

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

CICSplex SM CMCI REST API リファレン
ス



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 CMCI REST API Reference

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
第 1 章 CMCI REST API の概要	1
CMCI 要求.....	3
CMCI DELETE 要求数.....	3
CMCI DELETE 要求 URI.....	4
CMCI GET 要求数.....	5
CMCI GET 要求 URI.....	7
保持した結果セットを GET 要求のパフォーマンスを改善するために使用する	9
CMCI POST 要求数.....	11
CMCI POST 要求 URI.....	12
CMCI PUT 要求数.....	13
CMCI PUT 要求 URI.....	15
CMCI リソース名.....	17
CMCI XML.....	27
<request> エlement.....	27
<create> エlement.....	27
<update> エlement.....	28
<action> エlement.....	29
<delete> エlement.....	30
<parameter> エlement.....	31
<attributes> エlement.....	31
<response> エlement.....	32
<resultsummary> エlement.....	32
<records> エlement.....	34
<errors> エlement.....	35
<feedback> エlement.....	35
<installerror> エlement.....	36
<inconsistentscope> エlement.....	37
<inconsistentset> エlement.....	38
CMCI エスケープ・シーケンス.....	41
CMCI 問題判別.....	43
例: <resultsummary> 情報を使用した CICS 管理クライアント・インターフェース・エラーの診断....	43
例: <feedback> 情報を使用した CICS 管理クライアント・インターフェース・エラーの診断.....	44
例: CICS 管理クライアント・インターフェースのインストール・エラーの診断	45
CICS 管理クライアント・インターフェースのエラー・メッセージ.....	46
CMCI を使用したシステム初期設定パラメーターの照会.....	49
リソース定義との相互作用.....	53
CMCI を使用したリソース定義の作成.....	53
CMCI を使用したリソース定義の更新.....	54
CMCI を使用したリソース定義に対するアクションの実行.....	55
CMCI を使用したリソース定義の削除.....	56
特記事項.....	59
索引.....	65

この PDF について

この PDF は、CICS Transaction Server for z/OS の CICSplex SM エlement 用の CMCI REST アプリケーション・プログラミング・インターフェースの解説書です。この資料は、CICSplex SM と対話する HTTP ベースのシステム管理クライアントを作成しているシステム・プログラマーを対象としています。

本書で使用する用語および表記の詳細については、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。

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 CMCI REST API の概要

CICS 管理クライアント・インターフェース (CMCI) には、IBM CICS Explorer® などのシステム管理クライアント向けの REST アプリケーション・プログラミング・インターフェース (API) が用意されています。

クライアントは、CMCI に対して HTTP 要求を開始します。このインターフェースは、要求が有効であることを判別すると、CICSplex® SM API コマンドを構成します。スタンドアロン CICS 領域の場合には、CICS システム・コマンドを構成します。コマンドを実行した後、CMCI は HTTP 応答を作成します。要求が成功すると、応答は HTTP 200 (OK) 応答コードと結果セットが入った XML フィードの形式で、クライアントに戻されます。要求が失敗の場合は、OK ではない HTTP 応答コードと失敗の詳細で構成された応答となります。

CMCI の HTTP 要求と応答のフォーマットは、HTTP/1.1 プロトコルに基づきます。このプロトコルの詳細については、[HTTP プロトコル](#)を参照してください。

CMCI 要求の要素

CMCI 要求の形式は、HTTP ヘッダー、その後に URI (Universal Resource Identifier)、そして CICS または CICSplex SM リソースに加えられたすべての変更の詳細が入った XML 本体 (必要な場合) となります。

HTTP ヘッダー

ヘッダーには、以下の HTTP メソッドのいずれかが取り込まれます。

HTTP メソッド	説明
DELETE	CICSplex SM データ・リポジトリからリソースを除去するか、CSD からリソースを除去するか、またはインストール済みリソースを破棄します。
GET	CICSplex SM データ・リポジトリにあるリソースに関する情報を取り出すか、CSD にあるリソースに関する情報を取り出すか、またはインストール済みリソースに関する情報を取り出します。
POST	CICSplex SM データ・リポジトリにリソースを作成するか、CSD にリソースを作成します。
PUT	CICSplex SM データ・リポジトリにある既存のリソースを更新するか、CSD の既存のリソースを更新するか、または属性を設定してインストール済みリソースに対してアクションを実行します。また、CICSplex SM リソースおよび CSD リソースに対してもアクションを実行します。

URI

URI には、CICS または CICSplex SM リソースの名前が含まれます。これを使用して、指定されたリソースの 1 つ以上のインスタンスを識別するために照会のスコープと性質を詳細化する一連のパラメーターを指定します。GET 要求の URI では、API が結果のセットを保持するかまたは破棄するかも指定します。API が結果を保持する場合、新しい要求は、取り出し操作を繰り返さずに、保持された結果に対して作用できます。後続の要求を使用して、1 度に 1 つ以上のレコードを選択しながら、保持された結果のページを送ってゆくこともできます。保持されている結果セットに関する詳細は、[9 ページの『保持した結果セットを GET 要求のパフォーマンスを改善するために使用する』](#)を参照してください。

