実行して、更新されたワークロード定義がインストールされていることを確認します。

- メインメニューで、「**アクティブ・ワークロード (WLM)**」>「**定義**」をクリックします。これにより、「**アクティブ・ワークロード定義**」表形式ビューが開き、更新された定義が表示されます。TRGPAY02 が TRGPAY04 に置換されていることを確認します。



5. 以下の手順を実行して、トランザクション・グループ TRGPAY02 が非アクティブであり、トランザクション・グループ TRGPAY04 がアクティブになっていることを確認します。

- ・メインメニューから「**アクティブ・ワークロード (WLM)**」>「**トランザクション・グループ**」をクリックして、2つのトランザクション・グループの状況を表示します。

注:

ワークロード定義内の「**ユーザー ID**」、「**端末 LU 名**」、「**BTS プロセス・タイプ**」、または「**ターゲット・システム・セットのスコープ名**」の各フィールドを更新すると、ワークロード定義は動的に再インストールされません。ワークロード定義をアクティブ・ワークロードに再インストールするには、以下を実行する必要があります。

- ・「**アクティブ・ワークロード定義**」表形式ビューから「**破棄...**」ボタンをクリックして、アクティブ・ワークロード定義を破棄します。
- ・ワークロード管理の「**定義**」ビューから「**インストール...**」ボタンをクリックして、ワークロード定義をワークロードにインストールします。

代替方法として、変更済み属性を使用してワークロード定義を再インストールするには、バッチ API プログラムを使用して、影響のあるアプリケーションへのアクセスを無効にし、「**アクティブ・ワークロード定義**」ビューを破棄し、「**WLM 定義**」ビューをインストールし、影響のあるアプリケーションを再び有効にします。

## ワークロードからのアクティブ・トランザクションの破棄

この例では、Web ユーザー・インターフェース (WUI) を使用して、ワークロードからアクティブ・トランザクションを破棄する方法について説明します。

1. アクティブ・トランザクションを表示します。

- ・メインメニューから、「**アクティブ・ワークロード (WLM)**」>「**動的トランザクション**」をクリックします。

現行のコンテキストが PLXPROD1 ではない場合は、「**コンテキスト**」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。

- ・「**アクティブ・ワークロードの動的トランザクション**」表形式ビューに、ワークロード仕様 EYUWLS02 に関連付けられたアクティブ・トランザクションが表示されます。

2. 以下の手順を実行して、トランザクション PAY2 を破棄します。

- ・「**アクティブ・ワークロードの動的トランザクション**」ビューからトランザクション PAY2 を選択し、「**強制...**」ボタンをクリックします。「**強制**」確認ビューが表示されます。
- ・「**はい**」をクリックしてアクションを確認します。

## ワークロード仕様の更新

この例では、Web ユーザー・インターフェースを使用したワークロード仕様の更新方法について説明します。

### このタスクについて

この例では、更新するワークロード仕様は WLSPAY01 です。これは、[84 ページの『ワークロードの管理』](#)の例で作成したものです。

このタスクの影響について注意深く考慮してください。具体的には、更新されたワークロード仕様が即時に有効になることはありません。更新がすぐに有効になるためには、ワークロード仕様に関連付けられているルーティング領域と、ルーティング領域によるトランザクションのルーティング先であるターゲット領域の両方を停止してから再開する必要があります。

### プロシージャ

1. ワークロード仕様 WLSPAY01 を更新します。

- a. Web ユーザー・インターフェースのメインメニューから「管理」 > 「ワークロード・マネージャー 管理」 > 「仕様」をクリックして、「WLM 仕様」(WLMSPEC) 表形式ビューを開きます。  
 現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - b. WLSPAY01 のエントリーを選択して「更新」ボタンをクリックします。
  - c. 「アルゴリズム・タイプ」フィールドを、QUEUE から GOAL に変更します。
  - d. 「はい (Yes)」をクリックします。ビューが再表示され、更新された WLSPAY01 のエントリーが示されます。
2. 以下の手順を実行して、「アクティブ・ワークロード」ビューを表示します。
- a. メインメニューで、「アクティブ・ワークロード (WLM)」 > 「アクティブ・ワークロード」をクリックします。
  - b. 「アクティブ・ワークロード」表形式ビューで、ワークロード名 WLSPAY01 を入力し、「最新表示」をクリックします。
- 「アクティブ・ワークロード」表形式ビューが再表示されます。WLM 仕様 WLSPAY01 のアルゴリズム・タイプがまだ QUEUE であることが確認できます。これは、行われた変更を領域 CICSPT01 が認識していないことが原因です。

## 次のタスク

更新されたワークロード仕様を即時に有効にしたい場合は、「CICS 領域」ビューを使用して、ルーティング領域とルーティング先のターゲット領域を停止し、再開始してください。これらの領域は、すべて一緒にシャットダウンする必要があり、指定されたワークロードに参加している CICS エレメントを持つ各 CMAS がメッセージ EYUWM0410I を発行して、ワークロード仕様が正常に終了したことを裏付けるまでは再開始してはなりません。ワークロードのメンバーである CICS 領域が、メッセージ EYUWM0410I を受け取る前に再開始すると、既存のワークロード仕様が引き続き使用され、ワークロード仕様の更新は有効になりません。ルーティング領域が再開始された後、「アクティブ・ワークロード」テーブル表示形式ビューを表示すると、「アルゴリズム・タイプ」フィールドが GOAL に更新されていることが確認できます。

## ワークロード・ルーティングのターゲット領域を選択するためのリアルタイム分析の使用

ワークロード・ルーティング中のターゲット領域の選択に役立つデータを生成するには、CICSplex SM のリアルタイム分析機能を使用します。

処理をルーティングできるターゲット領域ごとに、一時ストレージ・キューの項目数がモニターされます。項目数が 50 を超えると、リアルタイム分析イベント通知 (重大度 HS) および外部メッセージが発行されます。イベント通知が発行されると、CICSplex SM のワークロード・ルーティング機能はその通知を受け、最適なターゲット領域を選択する際に、標準キュー・アルゴリズム基準と併せてその情報を使用します。

CICS システム・グループ CSGTGTS3 がすでに作成されていて、このグループに、4 つのターゲット領域 (CICSPA01、CICSPA02、CICSPA03、および CICSPA04) が含まれています。これらのターゲット領域は現在実行されています。これらのターゲット領域の間で現在処理をルーティングしているのは、ルーティング領域 CICSPT03 です。

1. 以下のようにして、リアルタイム分析評価定義を作成します。

- ・メインメニューで、「管理」 > 「RTA MAS リソース・モニター」 > 「評価」をクリックします。これにより、「評価定義」表形式ビューが開き、PLXPROD1 に作成済みの評価定義がリストされます。  
 現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
- ・「作成...」ボタンをクリックして、以下の情報を入力します。

名前	RTEPAY15
説明	TSQ NUMITEMS > 50
サンプリング間隔	300
リソース・テーブル名	MTSQGBL

評価リソースのインスタンス ID	*
結果セット内の結果の評価方式	ANY
分離タスク・インディケーター	NO
評価されたフィールド名	PUTQAUX
評価タイプ	VALUE
評価の論理演算子	GT
評価データ値	50
結果が基準と合う時に割り当てる重大度	HS
エクストラ情報を提供できるビュー名	MTSQGBL

- ・「はい」をクリックして新しい評価定義を作成します。

## 2. 以下のようにして、リアルタイム分析アクション定義を作成します。

- ・メインメニューで、「管理」 > 「RTA MAS リソース・モニター」 > 「アクション」をクリックします。これにより、「アクション定義」ビューが開きます。
- ・「作成...」 ボタンをクリックして、以下の情報を入力します。

アクション	RTAPAY15
説明	NUMITEMS IN TSQ > 50
イベント・オプションの生成	YES
有益な情報を提供できるビュー名	TSQ
アクションの優先順位	255
イベント発生時に送信するメッセージ	AUX TSQUEUE PUTQ ITEMS > 50
外部メッセージ・オプションの生成	YES
イベント発生時に送信した外部メッセージ	AUX TSQUEUE PUTQ ITEMS > 50
イベント切断時に送信した外部メッセージ	AUX TSQUEUE PUTQ ITEMS < 50
SNA 総称アラート・オプションの生成	NO
MVS 自動リスタート	NO

- ・「はい」をクリックして新しい RTA アクション定義を作成します。「アクション定義」表形式ビューが再表示されます。

## 3. 以下のとおり、分析定義を作成します。

- ・メインメニューで、「管理ビュー」 > 「RTA MAS リソース・モニター」 > 「定義」をクリックします。これにより、「RTA 定義」ビューが開きます。
- ・「作成...」 ボタンをクリックして、以下の情報を入力します。

名前	RTDPAY15
説明	TSQ NUMITEMS FOR WLM /RTA
実行評価変更ストリング	NO
分析間隔	300
アクション定義名	RTAPAY15
VLS 発生までの真の評価回数	0001
VLS 解決までの偽の評価回数	0001
LS 発生までの真の評価回数	001

LS 解決までの偽の評価回数	001
LW 発生までの真の評価回数	001
LW 解決までの偽の評価回数	001
HW 発生までの真の評価回数	001
HW 解決までの偽の評価回数	001
HS 発生までの真の評価回数	001
HS 解決までの偽の評価回数	0001
VHS 発生までの真の評価回数	0001
VHS 解決までの偽の評価回数	0001
VHS 解決までの偽の評価回数	0001
評価式	RTEPAY15

- ・「はい」をクリックして新しい分析定義を作成します。「分析定義」表形式ビューが再表示されます。

4. 以下のとおり、分析グループを作成します。

- ・メインメニューで、「管理ビュー」 > 「RTA MAS リソース・モニター」 > 「グループ」をクリックします。これにより、「RTA グループ」表形式ビューが開きます。
- ・「作成...」ボタンをクリックして、以下の情報を入力します。

RTA グループ

RTGPAY09

説明

RTA/WLM グループ

- ・「はい」をクリックして新しい分析グループを作成します。「RTA グループ」表形式ビューが再表示されます。

5. 以下のようにして、分析定義を分析グループに追加します。

- ・メインメニューで、「管理ビュー」 > 「RTA MAS リソース・モニター」 > 「定義」をクリックします。これにより、「RTA 定義」ビューが開きます。
- ・RTDPAY15のエントリーの隣にある「レコード」チェック・ボックスをクリックして、「RTA グループへの追加...」ボタンをクリックします。「RTA グループへの追加」ビューが表示されます。
- ・「リソース・グループ名」フィールドに RTGPAY09 と入力して、「はい」をクリックします。「RTA 定義」表形式ビューが再表示されます。

6. 以下のとおり、分析仕様を作成します。

- ・メインメニューで、「管理ビュー」 > 「RTA MAS リソース・モニター」 > 「仕様」をクリックします。これにより、「RTA 仕様」表形式ビューが開きます。
- ・「作成...」ボタンをクリックして、以下の情報を入力します。

RTA 仕様名

RTSPAY09

説明

RTA でのワークロード・ルーティング

- ・「はい」をクリックします。「RTA 仕様」表形式ビューが再表示されて、新規 RTA 仕様がリストされます。

7. 以下のようにして、分析グループを分析仕様に追加します。

- ・メインメニューで、「管理ビュー」 > 「RTA MAS リソース・モニター」 > 「グループ」をクリックします。これにより、「RTA グループ」表形式ビューが開きます。
- ・RTGPAY09のエントリーの隣にある「レコード」チェック・ボックスをクリックして、「RTA 仕様への追加...」ボタンをクリックします。「RTA 仕様への追加」ビューが表示されます。

- ・「**RTA 仕様名**」フィールドに RTSPAY09 と入力し、「はい」をクリックします。「**RTA グループ**」表形式ビューが再表示されます。

8. 以下のようにして、分析仕様のスコープを設定します。

- ・メインメニューで、「管理ビュー」 > 「**RTA MAS リソース・モニター**」 > 「仕様」をクリックします。これにより、「**RTA 仕様**」表形式ビューが開きます。
- ・RTGPAY09 のエントリーの隣にある「レコード」チェック・ボックスをクリックして、「**CICS グループとの関連付け...**」ボタンをクリックします。「**CICS グループとの関連付け**」ビューが表示されます。
- ・「**CICS システム・グループ**」フィールドに CSGTGTS3 と入力し、「強制」オプションを選択します。「はい」をクリックします。「**RTA 仕様**」表形式ビューが再表示されます。

9. 以下のようにして、CSGTGTS3 のターゲット領域でリアルタイム分析をアクティブにします。

- ・メインメニューで、「管理」 > 「**RTA システム使用可能性モニター**」 > 「**CICS システム定義**」をクリックします。これにより、「**CICS システム定義**」表形式ビューが開きます。
- ・CICS システム CICSPA01 のエントリーの隣にある「レコード」チェック・ボックスをクリックして、「更新...」ボタンをクリックし、以下の情報を入力します。

CICS システム定義名	CICSPA01
説明	システム A 上のターゲット領域 1
リアルタイム分析状況	YES
システム使用可能性モニター・イベントの重大度	NO
ストレージ不足 (SOS) イベントの重大度	NO
システム・ダンプ・イベントの重大度	NO
トランザクション・ダンプ・イベントの重大度	NO
CICS 最大タスク数到達イベントの重大度	NO
CICS 停止イベントの重大度	NO

- ・「はい」をクリックして、変更を確認します。CICS システム定義が更新されて、変更 (これは永続的な変更です) が即時に適用されます。CICS システム CICSPA01 を再始動する必要はありません。ターゲット領域 CICSPA02、CICSPA03、および CICSPA04 について、上記のステップを繰り返します。

10. ワークロード仕様 WLSPAY02 は、CSGTGTS3 内のターゲット領域の間でのワークロード・ルーティングにすでに使用されています。仕様を更新して、リアルタイム分析データを標準のターゲット領域選択基準に追加する必要があります。以下のようにして、ワークロード仕様を更新します。

- ・メインメニューで、「管理」 > 「**ワークロード・マネージャー管理**」 > 「仕様」をクリックします。これにより、「**WLM 仕様**」表形式ビューが開きます。
- ・WLSPAY02 のエントリーの隣にある「レコード」チェック・ボックスをクリックして、「更新...」ボタンをクリックし、以下の情報を入力します。

名前	WLSPAY02
説明	ターゲット領域選択に RTA を使用するワークロード
主検索基準	USERID
デフォルト・ターゲット・スコープ	CSGTGTS3
RTA イベント名	RTDPAY15
異常終了する確率の許容レベル	0
許容できる異常終了ロードしきい値	0
アルゴリズムのタイプ	QUEUE

- 「はい」をクリックして、仕様を更新します。

「**RTA イベント**」フィールドの値は、ステップ 97 ページの『3』で作成した分析定義の名前であることを注意してください。

ルーティング領域 CICSPT03 と CICS システム・グループ CSGTGTS3 内のターゲット領域が次に開始されると、ルーティング領域は、標準のキュー・アルゴリズム基準と、ターゲット領域を選択するための分析定義 RTDPAY15 の両方を使用して、トランザクションをターゲット領域の間でルーティングします。

## EXEC CICS START TERMID を使用した動的ルーティング

Web ユーザー・インターフェース (WUI) を使用して、端末 ID とトランザクション ID を指定する **EXEC CICS START** で開始されるトランザクションの動的ルーティング機能をセットアップし、GOAL アルゴリズムを使用してターゲット領域を選択します。

**注:** **EXEC CICS START TERMID** の動的ルーティングを試行する前にシステム要件を確認する必要があります。CICSplex SM ワークロード要件を参照してください。

この例では、CICSPA01 で実行されているプログラムが、端末 TRM1 に関連付けられた **EXEC CICS START** を使用して開始され、トランザクション PAY1 を実行します。このトランザクションには 2 秒の応答時間が必要です。端末 TRM1 は、領域 CICSPT01 に関連付けられています。トランザクション PAY1 は、CICSPT01 に接続されている任意の領域 (つまり、CICSPA01、CICSPA02、または CICSPA03) で実行できます。

この例では、前出の例の一部として既に作成されている環境が使用されます。作業は、CICS システム・グループ CSGTGTS1 内の、TOR CICSPT01 と、AOR の CICSPA01、CICSPA02、および CICSPA03 から成る CICSplex PLXPROD1で行います。トランザクション・グループ TRGPAY03 には、トランザクション PAY1、PAY2、PAY3、および PAY4 が関連付けられています。ワークロード定義 WLDAPAY03 は、グループ TRGPAY03 内のトランザクションを CICS システム・グループ CSGTGTS1 内のターゲット領域に経路指定する必要があることを CICSplex SM に指示するよう定義されました。

GOAL アルゴリズムを使用するため、必要な応答時間を持つサービス・クラスを MVS Workload Manager に定義し、そのサービス・クラスをトランザクション PAY1 に割り振る必要があります。例えば、次のように指定できます。

- 平均応答時間 2 秒のサービス・クラス Fast。
- Classification Subsystem CICS をトランザクション ID PAY1 およびサービス・クラス Fast に関連付ける分類規則。

サービス・クラスについて詳しくは、[ワークロード内の作業の管理](#)内の GOAL アルゴリズムに関する情報を参照してください。

CICSplex SM で以下を実行します。

### 1. ワークロード仕様を作成します。

- メインメニューで、「管理」 > 「ワークロード・マネージャー管理」 > 「仕様」をクリックします。これにより、「**WLM 仕様**」表形式ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。

- 「作成」ボタンをクリックして、以下の情報を入力します。

名前	WLSDYN01
説明	動的ルーティング START
主検索基準	USERID
デフォルト・ターゲット・スコープ	CSGTGTS1
異常終了する確率の許容レベル	0
許容できる異常終了ロードしきい値	0



アルゴリズムのタイプ	GOAL
------------	------

- 「はい」をクリックして、仕様を作成します。

以下を指定する必要があります。

- 仕様の名前 (この例では WLSDDYN01)。
- **主検索基準値**。この例では USERID が示されていますが、実際には、**主検索基準値**は、単純なワークロード・ルーティングには効果がないため、USERID を指定するか LUNAME を指定するかは問題ではありません。これは、ある種のワークロード分離にのみ使用されますが、CICSplex SM は、このワークロード仕様がワークロード分離に使用されないことをこの段階では認識していないため、値を指定する必要があります。
- **デフォルト・ターゲット・スコープ値**。これは、トランザクションを経路指定できる単一のターゲット領域、またはターゲット領域のグループです。
- **アルゴリズム・タイプ値**。ターゲット領域の選択基準は、トランザクションの応答時間が 2 秒であるという要件に基づいているため、この例では GOAL を使用します。

「はい」をクリックすると「**WLM 仕様**」表形式ビューが再表示され、新しいワークロード仕様 WLSDDYN01 のエントリーが示されます。

2. 次のステップは、CICSplex SM に、ワークロード要求を CSGTGTS1 内のターゲット領域にルーティングする領域について示すことです。ワークロード仕様をルーティング領域に関連付けるには、以下の手順を実行します。

- 「**WLM 仕様**」表形式ビューで、WLSDDYN01 仕様のエントリーの横にある「レコード」チェック・ボックスをクリックし、「**CICS システムの関連付け**」ボタンをクリックします。
- 「**CICS システム**」フィールドに WLSDDYN01 と入力し、「はい」をクリックします。「**WLM 仕様**」表形式ビューが再表示され、ルーティング領域とワークロード仕様との間の関連付けが作成されていることを確認できます。

3. 以下の手順を実行して、ルーティング領域でワークロード・ルーティングをアクティブにします。

- メインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックします。これにより、「**CICS システム定義**」表形式ビューが開きます。
- CICSPT01 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新」ボタンをクリックします。
- 「ワークロード・マネージャー状況」フィールドで、ドロップダウン・メニューから「はい」を選択します。この変更は、CICSPT01 を次回開始した時に有効になります。

4. ターゲット領域でワークロード・ルーティングをアクティブにします。

- メインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックします。これにより、「**CICS システム定義**」表形式ビューが開きます。
- CICSPTA01 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新」ボタンをクリックします。
- 「**AOR 動的ルーティング・モード**」フィールドで、ドロップダウン・メニューから「はい」を選択します。この変更は、ターゲット領域 CICSPTA01 を次回開始した時に有効になります。

ターゲット領域の CICSPTA02 および CICSPTA03 について、このステップを繰り返します。

5. プログラム定義をインストールします。

- メインメニューで、「管理」>「ベーシック CICS リソース管理」>「リソース定義」>「プログラム定義」をクリックします。これにより、「**プログラム定義**」表形式ビューが開きます。
- CICSPTA01 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新」ボタンをクリックします。
- ドロップダウン・ボックスから、「動的ルーティング状況」フィールドを「いいえ」に設定します。
- 「はい」をクリックして確定します。「**プログラム定義**」表形式ビューが再表示されます。
- CICSPTA01 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「インストール」ボタンをクリックします。

- ・「ターゲット・スコープ値」フィールドに CICSSPA01 と入力し、「はい」をクリックして確定します。

CICSSPA02 および CICSSPA03 について、このステップを繰り返します。

#### 6. トランザクション定義をインストールします。

- ・メインメニューで、「管理」>「ベーシック CICS リソース管理」>「リソース定義」>「トランザクション定義」をクリックします。これにより、「トランザクション定義」表形式ビューが開きます。
- ・「トランザクション定義」ビューで PAY1 を選択し、「インストール」ボタンをクリックします。「ターゲット・スコープ値」フィールドに CICSSPA01 と入力します。
- ・「はい」をクリックして確定します。