XML 本体

POST および PUT 要求には、XML 本体も組み込まれます。PUT 要求の場合、本体に収容される内容は、リソース属性に加えられる変更の詳細、またはターゲット・リソースに対して実行されるアクションです。POST 要求の場合、本体に取り込まれる内容は、新規リソース・インスタンスに設定する属性値です。

GET 要求および DELETE 要求では XML 本体は必要ありません。DELETE 要求に追加のパラメーターが必要な場合、それらのパラメーターは URI に含められる必要があり、オプションで XML 本体に追加す

することもできます。例えば、CSD リソース定義を削除する場合は、`PARAMETER=CSDGROUP(csdgrp)` を URI に含める必要があり、必要に応じて `<parameter name="CSD"/>` を XML 本体に追加できます。

CMCI 要求

CICS 管理クライアント・インターフェース (CMCI) を使用すると、インストール済みおよび定義 CICS および CICSplex SM リソースを管理する HTTP クライアント・アプリケーションを開発できます。CICS 管理クライアント・インターフェース 要求の形式は、HTTP ヘッダー、その後に URI (Universal Resource Identifier)、そして XML 本体 (適切な場合) となります。CICS 管理クライアント・インターフェース 要求には、DELETE、GET、POST、および PUT の 4 つがあります。

CMCI DELETE 要求数

CICS 管理クライアント・インターフェース (CMCI) は、HTTP DELETE メソッドを使用して、データ・リポジトリからリソースを除去するか、あるいは CICS または CICSplex SM からインストール済みリソースを破棄します。

クライアントは、以下の部分で構成される HTTP ヘッダーを使用して、DELETE 要求を形成します。

- メソッド名 (この場合は DELETE)
- 削除対象のリソースを特定する URI
- HTTP バージョン
- 許可証明書 (必要な場合)

インストール済みリソースまたは定義リソースに対して、DELETE 要求を実行できます。インターフェースは、操作リソースに対して作用する場合に CICS DISCARD コマンドを構成し、定義リソースに対して作用する場合に CICSplex SM REMOVE コマンドを構成します。

例えば、TR で始まる名前および P で始まるプログラムが含まれる、CICSplex PLEX1 内の全トランザクション定義を削除する場合、クライアントは以下の HTTP ヘッダーを作成します。

```
DELETE /CICSSystemManagement/CICSDefinitionTransaction/PLEX1?CRITERIA=NAME%3DTR%2A%20AND%20PROGRAM%3DP%2A HTTP/1.1
Host: example.com:23792
Authorization: Basic R1JFRDpQQVNTVzBSRA==
```

最初の行には、スペースで分離された以下の 3 つの部分があります。

- メソッド名
- 要求 URI。これは絶対パスで指定され、先頭が /CICSSystemManagement/ で、そのすぐ後にリソースを識別する外部名、ならびにそのリソースの削除対象となるインスタンスを特定するパラメーターおよびフィルターが続きます。URI を CICS 管理クライアント・インターフェースによる処理に適したものとするために、クライアントはスペースやアスタリスクなどの特定の制限付き文字をエスケープ・シーケンスに置換します。
- HTTP バージョン。CICS 管理クライアント・インターフェースへの接続時は、常に HTTP/1.1 です。

2 行目では、ターゲット・システムのホスト名およびポート番号を、URI を指定するのと同様にコロンの分離して特定します。この行の最後は、復帰、続いて改行 (LF) となります。

オプションの 3 行目には、許可証明書を含めます。CICS システム初期設定パラメーター **SEC** が YES に指定されて稼働しているシステムの場合、ユーザー ID およびパスワードを Base64 方式で基本認証ヘッダーに設定する必要があります。この例では、ユーザー ID は *FRED*、パスワードは *PASSWORD* です。

それから、別の行に最後の復帰および改行が置かれて、HTTP ヘッダーが完結します。

DELETE 要求に対する応答

DELETE 要求が完了すると、クライアントは、CMCI からの応答を受信します。応答は、HTTP ヘッダーと、結果サマリーおよび削除されたリソースの詳細を含む XML フィードで構成されています。

ヘッダーは、HTTP 応答コード、日時、ならびにサーバーおよびコンテンツ・タイプの詳細で構成されています。以下の例では、成功した DELETE 要求のヘッダーを示します。

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Tue, 02 Jun 2009 14:51:37 GMT
Server: IBM_CICS_Transaction_Server (zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

応答の本体は、要求に関する要約情報を表示する <resultsummary> エlement が収容された <response> ルート・Element で構成されます。例:

```
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int
http://example.com:30061/CICSSystemManagement/schema/CICSSystemManagement.xsd"
version="1.0" connect_version="0410">
<resultsummary api_response1="1024" api_response1_alt="OK"
api_response2="0" api_response2_alt="" recordcount="1" successcount="1" />
</response>
```