トランザクション定義を更新してこれらのフィールドの値を指定したり、トランザクション定義をインストールする時に上書き値を指定したりすることができます。

このタスク例では、要求側領域 CICSSPA01 で実行されているプログラムは、トランザクション ID PAY1 と端末 ID TRM1 を指定する **EXEC CICS START** コマンドを発行します。START コマンドは、指定された端末を所有する TOR である CICSPT01 に機能シップされます。CICSPT01 は、リージョン領域として機能し、ターゲット領域を選択する動的ルーティング出口を呼び出します。CICS システム・グループ CSGTGS1 内のすべての AOR は潜在的ターゲット領域です。実際のターゲット領域は、応答時間 2 秒の GOAL 基準に基づいて選択されます。CICSplex SM は、トランザクション PAY1 のサービス・クラス、およびそのサービス・クラスが割り振られるターゲット領域の ID を CICSplex SM 保守テーブルから入手します。

## インバウンド・クライアント DPL 要求の動的ルーティング

この例では、Web ユーザー・インターフェース (WUI) を使用して CICS クライアントからの DPL 要求の動的ルーティング機能をセットアップする方法と、GOAL アルゴリズムを使用してターゲット領域を選択する方法について説明します。

### 始める前に

インバウンド・クライアント DPL 要求の動的ルーティングを試行する前にシステム要件を確認する必要があります。[CICSplex SM ワークロード要件](#)を参照してください。

### このタスクについて

この例では、クライアント・プログラム PAYPROG1 を呼び出すためにトランザクション PAY1 を実行する要求が受信されます。PAYPROG1 は、サーバー・プログラム PAYPROG2 に **EXEC CICS LINK** コマンドを実行します。トランザクション PAY1 には 2 秒の応答時間が必要です。

この例では、100 ページの『[EXEC CICS START TERMID を使用した動的ルーティング](#)』で使用された環境が使用されます。

GOAL アルゴリズムを使用するため、必要な応答時間を持つサービス・クラスを MVS Workload Manager に定義し、そのサービス・クラスをトランザクションに割り振る必要があります。例えば、次のように指定できます。

- ・平均応答時間 2 秒のサービス・クラス Fast。
- ・Classification Subsystem CICS をトランザクション ID PAY1 およびサービス・クラス Fast に関連付ける分類規則。

サービス・クラスについて詳しくは、[ワークロード内の作業の管理](#)内の GOAL アルゴリズムに関する情報を参照してください。

CICSplex SM で以下の手順を実行します。

### 手順

1. プログラム PAYPROG1 の動的ルーティング状況値を設定します。
  - a) メインメニューで、「管理」>「ベーシック CICS リソース管理」>「リソース定義」>「プログラム定義」>をクリックします。  
これにより、「プログラム定義」表形式ビューが開きます。

- b) 現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - c) PAYPROG1 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新」ボタンをクリックします。ドロップダウン・ボックスから、「動的ルーティング状況」フィールドを「いいえ」に設定します。
  - d) 「はい」をクリックして確定します。
2. プログラム定義をインストールします。
- a) 「プログラム定義」表形式ビューで、PAYPROG1 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「インストール」ボタンをクリックします。
  - b) 「ターゲット・スコープ値」フィールドに CICSPA01 と入力します。
  - c) 「はい」をクリックして確定します。

このステップを繰り返して、ターゲット・スコープの CICSPA02 および CICSPA03 に PAYPROG1 をインストールします。

プログラム定義の定義およびインストールの詳細については、[BAS によるリソースの作成: PROGRAM リソース定義](#)を参照してください。

3. トランザクション定義をインストールします。
- a) メインメニューで、「管理」 > 「ベーシック CICS リソース管理」 > 「リソース定義」 > 「トランザクション定義」をクリックします。  
これにより、「トランザクション定義」表形式ビューが開きます。
  - b) 現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - c) 「トランザクション定義」ビューで、PAY1 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「インストール」ボタンをクリックします。
  - d) 「ターゲット・スコープ値」フィールドに CICSPA01 と入力します。
  - e) 「はい」をクリックして確定します。

このステップを繰り返して、ターゲット・スコープの CICSPA02 および CICSPA03 に PAY1 をインストールします。

トランザクション定義は、ミラー・プログラム DFHMIRS を指している必要があります。トランザクション定義の定義およびインストールの詳細については、[BAS によるリソースの作成: トランザクション・リソース定義](#)を参照してください。

## ピアツーピア DPL 要求の動的ルーティング

この例では、Web ユーザー・インターフェース (WUI) を使用してピアツーピア DPL 要求の動的ルーティングをセットアップする方法と、GOAL アルゴリズムを使用してターゲット領域を選択する方法について説明します。

**注:** システム要件を確認してから、ピアツーピア DPL 要求の動的ルーティングを試行する必要があります。[CICSplex SM ワークロード要件](#)を参照してください。

この例で、トランザクション PAY1 はプログラム PAYPROG1 を実行し、それにより **EXEC CICS LINK** コマンドがプログラム PAYPROG2 に発行されます。トランザクション PAY1 には応答時間 4 秒が必要です。

この例では、[100 ページの『EXEC CICS START TERMID を使用した動的ルーティング』](#)で使用された環境が使用されます。ただし、これはピアツーピア動的リンク要求であるため、関連があるのは AOR だけであり、それぞれの AOR が要求領域、ルーティング領域、またはターゲット領域として機能できます。

GOAL アルゴリズムを使用するため、必要な応答時間を持つサービス・クラスを MVS Workload Manager に定義し、そのサービス・クラスをトランザクションに割り振る必要があります。例えば、次のように指定できます。

- 平均応答時間 4 秒のサービス・クラス Medium。
- Classification Subsystem CICS をトランザクション ID PAY1 およびサービス・クラス Medium に関連付ける分類規則。

サービス・クラスについて詳しくは、[ワークロード内の作業の管理](#)内の GOAL アルゴリズムに関する情報を参照してください。

CICSplex SM で以下を実行します。

1. 最初のステップは、プログラム PAYPROG1 の動的ルーティング状況値を設定することです。このためには、以下の手順に従ってください。
  - ・メインメニューで、「管理」 > 「ベーシック CICS リソース管理」 > 「リソース定義」 > 「プログラム定義」をクリックします。これにより、「プログラム定義」表形式ビューが開きます。  
現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - ・PAYPROG1 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新...」ボタンをクリックします。
  - ・ドロップダウン・ボックスから、「動的ルーティング状況」フィールドを「いいえ」に設定します。
  - ・「はい」をクリックして確定します。

2. 以下の手順に従ってプログラム定義をインストールします。
  - ・「プログラム定義」表形式ビューで、PAYPROG1 のエントリーの横にある「レコード」チェック・ボックスをクリックし、「インストール...」ボタンをクリックします。
  - ・「ターゲット・スコープ値」フィールドに CICSPA01 と入力します。
  - ・「はい」をクリックして確定します。

このステップを繰り返して、ターゲット・スコープの CICSPA02 および CICSPA03 に PAYPROG1 をインストールします。

プログラム定義の定義およびインストールの詳細については、[BAS によるリソースの作成: PROGRAM リソース定義](#)を参照してください。

プログラム定義 PAYPROG2 をこれらの領域にインストールする必要はありませんが、インストールする場合は、PAYPROG2 を動的として定義する必要があります。

3. トランザクション定義をインストールする手順は、以下のとおりです。
  - ・メインメニューで、「管理」 > 「ベーシック CICS リソース管理」 > 「リソース定義」 > 「トランザクション定義」をクリックします。これにより、「トランザクション定義」表形式ビューが開きます。  
現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
  - ・「トランザクション定義」表形式ビューで、PAY2 のエントリーの横にある「レコード」チェック・ボックスをクリックし、「インストール...」ボタンをクリックします。
  - ・「ターゲット・スコープ値」フィールドに CICSPA01 と入力します。
  - ・「はい」をクリックして確定します。

トランザクション定義は、ミラー・プログラム DFHMIRS を指している必要があります。

このステップを繰り返して、ターゲット・スコープの CICSPA02 および CICSPA03 に PAY2 をインストールします。

## CICS BTS アクティビティのルーティング

Web ユーザー・インターフェース (WUI) を使用して、CICS BTS に関連したワークロードをルーティングします。

この例では、以前の例に使用されていた構成を使用します。作業は、CICS システム・グループ CSGTGTS1 内の AOR CICSPA01、CICSPA02、および CICSPA03 から成る CICSplex PLXPROD1 内で行います。これらの3つのシステムは、ルーティング領域とターゲット領域として機能する場合があります。この他に、BTS プロセス・タイプ用の RLS ファイルが、CICS システム・グループ CSGTGTS1 内のすべての CICS システムにアクセスできること、および CICS システム・グループ CSGTGTS1 内のすべてのシステムが相互接続されているものと想定されます。

注：CICS BTS アクティビティの動的ルーティングを試行する前に、システム要件を確認する必要があります。106 ページの『CICS BTS アクティビティの分離』および動的ルーティングのための CICS リリース要件を参照してください。

1. CSGTGTS1 内のシステムのワークロード・ルーティングをアクティブにします。

- メインメニューから「管理」>「トポロジー管理」>「システム定義」をクリックします。これにより、「システム定義」表形式ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

- CICSPA01 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「更新...」ボタンをクリックします。
- 「ワークロード・マネージャー状況」フィールドと「AOR 動的ルーティング・モード」フィールド両方のドロップダウン・ボックスから「はい」を選択します。これらの変更は、ターゲット領域 CICSPA01 を次回開始した時に有効になります。

ターゲット領域の CICSPA02 および CICSPA03 について、このステップを繰り返します。

2. ワークロード仕様を作成します。

- メインメニューで、「管理」>「ワークロード・マネージャー管理」>「仕様」をクリックします。これにより、「WLM 仕様」表形式ビューが開きます。
- 「作成...」ボタンをクリックして、以下の情報を入力します。

名前	WLSCBTS
説明	CICS BTS アクティビティのルーティング
主検索基準	USERID
デフォルト・ターゲット・スコープ	CSGTGTS1
異常終了する確率の許容レベル	0
許容できる異常終了ロードしきい値	0
アルゴリズムのタイプ	LNQUEUE

- 「はい」をクリックして確定します。「WLM 仕様」表形式ビューが再表示されます。

以下を指定する必要があります。

- 仕様の名前 (この例では WLSCBTS)。
- 主検索基準値。この例では USERID が示されていますが、実際には、主検索基準値は単純なワークロード・ルーティングには影響しないため、USERID を指定するか LUNAME を指定するかは問題ではありません。これは、ある種のワークロード分離にのみ使用されますが、CICSplex SM は、この段階ではこのワークロード仕様がワークロード分離に使用されないことを判断できないため、値を指定する必要があります。
- デフォルト・ターゲット・スコープ値。これは、トランザクションを経路指定できる単一のターゲット領域、またはターゲット領域のグループです。この例では、CSGTGTS1 を使用します。
- アルゴリズム・タイプ値。この例では、LNQUEUE を使用します。LNQUEUE (リンクに依存しないキュー) アルゴリズムは、ルーティング領域とターゲット領域間のリンクのタイプを無視しますが、それ以外は、QUEUE アルゴリズムと同じルーティング基準を使用します。

3. ワークロード仕様をルーティング領域スコープ CSGTGTS1 に関連付けます。

- WLSCBTS の項目の横にある「レコード」チェック・ボックスをクリックしてから、「CICS システムの関連付け...」ボタンをクリックします。
- 「CICS システム」フィールドに CSGTGTS1 と入力します。
- 「はい」をクリックして確定します。「WLM 仕様」表形式ビューが再表示されます。

4. システム初期設定パラメーター DSRTPGM が EYU9XLOP に設定されることを指定します。



これは、CICS システム・グループ内の各領域のシステム 初期設定パラメーターで行うか、「**CICS システム定義**」ビューを使用して行うことができます。

5. 以下の手順を実行して、ワークロード管理をアクティブにします。

- メインメニューから、「**CICSplex SM 操作**」>「**CICSplex 認知の MAS**」をクリックします。これにより、「**CICSplex 認知の MAS**」表形式ビューが開きます。
- CICS システム名 CICS001 をクリックして、CICS001 の「**CICSplex 認知の MAS**」詳細ビューを開きます。
- 「**ワークロード・マネージャー状況**」フィールドで、ドロップダウン・リストから「はい」を選択して、この CICS システムのワークロードの管理を開始します。
- 「**変更の適用**」をクリックします。

このステップを繰り返して、CICS002 および CICS003 のワークロード管理をアクティブにします。

6. BTS アクティビティが非同期で実行されるようにプログラミングします。

システム・グループ CSGTGS1 内のルーティング領域とターゲット領域を次回開始すると、BTS アクティビティは、ターゲット領域内にルーティングされます。

## CICS BTS アクティビティの分離

この例では、Web ユーザー・インターフェース (WUI) を使用して、CICS BTS に関連したワークロードを分離する方法について説明します。

この例では、BTS アクティビティ BTSACT1 は親和性 LIFETIME を持っており、トランザクション ID BTS1 およびプロセス・タイプ SALES の下で実行されます。この例では、[104 ページの『CICS BTS アクティビティのルーティング』](#)の例で使用されている構成を使用します。同様に、システム 初期設定パラメーター DSRTPGM は EYU9XLOP に設定されている必要があります。

**注:** CICS BTS アクティビティの動的ルーティングを試行する前に、システム 要件を確認する必要があります。[動的ルーティングのための CICS リリース要件](#)を参照してください。

1. トランザクション・グループを作成します。

- メインメニューから「**管理ビュー**」>「**ワークロード・マネージャー管理ビュー**」>「**トランザクション・グループ定義**」をクリックします。これにより、「**トランザクション・グループ定義**」表形式ビューが開きます。

現行のコンテキストが PLXPROD1 ではない場合は、「**コンテキスト**」フィールドに PLXPROD1 を指定して「**最新表示**」をクリックします。

- 「**作成...**」をクリックして、以下の情報を入力します。

名前	TRGCBTS1
説明	CICS BTS トランザクション・グループ
親和性の関係および存続時間チェック状況	ACTIVE
主検索基準	USERID
親和性の関係	BAPPL
親和性存続時間	ACTIVITY
異常終了する確率の許容レベル	0
許容できる異常終了ロードしきい値	0
アルゴリズムのタイプ	QUEUE

ブランクのままになっているフィールドはすべて無視することができます。

- 「はい」をクリックして「**トランザクション・グループ定義**」表形式ビューを再表示します。ビューには、トランザクション・グループ TRGCBTS1 の名前が表示されています。



「親和性の関係」フィールドと「親和性存続時間」フィールドに入力する必要がある点に注意してください。これらの値は、このグループ内のトランザクションが BTS 親和性を構成しており、それらのトランザクションが同じ BTS アプリケーションから開始されている間はこの親和性が継続することを CICSplex SM に示します。それらのいずれかが別の BTS アプリケーションから開始された場合、CICSplex SM は別のターゲット領域を選択することができます。もちろん、その 2 つ目のターゲット領域で同じタイプの親和性が動作します。

2. 以下の手順を実行して、グループ TRGCBTS1 内のトランザクションを識別します。

- 「トランザクション・グループ定義」ビューから TRGCBTS1 のエントリーを選択し、「トランザクションの追加...」をクリックして「トランザクションの追加」表形式ビューを開きます。
- 「トランザクション名」フィールドに BTS1 と入力し、「はい」をクリックして、トランザクションをトランザクション・グループに追加します。「トランザクション・グループ定義」ビューが再表示されます。

3. 以下のとおり、ワークロード定義を作成します。

- メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックします。これにより、「WLM 定義」表形式ビューが開きます。
- 「作成...」ボタンをクリックして、以下の情報を入力します。

ワークロード管理定義	WLDCBTS3
説明	CICS BTS アクティビティの分離
トランザクション・グループ名	TRGCBTS1
BTS プロセス・タイプ	SALES
ターゲット・システム・セットのスコープ名	CSGTGTS1

- 「はい」をクリックします。「WLM 定義」表形式ビューが再表示されます。

これらの値は、グループ TRGCBTS1 内にある、プロセス・タイプ SALES のトランザクションが、グループ CSGTGTS1 内のターゲット領域を使用する必要があることを CICSplex SM に示します。CICSplex SM は、トランザクションの開始時点で最も適切なターゲット領域を選択できます。

4. 以下のようにして、ワークロード定義をワークロード・グループに追加します。

- 「WLM 定義」表形式ビューで、WLDCBTS1 のエントリーの横にある「レコード」チェック・ボックスをクリックし、「WLM グループへの追加...」ボタンをクリックします。「WLM グループへの追加」ビューが表示されます。
- 「リソース・グループ名」フィールドに WLDCBTS1 と入力して、「はい」をクリックします。「WLM 定義」表形式ビューが再表示されます。

5. ワークロード・グループをワークロード仕様に追加します。

- メインメニューから「管理ビュー」>「ワークロード・マネージャー管理ビュー」>「グループ」をクリックします。これにより、「WLM 仕様」表形式ビューが開きます。
- 「WLM グループ」ビューで、WLDCBTS1 の項目の横にある「レコード」チェック・ボックスをクリックしてから、「WLM 仕様への追加...」ボタンをクリックします。「WLM 仕様への追加」ビューが表示されます。
- 「仕様名」フィールドに WLDCBTS1 と入力して、「はい」をクリックします。「WLM グループ」表形式ビューが再表示されます。

## Link3270 ブリッジ・ワークロードの管理

以下の例では、Link3270 ブリッジ・ワークロードを管理するための Web ユーザー・インターフェース (WUI) の使用方法について説明します。

Link3270 ブリッジ要求の場合、要求側領域内のクライアント・アプリケーションは、EXEC CICS LINK、EXCI、または ECI を使用して Link3270 ブリッジを呼び出し、ルーター領域で実行されているブリッジ・ルーター・プログラム DFHL3270 に通信域を渡します。Link3270 ブリッジ・ワークロードのルーティングに使用されるトランザクション ID は、通信域で DFHL3270 に渡される名前であり、必ずしも、端末または

ワークステーションで入力されるトランザクション名と同じではありません。ターゲット領域には、ターゲット・トランザクションが実行されるブリッジ環境が含まれています。

CICSplex SM が使用中であり、CICSplex SM ルーティング出口 EYU9XLOP が **DTRPGM** システム初期設定パラメーターとして指定されている場合、DFHL3270 は CICS 分散プログラム・リンクを使用して EYU9XLOP に制御を渡します。ミラー・プログラム DFHMIRS は、DFHDYPDS 通信域で以下の情報を EYU9XLOP に渡します。

- Link3270 ブリッジ要求タイプを示す DYRTYPE 値 8
- DYRBRTK フィールド内の 8 文字のブリッジ・トークン
- DYRTRAN フィールド内の、ターゲット領域で実行されるトランザクションのトランザクション ID。

Link3270 ブリッジについて詳しくは、[3270 ブリッジの概要](#)を参照してください。

CICS 分散プログラム・リンクについて詳しくは、[CICS 分散プログラム・リンク](#)を参照してください。

### Link3270 ブリッジ・ワークロードのルーティング

この例で、CICSPT01 はルーター領域であり、ターゲット領域は CICSSPA01、CICSSPA02、および CICSSPA03 です。

1. 現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。
2. すべての領域で **DTRPGM** システム初期設定パラメーターを EYU9XLOP に設定します。
3. 定義を更新します。

- a. Web ユーザー・インターフェースのメインメニューから「管理」>「トポロジ管理」>「システム定義」をクリックして、「**CICS システム定義**」表形式ビューを開きます。
- b. CICSPT01 のエントリーを選択し、「更新」をクリックして CICSPT01 の詳細ビューを開きます。
- c. 「ワークロード・マネージャー状況」フィールドにスクロールダウンし、メニューから「はい」を選択します。
- d. 「**AOR 動的ルーティング・モード**」フィールドにスクロールダウンし、メニューから「はい」を選択します。
- e. ビューの最下部で「はい」をクリックして、「**CICS システム定義**」表形式ビューに戻ります。CICS システム定義ビューの説明については、[CICSplex SM の管理](#)を参照してください。

この変更は、ターゲット領域 CICSSPA01 を次回開始した時に有効になります。

CICSSPA01、CICSSPA02、および CICSSPA03 の各領域に対してこの手順を繰り返します。

4. CICS システム・グループを作成します。

- a. メインメニューから「管理」>「トポロジ管理」>「システム・グループ」をクリックして、「システム・グループ定義」表形式ビューを開きます。
- b. ビューの最下部までスクロールし、「作成」をクリックして、CSGTGTS1 というシステム・グループを作成します。

この例の CSGTGTS1 には、ターゲットとして機能する領域が含まれます。（「システム・グループ定義」ビューの説明については、[CICSplex SM の管理](#)を参照してください。）

5. ターゲット領域を CSGTGTS1 に追加します。

6. ワークロード仕様の作成

- a. メインメニューから「管理」>「ワークロード・マネージャー管理」>「仕様」をクリックして、「**WLM 仕様**」表形式ビュー (WLMSPEC オブジェクト) を開きます。
- b. ビューの最下部までスクロールし、「作成」をクリックし、以下の情報を指定します。

**WLM 仕様名**

BRSPEC01

**説明**

Link3270 ブリッジ・ワークロード

## 主検索基準

USERID

## 自動親和性作成オプション

該当せず

## デフォルト・ターゲット・スコープ

BRITGTS1

## アルゴリズムのタイプ

QUEUE

残りのフィールドは空のままにするか、デフォルトを受け入れます。

以下を指定する必要があります。

- ワークロード仕様の名前。この例では、BRSPEC01 という名前です。
- **主検索基準値**。この例では USERID が示されていますが、実際には、**主検索基準値**は、単純なワークロード・ルーティングには影響しないため、USERID を指定するか LUNAME を指定するかは問題ではありません。これは、ある種のワークロード分離にのみ使用されますが、CICSplex SM は、このワークロード仕様がワークロード分離に使用されないことをこの段階では認識していないため、値を指定する必要があります。
- **デフォルト・ターゲット・スコープ**。これは、作業が経路指定される領域または領域のグループの名前 (この例では BRITGTS1) です。
- **アルゴリズム・タイプ**。この例では QUEUE が使用されていますが、Link3270 ブリッジ要求には、QUEUE、LNQUEUE、GOAL、または LINGOAL のいずれかを指定できます。

### 7. ワークロード仕様をルーティング領域スコープに関連付けます。

- 「**WLM 仕様**」表形式ビューで、BRSPEC01 のレコードを選択し、「**CICS システムの関連付け...**」ボタンをクリックします。
- 「**CICS システム**」フィールドにルーティング領域スコープを入力し、「はい」をクリックします。

領域を次に再開すると、ワークロードはターゲット領域全体でルーティングされます。ワークロード仕様 BRSPEC01 がアクティブであることを確認するには、「**アクティブ・ワークロード**」ビュー (WLMWORK オブジェクト) を使用できます。どのターゲット領域ワークロードがルーティングされているか確認するには、「**アクティブ・ワークロードのターゲット配布係数**」ビュー (WLMAWAOR オブジェクト) を使用できます。

## Link3270 ブリッジ・ワークロードの分離

ユーザー ID、トランザクション・グループ、または LU 名を使用して Link3270 ブリッジ・ワークロードを分離することができます。

## このタスクについて

LU 名による分離は、ブリッジが自動的に生成する NETNAME を上書きする場合のみ可能です。Link3270 ブリッジ・ワークロードを分離する際の LUNAME の使用に関する制約に注意してください。詳しくは、[Link3270 ブリッジのワークロードの分離](#)を参照してください。Link3270 ブリッジ・ワークロードの場合、LU 名は、ブリッジ機能自体の NETNAME ではなく、Link3270 ブリッジを開始したクライアント・トランザクションを実行している端末の 8 文字の NETNAME です。

ブリッジ機能 NETNAME で分離するには、EYU9WRAM モジュールを変更する必要があります。詳しくは、[Link3270 ブリッジのワークロードの分離](#)を参照してください。CICS は、同じブリッジ機能の下で実行されているすべてのトランザクションを同じターゲット領域に経路指定します。強制的に別の領域に経路指定することはできません。

以下の例では、ユーザー ID およびトランザクション・グループでブリッジ・ワークロードを分離する方法について説明します。この例では、[76 ページの『ワークロード管理定義の作成』](#)で説明されている構成と同じ構成が使用されます。

[76 ページの『ワークロード管理定義の作成』](#)で作成した定義に以下の定義を追加します。

## 手順

1. トランザクション・グループを作成します。

- a) Web ユーザー・インターフェースのメインメニューから「管理」>「ワークロード・マネージャー管理」>「トランザクション・グループ定義」をクリックします。

「トランザクション・グループ定義」ビューが開き、PLXPROD1 に既に定義されているすべてのトランザクションがリストされます。

現行のコンテキストが PLXPROD1 ではない場合は、「コンテキスト」フィールドに PLXPROD1 を指定して「最新表示」をクリックします。

- b) ビューの最下部までスクロールし、「作成」をクリックし、以下の情報を指定します。

**トランザクション・グループ名**

TRGBRIO1

### 説明

Link3270 ブリッジ・トランザクション・グループ

**親和性の関係および存続時間チェック状況**

ACTIVE

**主検索基準**

ユーザー ID

**自動親和性作成オプション**

該当せず

残りのフィールドはブランクのままにするか、デフォルトを受け入れることができます。

- c) 「はい」をクリックします。

「トランザクション・グループ定義」ビューが再表示され、今回はトランザクション・グループ TRGBRIO1 の名前が表示されています。

CICSplex SM は Link3270 ブリッジ・トランザクション間の親和性を処理しないため、「親和性の関係」フィールドと「親和性存続時間」フィールドはブランクのままにし、「自動親和性作成オプション」フィールドは「N/A」のままにします。

**注:** LUNAME で分離したい場合は、「トランザクション・グループ定義」作成ビューの主検索基準フィールドに LUNAME と入力する必要があります。

2. グループ TRGBRIO1 内のトランザクションを識別します。

- a) 「トランザクション・グループ定義」ビューで TRGPAY03 のエントリーを選択し、「トランザクションの追加」をクリックします。

- b) 「トランザクション名」フィールドに名前 BRI1 を入力し、「はい」をクリックして確定します。

「トランザクション・グループ定義」ビューが再表示されます。

3. 別の領域に経路指定したいその他のトランザクションに対して、前の 2 つのステップを繰り返してください。

4. ワークロード定義を作成します。

- a) メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックします。「ワークロード管理定義」ビューが開きます。

- b) ビューの最下部までスクロールし、「作成」をクリックし、以下の情報を指定します。

**ワークロード管理定義名**

WLDBRIO1

### 説明

Link3270 ブリッジ・アクティビティの分離

**トランザクション・グループ名**

TRGBRIO1

**端末 LU 名**

\*

#### ユーザー ID

BRIUSER1

#### BTS プロセス・タイプ

\*

#### ターゲット・システム・セットのスコープ名

CICSPA01

これらの値は、BRIUSER1 によって入力されたグループ TRGBRI01 内のトランザクションが CICSPA01 に経路指定されることを示しています。

- c) 「はい」をクリックして確定します。

注: LUNAME で分離したい場合は、「ワークロード管理定義」作成ビューの「端末 LU 名」フィールドに LUNAME を入力する必要があります。

5. 別の領域に経路指定するトランザクション用に作成したその他のトランザクション・グループに対して、前のステップを繰り返します。
6. ワークロード・グループを作成する。

トランザクションをルーティングするルーティング領域が開始された時にワークロード定義を自動的にインストールする場合は、ワークロード・グループが不可欠です。

- a) メインメニューから「管理」>「ワークロード・マネージャー管理」>「グループ」をクリックします。

「ワークロード管理グループ」ビューが開き、PLXPROD1 に既に作成されているすべてのワークロードがリストされます。

- b) 「作成」をクリックし、以下の情報を入力します。

#### ワークロード管理グループ名

WLGBRI01

#### 説明

WLDBRI01 のワークロード・グループ

- c) 「はい」をクリックして確定します。

「ワークロード管理グループ」ビューが再表示されます。

7. ワークロード定義 WLDBRI01 とその他の作成済みのすべてのワークロード定義を WLGBRI01 に追加します。

- a) メインメニューから「管理」>「ワークロード・マネージャー管理」>「定義」をクリックします。

「ワークロード管理定義」ビューが開きます。

- b) WLDBRI01 のエントリーを選択し、「WLM グループに追加」をクリックし、以下の情報を指定します。

#### ワークロード管理定義名

WLDPAY01

#### 説明

Link3270 ブリッジ・ワークロード定義

#### リソース・グループ名

WLGBRI01

- c) 「はい」をクリックして確定します。

- d) 以前に作成したすべての追加ワークロード定義に対して、ステップを繰り返します。

8. ワークロード・グループ WLGBRI01 をワークロード仕様 BRSPEC01 に追加します。

- a) メインメニューから「管理」>「ワークロード・マネージャー管理」>「グループ」をクリックします。

「ワークロード管理 (WLM) グループ」ビューが開きます。

- b) WLGBRI01 のエントリーを選択し、「WLM 仕様に追加」をクリックし、「仕様名」フィールドに BRSPEC01 と入力します。

- c) 「はい」をクリックして確定します。

9. 定義をアクティブ・ワークロードにインストールします。

- a) メインメニューから「管理」 > 「ワークロード・マネージャー管理」 > 「定義」をクリックします。  
「ワークロード管理定義」ビューが開きます。
- b) WLDBRI01 のエントリーを選択し、「インストール」をクリックしてアクティブ・ワークロードにインストールします。「はい」をクリックして確定します。  
作成したワークロード分離定義は即時に有効になります。
- c) 「はい」をクリックして確定します。

#### 次のタスク

「アクティブ・ワークロードにインストールされているワークロード定義 (Workload definition installed in active workload)」ビューを使用して、ワークロード定義がアクティブであることを確認できます。



## 第5章 アクティブ・ワークロード・ビュー

「アクティブ・ワークロード」ビューには、アクティブなワークロードや、それらのワークロードを構成するトランザクションとトランザクション・グループに関する情報が表示されます。注: ワークロード管理は CICSplex 全体ベースで実行します。これらのビューは、すべての有効なスコープを無視します。

### アクティブ・ワークロード - WLMWORK

「アクティブ・ワークロード」(WLMWORK) ビューは、コンテキストとして指定される CICSplex 中のアクティブ・ワークロードに関する情報を表示します。

#### 提供されるビュー

メインメニューからアクセスするには、

「アクティブ作業負荷ビュー (Active workload views)」 > 「アクティブ・ワークロード (Active workloads)」

表 8. 提供された「アクティブ・ワークロード」(WLMWORK) ビュー・セットのビュー

ビュー	注
アクティブ・ワークロード EYUSTARTWLMWORK.DETAILED	選択したアクティブ・ワークロードに関する詳細情報
アクティブ・ワークロード EYUSTARTWLMWORK.SET	アクティブなワークロードの属性を設定します。
アクティブ・ワークロード EYUSTARTWLMWORK.TABULAR	現行コンテキスト内のすべてのアクティブ・ワークロードに関するテーブル形式の情報

#### アクション

表 9. WLMWORK ビューに使用可能なアクション

アクション	説明
GET	オプション・パラメーター。 ワークロードを管理するすべての CMAS からレコードを要求する場合には、YES にします。 ワークロードを管理する 1 つの CMAS からレコードを要求する場合は、NO にします。 パラメーターを指定しない場合のデフォルト値は NO です。
SET	アクティブなワークロードの属性を設定します。

#### フィールド

表 10. WLMWORK ビューのフィールド

フィールド	属性名	説明
異常終了した確率の許容レベル	ABENDCRIT	ターゲット領域が正常でないと見なされる原因となる、デフォルトのトランザクション・グループに関連したトランザクションの異常終了の確率。 値 0 は、WLM がワークロードの異常終了の確率を計算していないという意味です。

表 10. WLMWORK ビューのフィールド (続き)		
フィールド	属性名	説明
ターゲット領域の許容負荷レベル	ABENDTHRESH	<p>ターゲット領域の負荷レベルが倍になる原因となる、デフォルトのトランザクション・グループに関連したトランザクションの異常終了の確率。</p> <p>値 0 は、WLM がワークロードの異常終了負荷を計算していないという意味です。</p>
自動親和性作成	AFFAUTO	<p>CICSplex SM が、インストールされているどのトランザクション・グループにも関連していないトランザクションの親和性の関係を自動的に作成するかどうかを示します。</p> <ul style="list-style-type: none"> <li>• YES <ul style="list-style-type: none"> <li>– 親和性は「親和性の関係」および「親和性存続時間」フィールドの値を使用して作成されます。</li> </ul> </li> <li>• NO <ul style="list-style-type: none"> <li>– 親和性は自動的に作成されません (ただし、動的ルーティング・プログラム EYU9WRAM のカスタマイズ版で作成することが可能です)。</li> </ul> </li> <li>• 該当なし <ul style="list-style-type: none"> <li>– 「親和性の関係」および「親和性存続時間」フィールドに値がないので、親和性は作成されません。</li> </ul> </li> </ul>
デフォルト親和性関係	AFFINITY	<p>インストールされているどのトランザクション・グループにも関連していないトランザクションに適用される、デフォルトの親和性の関係。親和性の関係値には、次のものがあります。</p> <ul style="list-style-type: none"> <li>• GLOBAL - すべての端末のすべてのユーザー。</li> <li>• LUNAME - 端末の論理装置名。</li> <li>• USERID - ユーザー ID</li> <li>• BAPPL - ビジネス・アプリケーション。</li> <li>• Locked - LOCKED の親和性は、動的にリンクされたプログラム間でのみ生じます。LOCKED の親和性は、呼び出し元に戻された後に保存される状態データを、呼び出されたプログラムが保持する際に作成されます。このタイプの親和性を持つプログラムは、作業単位が終了するまで、同じターゲット領域に送られます。LOCKED を使用できるのは、作業単位の親和性存続時間が関連付けられている動的プログラム・リンク (DPL) 要求の場合に限られます。</li> <li>• NONE - 親和性関係は定義されていません。</li> </ul>
デフォルト親和性存続時間	AFFLIFE	<p>デフォルトの親和性の関係と併用するデフォルト親和性存続時間。この値は、インストールされているどのトランザクション・グループにも関連していないトランザクションに適用されます。トランザクションの最初のインスタンスに続くインスタンスは、以下のように同じターゲット領域で実行されます。</p> <ul style="list-style-type: none"> <li>• DELIMIT - 疑似会話モードが END となるまで。</li> <li>• LOGON - 端末セッションの期間内。</li> <li>• PCONV - 疑似会話の期間内。</li> <li>• PERMANENT - ワークロードがアクティブである間。</li> <li>• SIGNON - 端末ユーザーがサインオフするまで。</li> <li>• SYSTEM - ターゲット領域が終了するまで。</li> <li>• ACTIVITY - CICS BTS の活動がアクティブである間。</li> <li>• PROCESS - CICS BTS プロセスがアクティブである間。</li> <li>• UOW - 作業単位がアクティブである間。</li> <li>• NONE - 親和性存続時間は定義されていません。</li> </ul>

表 10. WLMWORK ビューのフィールド (続き)

フィールド	属性名	説明
デフォルトのアルゴリズム・タイプ	ALGTYPE	<p>インストールされているどのトランザクション・グループにも関連していないすべてのトランザクションに適用されるアルゴリズム:</p> <ul style="list-style-type: none"> <li>• <b>QUEUE</b> - ルーティング評価の時点でルーティングの重みが最小だったターゲット領域にトランザクションを送信します。ターゲット領域のルーティングの重みは、以下の係数の組み合わせになります。 <ul style="list-style-type: none"> <li>- 現在のタスク・ロード</li> <li>- 現在のターゲットの正常性の状態</li> <li>- ルーターからターゲットまでのリンク速度</li> <li>- ターゲットでトランザクションが異常終了する可能性</li> <li>- ターゲットで未解決になっている RTA イベントの影響</li> </ul> </li> <li>• <b>LNQUEUE</b> - ルーティング評価の時点でルーティングの重みが最小だったターゲット領域にトランザクションを送信します。ターゲット領域のルーティングの重みは、以下の係数の組み合わせになります。 <ul style="list-style-type: none"> <li>- 現在のタスク・ロード</li> <li>- 現在のターゲットの正常性の状態</li> <li>- ターゲットで未解決になっている RTA イベントの影響</li> <li>- ターゲットでトランザクションが異常終了する可能性</li> </ul> </li> </ul> <p>注 - <b>LNQUEUE</b> アルゴリズムのルーティングの重みの計算では、ルーターからターゲットまでのリンク速度は、係数として使用されません。</p> <ul style="list-style-type: none"> <li>• <b>GOAL</b> - z/OS Workload Manager で事前定義されているトランザクションの平均応答時間の目標値に到達する可能性が最も高いターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- 設定されている GOAL の評価から特定のターゲット領域に解決できない場合は、残りのターゲット・セットに <b>QUEUE</b> アルゴリズムが適用されます。</li> </ul> </li> <li>• <b>LNGOAL</b> - z/OS Workload Manager で事前定義されているトランザクションの平均応答時間の目標値に到達する可能性が最も高いターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- 設定されている LNGOAL の評価から特定のターゲット領域に解決できない場合は、残りのターゲット・セットに <b>LNQUEUE</b> アルゴリズムが適用されます。</li> </ul> </li> </ul>
アクティブなターゲット領域	AORCNT	ワークロードのターゲット領域として識別される、アクティブな CICS システムの数。
デフォルト・ターゲット・スコープ	AORSCOPE	トランザクション・グループかプログラム・グループに関連していない動的トランザクションかプログラムのターゲットになる CICS システムか CICS システム・グループの名前。
説明	DESC	ワークロードの説明。
イベント名	EVENTNAME	<p>このワークロードのトランザクションのルーティングに影響するリアルタイム分析イベントの名前。リアルタイム分析イベントが生成される場合、WLM はターゲット選択基準の一部としてこの情報を使用します。</p> <p>ワークロード仕様の作成時に、オプションでイベントがワークロードに関連付けられます。このフィールドがブランクの場合は、どのイベントもワークロードと関連付けられません。</p> <p>イベント名は、インストール済みのリアルタイム分析定義 (RTADEF) または状況定義 (STATDEF) と同じです。</p>

表 10. WLMWORK ビューのフィールド (続き)

フィールド	属性名	説明
最適化の状況	OWSTATE	<p>このフィールドは、現在のワークロードの sysplex 最適化状況を報告します。値は、すべてのワークロード・ルーター最適化状況と、すべてのワークロード・ターゲット最適化状況を融合したものです。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ACTIVE <ul style="list-style-type: none"> <li>- ワークロードのすべてのターゲットとルーターは最適化されたワークロード状態で実行されています。</li> </ul> </li> <li>• PARTIAL <ul style="list-style-type: none"> <li>- 少なくとも 1 つのターゲットと 1 つルーターが最適化されたワークロード・モードで実行されています。アクティブなルーティング領域とアクティブなターゲット領域のハイパーリンクを使用して、最適化された状態で実行されない領域を判別します。</li> </ul> </li> <li>• INACTIVE <ul style="list-style-type: none"> <li>- ワークロードは、以下の 1 つ以上の理由で最適化された状態で実行されていません。 <ul style="list-style-type: none"> <li>- 最適化された状態で実行されているルーティング領域がワークロードにない。</li> <li>- 最適化された状態で実行されているターゲット領域がワークロードにない。</li> <li>- 最適化された状態で実行されている領域がワークロードにない。</li> <li>- 最適化されたワークロードのルーティング機能が現行のターゲット領域に対して無効である。</li> <li>- このワークロードの CICSplex 定義または CICS システム 定義で RS サーバーの更新頻度として 0 の値が指定され、ワークロードが非最適化状態であると示されている。</li> </ul> </li> </ul> </li> </ul>
CMAS のレポート	RPTCMAS	<p>このレコードの情報を提供した CMAS の名前。</p> <p>GET 操作のオプション・パラメーター EXPLODE が指定されていないか NO として指定されている場合、この属性はブランクに設定され 1 つのレコードのみがワークロードに対して戻されます。</p> <p>GET 操作のオプション・パラメーター EXPLODE が YES として指定されている場合、この属性は、このレコードの情報を提供した CMAS の名前で設定されます。ワークロードを管理する各 CMAS からワークロードのレコードが 1 つずつ返されます。</p> <p>IBM 分散 WUI のメニューやビューを使用して WLMWORK 表形式ビューを要求する場合、EXPLODE パラメーターは YES に設定されます。</p>
共用状況	SHARED	<p>ワークロードが、CICS BTS 関連の定義のインストールをサポートしていないバックレベルの CMAS と共用されているかどうかを示します。</p>

表 10. WLMWORK ビューのフィールド (続き)		
フィールド	属性名	説明
ワークロードの状況	STATUS	<p>ワークロードの状況は次のいずれかになります。</p> <ul style="list-style-type: none"> <li>• <b>ACTIVE</b> - このワークロードが依存するすべてのターゲットとルーティング領域の接続は有効です。</li> <li>• <b>FROZEN</b> - このワークロードが従属しているターゲットまたはルーティング領域の接続は失われています。</li> </ul> <p>ワークロードが凍結されている間、次のようになります。</p> <ul style="list-style-type: none"> <li>• トランザクションは引き続き既存の親和性に従って経路指定されます。</li> <li>• 他の CMAS に関係していない限り、新しい親和性を作成できます。他の CMAS に通知する必要があるグローバルまたは永続的な親和性は作成できません。</li> <li>• ワークロードのどの局面でもインストールおよび破棄アクション・コマンドは使用できません。</li> </ul> <p>接続が失われた原因は、ターゲットまたはルーティング領域、ワークロードを管理する CMAS の 1 つ、または CMAS 間の接続である可能性があります。ワークロードのフリーズ理由を判別するには、以下のようになります。</p> <ul style="list-style-type: none"> <li>• 「アクティブ・ワークロードのターゲット領域 (WLMAWAOR)」ビューと「アクティブ・ワークロードのルーティング領域 (WLMAWTOR)」ビューの接続状況を確認して、影響を受けているターゲット領域またはルーティング領域を判別します。</li> <li>• MAS ビューを確認して、ターゲットまたはルーティング領域の状況を判別します。</li> <li>• ターゲットまたはルーティング領域のすべての接続状況を確認します。</li> </ul> <p>失われた接続を判別し、接続を再確立するための適切なアクションを実行します。</p>
アクティブなルーティング領域	TORCNT	ルーティング領域として識別される、ワークロードに関連したアクティブな CICS システムの数。
アクティブ・トランザクション・グループの親和性	WAFFCNT	ワークロードに関連したアクティブなトランザクション・グループの親和性の数。
アクティブ・ワークロード定義	WDFCNT	ワークロードに関連したアクティブなワークロード定義の数。
名前	WORKLOAD	ワークロードの名前。これは、インストール済みのワークロード仕様定義の名前でもあります。
所有システム ID	WRKLOWNER	ワークロードを作成した CMAS の CICS システム ID。
アクティブ・トランザクション・グループ	WTGPCNT	ワークロードに関連したアクティブなトランザクション・グループの数。
アクティブ動的トランザクション	WTRNCNT	ワークロードに関連したアクティブな動的トランザクションの数。