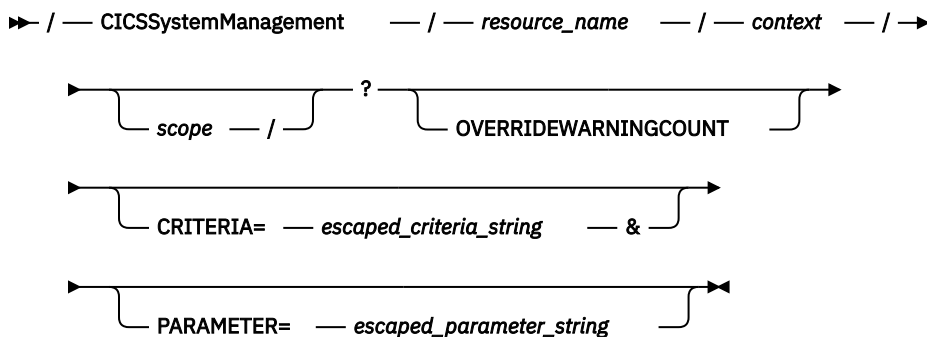
セキュリティの考慮事項

HTTP メソッドを使用する場合、一部のファイアウォールで、セキュリティ上の考慮事項が原因でファイアウォール経由の HTTP PUT トラフィックまたは DELETE トラフィックが許可されないことに注意してください。この制限に対応するために、[CMCI POST 要求](#)を使用して、DELETE 要求を POST 要求によってトンネル伝送することができます。

CMCI DELETE 要求 URI

DELETE 要求の URI により、操作で削除されるリソースのインスタンスを特定します。

以下のダイアグラムでは、DELETE 要求の URI 構文を示します。



オプション

/CICSSystemManagement

要求が CICS 管理クライアント・インターフェースにアクセスすることを示します。

CICSSystemManagement は大/小文字が区別されます。

resource_name

削除される CICS または CICSplex SM リソースに関連付けられる外部リソース名を指定します。例えば、リソース名 CICSLocalFile を指定すると、要求を CICSplex SM LOCFILE リソース・テーブルに関連付け、CICSResourceAssignmentDefinition を指定すると、要求を RASGNDEF リソース・テーブルに関連付けます。外部リソース名の完全なリストについては、[17 ページの『CMCI リソース名』](#)を参照してください。

HTTP DELETE メソッドは、すべてのリソースに対して有効であるわけではありません。特定のリソースにおいてこの HTTP メソッドが有効であるかどうかを判別するには、[CICSplex SM リソース・テーブル](#)にあるリソースの説明を参照してください。

リソース名は大/小文字が区別されません。正しくないリソース名がクライアントによって指定されると、HTTP 404 (Not Found) 応答が返されます。

context

CMCI が CICSplex SM 環境にインストールされている場合、*context* は、要求に関連付けられる CICSplex または CMAS の名前 (例えば PLEX1) です。

CMCI が単一サーバーとしてインストールされている場合、*context* は、要求に関連付けられる CICS 領域のアプリケーション ID です。

コンテキストの値にスペースを含めることはできません。コンテキストは大/小文字が区別されません。

scope

要求に関連付けられた CICSplex、CICS グループ、CICS 領域、または論理的スコープの名前を指定します。スコープはコンテキストのサブセットで、これにより、要求を特定の CICS システムまたはリソースに限定します。スコープは必須ではありません。これが指定されない場合、要求はコンテキストの値によってのみ制限されます。スコープにスペースを含めることはできません。大/小文字の区別はありません。

OVERRIDEWARNINGCOUNT

警告カウント制限メカニズムをバイパスして、デフォルトの警告カウント制限が指定されていない場合と同様に要求を実行できるようにします。このオプションが有効になるのは、CICSplex SM Web ユーザー・インターフェースのサーバー 初期設定パラメーター **RESOURCELIMIT** が WARNING に設定されている場合だけです。

照会パラメーター

要求のスコープおよび特性を絞り込みます。照会セクションの各構成要素を任意の順序で配置できますが、1 つの URI 内でそれぞれ 1 回だけ使用できます。各照会パラメーターは、アンパーサンド (&) で分離されます。照会パラメーター値には大/小文字の区別はありませんが、特定の属性値では、大文字を適正に使用する必要があります。なぜなら、TRANID および DESC など、属性によっては大/小文字混合の値を保持できるためです。照会パラメーターは以下のとおりです。

CRITERIA=escaped_criteria_string

要求で返されるデータをフィルタリングする論理式のストリングです。CRITERIA パラメーターの値を構成するストリングは、CICSplex SM アプリケーション・プログラミング・インターフェースのフィルター式と同じ規則に従います。CICSplex SM API を使用してフィルター式を指定する方法については、[フィルター式の作成方法を参照してください](#)。

特定の制限付き文字については、エスケープ・シーケンスで置換して、サーバーで正しく解釈されるようにする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

PARAMETER=escaped_parameter_string

parameter_name(data_value) 形式の 1 つ以上のパラメーターおよび値からなるストリングで、要求を絞り込みます。これらのパラメーターを指定する規則は、CICSplex SM アプリケーション・プログラミング・インターフェースのものと 同じです。CICSplex SM API を使用してパラメーター式を指定する方法については、[リソースに対するアクションの実行を参照してください](#)。

基準ストリングと同様、特定の文字をエスケープ文字でエンコードする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

CMCI GET 要求数

CICS 管理クライアント・インターフェース (CMCI) では、HTTP GET メソッドを使用して、CICS または CICSplex SM からリソースを取得します。

クライアントは、以下の部分から GET 要求を形成します。

- HTTP メソッド (この場合は GET)
- URI。これにより、取得するリソースを指定し、結果を後で使用するために保管するか破棄するかを示します。
- HTTP バージョン