## アクティブなルーティング領域 - WLMAWTOR

「アクティブ・ワークロード内のルーティング領域」(WLMAWTOR) ビューは、コンテキストとして識別される CICSplex 中のワークロードに関連したすべてのアクティブなルーティング領域に関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「**アクティブ作業負荷ビュー (Active workload views)**」>「**アクティブなルーティング領域 (Active routing regions)**」

表 11. 提供されている「アクティブ・ワークロードのルーティング領域」(WMAWTOR) ビュー・セットのビュー	
ビュー	注
アクティブ・ワークロードのルーティング領域 EYUSTARTWMAWTOR.DETAILED	選択したアクティブ・ルーティング領域に関する詳細情報
アクティブ・ワークロードのルーティング領域 EYUSTARTWMAWTOR.TABULAR	ワークロードに関連したすべてのアクティブ・ルーティング領域に関するテーブル形式の情報

## アクション

表 12. WMAWTOR ビューに使用可能なアクション	
アクション	説明
GET	ワークロード仕様の名前。ワークロード名 '*' を指定すると、すべてのワークロードにおけるすべてのルーティング領域に関する詳細を戻します。他の種類の汎用ワークロード名を指定することはできません。

## フィールド

表 13. WMAWTOR ビューのフィールド		
フィールド	属性名	説明
ルーティング領域コンタクト状況	CON_STATUS	<p>ルーティング領域と CMAS の間の接続の状況。値は次のとおりです。</p> <ul style="list-style-type: none"> <li>LOSTCON - ルーティング領域と CMAS の間の接続は失われています。</li> <li>N_A - ルーティング領域と、ワークロードを管理する CMAS との間の接続は使用可能です。</li> </ul>
最適化の状況	OWSTATE	<p>このフィールドは、最適化されたワークロード・ルーティング機能の現行ルーティング領域の状況を報告します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>ACTIVE <ul style="list-style-type: none"> <li>このルーター領域は、最適化されたワークロード状態で実行されています。</li> </ul> </li> <li>INACTIVE <ul style="list-style-type: none"> <li>この領域は最適化されたワークロード状態で実行可能ですが、以下の1つ以上の理由で、現在最適化されていません。 <ul style="list-style-type: none"> <li>領域に RS サーバーへの接続がない。</li> <li>領域は RS サーバーに接続されているが、サーバーがカップリング・ファシリティーに接続できていない。</li> <li>最適化されたワークロードのルーティング機能が現行のターゲット領域に対して無効である。</li> <li>この領域の RS サーバー更新頻度の値が 0 である、すなわち、この領域の最適化機能が使用可能にされていない。</li> </ul> </li> </ul> </li> <li>N_A <ul style="list-style-type: none"> <li>このルーター領域は、最適化されたワークロード管理を提供するように構成されていません。最適化されていない WLM ルーティング決定しか行えません。</li> </ul> </li> </ul> <p>WUI が使用している CMAS が領域状況 (RS) サーバーに接続されていない場合、リモート領域の最適化状況は更新されず、デフォルトの INACTIVE になります。そのような状態になっている場合は、すべての報告 CMAS を RS サーバーに接続するか、「アクティブなターゲット 領域」ビューを使用して最適化状況をチェックしてください。</p>



表 13. WLMAWTOR ビューのフィールド (続き)		
フィールド	属性名	説明
CMAS のレポート	RPTCMAS	このレコードの情報を提供した CMAS の名前。  GET 操作のオプション・パラメーター EXPLODE が指定されていないか NO として指定されている場合、この属性はブランクに設定され 1 つのレコードのみがルーティング領域に対して戻されます。  GET 操作のオプション・パラメーター EXPLODE が YES として指定されている場合、この属性は、このレコードの情報を提供した CMAS の名前で設定されます。ルーティング領域に関連するワークロードを管理する各 CMAS からルーティング領域のレコードが 1 つずつ返されます。  IBM 分散 WUI のメニューやビューを使用して WLMAWTOR 表形式ビューを要求する場合、EXPLODE パラメーターは YES に設定されます。
ルーティング領域名	TOR	ルーティング領域の働きをしており、ワークロードに関連しているアクティブな CICS システムの名前。
コントロール CMAS 名	TOROWNER	コントロール CMAS の名前。
ワークロード名	WORKLOAD	ワークロード仕様の名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## アクティブ・ワークロードのターゲット配布係数 - WLMAWAOR

「アクティブなワークロードでのターゲット領域」(WLMAWAOR) ビューは、コンテキストとして識別される CICSplex 中のワークロードに関連したすべてのターゲット領域に関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「アクティブ作業負荷ビュー (Active workload views)」 > 「アクティブ・ワークロードのターゲット配布係数 (Active workload target distribution factors)」

表 14. 提供されている「アクティブなワークロードでのターゲット領域」(WLMAWAOR) ビュー・セットのビュー	
ビュー	注
アクティブなワークロードでのターゲット領域 EYUSTARTWLMAWAOR.ACTIVATE	指定されたターゲット領域が認識され、ワークロード管理に参加できるようにします。
アクティブなワークロードでのターゲット領域 EYUSTARTWLMAWAOR.DETAILED	選択したターゲット領域に関する詳細情報
アクティブなワークロードでのターゲット領域 EYUSTARTWLMAWAOR.QUIESCE	指定されたターゲット領域がワークロード管理から除去されるようにします。  この場合、追加のトランザクションは、古いトランザクションと親和性を共用しない限り、このターゲット領域に経路指定されません。例えば、ターゲット領域の静止の開始時に疑似会話に関与している場合、疑似会話を構成するトランザクションは、親和性存続時間が終了するまで引き続き同じターゲット領域に経路指定されます。親和性がアクティブでなくなると、ターゲット領域は完全に静止し、それ以降のトランザクションはターゲット・スコープ内の別のターゲット領域に経路指定されます。
アクティブなワークロードでのターゲット領域 EYUSTARTWLMAWAOR.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したターゲット領域に関するテーブル形式の情報。
アクティブなワークロードでのターゲット領域 EYUSTARTWLMAWAOR.TABULAR2	コンテキストと識別される CICSplex 内のワークロードに関連したターゲット領域に関するテーブル形式の情報。

## アクション

表 15. WLMWAOR ビューに使用可能なアクション	
アクション	説明
ACTIVATE	指定されたターゲット領域が認識され、ワークロード管理に参加できるようにします。
QUIESCE	指定されたターゲット領域がワークロード管理から除去されるようにします。  この場合、追加のトランザクションは、古いトランザクションと親和性を共用しない限り、このターゲット領域に経路指定されません。例えば、ターゲット領域の静止の開始時に疑似会話に参与している場合、疑似会話を構成するトランザクションは、親和性存続時間が終了するまで引き続き同じターゲット領域に経路指定されます。親和性がアクティブでなくなると、ターゲット領域は完全に静止し、それ以降のトランザクションはターゲット・スコープ内の別のターゲット領域に経路指定されます。

## フィールド

表 16. WLMWAOR ビューのフィールド		
フィールド	属性名	説明
ターゲット領域名	AOR	ターゲット領域の働きをしており、ワークロードに関連しているアクティブな CICS システムの名前。
カップリング・ファシリティ更新数	CFUPDCNT	これは、RS サーバーがこのターゲット領域の z/OS カップリング・ファシリティに対して行った更新呼び出し回数です。  RS サーバーまたは CICS 領域が開始されるたびに、この値はリセットされます。
ターゲット領域コンタクト状況	CON_STATUS	ターゲット領域と CMAS の間の接続の状況。値は次のとおりです。 <ul style="list-style-type: none"> <li>LOSTCON - ターゲット領域と CMAS の間の接続は失われています。</li> <li>N_A - ターゲット領域と、ワークロードを管理する CMAS との間の接続は使用可能です。</li> </ul>
CICSplex SM バージョン	CPSMVER	ターゲット領域の CICSplex SM バージョンです。
アクティブな RTA イベント	EVENTS	現在の領域の未解決の RTA イベントの標識です。これにはワークロード仕様が関連付けられます。ただし、トランザクション名 (TRANNAME) パラメータが指定されていて値が '*' でない場合は、その特定のトランザクションを管理する現在のワークロードの TRANGRP が関連付けられます。この値は現在の領域をターゲットとするワークロードごとに異なる可能性があり、それに伴いその領域のルーティングの重みも異なります。
領域のダンプ	HLTHDUMP	これはルーティング・ターゲットが現在トランザクション・ダンプまたはシステム・ダンプを発行しているかどうかを示します。
領域が MAXTASKS にある	HLTHMAXT	これはルーティング・ターゲットがその最大タスクしきい値に達したタスク・レートで、現在実行されているかどうかを示します。
MAS が応答していない	HLTHNRM	このフィールドは、ルーティング・ターゲットが現時点で応答していないことを示します。
ストレージ不足	HLTHSOS	このフィールドは、現在時刻におけるこのルーティング・ターゲットの SOS 状況を示します。CICS 領域では瞬時に SOS 状況が切り替わります。
領域の停止	HLTHSTALL	これは、ルーティング・ターゲットが停止状態になっているかどうかを示します。領域が停止している場合、アクティブ・タスク・ビューを使用して、中断されているタスクとその原因を判別します。注: ここでの停止とは、タスクが特定の時間同じリソースを待機していることを意味しています。
領域の最大タスク数	MAXTASKS	これはこのルーティング・ターゲットで同時に実行できるアクティブ・タスクの、定義された最大数です。

フィールド	属性名	説明
最適化の状況	OWSTATE	<p>このフィールドは、最適化されたワークロード・ルーティング機能の現在のターゲット領域の状況を報告します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ACTIVE <ul style="list-style-type: none"> <li>- このターゲット領域は、最適化されたワークロード状態で実行されています。</li> </ul> </li> <li>• INACTIVE <ul style="list-style-type: none"> <li>- この領域は最適化されたワークロード状態で実行可能ですが、以下の1つ以上の理由で、現在最適化されていません。 <ul style="list-style-type: none"> <li>- 領域に RS サーバーへの接続がない。</li> <li>- 領域は RS サーバーに接続されているが、サーバーが z/OS カップリング・ファシリティに接続できていない。</li> <li>- 最適化されたワークロードのルーティング機能が現行のターゲット領域に対して無効である。</li> <li>- この領域の RS サーバー更新頻度の値が 0 である、すなわち、この領域の最適化機能が使用可能にされていない。</li> </ul> </li> </ul> </li> <li>• N_A <ul style="list-style-type: none"> <li>- このターゲット領域は、最適化されたワークロード管理を提供するように構成されていません。最適化されていない WLM ルーティング決定しか行えません。</li> </ul> </li> </ul> <p>WUI が使用している CMAS が領域状況 (RS) サーバーに接続されていない場合、リモート領域の最適化状況は更新されず、デフォルトの INACTIVE になります。そのような状態になっている場合は、すべての報告 CMAS を RS サーバーに接続するか、「アクティブなターゲット領域」ビューを使用して最適化状況をチェックしてください。</p>
RS サーバー読み取り間隔	READRS	<p>最適化モードで実行されるワークロードにおいて、この値は CICS CFDT サーバーからのターゲット領域状況の最新表示の最小間隔を指定します。これらのリフレッシュ要求は、動的ルーティング要求のターゲット候補としてこの領域を評価しているルーティング領域によって発行されます。</p> <p>値の範囲は 0 から 2000 で、ミリ秒単位で示されます。</p> <ul style="list-style-type: none"> <li>• 最適化状況がアクティブに設定されていないとき、この値は 0 に設定され、WLM はそれを無視します。最適化の状況が「アクティブ」に設定されている場合の値 0 は、ターゲット領域の状況を調べるたびに、その領域の状況更新をルーティング領域が要求することを意味します。</li> <li>• この領域の状況を最新表示するまでの最小時間間隔を指定する場合は、1 から 2000 までの値を使用します。</li> </ul> <p>間隔値が小さい場合、より大きい値の場合よりも頻繁に、CFDT サーバーをポーリングして状況を更新します。ワークロードが QUEUE モードの場合は、これによる結果として、ワークロードのターゲット・スコープ内の CICS 領域にわたってタスク・ロードがより均等に分散されます<b>(他の正常性およびリンクの要因がすべて同等であると仮定した場合)</b>。しかし、RS サーバーの使用率がそれに対応して増加し、その結果 z/OS カップリング・ファシリティの使用率が増加することがあります。</p> <p>値 1000 は、1 秒の間隔を表します。デフォルト値は 200 ミリ秒です。</p> <p>この値は CICSplex ビューまたは CICS システム定義ビューに認識されている MAS を使用して、この特定の領域に対応するように変更される場合があります。</p>

表 16. WLMWAOR ビューのフィールド (続き)		
フィールド	属性名	説明
領域の WLM ルーティングの重み	ROUTEWGHT	<p>トランザクション名 (TRANNAME) パラメーターで指定されているトランザクション名に基づいて計算される現在のルーティング・ターゲットのルーティングの重みです。トランザクション名の値が '*' の場合は、ワークロード仕様により提示されるデフォルト・ルートのルーティングの重みになります。この値はルーターからのリンク重みと結合され、この CICS 領域の正規化された全体のルーティングの重みを決定します。その値は、他のルーティング・ターゲット候補の同じ値と比較されます。親和性が未解決でないことを前提として、最も低いルーティングの重みを持つターゲットが実際のルーティング・ターゲットとして選択されます。</p> <p>このターゲット領域に関連したワークロードで、RTA MRM イベントが特定のトランザクション・コードの経路決定に影響を及ぼすように指定する場合、未解決の RTA イベントの影響のため、異なるトランザクション名は異なるルーティングの重みになります。</p> <p>注: 2147483647 は、ターゲットが静止しているか、z/OS WLM ヘルスが 0 であることを示す特殊値です。この値のターゲット領域は、ワークロード管理から除去されます。この場合、追加のトランザクションは、このターゲット領域に経路指定されません。</p> <p>一般的に、リンク重みは遅い接続よりも、より速い接続を好むために使用されます。重み付けは次のとおりです。</p> <ul style="list-style-type: none"> <li>• MRO は XCF (クロス・カップリング・ファシリティー) 接続より速いと見なされます。</li> <li>• XCF は IPIC より速いと見なされます。</li> <li>• IPIC は VTAM より速いと見なされます。</li> <li>• VTAM は間接接続より速いと見なされます。</li> </ul>
ルーティング・ロード	ROUTINGLOAD	<p>これは、ターゲット領域の WLM 経路の決定で使用するルーティング・ロードです。これは、この照会時点での実際の CICS タスク数と一致しないことに注意してください。</p> <p>アクティブなシステムの最適化された WLM ターゲット領域の場合、これは、「タスク数の増分値」、「RS サーバー下部層」および「RS サーバー上部層」を考慮に入れた、ターゲット領域での概算のタスク・ロードになります。この値の精度は、「RS サーバー読み取り間隔」設定と「RS サーバー更新頻度」設定にも影響を受けます。</p> <p>RS サーバーに接続されておらず、そのため最適化されていない WLM 状態にあるターゲット領域の場合、実際のタスク数と比較すると、この値はあまり正確でないようにみえます。それは、これがこの領域の正規化されたルーティング・ロードであるためです。最適化されていない WLM 状態で実行しているとき、この値は MAX TASK フラグを設定せずに MAX TASK 値を超えることができ、MAX TASK 標識が実際のタスク数から取得されたターゲット領域のヘルス状態に関連しているとき、この値は正しい動作であることを注意してください。</p> <p>領域が複数のワークロードのターゲットになり、異なるルーティングの重み値を示す場合があります。これは異常な動作ではありません。ワークロード・マネージャー内の同期待ち時間の症状です。ワークロードの最適化は、この効果を無効にします。ワークロード・マネージャーのランタイム・プロセスは、ロード値におけるこの不均衡が生じて、正常に実行し続けます。このようにロードで観測される相違を抑制するには、ワークロードに関係するすべての MAS と CMAS が、接続可能な領域状況サーバーを使用できるようにしてみてください。</p> <p>CICS TS リリースが異なる、複数のターゲット領域を含むワークロードの場合、CICS 領域でのアクティブな CICSplex SM タスクのデフォルトの数が異なり、その他のタスクが実行中でない場合、ターゲット領域への作業の配分が変わる可能性があります。CICS TS V5.4 のターゲット領域では、CICSplex SM タスクの数はデフォルトでは 0 です。これに対し CICS TS V5.3 以前のターゲット領域では、デフォルトで、アクティブな CICSplex SM タスクの数が 3 から 19 までの任意の数となります。このため、CICS TS V5.4 より前のリリースのターゲット領域は、ルーティング・アルゴリズムにおいて、より優先度の低い領域と見なされる可能性があります。</p>

表 16. WLMAWAOR ビューのフィールド (続き)		
フィールド	属性名	説明
CMAS のレポート	RPTCMAS	<p>このレコードの情報を提供した CMAS の名前。</p> <p>GET 操作のオプション・パラメーター EXPLODE が指定されていないか NO として指定されている場合、この属性はブランクに設定され 1 つのレコードのみがターゲット領域に対して戻されます。</p> <p>GET 操作のオプション・パラメーター EXPLODE が YES として指定されている場合、この属性は、このレコードの情報を提供した CMAS の名前で設定されます。ターゲット領域に関連するワークロードを管理する各 CMAS からターゲット領域のレコードが 1 つずつ返されます。</p> <p>IBM 分散 WUI メニューおよびビューを使用して WLMAWAOR 表形式ビューを要求する場合、EXPLODE パラメーターは YES に設定されます。</p>
RS サーバーのプール名	RSPPOOLID	<p>sysplex 最適化ワークロードでは、領域状況データはカップリング・ファシリティー・データ・テーブル内で保守されます。このテーブルは、この属性で識別される CFDT プールに含まれます。最適化されたワークロードをアクティブ化する場合、ここで指定したプール名のデータを管理する領域状況サーバーがアクティブである必要があります。</p> <p>ご使用の CICSplex ID がシスプレックス内で固有でない場合は、RS サーバーのプール名を固有であるようにしてください。シスプレックスで固有の CICSplex ID を複数使用している場合は、それらすべてに対して同じ RS サーバーのプールを共有しても構いません。</p> <p>この属性は、現在のターゲット領域の親 CICSplex 定義で定義されます。デフォルト名は DFHRSTAT です。CICSplex データ・テーブルを収容するため、既存の CFDT プールを使用することができます。行う場合は、最適化ワークロードのスループットが、指定されたプール名へのユーザー・アプリケーション活動によって妨げられることがあることに注意してください。さらに、プールに対するアプリケーションのスループットがシスプレックスの最適化されたワークロードによって妨げられる可能性もあります。最適化ワークロード機能に対して個別の RS サーバーとプール名を定義することをお勧めします。</p>
ターゲット領域状況	STATUS	<p>ワークロードに関連したターゲット領域の現在の状況。次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ACTIVE - ターゲットはワークロード管理に参加できます。</li> <li>• QUIESCING - ターゲットは静止しています。新しいトランザクションはこのターゲットにルーティングされません。現在このターゲットにルーティングされているトランザクションは、親和性関係が存在しなければ別のターゲットにルーティングされます。親和性の関係がある場合、親和性存続時間の有効期限が切れるまで、引き続きトランザクションはこのターゲットにルーティングされます。ターゲットの親和性存続時間が PERMANENT の場合、無期限に QUIESCING 状態になることに注意してください。</li> <li>• QUIESCED - ターゲットはワークロード管理に参加できません。</li> </ul> <p>入力値: ACTIVE   QUIESCE</p>
タスク数の増分値	TASKINC	<p>これは、RS サーバー更新頻度の値を、現在のターゲット領域の実際のタスク数の値へ変換したものです。現在のタスク数がこの値の係数である境界に達するたびに、更新が RS サーバーに送信されます。</p>
タスク・ロード・パーセンテージ	TASKLOAD	<p>これは、このターゲット領域の正規化されたタスク・ロードです。現在のタスクのルーティング数を領域の MAXTASKS 値で除算し、100 で乗算して、計算されます。</p>

表 16. WLMWAOR ビューのフィールド (続き)		
フィールド	属性名	説明
RS サーバー 更新頻度	UPDATERS	<p>最適化モードで実行されているワークロードの場合、この値は、このターゲット領域内のタスク・ロードの値を変更するために CICS CFDT (RS) サーバーが呼び出される頻度を指定します。</p> <p>値の範囲は 0 から 25 です。</p> <ul style="list-style-type: none"> <li>最適化状況がアクティブに設定されていないとき、この値は 0 に設定され、WLM はそれを無視します。最適化状況がアクティブに設定されているとき、値が 0 の場合、タスク・ロード数が増えられても、RS サーバーには通知されないため、このターゲット領域に対して最適化されたワークロード機能が無効になります。</li> <li>1 から 25 の間の値が、演算比率として領域の MAXTASKS 設定に適用されます。その結果生じたタスク数の値は、RS サーバーへの更新呼び出しを行うための数値しきい値として使用されます。</li> </ul> <p>例えば、MAXTASKS 設定値が 120 で、この属性値が 20 に設定されている場合、領域のタスク数が次のように変更したとき、RS サーバーが WLM ロード数の更新のため呼び出されます。</p> <ul style="list-style-type: none"> <li>タスク 23 と 24 (20%)</li> <li>47 および 48 タスク - (40%)</li> <li>タスク 71 と 72 (60%)</li> <li>95 タスクから 96 タスクに - (80%)、</li> <li>119 タスクから 120 タスクに - (100%)。</li> </ul> <p>RS サーバーは、タスク・ロードがこれらの境界をまたいで増加または減少したときに更新されます。</p> <p>1 から 25 スケールのうちの小さい値が指定されると、タスク・ロード範囲全体を通して RS サーバーを更新する頻度が増します。ワークロードが QUEUE モードの場合は、これによる結果として、ワークロードのターゲット・スコープ内の CICS 領域にわたってタスク・ロードがより均等に分散されます (他の正常性およびリンクの要因がすべて同等であると仮定した場合)。しかし、RS サーバーの使用率がそれに対応して増加し、その結果 z/OS カップリング・ファシリティーの使用率が増加することがあります。</p> <p>この値は CICSplex ビューまたは CICS システム定義ビューに認識されている MAS を使用して、この特定の領域に対応するように変更される場合があります。</p>
z/OS WLM ヘルス	WLMHLTH	<p>領域の z/OS WLM 正常性オープン状態を表すパーセント値。この値は、その領域に向けて CPSPM WLM ルーティング領域から送信される作業の量に影響を与える可能性があります。</p> <p>値ゼロ (0) は、作業がその領域にルーティングされないことを示しています。この領域に関連付けられているワークロード親和性はすべて、そのまま残されて受け入れられます。しかし、この値がゼロのとき、それらの親和性に基づくルーティングは失敗します。なお、ルーティングを試みる方法に応じて、またカスタム WRAM プログラムが有効かどうかに応じて、示されるルーティングの失敗が異なります。ユーザーが、親和性 AOR を使用できないことを示すメッセージ EYUWR0003W を受け取る場合もあれば、SYSIDERR を受け取る場合もあります。</p> <p>1 から 99 まで (両端を含む) の値は、WLMHLTH 値と反比例する、追加の重みが領域に加えられることを示しています。つまり、WLMHLTH の値が小さいほど、より大きな追加の重みが割り当てられます。結果として、AORSCOPE 内の他の領域に割り当てられる重みに応じて、WLMHLTH 値の増減に合わせて、領域で受け取るワークロードが増減する可能性があります。</p> <p>値 100 は、フルオープンな z/OS WLM 正常性オープン状態に領域が達しているか、または SIT において WLMHEALTH=OFF で領域が稼働しているか、または領域が CICS TS V5.3 以下で稼働していることを示しています。値 100 の領域は正常と見なされ、その領域に追加の重みは加えられません。</p>



表 16. WLMWAOR ビューのフィールド (続き)		
フィールド	属性名	説明
タスク・ロード・キュー・モード	WLMQMODE	<p>この属性によって、ターゲット CICS 領域のキューに入れられたタスク・ロードの評価方法を次のように指定できます。</p> <ul style="list-style-type: none"> <li>• <b>MAXTASK</b> - アクティブ・タスクおよび MAXTASK のキューに入れられたタスクの両方が、領域のタスク・ロード評価に含まれること指定します。</li> <li>• <b>ALL</b> - アクティブ・タスク、MAXTASK 限界のキューに入ったタスク、および TRANCLASS 限界のためにキューに入ったタスクを領域のタスク・ロード評価に含めることを指定します。</li> <li>• <b>N_a</b> - 現在の領域がこの属性の使用をサポートするのに十分高いレベルでないことを指定します。これは変更できず、WLM 経路決定プロセスで無視されます。</li> </ul> <p>デフォルト値は <b>ALL</b> です。この値は、CICSplex SM CICS システム定義オブジェクト内に指定されます。この値を変更する場合は、現在の領域のシステム定義を変更した後、再始動する必要があります。アクティブな CICS 領域におけるこの値は、CICSplex SM MAS オブジェクトを使って変更できます。それは、その領域の再始動時に、システム定義で指定されている値に戻ります。</p>
タスク・ロード・ヘルスしい値	WLMTHRS	<p>これは現在の領域タスク・ロードの比率しい値を指定します。これは現在のタスク数を最大タスク数で割ることで算出されます。ターゲット領域の負荷がそのしい値に達すると、WLM は、その領域が比較的正常でないと思えます。これにより、その領域を評価するときに、より高いリンク重みが WLM ルーティング・アルゴリズムに適用されるようになります。</p> <p>ターゲット・スコープに、ルーターに対するローカル領域およびリモート領域が含まれる場合、WLM はリモート領域よりもローカル領域を優先します。この属性を適用すると、ローカル領域でこのロードしい値に達したときに、WLM はローカルではなくリモートのターゲットを優先するようになります。スコープ内のすべてのターゲットがこのロードしい値に達すると、WLM は再びローカルのターゲットを優先するようになります。</p> <p>値の範囲は 1 から 100 で、デフォルト値は 60 です。この値は、CICSplex SM CICS システム定義オブジェクト内に指定されます。この値を変更する場合は、現在の領域のシステム定義を変更した後、再始動する必要があります。アクティブな CICS 領域におけるこの値は、CICSplex SM MAS オブジェクトを使って変更できます。それは、その領域の再始動時に、システム定義で指定されている値に戻ります。</p> <p>この値が 0 に設定されている場合、現在の領域はこの属性の使用をサポートするのに十分高いレベルではありません。これは変更できず、WLM 経路決定プロセスで無視されます。</p> <p>注: この値は、リンク中立な動的ルーティング要求のルーティング要因に適用されると、ヌルになります。LNQUEUE アルゴリズムと LINGOAL アルゴリズムでは、リンク重み自体が無視されるからです。</p>
ワークロード名	WORKLOAD	ワークロード仕様の名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## アクティブなターゲット領域 - WLMATARG

「アクティブなワークロードでのターゲット領域」(WLMATARG) ビューは、コンテキストとして識別される CICSplex 中のワークロードに関連したすべてのターゲット領域に関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「アクティブ・ワークロード・ビュー」 > 「アクティブなターゲット領域」

表 17. 提供されている「アクティブなターゲット領域」(WLMATARG) ビュー・セットのビュー	
ビュー	注
アクティブなターゲット領域 EYUSTARTWLMATARG.DETAILED	選択したターゲット領域に関する詳細情報
アクティブなターゲット領域 EYUSTARTWLMATARG.RESET	現在のターゲット領域の動的ルーティング統計をゼロにリセットします。統計をリセットした後、パフォーマンス目的のため、また正確を期すために、このビュー・セットのリフレッシュ・ボタンを使用して、統計に対する変更を反映することをお勧めします。
アクティブなターゲット領域 EYUSTARTWLMATARG.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したターゲット領域に関するテーブル形式の情報。
アクティブなターゲット領域 EYUSTARTWLMATARG.TABULAR2	コンテキストと識別される CICSplex 内のワークロードに関連したターゲット領域に関するテーブル形式の情報。ルーティング統計は、CICS TS V4.1 CMAS (またはそれ以上) に接続された領域でのみ収集可能です。CICS TS V4.1 より前の CMAS に接続されたターゲットまたはルーターでは、いくつかの統計がゼロと表示されることがあります。
アクティブなターゲット領域 EYUSTARTWLMATARG.TABULAR3	コンテキストと識別される CICSplex 内のワークロードに関連したターゲット領域に関するテーブル形式の情報。ルーティング統計は、CICS TS V4.1 CMAS (またはそれ以上) に接続された領域でのみ収集可能です。CICS TS V4.1 より前の CMAS に接続されたターゲットまたはルーターでは、いくつかの統計がゼロと表示されることがあります。

## アクション

表 18. WLMATARG ビューに使用可能なアクション	
アクション	説明
RESET	現在のターゲット領域の動的ルーティング統計をゼロにリセットします。統計をリセットした後、パフォーマンス目的のため、また正確を期すために、このビュー・セットのリフレッシュ・ボタンを使用して、統計に対する変更を反映することをお勧めします。

## フィールド

表 19. WLMATARG ビューのフィールド		
フィールド	属性名	説明
ターゲット領域名	AOR	WLM ターゲット領域の働きをしている、アクティブな CICS システムの名前。
RS サーバーの最下位層	BOTRSUPD	これは、最下位層のタスク・ロード範囲 (ゼロからこの値まで) です。  値の範囲は 1 から 25 までとなります。領域のタスク・ロードがこの範囲内に入る場合は、タスク・ロードの変更ごとに、タスク・ロードがカップリング・ファシリティにブロードキャストされます。ロードがこの値に達すると、「RS サーバー更新頻度」のタスク規則が活動化されます。  ここに示す CICSplex レベルの値は、個々の CICS 領域ベースで値を微調整できるように、CICS 定義レベルでオーバーライドされる場合があります。
カップリング・ファシリティ読み取り数	CFREADCT	この領域の状況に関してルーティング領域が z/OS カップリング・ファシリティに対して実行した読み取り呼び出しの回数です。
カップリング・ファシリティ更新数	CFUPDCNT	これは、RS サーバーがこのターゲット領域のカップリング・ファシリティに対して行った更新呼び出し回数です。  RS サーバーまたは CICS 領域が開始されるたびに、この値はリセットされます。

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
ターゲット領域コンタクト状況	CON_STATUS	ターゲット領域と CMAS の間の接続の状況。値は次のとおりです。 <ul style="list-style-type: none"> <li>• LOSTCON - ターゲット領域と CMAS の間の接続は失われています。</li> <li>• N_A - ターゲット領域と、ワークロードを管理する CMAS との間の接続は使用可能です。</li> </ul>
CICSplex SM バージョン	CPSMVER	ターゲット領域の CICSplex SM バージョンです。
ターゲット領域の RTA イベント数	EVENTS	これは、現在の領域と関連付けられている未解決の RTA イベントの総数です。 この値は、報告 CMAS が所有 CMAS でない場合、ゼロ (0) に設定されます。
領域のダンプ	HLTHDUMP	これはルーティング・ターゲットが現在トランザクション・ダンプまたはシステム・ダンプを発行しているかどうかを示します。
領域が MAXTASKS にある	HLTHMAXT	これはルーティング・ターゲットがその最大タスクしきい値に達したタスク・レートで、現在実行されているかどうかを示します。
MAS が応答していない	HLTHNRM	このフィールドは、ルーティング・ターゲットが現時点で応答していないことを示します。
ストレージ不足	HLTHSOS	このフィールドは、現在時刻におけるこのルーティング・ターゲットの SOS 状況を示します。CICS 領域では瞬時に SOS 状況が切り替わります。
領域の停止	HLTHSTALL	これは、ルーティング・ターゲットが停止状態になっているかどうかを示します。領域が停止している場合、アクティブ・タスク・ビューを使用して、中断されているタスクとその原因を判別します。注: ここでの停止とは、タスクが特定の時間同じリソースを待機していることを意味しています。
領域の最大タスク数	MAXTASKS	これはこのルーティング・ターゲットで同時に実行できるアクティブ・タスクの、定義された最大数です。
所有 CMAS 名	OWNINGCMAS	ターゲット領域の管理に参加している CMAS の名前。
最適化の状況	OWSTATE	このフィールドは、最適化されたワークロード・ルーティング機能の現在のターゲット領域の状況を報告します。 可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• ACTIVE <ul style="list-style-type: none"> <li>- このターゲット領域は、最適化されたワークロード状態で実行されています。</li> </ul> </li> <li>• INACTIVE <ul style="list-style-type: none"> <li>- この領域は最適化されたワークロード状態で実行可能ですが、以下の 1 つ以上の理由で、現在最適化されていません。 <ul style="list-style-type: none"> <li>- 領域に RS サーバーへの接続がない。</li> <li>- 領域は RS サーバーに接続されているが、サーバーが z/OS カップリング・ファシリティに接続できていない。</li> <li>- 最適化ワークロード・ルーティング機能が現在のターゲット領域機能で無効になっている。</li> <li>- この領域の RS サーバーの更新頻度値が 0 である。これは、この領域の最適化機能が使用可能でないことを意味します。</li> </ul> </li> </ul> </li> <li>• N_A <ul style="list-style-type: none"> <li>- このターゲット領域は、最適化されたワークロード管理を提供するように構成されていません。最適化されていない WLM ルーティング決定しか行えません。</li> </ul> </li> </ul> 報告 CMAS が領域状況 (RS) サーバーに接続されていない場合、この値は INACTIVE に設定されます。

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
RS サーバー 読み取り間隔	READRS	<p>最適化モードで実行されている領域の場合、この値は、CICS CFDT サーバーからのターゲット領域の状況のリフレッシュから次のリフレッシュまでの最小間隔を指定します。これらのリフレッシュ要求は、動的ルーティング要求のターゲット候補としてこの領域を評価しているルーティング領域によって発行されます。</p> <p>値の範囲は 0 から 2000 で、ミリ秒単位で示されます。</p> <ul style="list-style-type: none"> <li>最適化の状況が「アクティブ」に設定されていない場合、この値は 0 に設定され、WLM はこれを無視します。最適化の状況が「アクティブ」に設定されている場合の値 0 は、ターゲット領域の状況を調べるたびに、その領域の状況更新をルーティング領域が要求することを意味します。</li> <li>この領域の状況を最新表示するまでの最小時間間隔を指定する場合は、1 から 2000 までの値を使用します。</li> </ul> <p>間隔値が小さい場合、より大きい値の場合よりも頻繁に、CFDT サーバーをポーリングして状況を更新します。ワークロードが QUEUE モードにある場合は、この値を設定することで、ワークロードの対象有効範囲内の CICS 領域全体にタスク・ロードがより均等に分散されます (<b>他の正常性および関連する要因がすべて同等であると仮定した場合</b>)。しかし、RS サーバーの使用率がそれに対応して増加し、その結果 z/OS カップリング・ファシリティの使用率が増加することがあります。</p> <p>値 1000 は、1 秒の間隔を表します。デフォルト値は 200 ミリ秒です。</p> <p>この値は、「CICSplex 認知の MAS」ビュー、または「CICS システム定義名」ビューを使ってこの特定の領域に合わせて変更することができます。</p>
領域の WLM ルーティングの重み	ROUTEWGHT	<p>未解決の RTA MRM イベント、ルーター領域のリンク速度、トランザクションの異常終了の可能性を考慮に入れない、現在のルーティング・ターゲットのルーティングの重みの計算値です。領域のトランザクション・ロードとヘルスの状態だけが計算され、この値に組み込まれます。正常性の状態は、SOS、ダンプ、停止、MAXTASK、z/OS WLM 正常性の各状態の組み合わせです。トランザクション・ロードは、「タスク・ロード・パーセンテージ」属性で示されている値です。ワークロード関連のパラメーターは、この重み値に組み込まれません。特定のワークロードのルーティングの重みを確認するには、WLMAWAOR の「アクティブなワークロードでのターゲット領域」ビューで返される対応属性を使用します。</p> <p>動的ルーティング要求に親和性が適用されない場合は、ルーティング・スコープに含まれているすべてのターゲット領域のルーティングの重みが比較され、重み値が最も小さい領域が選択されます。同じ最小ウェイト値の領域がいくつかある場合は、そのセットの中のいずれかがルーティング・ターゲットとしてランダムに選択されます。</p> <p>注: 2147483647 は、ターゲットが静止しているか、z/OS WLM ヘルスが 0 であることを示す特殊値です。この値のターゲット領域は、ワークロード管理から除去されます。この場合、追加のトランザクションは、このターゲット領域に経路指定されません。</p>

表 19. WLMATARG ビューのフィールド (続き)

フィールド	属性名	説明
ルーティング・ロード	ROUTINGLOAD	<p>これは、ターゲット領域の WLM 経路の決定で使用するルーティング・ロードです。これは、この照会時点での実際の CICS タスク数と一致しないことに注意してください。</p> <p>アクティブなシステムの最適化された WLM ターゲット領域の場合、これは、「タスク数の増分値」、「RS サーバー下部層」および「RS サーバー上部層」を考慮に入れた、ターゲット領域での概算のタスク・ロードになります。この値の精度は、「RS サーバー読み取り間隔」設定と「RS サーバー更新頻度」設定にも影響を受けます。</p> <p>RS サーバーに接続されておらず、そのため最適化されていない WLM 状態にあるターゲット領域の場合、実際のタスク数と比較すると、この値はあまり正確でないようにみえます。それは、これがこの領域の正規化されたルーティング・ロードであるためです。最適化されていない WLM 状態で実行しているとき、この値は MAX TASK フラグを設定せずに MAX TASK 値を超えることができ、MAX TASK 標識が実際のタスク数から取得されたターゲット領域のヘルス状態に関連しているとき、この値は正しい動作であることに注意してください。</p> <p>領域が複数のワークロードのターゲットになり、異なるルーティングの重み値を示す場合があります。これは異常な動作ではありません。ワークロード・マネージャー内の同期待ち時間の症状です。ワークロードの最適化は、この効果を無効にします。ワークロード・マネージャーのランタイム・プロセスは、ロード値におけるこの不均衡が生じてても、正常に実行し続けます。このようにロードで観測される相違を抑制するには、ワークロードに関係するすべての MAS と CMAS が、接続可能な領域状況サーバーを使用できるようにしてみてください。</p> <p>CICS TS リリースが異なる、複数のターゲット領域を含むワークロードの場合、CICS 領域でのアクティブな CICSplex SM タスクのデフォルトの数が異なっており、その他のタスクが実行中でない場合、ターゲット領域への作業の配分が変わる可能性があります。CICS TS V5.4 のターゲット領域では、CICSplex SM タスクの数はデフォルトでは 0 です。これに対し CICS TS V5.3 以前のターゲット領域では、デフォルトで、アクティブな CICSplex SM タスクの数が 3 から 19 までの任意の数となります。このため、CICS TS V5.4 より前のリリースのターゲット領域は、ルーティング・アルゴリズムにおいて、より優先度の低い領域と見なされる可能性があります。</p>
報告 CMAS 名	RPTINGCMAS	このレコードを指定した CMAS の名前。

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
RS サーバーのプール名	RSPOOLID	<p>sysplex 最適化ワークロードでは、領域状況データはカップリング・ファシリティー・データ・テーブル内で保守されます。このテーブルは、この属性で識別される CFDT プールに含まれます。最適化されたワークロードをアクティブ化する場合、ここで指定したプール名のデータを管理する領域状況サーバーがアクティブである必要があります。</p> <p>ベスト・プラクティスとして、領域状況サーバーとそのプール名は、ユーザー・アプリケーションのために使用するものとは別にする必要があります。領域状況サーバーやプールを、ユーザー・アプリケーションと共用しないでください。この属性は、現在のターゲット領域の親 CICSplex 定義で定義されます。CICSplex SM によって実装されるデフォルトのプール名は DFHRSTAT です。デフォルトの領域状況サーバー・プール名を使用することを強くお勧めします。</p> <p>以下のシナリオでは、デフォルト以外のプール名を使用することもできます。</p> <ul style="list-style-type: none"> <li>領域状況サーバーと同じシスプレックスで、既に重複する CICSplex 名を使用している。</li> <li>領域状況サーバーと同じシスプレックスで、重複する CICSplex 名を使用する予定である。</li> <li>デフォルトのプール名の使用を禁止する厳格な命名規則ポリシーがある。</li> <li>同一のシスプレックスで複数の環境 (テスト環境と実稼働環境など) が稼働するので、追加の分離が必要である。</li> </ul> <p>デフォルト名の DFHRSTAT を使用しない場合は、CICSplex で他の領域を開始する前に、名前を変更する必要があります。</p> <p>CICSplex SM は、CICSplex がアクティブである間のプール名の変更を禁止しません。</p> <p>CICSplex がアクティブである間に変更した場合は、CICSplex 内のすべての CMAS 領域と MAS 領域 (ルーターとターゲットの両方) をできるだけ早く再始動してください。</p> <p>すべてのルーターとターゲットが新しい領域状況サーバー・プールに転送されるまで、WLM 最適化は非活動化されます。移行が完了した一部の領域で最適化が行われているかのように表示されることもありますが、ルーティング要求が最適化されることはありません。</p> <p>つまり、CICSplex 内のすべての領域が再始動されるまでは、CICSplex SM の WLM ビューに、矛盾するデータが表示されることになります。</p>
ルート異常終了	RTABEND	<p>この領域の選択後にルート異常終了が返された回数。</p> <p>この値は、ルーティング領域 (動的ルーティング) およびターゲット領域 (分散ルーティング) が、報告 CMAS に接続していたときに、これらの領域によって処理されたすべての要求の合計です。</p>
ルート完了	RTCOMPLETE	<p>この領域でルート完了が実行された回数。ルート完了は、トランザクション要求がターゲット領域の待機キューにディスパッチされた後に発生します。</p> <p>この値は、ルーティング領域が報告 CMAS に接続していたときに、それらの領域によって処理されたすべての要求の合計です。</p>
ルート・エラー	RTERROR	<p>現在の領域を選択することでルート・エラーが発生した回数。ルート・エラーは、CICSplex SM がターゲット領域を選択し、作業をその領域にルーティングするときに CICS でエラーになる場合に発生します。例えば、接続がサービス休止中であつたり、LU 6.2 接続ですぐに使用できるセッションがなかったりするような場合です。これが生じると、CICSplex SM は別のターゲットの選択を試みます。</p> <p>この値は、ルーティング領域が報告 CMAS に接続していたときに、それらの領域によって処理されたすべての要求の合計です。</p>
ルート開始	RTINITIATE	<p>ルーティングされたトランザクションがターゲット領域で実際に開始された回数。</p> <p>この値は、ターゲット領域が報告 CMAS に接続していたときに、この領域によって処理されたすべての要求の合計です。</p>