- 許可証明書 (必要な場合)

例えば、CICSplex PLEX1 内の CICS ローカル・ファイルすべてを取得する場合、クライアントは以下の要求を構成することができます。

```
GET /CICSSystemManagement/CICSLocalFile/PLEX1/ HTTP/1.1
Host: example.com:22958
Authorization: Basic RIJFRDpQQVNTVzBSRA==
```

HTTP ヘッダーの最初の要求行には、以下に示す 3 つの部分があります。各部はスペースで分離され、末尾は復帰、次いで改行 (LF) です。

- HTTP メソッド名
- 要求 URI。これは絶対パスで指定され、先頭が /CICSSystemManagement/ で、そのすぐ後にリソースを識別する外部名、およびそのリソースの取得対象となるインスタンスを特定するパラメーターおよびフィルターが続きます。

GET 要求は、CICS または CICSplex SM リソースに対して直接作用するか、または以前の GET 要求から保持された結果に対して作用することができます。通常、結果は各要求の最後に廃棄されますが、GET 要求 URI で NODISCARD オプションを指定することにより結果セットを保持できます。このオプションを使用した場合、CMCI は、すべての結果セットの状況を検査し、15 分以上未使用であった結果セットのみを廃棄します。

要求がリソースに対して直接作用している場合は、リソース名を指定し (操作リソースまたは定義リソースのいずれか)、コンテキストおよびオプションでスコープを追加した後、1 つ以上の照会パラメーターを含めることによって結果をさらに詳細化、ソート、およびフィルタリングします。1 つ以上の属性によって結果をソートするには、ORDERBY パラメーターを GET 要求 URI で指定します。

保持されている結果セットに対して要求が作用する場合は、リソース名を CICSResultCache で置換し、キャッシュ・トークンを指定することにより、結果を識別します。**index** オプションと **count** オプションの値を追加することにより、保持されている結果に含まれる 1 つ以上のレコードに要求を絞り込むことができます。

- HTTP バージョン。CICS 管理クライアント・インターフェースへの接続時は、常に HTTP/1.1 です。

2 行目では、ターゲット・システムのホスト名およびポート番号を、URI を指定するのと同様にコロンで分離して特定します。この行の最後は、復帰、続いて改行 (LF) となります。

オプションの 3 行目には、許可証明書を含めます。CICS システム初期設定パラメーター **SEC** が YES に指定されて稼働しているシステムの場合、ユーザー ID およびパスワードを Base64 方式で基本認証ヘッダーに設定する必要があります。この例では、ユーザー ID は *FRED*、パスワードは *PASSWORD* です。

それから、別の行に最後の復帰および改行が置かれて、HTTP ヘッダーが完結します。

GET 要求に対する応答

GET 要求が完了すると、クライアントは、CICS 管理クライアント・インターフェースからの応答を受信します。応答は、HTTP ヘッダー、結果サマリーを含む XML フィード、および取得されたリソースの詳細で構成されています。しかし、URI に SUMMONLY パラメーターが含まれるなら、応答は HTTP ヘッダーおよび結果サマリーのみで構成されます。

ヘッダーは、HTTP 応答、日時、ならびにサーバーおよびコンテンツ・タイプの詳細で構成されています。以下の例では、成功した GET 要求のヘッダーを示します。

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Wed, 06 Aug 2008 08:32:00 GMT
Server: IBM_CICS_Transaction_Server (zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

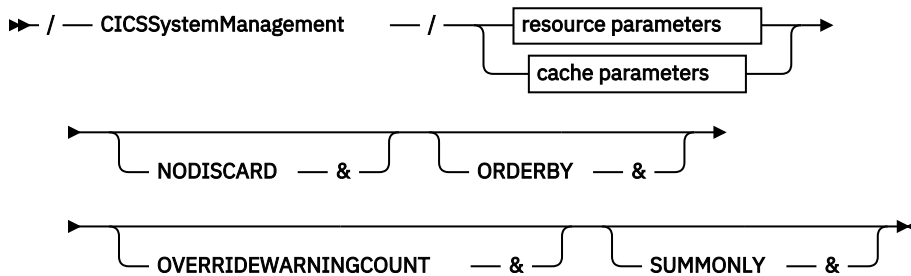
応答の本体は、要求に関する要約情報を表示する <resultsummary> エレメントが収容された <response> ルート・エレメント、および取得されたリソースの詳細が収容された <records> エレメント (通常は含まれる) で構成されます。URI で **SUMMONLY** パラメーターを指定すると、応答の <records> エレメントを省くことができます。要求がエラーで完了すると、診断情報が <errors> エレメントに格納されます。

CMCI GET 要求 URI

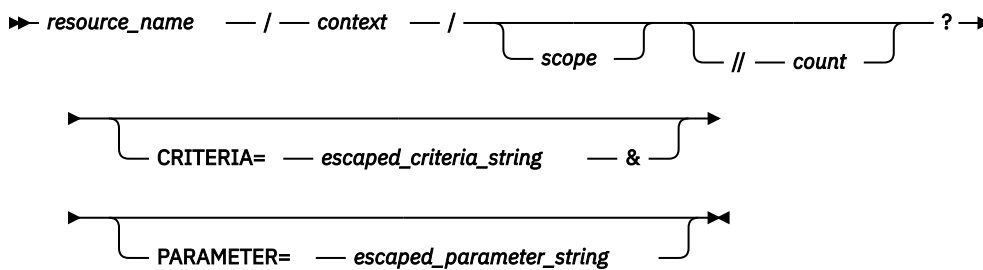
GET 要求の URI により、操作で取得されるリソースのインスタンスを特定します。

以下のダイアグラムでは、GET 要求の URI 構文を示します。

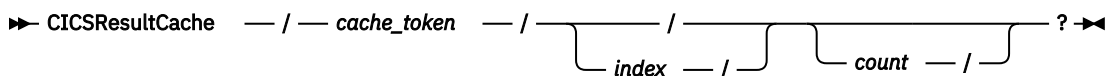
URI 構文



リソース・パラメーター:



キャッシュ・パラメーター:



オプション

/CICSSystemManagement

要求が CICS 管理クライアント・インターフェースにアクセスすることを示します。

CICSSystemManagement は大/小文字が区別されます。

resource_name

照会対象の CICS または CICSplex SM リソースに関連付けられる外部リソース名を指定します。例えば、リソース名 CICSRegion を指定すると、要求を CICSplex SM CICSRegion リソースに関連付け、CICSResourceAssignmentDefinition を指定すると、要求を RASGNDEF リソースに関連付けます。CICS 管理クライアント・インターフェース 要求における外部リソース名の完全なリストについては、[17 ページの『CMCI リソース名』](#)を参照してください。

HTTP GET メソッドは、すべてのリソースに対して有効であるわけではありません。特定のリソースにおいてこの HTTP メソッドが有効であるかどうかを判別するには、[CICSplex SM リソース・テーブル](#)にあるリソースの説明を参照してください。

リソース名は大/小文字が区別されません。正しくないリソース名がクライアントによって指定されると、HTTP 404 (Not Found) 応答が返されます。

context

CMCI が CICSplex SM 環境にインストールされている場合、context は、要求に関連付けられる CICSplex または CMAS の名前 (例えば PLEX1) です。CICSplex または CMAS のどちらを指定するかを判別するには、[CICSplex SM リソース・テーブル](#)にある関連リソース・テーブルを参照してください。

CMCI が単一サーバーとしてインストールされている場合、context は、要求に関連付けられる CICS 領域のアプリケーション ID です。

コンテキストの値にスペースを含めることはできません。コンテキストは大/小文字が区別されません。

scope

照会に関連付けられた CICSplex、CICS グループ、CICS 領域、または論理的スコープの名前を指定します。スコープはコンテキストのサブセットで、これにより、要求を特定の CICS システムまたはリソースに限定します。スコープは必須ではありません。スコープを省くと、要求はコンテキストのみの値で制限されます。スコープの値にスペースを含めることはできません。スコープは大/小文字が区別されません。

CICSResultCache/cache_token

保持された結果セットを格納する結果キャッシュを指定します。キャッシュ・トークンを指定すると、要求を、前の GET 要求から保持されたレコードに制限します。CMCI は各 CMCI 要求の完了時に、保存された結果セットの状況を確認します。15 分以上未使用の状態だった結果セットは、`cache_token` と合わせて廃棄されます。

index

保持された結果 (キャッシュ・トークン・パラメーターで指定) 内のレコードを特定します。各レコードは、保持された結果セット内での位置によって識別されます。**index** は、1 から 2 147 483 647 の範囲 (両端を含む) の整数でなければなりません。ゼロという値は許可されていません。**index** を指定しない場合、この位置にスラッシュを挿入する必要があります。

index オプションを指定することで、要求されるレコードの数を変更されることはなく、またそれが警告カウント・メカニズムに影響することはありません。

count

結果キャッシュ内の 1 番目のレコードまたは **index** パラメーターで指定されたレコードから始まる、結果キャッシュ内にあるレコードのサブセットを特定します。負の数値の場合、最終レコードから逆方向にカウントします。例えば、-1 は最終レコード、-2 は最後から 2 番目のレコードを表し、以降同様です。**count** は、-2,147,483,647 から 2,147,483,647 まで (両端を含む) の範囲の整数でなければなりません。ゼロの値は許可されません。リソース・パラメーターのケースでは、**count** の前に 2 本のスラッシュを付ける必要があります。

count オプションを指定することで、要求に対して表示されるレコードの数を制限することができますが、それによって要求されるレコードの数を変更されることはなく、またそれが警告カウント・メカニズムに影響することはありません。

照会パラメーター

要求のスコープおよび特性を絞り込みます。照会セクションの各構成要素を任意の順序で配置できますが、1 つの URI 内でそれぞれ 1 回だけ使用できます。各照会パラメーターは、アンパーサンド (&) で分離されます。照会パラメーター値には大/小文字の区別はありませんが、特定の属性値では、大文字を適正に使用する必要があります。なぜなら、TRANID および DESC など、属性によっては大/小文字混合の値を保持できるためです。照会パラメーターは以下のとおりです。

CRITERIA=escaped_criteria_string

要求で返されるデータをフィルタリングする論理式のストリングです。CRITERIA パラメーターの値を構成するストリングは、CICSplex SM アプリケーション・プログラミング・インターフェースのフィルター式と同じ規則に従います。CICSplex SM API を使用してフィルター式を指定する方法については、[フィルター式の作成方法を参照してください](#)。