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
ルート通知	RTNOTIFY	この領域でルート通知が実行された回数。ルート通知は通常、静的ルーティングの開始時に発生します。  この値は、ルーティング領域が報告 CMAS に接続していたときに、それらの領域によって処理されたすべての要求の合計です。
ルート選択	RTSELECT	現在の領域がルート選択処理においてルーティング・ターゲットとして選択された回数。  この値は、ルーティング領域が報告 CMAS に接続していたときに、それらの領域によって処理されたすべての要求の合計です。
ルート終了	RTTERMINATE	この領域でルート終了が実行された回数。ルート終了は、ターゲット領域でトランザクションが終了するときに発生します。  この値は、ルーティング領域 (動的ルーティング) およびターゲット領域 (分散ルーティング) が、報告 CMAS に接続していたときに、これらの領域によって処理されたすべての要求の合計です。
ターゲット領域状況	STATUS	ワークロードに関連したターゲット領域の現在の状況。次のいずれかです。  <ul style="list-style-type: none"> <li>• ACTIVE - ターゲットはワークロード管理に参加できます。</li> <li>• QUIESCING - ターゲットは静止しています。新しいトランザクションはこのターゲットにルーティングされません。現在このターゲットにルーティングされているトランザクションは、親和性関係が存在しなければ別のターゲットにルーティングされます。親和性の関係がある場合、親和性存続時間の有効期限が切れるまで、引き続きトランザクションはこのターゲットにルーティングされます。ターゲットの親和性存続時間が PERMANENT の場合、無期限に QUIESCING 状態になることに注意してください。</li> <li>• QUIESCED - ターゲットはワークロード管理に参加できません。</li> </ul> 入力値: ACTIVE   QUIESCE
タスク数の増分値	TASKINC	これは、RS サーバー更新頻度の値を、現在のターゲット領域の実際のタスク数の値へ変換したものです。現在のタスク数がこの値の係数である境界に達するたびに、更新が RS サーバーに送信されます。
タスク・ロード・パーセンテージ	TASKLOAD	これは、このターゲット領域の正規化されたタスク・ロードです。現在のタスクのルーティング数を領域の MAXTASKS 値で除算し、100 で乗算して、計算されます。
RS サーバーの最上位層	TOPRSUPD	これは、領域の MAXTASKS 設定に対する演算比率です。その結果生じたタスク数の値は、領域の MAXTASKS 設定から減算されてタスク・ロードの「最上位層」を確立します。領域内のタスク・ロードが MAXTASKS 限界値に達した場合、タスク・ロードがこの値よりも小さい値に戻った後に、領域の MAXTASKS 状態をオフに切り替えて、カップリング・ファシリティーに送る必要があります。  ここに示す CICSplex レベルの値は、個々の CICS 領域ベースで値を微調整できるように、CICS 定義レベルでオーバーライドされる場合があります。



表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
RS サーバー 更新頻度	UPDATERS	<p>最適化モードで実行されている領域の場合、この値は、このターゲット領域内のタスク・ロードの値を変更するために CICS CFDT (RS) サーバーが呼び出される頻度を指定します。</p> <p>値の範囲は 0 から 25 です。</p> <ul style="list-style-type: none"> <li>最適化の状況が「アクティブ」に設定されていない場合、この値は 0 に設定され、WLM はこれを無視します。最適化状況が「アクティブ」に設定されている際の値 0 は、タスク・ロード・カウントの変更が RS サーバーには全く通知されないことを意味します。この場合、このターゲット領域の最適化ワークロード機能が使用不可になります。</li> <li>1 から 25 の間の値が、演算比率として領域の MAXTASKS 設定に適用されます。その結果生じたタスク数の値は、RS サーバーへの更新呼び出しを行うための数値しきい値として使用されます。</li> </ul> <p>例えば、MAXTASKS 設定値が 120 で、この属性値が 20 に設定されている場合、領域のタスク数が次のように変更したとき、RS サーバーが WLM ロード数の更新のため呼び出されます。</p> <ul style="list-style-type: none"> <li>タスク 23 と 24 (20%)</li> <li>47 および 48 タスク - (40%)</li> <li>タスク 71 と 72 (60%)</li> <li>タスク 95 と 96 (80%)</li> <li>119 タスクから 120 タスクに - (100%)。</li> </ul> <p>RS サーバーは、タスク・ロードがこれらの境界をまたいで増加または減少したときに更新されます。</p> <p>1 から 25 スケールのうちの小さい値が指定されると、タスク・ロード範囲全体を通して RS サーバーを更新する頻度が増します。ワークロードが QUEUE モードにある場合は、この値を設定することで、ワークロードの対象有効範囲内の CICS 領域全体にタスク・ロードがより均等に分散されます (<b>他の正常性および関連する要因がすべて同等であると仮定した場合</b>)。しかし、RS サーバーの使用率がそれに対応して増加し、その結果 z/OS カップリング・ファシリティーの使用率が増加することがあります。</p> <p>この値は、「CICSplex 認知の MAS」ビュー、または「CICS システム定義名」ビューを使ってこの特定の領域に合わせて変更することができます。</p>
z/OS WLM ヘルス	WLMHLTH	<p>領域の z/OS WLM 正常性オープン状態を表すパーセント値。この値は、その領域に向けて CPSM WLM ルーティング領域から送信される作業の量に影響を与える可能性があります。</p> <p>値ゼロ (0) は、作業がその領域にルーティングされないことを示しています。この領域に関連付けられているワークロード親和性はすべて、そのまま残されて受け入れられます。しかし、この値がゼロのとき、それらの親和性に基づくルーティングは失敗します。なお、ルーティングを試みる方法に応じて、またカスタム WRAM プログラムが有効かどうかに応じて、示されるルーティングの失敗が異なります。ユーザーが、親和性 AOR を使用できないことを示すメッセージ EYUWR0003W を受け取る場合もあれば、SYSIDERR を受け取る場合もあります。</p> <p>1 から 99 まで (両端を含む) の値は、WLMHLTH 値と反比例する、追加の重みが領域に加えられることを示しています。つまり、WLMHLTH の値が小さいほど、より大きな追加の重みが割り当てられます。結果として、AORSCOPE 内の他の領域に割り当てられる重みに応じて、WLMHLTH 値の増減に合わせて、領域で受け取るワークロードが増減する可能性があります。</p> <p>値 100 は、フルオープンな z/OS WLM 正常性オープン状態に領域が達しているか、または SIT において WLMHEALTH=OFF で領域が稼働しているか、または領域が CICS TS V5.3 以下で稼働していることを示しています。値 100 の領域は正常と見なされ、その領域に追加の重みは加えられません。</p>

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
最適化の使用可能化の状態	WLMOPTEN	<p>このフィールドは、最適化されたワークロード・ルーティング機能が現在のターゲット領域でサポートされるかどうかを報告します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>ENABLED</b> <ul style="list-style-type: none"> <li>– このターゲット領域は、最適化されたワークロード・ルーティング要求をサポートします。</li> </ul> </li> <li>• <b>DISABLED</b> <ul style="list-style-type: none"> <li>– この領域では、最適化されたワークロード・ルーティング要求がサポートされず、現在の領域の状態を照会またはブロードキャストするために領域状況サーバーに通信しません。</li> </ul> </li> </ul> <p>この値は、「CICSplex 認知の MAS」ビュー、または「CICS システム定義名」ビューを使ってこの特定の領域に合わせて変更することができます。</p>
タスク・ロード・キュー・モード	WLMQMODE	<p>この属性によって、ターゲット CICS 領域のキューに入れられたタスク・ロードの評価方法を次のように指定できます。</p> <ul style="list-style-type: none"> <li>• <b>MAXTASK</b> - アクティブ・タスクおよび MAXTASK のキューに入れられたタスクの両方が、領域のタスク・ロード評価に含まれること指定します。</li> <li>• <b>ALL</b> - アクティブ・タスク、MAXTASK 限界のキューに入ったタスク、および TRANCLASS 限界のためにキューに入ったタスクを領域のタスク・ロード評価に含めることを指定します。</li> <li>• <b>N_a</b> - 現在の領域がこの属性の使用をサポートするのに十分高いレベルでないことを指定します。これは変更できず、WLM 経路決定プロセスで無視されます。</li> </ul> <p>デフォルト値は <b>ALL</b> です。この値は、CICSplex SM CICS システム定義オブジェクト内に指定されます。この値を変更する場合は、現在の領域のシステム定義を変更した後、再始動する必要があります。アクティブな CICS 領域におけるこの値は、CICSplex SM MAS オブジェクトを使って変更できます。それは、その領域の再始動時に、システム定義で指定されている値に戻ります。</p>
タスク・ロード・ヘルスしい値	WLMTHRSH	<p>これは現在の領域タスク・ロードの比率しい値を指定します。これは現在のタスク数を最大タスク数で割ることで算出されます。ターゲット領域の負荷がそのしい値に達すると、WLM は、その領域が比較的正常でないと思なします。これにより、その領域を評価するときに、より高いリンク重みが WLM ルーティング・アルゴリズムに適用されるようになります。</p> <p>ターゲット・スコープに、ルーターに対するローカル領域およびリモート領域が含まれる場合、WLM はリモート領域よりもローカル領域を優先します。この属性を適用すると、ローカル領域でこのロードしい値に達したときに、WLM はローカルではなくリモートのターゲットを優先するようになります。スコープ内のすべてのターゲットがこのロードしい値に達すると、WLM は再びローカルのターゲットを優先するようになります。</p> <p>値の範囲は 1 から 100 で、デフォルト値は 60 です。この値は、CICSplex SM CICS システム定義オブジェクト内に指定されます。この値を変更する場合は、現在の領域のシステム定義を変更した後、再始動する必要があります。アクティブな CICS 領域におけるこの値は、CICSplex SM MAS オブジェクトを使って変更できます。それは、その領域の再始動時に、システム定義で指定されている値に戻ります。</p> <p>この値が 0 に設定されている場合、現在の領域はこの属性の使用をサポートするのに十分高いレベルではありません。これは変更できず、WLM 経路決定プロセスで無視されます。</p> <p>注: この値は、リンク中立な動的ルーティング要求のルーティング要因に適用されると、ヌルになります。LNQUEUE アルゴリズムと Lngoal アルゴリズムでは、リンク重み自体が無視されるからです。</p>

表 19. WLMATARG ビューのフィールド (続き)		
フィールド	属性名	説明
ワークロード数	WRKLDCNT	<p>これは、この領域を動的ルーティング・ターゲットとして指定したワークロードの数です。</p> <p>次の場合、ワークロード数はゼロ (0) に設定されます。</p> <ul style="list-style-type: none"> <li>報告 CMAS が所有 CMAS でない。</li> <li>所有 CMAS が CICS TS 4.1 より前のバージョンである。</li> <li>所有 CMAS がワークロードに参加しているが、それに接続するルーターがない。</li> </ul>

## 定義 - WLMWDEF

「アクティブ・ワークロード定義」(WLMWDEF) ビューは、コンテキストとして指定される CICSplex 中のワークロードにインストールされているアクティブなワークロード定義に関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「アクティブ・ワークロード・ビュー」 > 「定義」

表 20. 提供されている「アクティブ・ワークロード定義」(WLMWDEF) ビュー・セットのビュー	
ビュー	注
アクティブ・ワークロード定義 EYUSTARTWLMWDEF.DETAILED	選択したワークロード定義に関する詳細情報
アクティブ・ワークロード定義 EYUSTARTWLMWDEF.DISCARD	アクティブなワークロード定義を破棄します。
アクティブ・ワークロード定義 EYUSTARTWLMWDEF.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したアクティブ・ワークロード定義に関するテーブル形式の情報

### アクション

表 21. WLMWDEF ビューに使用可能なアクション	
アクション	説明
DISCARD	アクティブなワークロード定義を破棄します。

### フィールド

表 22. WLMWDEF ビューのフィールド		
フィールド	属性名	説明
ターゲット・システム・セットのスコープ名	AORSCOPE	ワークロード定義に関連付けられたトランザクションが誘導される CICS システムまたは CICS システム・グループの名前。
説明	DESC	ワークロード定義の説明。
端末 LU 名	LUNAME	トランザクションとワークロード定義の突き合わせに使用される論理装置名。
ワークロード定義	NAME	ワークロード定義の名前。

表 22. WLMWDEF ビューのフィールド (続き)		
フィールド	属性名	説明
プロセス・タイプ	PROCESSTYPE	トランザクションとワークロード定義の突き合わせに使用される特定または総称プロセス・タイプ。  「端末 LU 名」と「ユーザー ID」は両方とも「*」に設定する必要があります。「トランザクション・グループ」にはブランクまたは特定の名前を指定できます。ワークロードは、「トランザクション・グループ」と「プロセス・タイプ」、または「トランザクション・グループ」、「端末 LU 名」、および「ユーザー ID」でのみ区切ることができます。
トランザクション・グループ	TRANGRP	ワークロード定義に関連付けられたトランザクション・グループの名前。  このフィールドがブランクの場合、トランザクション・グループが定義されなかったことを意味します。ワークロード定義は、そのワークロード仕様に関連したデフォルトのトランザクション・グループを使用しています。  注: このフィールドがブランクの場合、WLMATGRP ビューへのハイパーリンクは無効です。
ユーザー ID	USERID	トランザクションとワークロード定義の突き合わせに使用されるユーザー ID。
ワークロード名	WORKLOAD	ワークロード定義に関連付けられているワークロード仕様の名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## トランザクション・グループ - WLMATGRP

「アクティブ・ワークロードのトランザクション・グループ」(WLMATGRP) ビューは、コンテキストとして指定される CICSplex 中のワークロードにインストールされているトランザクション・グループに関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「**アクティブ作業負荷ビュー (Active workload views)**」 > 「**トランザクション・グループ (Transaction groups)**」

表 23. 提供されている「アクティブ・ワークロードのトランザクション・グループ」(WLMATGRP) ビュー・セットのビュー	
ビュー	注
アクティブ・ワークロードのトランザクション・グループ EYUSTARTWLMATGRP.ACTIVE	ワークロードをアクティブに設定します。
アクティブ・ワークロードのトランザクション・グループ EYUSTARTWLMATGRP.DETAILED	選択したトランザクション・グループに関する詳細情報
アクティブ・ワークロードのトランザクション・グループ EYUSTARTWLMATGRP.DORMANT	ワークロードを休止に設定します。
アクティブ・ワークロードのトランザクション・グループ EYUSTARTWLMATGRP.SET	トランザクション・グループの属性を設定します。
アクティブ・ワークロードのトランザクション・グループ EYUSTARTWLMATGRP.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したトランザクション・グループに関するテーブル形式の情報

## アクション

表 24. WLMATGRP ビューに使用可能なアクション	
アクション	説明
ACTIVE	ワークロードをアクティブに設定します。
DORMANT	ワークロードを休止に設定します。
GET	(必須) アクティブなワークロードの名前。
SET	トランザクション・グループの属性を設定します。

## フィールド

表 25. WLMATGRP ビューのフィールド		
フィールド	属性名	説明
許容できる異常終了レベル	ABENDCRIT	ターゲット領域が正常でないと見なされる原因となる、デフォルトのトランザクション・グループに関連したトランザクションの異常終了の確率。
ターゲット領域の許容負荷レベル	ABENDTHRESH	ターゲット領域の負荷レベルが倍になる原因となる、デフォルトのトランザクション・グループに関連したトランザクションの異常終了の確率。
自動親和性作成	AFFAUTO	<p>CICSplex SM が、トランザクション・グループに関連付けられたトランザクションの親和性の関係を自動的に作成するかどうかを示します。</p> <ul style="list-style-type: none"> <li>• YES <ul style="list-style-type: none"> <li>– 「親和性の関係」および「親和性存続時間」フィールド内の値を使用して親和性が作成されます。</li> </ul> </li> <li>• NO <ul style="list-style-type: none"> <li>– 親和性は自動的に作成されません (ただし、動的ルーティング・プログラム EYU9WRAM のカスタマイズ版で作成することが可能です)。</li> </ul> </li> <li>• 該当なし <ul style="list-style-type: none"> <li>– 「親和性の関係」および「親和性存続時間」フィールドに値がないため、親和性は作成されません。</li> </ul> </li> </ul>
親和性の関係	AFFINITY	<p>トランザクション・グループのオカレンスを処理する際に使用する親和性の関係。親和性の関係値は、次のとおりです。</p> <ul style="list-style-type: none"> <li>• GLOBAL - すべての端末のすべてのユーザー。</li> <li>• LUNAME - 端末の論理装置名。</li> <li>• NONE - 親和性は定義されていません。</li> <li>• N/A - このフィールドは適用外です。</li> <li>• USERID - ユーザー ID</li> <li>• BAPPL - CICS BTS ビジネス・アプリケーション。</li> <li>• Locked - LOCKED の親和性は、動的にリンクされたプログラム間でのみ生じます。LOCKED の親和性は、呼び出し元に戻された後に保存される状態データを、呼び出されたプログラムが保持する際に作成されます。このタイプの親和性を持つプログラムは、作業単位が終了するまで、同じターゲット領域に送られます。LOCKED を使用できるのは、作業単位の親和性存続時間が関連付けられている動的プログラム・リンク (DPL) 要求の場合に限られます。</li> </ul>

表 25. WLMATGRP ビューのフィールド (続き)

フィールド	属性名	説明
親和性存続時間	AFFLIFE	<p>トランザクション・グループのオカレンスを処理する際に使用する親和性存続時間。親和性存続時間の値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• DELIMIT - 疑似会話モードが END となるまで。</li> <li>• LOGON - 端末セッションの期間内。</li> <li>• PCONV - 疑似会話の期間内。</li> <li>• PERMANENT - ワークロードがアクティブである間。</li> <li>• SIGNON - ユーザー・セッションがアクティブである間。</li> <li>• SYSTEM - トランザクションの経路指定先の AOR がアクティブである間。</li> <li>• ACTIVITY - CICS BTS の活動がアクティブである間。</li> <li>• PROCESS - CICS BTS プロセスがアクティブである間。</li> <li>• UOW - 作業単位がアクティブである間。</li> <li>• NONE - 親和性存続時間は定義されていません。</li> <li>• N/A - このフィールドは適用外です。</li> </ul>
アルゴリズムのタイプ	ALGTYPE	<p>現在のトランザクション・グループにあるトランザクションの送信先として最適なターゲット領域を「ターゲット・スコープ」で選択するときを使用されるアルゴリズム。有効なオプションは、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>QUEUE</b> - 以下の組み合わせが最適なターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- 正常性 (MaxTask、ストレージ不足、ダンプ、停止)</li> <li>- タスク・キューの項目数 (または負荷)</li> <li>- ルーティング領域からのリンク速度</li> <li>- 異常終了の可能性 (計算されている場合)</li> <li>- RTA イベントの影響 (定義されている場合)</li> </ul> </li> <li>• <b>LNQUEUE</b> - 以下の組み合わせが最適なターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- 正常性 (MaxTask、ストレージ不足、ダンプ、停止)</li> <li>- タスク・キューの項目数 (または負荷)</li> <li>- 異常終了の可能性 (計算されている場合)</li> <li>- RTA イベントの影響 (定義されている場合)</li> </ul> <p>注 - <b>LNQUEUE</b> アルゴリズムのルーティングの重みの計算では、ルーターからターゲットまでのリンク速度は、係数として使用されません。</p> </li> <li>• <b>GOAL</b> - 以下のターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- このトランザクションや他のトランザクションに対して MVS ワークロード管理クラスで設定されている応答時間の目標値にトランザクションが達する可能性が最も高い領域 <ul style="list-style-type: none"> <li>- ターゲットを特定できない場合は、ターゲット領域の残りのセットに <b>QUEUE</b> アルゴリズムを適用します。</li> </ul> </li> </ul> </li> <li>• <b>LNGOAL</b> - 以下のターゲット領域にトランザクションを送信します。 <ul style="list-style-type: none"> <li>- このトランザクションや他のトランザクションに対して MVS ワークロード管理クラスで設定されている応答時間の目標値にトランザクションが達する可能性が最も高い領域 <ul style="list-style-type: none"> <li>- ターゲットを特定できない場合は、ターゲット領域の残りのセットに <b>LNQUEUE</b> アルゴリズムを適用します。</li> </ul> </li> </ul> </li> <li>• <b>INHERIT</b> - このトランザクション・グループに関連する WLM 仕様 (WLMSPEC) で指定されているタイプのアルゴリズムを使用します。</li> </ul>

表 25. WLMATGRP ビューのフィールド (続き)		
フィールド	属性名	説明
イベント名	EVENTNAME	このワークロード中のトランザクションのルーティングに影響する可能性のある分析定義 (RTADEF) か状況定義 (STATDEF) の名前。この定義によってリアルタイム分析イベントが生成される場合、WLM は AOR 選択基準の一部としてこの情報を使用します。  ワークロード仕様の作成時に、オプションで分析定義か状況定義がワークロードに関連付けられます。このフィールドがブランクの場合、分析または状況定義はワークロードに関連付けられません。  注: EVENT ビューは、指定されたタイプのイベントが生じた場合のみ表示されます。
主検索基準	FILTER	トランザクション・グループに関連付けられたトランザクションの 1 次検索条件として、ユーザー名 (USERID) と論理装置名 (LUNAME) のどちらを使用するかを指定します。
トランザクション・グループの状況	STATUS	トランザクション・グループに関連付けられたトランザクションの AOR を選択する方法を示します。  <ul style="list-style-type: none"> <li>ACTIVE - 関連付けられたワークロード定義で識別される AOR スコープから AOR を選択します。</li> <li>DORMANT - CICS に定義した時、トランザクションに関連付けられた AOR を使用します。</li> </ul>
アクティブ・トランザクション・カウント	TRANCNT	このトランザクション・グループに関連したアクティブなトランザクションの数。
トランザクション・グループ	TRANGRP	トランザクション・グループの名前。
ワークロード	WORKLOAD	アクティブなトランザクション・グループのあるワークロードの名前。
ワークロード所有者	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## 動的トランザクション - WLMATRAN