特定の制限付き文字については、エスケープ・シーケンスで置換して、サーバーで正しく解釈されるようにする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

PARAMETER=escaped_parameter_string

parameter_name(data_value) 形式の 1 つ以上のパラメーターおよび値からなるストリングで、要求を絞り込みます。これらのパラメーターを指定する規則は、CICSplex SM アプリケーション・プログラミング・インターフェースのものと 同じです。CICSplex SM API を使用してパラメーター式を指定する方法については、[リソースに対するアクションの実行を参照してください](#)。

特定の文字をエスケープ文字でエンコードする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

NODISCARD

要求の後、結果セットが保持されることを示します。この属性は、値なしで指定します。NODISCARD を指定しない場合、結果セットは破棄されます。NODISCARD は、リソース名または保持されたキャッシュに作用する要求に対して使用できます。CMCI は各要求の完了時に、保存された結果セットの状況を確認し、15 分以上未使用であったものを廃棄します。

ORDERBY

指定したフィールドによって結果セットがソートされることを示します。この属性は論理式のストリングによって指定されます。**ORDERBY** パラメーターの値を構成するストリングは、CICSplex SM アプリケーション・プログラミング・インターフェースの場合と同じ規則に従います。EYUDA および CVDA 値の場合、フィールドは表示される文字値ではなく数値でソートされます。CICSplex SM API を使用してレコードをソートする方法については、[Sorting the records in a result set](#) を参照してください。

特定の文字をエスケープ文字でエンコードする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』](#)を参照してください。

このオプションを使用する各要求には、CICS によって自動的に管理される、L8 モード・オープン TCB および L9 モード・オープン TCB のプール内の L8 TCB が必要です。

OVERRIDEWARNINGCOUNT

警告カウント制限メカニズムをバイパスして、デフォルトの警告カウント制限が指定されていない場合と同様に要求を実行できるようにします。このオプションが有効になるのは、CICSplex SM Web ユーザー・インターフェースのサーバー 初期設定パラメーター **RESOURCELIMIT** が WARNING に設定されている場合だけです。

SUMMONLY

要求で結果サマリーのみが返されることを意味します。明細レコードは応答に含まれません。

SUMMONLY オプションを指定することで、要求に対してレコードは表示されなくなりますが、それによって要求されるレコードの数が増えることはなく、またそれが警告カウント・メカニズムに影響することはありません。**SUMMONLY** を含む要求は、要求されるレコードの数が警告カウント値を超えると失敗します。

保持した結果セットを GET 要求のパフォーマンスを改善するために使用する

HTTP GET 要求のパフォーマンスは、NODISCARD オプションを使用して、初期の要求が完了した後に結果セットをメモリー内に保持するようにサーバーに指示することにより改善できます。後続の要求は、CMAS を照会する代わりに、保持されている結果セットに対して実行でき、サーバーの負荷を削減できます。

このタスクについて

通常、CMCI ガーベッジ・コレクションはそれぞれの GET 要求の後で結果セットを廃棄します。ただし、NODISCARD オプションを使用すると、結果セットは保持されます。各要求の最後に、CMCI は、すべての結果セットの状況を検査し、15 分以上未使用であった結果セットのみを廃棄します。

保持されている結果セットを使用すると、1 つの大きな XML 応答文書の代わりに、複数の小さな応答文書で、クライアントが照会結果を要求できるようになります。小さな XML 文書で処理を行うと、大量の結果をレンダリング、送信、および解析する必要がなくなり、サーバー、ネットワーク、およびクライアントの負荷が削減されます。例えば、保持されている結果で処理を行うことで、クライアントは照会済みのリソースをページ単位で表示できます。各ページには、数千のレコードを含んでいる元の結果セットから 20 程度の結果だけが表示されます。ただし、保持されている結果セットのスナップショット的性質により、急速に変化する照会データ、または可能な限り最新の状況をクライアントが取得する必要がある場合には適しません。

サーバー上でセキュリティが有効になっている場合は、自分が作成した保持されている結果セットにのみアクセスできます。他のユーザーが作成した保持されている結果セットにアクセスを試みると、HTTP 403 エラーになります。

保持された結果セットを処理するには、以下のようにします。

手順

1. 最初の HTTP GET 要求を発行して、要求 URI に NODISCARD オプションおよび SUMMONLY オプションを指定します。

CICS 管理クライアント・インターフェースは、要求が成功したと仮定して、照会結果を GCDSA 動的ストレージ域に格納し、後続の要求で保持されている結果セットにアクセスするために使用できるトークンをクライアントに返します。トークンは、GET 要求応答の <resultsummary> エlement に **cachetoken** 属性の値として返されます。SUMMONLY を指定すると、サーバーからの応答の本体にレコードが含まれなくなります。

2. 2 番目の GET 要求を発行します。この中で、外部リソース名の代わりに CICSResultCache を指定し、続いて、最初の要求に対する応答で返された **cachetoken** 属性の値を指定します。

認識されない、または削除されたキャッシュ・トークンを指定した場合、リソースが検出されない状態として処理され、サーバーは HTTP 404 エラーを返します。

要求は、**index**、**count** および **orderby** オプションを指定して、さらに詳細化できます。**index** は、結果セットの中から 1 つのレコードを、保持された結果セット内の位置で指定します。**count** は、保持されている結果セットの最初のレコードから、または **index** で特定されるレコードの位置からレコードのサブセットを特定します。**orderby** は、どのフィールド (複数可) によって結果セットをソートするかを指定します。**orderby** オプションに指定できるフィールドの最大数は 32 です。

この要求に続く要求では、NODISCARD オプションを指定することで、結果セットを継続して使用できます。NODISCARD を指定しないと、結果セットはサーバーによって要求の最後に削除されます。

クライアントは、単一のキャッシュ・トークンに対する複数の同時要求をサーバーに発行することもできます。例えば、マルチスレッドのクライアント・アプリケーションは、レコード 1 から 10、11 から 20、および 21 から 30 の情報を同時に取得できます。ただし、1 つ以上の同時要求で NODISCARD オプションを指定しないと、予測不能な結果が生じます。

例

システム内にインストールされたすべてのプログラムを表示するために GET 要求を使用すると、何千ものレコードがクライアントに返される場合がありますが、保持された結果セットを使用して結果を迅速にページングすることができます。

以下の要求を使用すると、システムにインストールされたすべての PROGRAM リソースが識別されますが、SUMMONLY オプションが指定されているので、レコードは返されず、識別されたリソースの要約だけが返されます。NODISCARD オプションによって、この結果セットを識別するためのキャッシュ・トークンが返されます。

```
http://hostname.example.com:12345/CICSSystemManagement/CICSProgram/CICSPLEX?NODISCARD&SUMMONLY
```

サーバーは次の応答を返します。

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
connect_version="0410" version="1.0" xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int
http://hostname.mycompany.com:12345/CICSSystemManagement/schema/CICSSystemManagement.xsd">
<resultsummary api_response1="1024" api_response1_alt="0K" api_response2="0"
api_response2_alt="" cachetoken="C3D526A2F224FFE1" recordcount="1662"/>
</response>
```

照会に成功したことは、api_response1 および api_response2 属性がそれぞれ 1024 および 0 であることを確認することによって検証できます。

<resultsummary> Element は、cachetoken 値が C3D526A2F224FFE1 である結果セットを示しています。recordcount 属性は、結果セットに合計 1662 個のレコードがあることを示しています。

この保持された結果セット内の最初の 10 個のレコードを調べるには、次の要求を発行できます。

```
http://hostname.example.com:12345/CICSSystemManagement/CICSResultCache/C3D526A2F224FFE1/1/10?NODISCARD
```

サーバーは次の応答を返します。


```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
connect_version="0410" version="1.0" xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int
http://hostname.mycompany.com:12345/CICSSystemManagement/schema/CICSSystemManagement.xsd">
<resultsummary api_response1="1024" api_response1_alt="OK" api_response2="0" api_response2_alt=""
cachetoken="C3D526A2F224FFE1" displayed_recordcount="10" recordcount="1662"/>
<records>
  <cicsprogram _keydata="C3C5C5C3C2D3C4E8" aloadtime="00:00:00.000" ... usecount="0" usefetch="0.000" />
  <cicsprogram _keydata="C3C5C5C3C3C9C3E2" aloadtime="00:00:00.000" ... usecount="0" usefetch="0.000" />
  <cicsprogram _keydata="C3C5C5C3D4C94040" aloadtime="00:00:00.000" ... usecount="0" usefetch="0.000" />
  ...
</records>
</response>
```

displayed_recordcount 属性は、<records> エlement に 10 個の <cicsprogram> 子Element があることを示しています。cachetoken 属性は、将来使用するために結果セットがサーバーに保持されていることを示しています。

この後クライアントは、次の要求を発行することにより、レコード 11 から始まる次の 10 個のレコードを取得できます。

```
http://hostname.example.com:12345/CICSSystemManagement/CICSResultCache/C3D526A2F224FFE1/11/10?NODISCARD
```

保持された結果セットの確認が終わったことをサーバーに知らせるには、NODISCARD オプションを指定せずに結果セットにアクセスします。次の要求では再度 SUMMONLY パラメーターを使用し、サーバーがこの要求でレコードを返さないようにしています。

```
http://hostname.example.com:12345/CICSSystemManagement/CICSResultCache/C3D526A2F224FFE1?SUMMONLY
```

サーバーは次の応答を返します。

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
connect_version="0410" version="1.0" xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int
http://hostname.mycompany.com:12345/CICSSystemManagement/schema/CICSSystemManagement.xsd">
<resultsummary api_response1="1024" api_response1_alt="OK" api_response2="0"
api_response2_alt="" recordcount="1662"/>
</response>
```

<resultsummary> Element に cachetoken 属性がないということは、結果セットがもはやサーバーに格納されていないことを示しています。保持された結果セットを使用するためにさらに要求を発行しようとすると、サーバーは HTTP 404 Not Found エラーを返します。

CMCI POST 要求数

CICS 管理クライアント・インターフェース (CMCI) では、HTTP POST メソッドを使用して、CICS または CICSplex SM 内にリソースを作成します。