「アクティブ・ワークロードの動的トランザクション」(WLMATRAN) ビューは、コンテキストとして識別される CICSplex 中のワークロードに関連したインストール済みの動的トランザクション定義に関する情報を表示します。

### 提供されるビュー

メインメニューからアクセスするには、

「アクティブ・ワークロード・ビュー」 > 「動的トランザクション」

表 26. 提供されている「アクティブ・ワークロードの動的トランザクション」(WLMATRAN) ビュー・セットのビュー	
ビュー	注
アクティブ・ワークロードの動的トランザクション EYUSTARTWLMATRAN.DETAILED	選択したアクティブ・トランザクションに関する詳細情報
アクティブ・ワークロードの動的トランザクション EYUSTARTWLMATRAN.FORCE	トランザクション・グループからトランザクションを破棄します。
アクティブ・ワークロードの動的トランザクション EYUSTARTWLMATRAN.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したアクティブ・トランザクションに関するテーブル形式の情報

### アクション

表 27. WLMATRAN ビューに使用可能なアクション	
アクション	説明
FORCE	トランザクション・グループからトランザクションを破棄します。



## フィールド

表 28. WLMATRAN ビューのフィールド		
フィールド	属性名	説明
疑似会話の開始または終了	PSEUDOMODE	トランザクションが疑似会話の中の最初 (START) または最後 (END) のトランザクションであるかどうかを示します。このフィールドがブランクの場合、トランザクションは疑似会話中の最初でも最後でもないことを意味します。
トランザクション・グループ	TRANGRP	トランザクション・グループの名前。
トランザクション	TRANID	アクティブなトランザクションの ID。
ワークロード名	WORKLOAD	アクティブなトランザクション・グループのあるワークロードの名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## トランザクション・グループの親和性 - WLMATAFF

「**アクティブ・ワークロードのトランザクション・グループの親和性**」(WLMATAFF) ビューは、コンテキストとして指定される CICSplex 中のワークロードにインストールされているトランザクション・グループのアクティブな親和性に関する情報を表示します。トランザクション・グループに関連した最初のトランザクションがターゲット領域に経路指定されたときに、親和性がアクティブになります。

### 提供されるビュー

メインメニューからアクセスするには、

「**アクティブ作業負荷ビュー (Active workload views)**」 > 「**トランザクション・グループ親和性 (Transaction group affinities)**」

表 29. 提供されている「アクティブ・ワークロードのトランザクション・グループの親和性」(WLMATAFF) ビュー・セットのビュー	
ビュー	注
アクティブ・ワークロードのトランザクション・グループの親和性 EYUSTARTWLMATAFF.DETAILED	選択したアクティブ親和性に関する詳細情報
アクティブ・ワークロードのトランザクション・グループの親和性 EYUSTARTWLMATAFF.FORCE	親和性の項目を破棄します。 影響のあるトランザクション・グループ中の次のトランザクションが検出されると、親和性が再確立されます。
アクティブ・ワークロードのトランザクション・グループの親和性 EYUSTARTWLMATAFF.TABULAR	コンテキストと識別される CICSplex 内のワークロードに関連したトランザクション・グループのアクティブな親和性に関するテーブル形式の情報

### アクション

表 30. WLMATAFF ビューに使用可能なアクション	
アクション	説明
FORCE	親和性の項目を破棄します。 影響のあるトランザクション・グループ中の次のトランザクションが検出されると、親和性が再確立されます。

## フィールド

表 31. WLMATAFF ビューのフィールド		
フィールド	属性名	説明
親和性の関係	AFFINITY	<p>トランザクション・グループに関連したトランザクションを処理する際に使用する親和性の関係。親和性の関係値は、次のとおりです。</p> <ul style="list-style-type: none"> <li>GLOBAL - すべての端末のすべてのユーザー。</li> <li>LUNAME - 端末の論理装置名。</li> <li>USERID - ユーザー ID。</li> <li>BAPPL - CICS BTS ビジネス・アプリケーション。</li> <li>Locked - LOCKED の親和性は、動的にリンクされたプログラム間でのみ生じます。LOCKED の親和性は、呼び出し元に戻された後に保存される状態データを、呼び出されたプログラムが保持する際に作成されます。このタイプの親和性を持つプログラムは、作業単位が終了するまで、同じターゲット領域に送られます。LOCKED を使用できるのは、作業単位の親和性存続時間が関連付けられている動的プログラム・リンク (DPL) 要求の場合に限られます。</li> <li>NONE - 親和性関係は定義されていません。</li> </ul>
親和性のクリエイターを示すキー	AFFKEY	<p>トランザクション・グループとその場所からなる、トランザクションの起動元を示す最大 64 文字の文字列。親和性の関係と存続時間の値に応じて、このキーはユーザー ID、論理装置名、ルーティング領域名、および端末 ID のいずれかの組み合わせになり、形式は <code>userid.luname.rtrname.termid</code> になります。これらの値のうち 1 つ以上が親和性に適用されない場合、キーのその部分はブランクのままになります。</p> <p>BAPPL 親和性の場合、このフィールドには CICS BTS 活動の活動 ID が入ります。</p> <p>LOCKED 親和性の場合、このフィールドにはルーティング領域名とネットワーク ID およびローカル作業単位 ID の組み合わせが入ります。</p> <p>グローバル親和性では、<b>GA</b> という値が表示されます。</p>
ターゲット領域	AOR	トランザクション・グループに関連付けられているトランザクションの経路指定先の CICS システム名。
BTS 活動 ID	CBTSKEY	親和性キーの 16 進数表現。通常の文字表現ではこのキーの一部を表示できない場合があるので、このキーはこの追加形式で表示されます。
親和性存続時間	LIFETIME	<p>トランザクション・グループに関連したトランザクションを処理する際に使用する親和性存続時間。親和性存続時間の値は次のとおりです。</p> <ul style="list-style-type: none"> <li>DELIMIT - 疑似会話モードが END になるまで。</li> <li>LOGON - 端末セッションの期間内。</li> <li>NONE - 親和性存続時間は定義されていません。</li> <li>PCONV - 疑似会話の期間内。</li> <li>PERMANENT - ワークロードがアクティブである間。</li> <li>SIGNON - ユーザー・セッションがアクティブである間。</li> <li>SYSTEM - ターゲット領域がアクティブである間。</li> <li>ACTIVITY - CICS BTS の活動がアクティブである間。</li> <li>PROCESS - CICS BTS プロセスがアクティブである間。</li> <li>UOW - 作業単位がアクティブである間。</li> </ul>
ローカル作業単位 ID	LOCUOWID	CICS ローカル作業単位 ID の 16 進数表記。通常の文字表現ではこのキーの一部を表示できない場合があるので、このキーはこの追加形式で表示されます。
端末 LU 名	LUNAME	トランザクション・グループに関連付けられているトランザクションの経路指定先の CICS システムの端末の論理装置名。
ネットワーク作業単位 ID	NETUOWID	CICS ネットワーク作業単位 ID の 16 進数表記。通常の文字表現ではこのキーの一部を表示できない場合があるので、このキーはこの追加形式で表示されます。
CBTS 活動の活動 ID	RESERVED2	CBTS 活動の ID。

表 31. WLMATAFF ビューのフィールド (続き)		
フィールド	属性名	説明
ルーティング領域の端末	TERMID	トランザクション・グループに関連付けられているトランザクションの経路指定先の CICS システムの端末 ID。
ルーティング領域	TORNAME	トランザクション・グループに関連付けられているトランザクションの経路指定先の CICS システム名。
トランザクション・グループ	TRANGRP	アクティブなトランザクション・グループの名前。値 <b>\$\$DTGA\$\$</b> は、トランザクション・グループに関連付けられていないトランザクションで使用するデフォルトのトランザクション・グループを表します。
ユーザー ID	USERID	ユーザー ID。
ワークロード名	WORKLOAD	アクティブなトランザクション・グループのあるワークロードの名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。

## 完全なアクティブ・ルーティング領域の詳細 - WLMAROUT

「全アクティブ・ルーティング領域」(WLMAROUT) ビューには、コンテキストとして識別されている CICSplex 内のすべてのアクティブ・ルーティング領域に関する情報が表示されます。

### 提供されるビュー

メインメニューからアクセスするには、

「**アクティブ・ワークロード・ビュー**」 > 「**完全なアクティブ・ルーティング領域の詳細**」

表 32. 提供されている「 <b>アクティブなルーティング領域</b> 」(WLMAROUT) ビュー・セットのビュー	
ビュー	注
アクティブなルーティング領域 EYUSTARTWLMAROUT.DETAILED	選択したアクティブ・ルーティング領域に関する詳細情報
アクティブなルーティング領域 EYUSTARTWLMAROUT.TABULAR	ワークロードに関連したすべてのアクティブ・ルーティング領域に関するテーブル形式の情報

### アクション

表 33. WLMAROUT ビューに使用可能なアクション	
アクション	説明
GET	ワークロード仕様の名前。ワークロード名 '*' を指定すると、すべてのワークロードにおけるすべてのルーティング領域に関する詳細を戻します。他の種類の汎用ワークロード名を指定することはできません。

## フィールド

表 34. WLMAROUT ビューのフィールド		
フィールド	属性名	説明
異常終了の補正状況	ABDCMPSTATE	<p>このフィールドは、TOR が現在ルーティングしているトランザクションの異常終了の補正状況を報告します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• YES <ul style="list-style-type: none"> <li>– このルーター領域は、異常終了の補正が有効な状態で、トランザクションを実行しています。</li> </ul> </li> <li>• NO <ul style="list-style-type: none"> <li>– このルーター領域は、有効な異常終了の補正なしで、トランザクションを実行しています。</li> </ul> </li> </ul>
ルーティング領域コンタクト状況	CON_STATUS	<p>ルーティング領域と CMAS の間の接続の状況。値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• LOSTCON - ルーティング領域と CMAS の間の接続は失われています。</li> <li>• N_A - ルーティング領域と、ワークロードを管理する CMAS との間の接続は使用可能です。</li> </ul>
FMID 順序番号	FMIDSEQ	<p>現行領域が実行されているオペレーティング・システム・バージョンの FMID 値のシーケンス番号部分。</p>
MVS システム名	MVSSYSNAME	<p>このルーティング領域が実行されているオペレーティング・システムの ID。</p>
最適化の状況	OWSTATE	<p>このフィールドは、最適化されたワークロード・ルーティング機能の現行ルーティング領域の状況を報告します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ACTIVE <ul style="list-style-type: none"> <li>– このルーター領域は、最適化されたワークロード状態で実行されています。</li> </ul> </li> <li>• INACTIVE <ul style="list-style-type: none"> <li>– この領域は最適化されたワークロード状態で実行可能ですが、以下の 1 つ以上の理由で、現在最適化されていません。 <ul style="list-style-type: none"> <li>- 領域に RS サーバーへの接続がない。</li> <li>- 領域は RS サーバーに接続されているが、サーバーがカップリング・ファシリティーに接続できていない。</li> <li>- 最適化されたワークロードのルーティング機能が現行のターゲット領域に対して無効である。</li> <li>- この領域の RS サーバー 更新頻度の値が 0 である、すなわち、この領域の最適化機能が使用可能にされていない。</li> </ul> </li> </ul> </li> <li>• N_A <ul style="list-style-type: none"> <li>– このルーター領域は、最適化されたワークロード管理を提供するように構成されていません。最適化されていない WLM ルーティング決定しか行えません。</li> </ul> </li> </ul> <p>WUI が使用している CMAS が領域状況 (RS) サーバーに接続されていない場合、リモート領域の最適化状況は更新されず、デフォルトの INACTIVE になります。そのような状態になっている場合は、すべての報告 CMAS を RS サーバーに接続するか、「アクティブなターゲット 領域」ビューを使用して最適化状況をチェックしてください。</p>
報告 CMAS 名	RPTINGCMAS	<p>このレコードを指定した CMAS の名前。</p>
作業単位の完了	RTUOWEND	<p>動的プログラム・リンクの呼び出し側プログラムが作業単位を完了した回数。</p> <p>これらの完了は、CICS SYNCPOINT コマンドまたは通常のタスク終了の結果である可能性があります。</p>
ルーターが開始された現地時間	STRTTIME	<p>この CICS の実行が開始された時刻。</p>

表 34. WLMAROUT ビューのフィールド (続き)		
フィールド	属性名	説明
ルーティング領域名	TOR	ルーティング領域の働きをしており、ワークロードに関連しているアクティブな CICS システムの名前。
コントロール CMAS 名	TOROWNER	コントロール CMAS の名前。
ワークロード名	WORKLOAD	ワークロード仕様の名前。
ワークロード所有者のシステム ID	WRKLOWNER	ワークロードを作成した CMAS の 4 文字の CICS システム ID。



## 第 6 章 CICSplex SM API サンプル・プログラム

CICSplex SM には、いくつかのサンプル・プログラムが用意されています。

各サンプル・プログラムは、配布されたときの言語のいずれかで、ここに示されます。各言語で提供されるサンプル・プログラムのリストおよびそれらが配布されるライブラリーについては、[サンプル・プログラム](#)を参照してください。

注：その他のサンプル CICSplex SM API プログラムを、IBM CICS を通じて入手できます。

### EYU#API1

プログラム EYU#API1 は、TSO 環境用に REXX で作成されます。

#### EYU#API1 について

このプログラムは、以下の処理を行います。

- API への接続を確立します。
- DFH、EYU、または IBM で始まらないすべての PROGRAM リソース・テーブル・レコードを含む結果セットを作成します
- 結果セットに各レコードを取得します。
- CICS CVDA 属性をすべて、意味のある文字値に変換します。
- 端末上に各レコードを表示して、プログラム名、言語、使用可能状況、および CEDF 状況を示します。
- API 接続を終了します。

使用されるコマンド：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_INT0_OBJECTLEN = 136          /* LENGTH OF PROGRAM TABLE */
Do III = 1 To W_RECcnt
  XX = EYUAPI('FETCH INTO(W_INT0_OBJECT)' ,
              'LENGTH(W_INT0_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_INT0_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 SCRNL0G
NO_CONNECT:
  W_MSG_TEXT = 'ERROR CONNECTING TO API.'
  Signal SCRNL0G
NO_GET:
  W_MSG_TEXT = 'ERROR GETTING RESOURCE TABLE.'
  Signal SCRNL0G
NO_FETCH:
  W_MSG_TEXT = 'ERROR FETCHING RESULT SET.'
  Signal SCRNL0G
SCRNL0G:
  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

```

EYUxAPI1 の C/370、COBOL、および PL/I の各バージョンは、CICS 環境用に作成されているため、**EXEC CICS SEND** コマンドをコメント化し、上記の言語固有の出力ステートメントをアンコメントすることにより、MVS バッチ環境で実行するよう変換することができます。

## EYUCAPI2

プログラム EYUCAPI2 は、CICS 環境用に C で作成されます。

### EYUxAPI2 について

このプログラムは、以下の処理を行います。

- API への接続を確立します。
- アセンブラーの言語属性を持つ PROGRAM リソース・テーブル・レコードを識別するフィルターを定義します。
- DFH、EYU、または IBM で始まらないすべての PROGRAM リソース・テーブル・レコードを含む結果セットを作成します
- 指定されたフィルター (LANGUAGE=ASSEMBLER) に一致する結果セット内のそれらのレコードにマークを付けます。
- マークを付けられたレコードを新しい結果セットにコピーします。
- マークを付けられたレコードを元の結果セットから削除します。
- 結果セット (LANGUAGE=ASSEMBLER および LANGUAGE≠ ASSEMBLER) ごとに、以下のようになります。
  - 各レコードを取り出します。
  - CICS CVDA 属性をすべて変換します。
  - 端末で各レコードを表示します。
- API 接続を終了します。

使用されるコマンド: 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 COPIed 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_INT0_OBJECT;
int W_INT0_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_INT0_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_INT0_OBJECT)
                LENGTH(W_INT0_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_INT0_OBJECT.PROGRAM,8);
        EXEC CPSM TRANSLATE OBJECT("PROGRAM ")
                ATTRIBUTE("LANGUAGE ")
                FROMCV(W_INT0_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_INTD_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_INTD_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 SCRNL0G;
NO_FILTER:
    strcpy(W_MSG_TEXT,"ERROR CREATING FILTER.\n");
    goto SCRNL0G;
NO_GET:
    strcpy(W_MSG_TEXT,"ERROR GETTING RESOURCE TABLE.\n");
    goto SCRNL0G;
NO_MARK:
    strcpy(W_MSG_TEXT,"ERROR MARKING RESULT SET.\n");
    goto SCRNL0G;
NO_COPY:
    strcpy(W_MSG_TEXT,"ERROR COPYING RESULT SET.\n");
    goto SCRNL0G;
NO_DELETE:
    strcpy(W_MSG_TEXT,"ERROR DELETING FROM RESULT SET.\n");
    goto SCRNL0G;
NO_LOCATE:
    strcpy(W_MSG_TEXT,"ERROR LOCATING TO TOP OF RESULT SET.\n");
    goto SCRNL0G;
NO_FETCH:
    strcpy(W_MSG_TEXT,"ERROR FETCHING RESULT SET.\n");
    goto SCRNL0G;
NO_TRANSLATE:
    strcpy(W_MSG_TEXT,"ERROR TRANSLATING ATTRIBUTE\n");
    goto SCRNL0G;
SCRNL0G:
    /* 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;
}

```

EYUxAPI2 の C、C++、COBOL、および PL/I の各バージョンは、CICS 環境用に作成されているため、**EXEC CICS SEND** コマンドをコメント化し、上記の言語固有の出力ステートメントをアンコメントすることにより、MVS バッチ環境で実行するよう変換することができます。



## EYUAAPI3

プログラム EYUAAPI3 は、MVS バッチ環境用にアセンブラーで作成されます。

### EYUAAPI3 について

このプログラムは、以下の処理を行います。

- コンテキストを既存の CICSplex に設定して、API への接続を確立します。
- 推奨された新規 CICSplex 名が CICSplex SM に対して CICSplex、CMAS、CICS システム、または CICS システム・グループに対してまだ定義されていないことを検証します。
- 既存の CICSplex 定義用の CPLEXDEF リソース・テーブル・レコードが含まれる結果セットを作成し、そのレコードを取り出します。
- 既存のレコードをモデルとして使用して、新しい CPLEXDEF リソース・テーブル・レコードを作成します。
- 既存の CICSplex に関連付けられた CICSplex リソース・テーブル・レコードが含まれる結果セットを作成し、それらのレコードを取得します。
- 既存のレコードをモデルとして使用して、新しい CICSplex リソース・テーブル・レコードを作成します。
- CICS システム、CICS システム・グループ、ワークロード管理定義、リアルタイム分析定義、およびリソース・モニター定義を含め、既存の CICSplex に関連付けられたすべてのリソース・テーブル・レコードを順次取り出します。
- 既存のレコードをモデルとして使用して、新しい CICSplex 用の必要なリソース・テーブル・レコードをすべて作成します。
- 必要なリソース・テーブル・レコードがすべて作成される前にエラーが発生した場合は、その新規 CICSplex 定義を除去します。
- API 処理スレッドを切り離します。

使用されるコマンド: 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
      CONTEXT(CONTEXT)
      VERSION(=CL4'0130')
      THREAD(THREAD)
      RESPONSE(RESPONSE)
      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
      CONTEXT(NEWPLEX)
      THREAD(THREAD)
      RESPONSE(RESPONSE)
      REASON(REASON)
CLC    RESPONSE,EYUVALUE(OK)    RESPONSE OK?
BE     ERRNISC                  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    OH
*
*      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 ERNMISS             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 OH
*-----*
* 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. 変数
* 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(ADDCLLEN),ADDC      Build most of parm
MVC    PARMLLEN,=A(ADDCLLEN)        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    Add CMAS name to PARM      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
      PARMLLEN(PARMLLEN)               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

```

NODATA	BAS	R10,DISCARD	Discard the GET result set	
	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
*                  OBJECT(OBJECT)
*                  CRITERIA(CRITERIA)
*                  LENGTH(CRITLEN)
*                  COUNT(COUNT)
*                  RESULT(RESULT)
*                  THREAD(THREAD)
*                  CONTEXT(CONTEXT)
*                  RESPONSE(RESPONSE)
*                  REASON(REASON)
*
*      GETNOCRT BR      R10
*              DS      0H
*              EXEC     CPSM GET
*                  OBJECT(OBJECT)
*                  COUNT(COUNT)
*                  RESULT(RESULT)
*                  THREAD(THREAD)
*                  CONTEXT(CONTEXT)
*                  RESPONSE(RESPONSE)
*                  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
*                  OBJECT(OBJECT)
*                  DATALENGTH(RECLEN)
*                  RESULT(RESULT)
*                  THREAD(THREAD)
*                  RESPONSE(RESPONSE)
*                  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
      ALL
      INTO(0(,R2))
      LENGTH(LENGTH)
      COUNT(COUNT)
      RESULT(RESULT)
      THREAD(THREAD)
      RESPONSE(RESPONSE)
      REASON(REASON)
CLC    RESPONSE,EYUVALUE(OK)
BNE    ERRFETCH
BR      R10

```

		REASON(REASON)		
CRENOPRM	B	DS	CRECHKRR	
	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	



R13	EQU	13
R14	EQU	14
R15	EQU	15
	END	EYUAAPI3

## EYULAPI4

プログラム EYULAPI4 は、CICS 環境用に COBOL で作成されます。

### EYULAPI4 について

このプログラムは、以下の処理を行います。

- API への接続を確立します。
- 1 というバージョンを指定する TS モデル (TSMDEF) 用の BAS 定義を作成します。
- 以前に定義された TSMDEF を含む結果セットを作成します。
- PERFORM OBJECT コマンドを発行して、TSMDEF をターゲット・スコープにインストールします。
- API 接続を終了します。
- BAS エラーは、BINCONRS、BINCONSC、および BINSTERR のリソース・テーブル・レコードを使用して処理されます。

**使用されるコマンド:** 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, SERIALY 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-PARMLN        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.
*
        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.
*
        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 PICZZZZZZ9.
        STRING 'Installing ' DELIMITED BY SIZE
            PICZZZZZZ9 DELIMITED BY SIZE
            ' TSMDEF Entries...' DELIMITED BY SIZE
            INTO 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-PARMLN.

        EXEC CPSM PERFORM OBJECT('TSMDEF')
            ACTION('INSTALL')
            PARM(W-PARM)
            PARMLN(W-PARMLN)
            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 SCRNL0G2.

```

```

*****
* 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 SCRNL0G2.
    MOVE W-REASON TO PICZZZ9.
    STRING '*** REASON=' DELIMITED BY SIZE PICZZZ9
    DELIMITED BY SIZE BLNKPAD DELIMITED BY SIZE INTO W-TEXT.
    PERFORM SCRNL0G2.
    MOVE '*** Unexpected error condition arose' TO W-TEXT.
    PERFORM SCRNL0G2.
* 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 SCRNL0G2.
    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 DISPFEEED
        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 SCRNL0G2
    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 SCRNL0G2.
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-FBUBFF) 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 SCRNLG2.
    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 DISPFEEED
        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 SCRNLG2.
    EGETFER.
    EXIT.

*****
* This subroutine displays FEEDBACK information *
*****
    DISPFEEED.
        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 SCRNLG2.
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 SCRNLG2.
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 SCRNLG2.
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 SCRNLG2.
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 SCRNLG2.
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 SCRNLG2.
MOVE CEIBFN OF BINSTERR TO PICZZZ9.
MOVE SPACES TO W-TEXT.
STRING ' EibFn=' PICZZZ9
DELIMITED BY SIZE INTO W-TEXT END-STRING.
PERFORM SCRNLG2.
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 SCRNLG2.
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 SCRNL0G2.
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 ' CandAss0=' CHARV
DELIMITED BY SIZE INTO W-TEXT END-STRING.
PERFORM SCRNL0G2.
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 SCRNL0G2.
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 ' ExistAss0=' CHARV
DELIMITED BY SIZE INTO W-TEXT END-STRING.
PERFORM SCRNL0G2.
EXIT.

```

```

*****
* This subroutine displays BINCONSC information *
*****
DISPBISC.
MOVE W-BUFFER TO BINSTERR.
MOVE ERRORP 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 SCRNL0G2.
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 SCRNL0G2.
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 SCRNL0G.
NO-CREATE.
      MOVE 'ERROR CREATING DEFINITION.' TO W-MSG-TEXT.
      GO TO SCRNL0G.
NO-GET.
      MOVE 'ERROR GETTING RESOURCE TABLE.' TO W-MSG-TEXT.
      GO TO SCRNL0G.
NO-INSTALL.
      MOVE 'ERROR INSTALLING RESULT SET.' TO W-MSG-TEXT.
      GO TO SCRNL0G.
NO-TRANSLATE.
      MOVE 'ERROR TRANSLATING ATTRIBUTE.' TO W-MSG-TEXT.
      GO TO SCRNL0G.
SCRNL0G.
*      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.
SCRNL0G2.
*      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.

```

EYUxAPI4 の COBOL バージョンは、CICS 環境用に作成されているため、**EXEC CICS SEND** コマンドをコメント化し、上記の言語固有の出力ステートメントをアンコメントして、MVS/ESA バッチ環境で実行できるよう変換することができます。



## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができません。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス涉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様自身の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive, MD-NC119 Armonk,*

*NY 10504-1785*

*United States of America*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関す



る実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

## プログラミング・インターフェース情報

CICS には、プログラミング・インターフェースと見なすことのできる資料と、プログラミング・インターフェースと見なすことのできない資料があります。

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- [アプリケーションの開発](#)
- [システム・プログラムの開発](#)
- [CICS TS セキュリティー](#)
- [外部インターフェースに向けた開発](#)
- [アプリケーション開発のリファレンス](#)
- [リファレンス: システム・プログラミング](#)
- [リファレンス: 接続](#)

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- [トラブルシューティングおよびサポート](#)
- [CICS TS 診断参照](#)

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- [アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス](#)
- [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
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSplex SM Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM アプリケーション・プログラミング・ガイドおよび CICSplex SM アプリケーション・プログラミング・リファレンス
- CICS における Java アプリケーション

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSplex SM Problem Determination Guide

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com)<sup>®</sup> は、世界の多くの国で登録された International Business Machines Corporation の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Linux<sup>®</sup> は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

## 製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

### 適用範囲

IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

### 個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

## 商用使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

## 権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM これらの資料の内容 についていかなる保証もしません。これらの資料は、特定物として現存するままの状態 で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

## IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (ソフトウェア・オファリング) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

### CICSplex SM Web ユーザー・インターフェース (メイン・インターフェース) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

### CICSplex SM Web ユーザー・インターフェース (データ・インターフェース) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名またはその他の個人情報を、セッションごとの Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

### CICSplex SM Web ユーザー・インターフェース (「Hello World」 ページ) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

### CICS Explorer の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの設定および持続的な設定を使用して収集する場合があります。これらの設定を無効にすることはできませんが、ユーザー・パスワードの暗号化形式でのディスクへの保管は、サインオン中にチェック・ボックスにチェック・マークを付けることによるユーザーの明示的な操作によってのみ有効化することができます。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』 (<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビー

コン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』 (<http://www.ibm.com/software/info/product-privacy>) を参照してください。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。  
なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

「アクティブなターゲット領域」ビュー  
一般 (WLMATARG) [125](#)  
「アクティブなルーティング領域」ビュー  
一般 (WLMAROUT) [141](#)  
「アクティブなワークロードでのターゲット領域」ビュー  
一般 (WLMAWAOR) [119](#)  
アクティブ・ワークロード [71](#)  
アクティブ・ワークロード定義 [71](#)  
アクティブ・ワークロード定義の更新、タスク例 [94](#)  
「アクティブ・ワークロード定義」ビュー  
一般 (WLMAWDEF) [134](#)  
アクティブ・ワークロード内のターゲット領域の静止、タスク例 [88](#)  
アクティブ・ワークロード内のルーティング領域 (Routing regions in an active workload) [71](#)  
アクティブ・ワークロードのターゲット配布係数 [71](#)  
アクティブ・ワークロードの動的トランザクション [71](#)  
「アクティブ・ワークロードの動的トランザクション」ビュー  
一般 (WLMATRAN) [138](#)  
アクティブ・ワークロードのトランザクション・グループ [70](#)  
アクティブ・ワークロードのトランザクション・グループの親和性 [70](#)  
「アクティブ・ワークロードのトランザクション・グループの親和性」ビュー  
一般 (WLMATAFF) [139](#)  
アクティブ・ワークロードのトランザクション・グループ・ビュー  
一般 (WLMATGRP) [135](#)  
「アクティブ・ワークロードのルーティング領域」ビュー  
一般 (WLMAWTOR) [117](#)  
アクティブ・ワークロード・ビュー  
一般 (WLMAWORK) [113](#)  
アクティブ・ワークロードへのルーティング領域の追加、タスク例 [88](#)  
アフィニティー 存続時間 [50](#)  
異常終了の確率 [32](#)  
異常終了の正常性  
ABENDCRIT [32](#)  
異常終了補正  
ワークロード管理に直接アクセスする場合 [55](#)  
異常終了ロード  
ABENDTHRESH [32](#)  
インバウンド・クライアント DPL 要求の動的ルーティング、タスク例 [102](#)  
エンタープライズ Bean  
ワークロード分離 [64](#)  
ワークロード・ルーティング [19, 59](#)  
オブジェクト  
DTRINGRP [70](#)  
LNKSWSCG [70](#)  
LNKSWSCS [70](#)

オブジェクト (続き)

TRANGRP [70](#)  
WLMATAFF [70](#)  
WLMATGRP [70](#)  
WLMATRAN [71](#)  
WLMAWAOR [71](#)  
WLMAWDEF [71](#)  
WLMAWORK [71](#)  
WLMAWTOR [71](#)  
WLMDEF [71](#)  
WLMGROUP [71](#)  
WLMINGRP [71](#)  
WLMINSPC [71](#)  
WLMSPEC [71](#)

親和性、トランザクション間 [64](#)

親和性、ワークロード  
説明 [29](#)

## [カ行]

疑似会話型トランザクションの受け入れ、タスク例 [92](#)  
キュー・アルゴリズム

説明 [19](#)

ゴール・アルゴリズム

説明 [19](#)

コピーブック [37](#)

## [サ行]

作成、EYU9WRAM のユーザー置換 [52](#)

サンプル・プログラム

リスト [145](#)

自動インストール

ワークロード定義の [69](#)

除去

仕様からのルーティング領域のスコープ [37](#)

トランザクション親和性 [50](#)

スコープ

関連付け

ターゲット領域として、トランザクション・グループに [81](#)

ルーティング領域として、ワークロード仕様への [77](#)

スコープとの関連の更新

トランザクション・グループ

説明 [18](#)

スコープの関連付け

ターゲット領域として、トランザクション・グループに [81](#)

ターゲット領域として、ワークロード仕様への [77](#)

## [タ行]

ターゲット領域

リリース要件 [37](#)

割り当て



ターゲット領域 (続き)

割り当て (続き)

トランザクション・グループ [81](#)

ターゲット領域スコープからの領域の除去、タスク例 [87](#)

ターゲット領域スコープへの領域の追加、タスク例 [86](#)

ターゲット領域の選択

セットからの [46](#)

通知 (ルート) [46](#)

特定の [48](#)

トランザクション異常終了 [46](#)

トランザクション開始 [46](#)

トランザクション終了 [46](#)

ルーティング試行完了 [46](#)

ルート選択 [46](#)

ルート選択エラー [46](#)

ターゲット領域を選択するためのリアルタイム分析の使用、タスクの例 [96](#)

タスク、例

アクティブ・ワークロード定義の更新 [94](#)

アクティブ・ワークロード内のターゲット領域の静止

[88](#)

アクティブ・ワークロードへのルーティング領域の追加

[88](#)

インバウンド・クライアント DPL 要求の動的ルーティング [102](#)

疑似会話型トランザクションの受け入れ [92](#)

ターゲット領域スコープからの領域の除去 [87](#)

ターゲット領域スコープへの領域の追加 [86](#)

ターゲット領域を選択するためのリアルタイム分析の使用 [96](#)

特定のターゲット領域への特定のトランザクションのルーティング [89](#)

ピアツーピア DPL 要求の動的ルーティングの使用 [103](#)

ユーザーから特定のターゲット領域へのトランザクションのルーティング [91](#)

ワークロードからのアクティブ・トランザクションの破棄 [95](#)

ワークロード仕様の更新 [95](#)

ワークロード定義の非アクティブ化 [93](#)

ワークロードの管理 [84](#)

CICS BTS アクティビティの分離 [106](#)

CICS BTS アクティビティのルーティング [104](#)

EXEC CICS START TERMID を使用した動的ルーティング [100](#)

Link3270 ブリッジ・ワークロードの管理 [107](#)

タスクの例

アクティブ・ワークロード定義の更新 [94](#)

アクティブ・ワークロード内のターゲット領域の静止

[88](#)

アクティブ・ワークロードへのルーティング領域の追加 [88](#)

インバウンド・クライアント DPL 要求の動的ルーティング [102](#)

疑似会話型トランザクションの受け入れ [92](#)

ターゲット領域スコープからの領域の除去 [87](#)

ターゲット領域スコープへの領域の追加 [86](#)

ターゲット領域を選択するためのリアルタイム分析の使用 [96](#)

特定のターゲット領域への特定のトランザクションのルーティング [89](#)

ピアツーピア DPL 要求の動的ルーティング [103](#)

ユーザーから特定のターゲット領域へのトランザクションのルーティング [91](#)

タスクの例 (続き)

ワークロードからのアクティブ・トランザクションの破棄 [95](#)

ワークロード仕様の更新 [95](#)

ワークロード定義の非アクティブ化 [93](#)

ワークロードの管理 [84](#)

CICS BTS アクティビティの分離 [106](#)

CICS BTS アクティビティのルーティング [104](#)

EXEC CICS START TERMID を使用した動的ルーティング [100](#)

Link3270 ブリッジ・ワークロードの管理 [107](#)

突き合わせ基準、トランザクション [81](#)

動的トランザクション・ルーティング [65](#)

動的ルーティング

追加サポートの要求

異常終了の補正に関する考慮事項 [55](#)

概要 [52](#)

サンプルの呼び出しシーケンス [56](#)

処理に関する考慮事項 [54](#)

トランザクション親和性に関する考慮事項 [55](#)

ルート・エラーに関する考慮事項 [54](#)

CICSplex SM ワークロード管理の呼び出し [53](#)

EYURWTRA 連絡域 [53](#)

プログラム、EYU9XLOP [5](#)

CICSplex SM 処理のカスタマイズ

サンプル・プログラム [37](#)

トランザクション・ルーティングのアクション [48](#)

非動的トランザクション [45](#)

ユーザー置換プログラムのインストール [52](#)

CICSplex SM 処理の概要 [35](#)

DTR プログラムの変更 [44](#)

CICSplex SM 処理のカスタマイズ

ターゲット領域の選択 [46](#)

トランザクション・ルーティングのアクション [47](#)

リリース要件 [37](#)

CICSplex SM データ域 [52](#)

動的ワークロード・ルーティング [8](#)

特定のターゲット領域への特定のトランザクションのルーティング、タスク例 [89](#)

トランザクション

突き合わせ基準 [81](#)

トランザクション異常終了 [48](#)

トランザクション親和性の作成 [49](#)

トランザクション開始 [48](#)

トランザクション間の類縁性

削除 [50](#)

作成 [49](#)

識別 [66](#)

説明 [29](#)

定義 [64](#)

トランザクション・グループ

作成 [79](#)

説明 [18](#)

ターゲット領域との関連付け [81](#)

突き合わせ基準 [81](#)

トランザクション・グループ内のトランザクション・ビュー [70](#)

トランザクション・グループ・ビュー [70](#)

トランザクション終了 [48](#)

トランザクションの類縁性

削除 [50](#)

作成 [49](#)

説明 [29](#)

## [ハ行]

ピアツーピア DPL 要求の動的ルーティング、タスク例 [103](#)  
百分位数ゴール [62](#), [63](#)  
ビュー  
    その要約  
        定義を作成するための [74](#)  
ビューの要約  
    定義の作成用 [74](#)  
プログラム、サンプル  
    リスト [145](#)  
プロセス・タイプ (BTS) [18](#)  
分散プログラム・リンク (DPL)  
    リリース要件 [37](#)

## [ヤ行]

ユーザーから特定のターゲット領域へのトランザクションの  
ルーティング、タスク例 [91](#)  
ユーザー置換プログラムのインストール [52](#)  
要求領域  
    リリース要件 [37](#)

## [ラ行]

リリース要件 [37](#)  
リンクに依存しないキュー・アルゴリズム [62](#)  
リンクに依存しないゴール・アルゴリズム [63](#)  
ルーティング、ワークロード [59](#)  
ルーティング・アルゴリズム [67](#)  
ルーティング試行完了 [48](#)  
ルーティング領域  
    ワークロード管理のアクティブ化  
        CICS システムがアクティブであるとき [33](#)  
        CICS システムがアクティブでないとき [33](#)  
        CMAS がアクティブでない場合 [34](#)  
    ワークロード仕様への関連付け [77](#)  
ルート選択 [48](#)  
ルート選択エラー [48](#)  
ルート・フィールド [82](#)  
連絡域  
    EYURWCOM [52](#)  
    EYURWTRA [53](#)

## [ワ行]

ワークロード親和性  
    説明 [29](#)  
ワークロードからのアクティブ・トランザクションの破棄、  
タスク例 [95](#)  
ワークロード管理  
    説明 [18](#)  
    ルーティング領域でのアクティブ化  
        CICS システムがアクティブであるとき [34](#)  
        CMAS が使用不可である場合 [34](#)  
ワークロード管理 (WLM)  
    トランザクション間の類縁性 [64](#)  
    ワークロード・バランシング  
        異常終了補正 [61](#)  
        キュー・アルゴリズム [61](#)  
        ゴール・アルゴリズム [62](#)  
    ワークロード分離 [64](#), [68](#)  
    ワークロード・ルーティング

ワークロード管理 (WLM) (続き)  
    ワークロード・ルーティング (続き)  
        異常終了補正 [62](#)  
        リンクに依存しないキュー・アルゴリズム [62](#)  
        リンクに依存しないゴール・アルゴリズム [63](#)  
ワークロード管理: CICSplex SM [1](#), [11](#)  
ワークロード管理コンポーネント [6](#)  
ワークロード管理のアクティブ化 [33](#)  
「ワークロード管理」ビュー  
    定義との関係 [72](#)  
ワークロード機能 [65](#)  
ワークロード・グループ  
    説明 [18](#)  
    ワークロード・グループ内のワークロード定義ビュー [71](#)  
    ワークロード・グループ・ビュー [71](#)  
    ワークロード仕様 [67](#)  
    ワークロード仕様内のワークロード・グループ・ビュー [71](#)  
    ワークロード仕様の更新、タスク例 [95](#)  
    ワークロード仕様ビュー [71](#)  
    ワークロード定義  
        作成 [80](#)  
        説明 [18](#)  
    ワークロード定義の非アクティブ化、タスク例 [93](#)  
    ワークロード定義ビュー [71](#)  
    ワークロードの管理、タスク例 [84](#)  
    ワークロード・ビュー [82](#)  
    ワークロード分離  
        説明 [22](#)  
    ワークロード・マネージャー [1](#)  
    ワークロード・ルーティング  
        実装 [67](#)  
        説明 [19](#)

## A

APPC [50](#)

## C

CICS BTS アクティビティの分離、タスク例 [106](#)  
CICS BTS アクティビティの平衡化、タスク例 [104](#)  
CICS ビジネス・トランザクション・サービス (BTS)  
    親和性の処理 [18](#)  
    リリース要件 [30](#), [37](#)  
    BTS セット [18](#)  
    CICSplex に関する考慮事項 [18](#)

## D

DSRTPGM SIT パラメーター [33](#)  
DTRINGRP オブジェクト [70](#)  
DTRPGM SIT パラメーター [33](#)  
DTRPROG API [44](#)

## E

EXEC CICS START TERMID を使用した動的ルーティング、タ  
スク例 [100](#)  
EYU9WRAM ルーティング・アクション・モジュール  
    コピーブック [37](#)  
    作成、ユーザー置換 [52](#)  
EYU9XLOP [5](#)  
EYU9XLOP 動的ルーティング・プログラム [33](#), [53](#)



EYURWCOD コピーブック [37](#), [52](#)  
EYURWCOM 連絡域 [37](#), [52](#)  
EYURWSVD コピーブック [37](#), [52](#)  
EYURWSVE コピーブック [37](#), [52](#)  
EYURWTRA 連絡域 [53](#)

## [特殊文字]

オブジェクト・モデル、推奨される使用法  
ワークロード管理 [69](#)

## L

Link3270 ブリッジ・ワークロードの管理、タスク例 [107](#)  
LNGOAL ルーティング・アルゴリズム [63](#)  
LNKSWSCG オブジェクト [70](#)  
LNKSWSCS オブジェクト [70](#)  
LNQUEUE ルーティング・アルゴリズム [62](#)  
LUTYPE6.2 [50](#)

## S

SM\_BALANCE [45-47](#)  
SM\_CREAFF [49](#)  
SM\_DELAFF [50](#)  
SM\_ROUTE [45](#), [48](#)  
SM\_SCOPE [45-47](#)  
START コマンド  
    動的ルーティング [19](#)  
START コマンド、動的ルーティング  
    リリース要件 [37](#)

## T

TRANGRP オブジェクト [70](#)

## W

Web ユーザー・インターフェース  
    ワークロード仕様  
        スコープ [77](#)  
        ワークロード定義 [80](#)  
WLM 仕様と CICS システムとの間のリンク (WLM specifications to CICS system links) [70](#)  
WLM 仕様とシステム・グループとの間のリンク [70](#)  
WLMAROUT ビュー [141](#)  
WLMATAFF オブジェクト [70](#)  
WLMATAFF ビュー [139](#)  
WLMATARG ビュー [125](#)  
WLMATGRP オブジェクト [70](#)  
WLMATGRP ビュー [135](#)  
WLMATRAN オブジェクト [71](#)  
WLMATRAN ビュー [138](#)  
WLMAWAOR オブジェクト [71](#)  
WLMAWAOR ビュー [119](#)  
WLMAWDEF オブジェクト [71](#)  
WLMAWDEF ビュー [134](#)  
WLMAWORK オブジェクト [71](#)  
WLMAWORK ビュー [113](#)  
WLMAWTOR オブジェクト [71](#)  
WLMAWTOR ビュー [117](#)  
WLMDEF オブジェクト [71](#)  
WLMGROUP オブジェクト [71](#)  
WLMINSPC オブジェクト [71](#)  
WLMSPEC オブジェクト [71](#)