クライアントは、以下の部分から POST 要求を形成します。

- HTTP メソッド (この場合は POST)
- URI。これにより、取得するリソースを指定し、結果を後で使用するためにキャッシュするか破棄するかを示します。
- HTTP バージョン
- 許可証明書 (必要な場合)
- 作成するリソースの詳細を収容する XML 本体

例:

```
POST /CICSSystemManagement/CICSLocalFile/PLEX1/ HTTP/1.1
Host: example.com:22958
Authorization: Basic R1JFRDpQQVNTVzBSRA==
```

```
<request>
<create>
```

```
<parameter name="RESGROUP" value="BASIC" />
<attributes name="FILE1" defver="1" />
</create>
</request>
```

最初の行は次の 3 つの部分で構成されており、それぞれスペースで区切られています。

- HTTP メソッド名
- 要求 URI。これは絶対パスで指定され、先頭が /CICSSystemManagement/ で、そのすぐ後にリソースを識別する外部名、および作成されるリソースのインスタンスを特定するパラメーターおよびフィルターが続きます。
- HTTP バージョン。CICS システム 管理クライアント API への接続時は、常に HTTP/1.1 です。

2 行目は、ターゲット・システムのホスト名とポート番号を、URI を指定するのと同様にコロンで分離して指定します。この行の最後は、復帰、続いて改行 (LF) となります。

オプションの 3 行目には、許可証明書を含めます。CICS システム 初期設定パラメーター **SEC** が YES に指定されて稼働しているシステムの場合、ユーザー ID およびパスワードを Base64 方式で基本認証ヘッダーに設定する必要があります。この例では、ユーザー ID は *FRED*、パスワードは *PASSWORD* です。

それから、別の行に最後の復帰および改行が置かれて、HTTP ヘッダーが完結します。

要求の残りの部分は、新規リソースの属性を指定する XML 本体です。この本体は、1 つの <attributes> 子要素が入っている XML <request> エレメントで構成されます。

POST 要求に対する応答

POST 要求が完了すると、クライアントは、クライアント API からの応答を受信します。応答は、HTTP ヘッダー、結果サマリーを含む XML フィード、および作成されたリソースの詳細で構成されています。しかし、要求に SUMMONLY パラメーターが含まれるなら、応答は HTTP ヘッダーおよび結果サマリーのみで構成されます。

ヘッダーは、HTTP 応答、日時、ならびにサーバーおよびコンテンツ・タイプの詳細で構成されています。以下の例では、成功した要求のヘッダーを示します。

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Wed, 06 Aug 2008 08:32:00 GMT
Server: IBM_CICS_Transaction_Server (zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

応答の本体は、要求に関する要約情報を表示する <resultsummary> エレメントが収容された <response> ルート・エレメント、および新規リソースの詳細が収容された <records> エレメント (通常は含まれる) で構成されます。しかし、URI で SUMMONLY パラメーターを指定すると、応答の <records> エレメントを省くことができます。要求がエラーで完了すると、診断情報が <errors> エレメントに格納されます。

セキュリティの考慮事項

HTTP メソッドを使用する場合、一部のファイアウォールで、セキュリティ上の考慮事項が原因でファイアウォール経由の HTTP PUT トラフィックまたは DELETE トラフィックが許可されないことに注意してください。この制限に対応するために、X-HTTP-Method-Override HTTP ヘッダー・フィールドを使用して、PUT 要求または DELETE 要求を POST 要求によってトンネル伝送することができます。例えば、次のようになります。

```
X-HTTP-Method-Override: PUT
```

CMCI POST 要求 URI

POST 要求の URI により、操作で作成されるリソースを特定します。

以下のダイアグラムでは、POST 要求の URI 構文を示します。

➡ / — CICSSystemManagement — / — resource_name — / — context — / — ? ➡



POST 要求では、基準ストリングもパラメーター・ストリングも指定しません。作成するリソースに関する情報はすべて、要求の本体に組み込みます。

オプション

/CICSSystemManagement

要求が CICS 管理クライアント・インターフェースにアクセスすることを示します。
CICSSystemManagement は大/小文字が区別されます。

resource_name

作成する CICS または CICSplex SM リソースに関連付けられる外部リソース名を指定します。例えば、リソース名 CICSRegion を指定すると、要求を CICSplex SM CICSRegion リソース・テーブルに関連付け、CICSResourceAssignmentDefinition を指定すると、要求を RASGNDEF リソース・テーブルに関連付けます。外部リソース名の完全なリストについては、[17 ページの『CMCI リソース名』](#)を参照してください。

HTTP POST メソッドは、すべてのリソースに対して有効であるわけではありません。特定のリソースにおいてこの HTTP メソッドが有効であるかどうかを判別するには、[CICSplex SM リソース・テーブル](#)にあるリソースの説明を参照してください。

リソース名は大/小文字が区別されません。正しくないリソース名がクライアントによって指定されると、HTTP 404 (Not Found) 応答が返されます。

コンテキスト

要求に関連付けられた CICSplex または CMAS の名前を指定します。例えば、PLEX1 とします。このパラメーターは、CICSplex SM とやり取りする URI で必要です。コンテキストにスペースを含めることはできません。大/小文字の区別はありません。

SUMMONLY

要求で結果サマリーのみが返されることを意味します。応答に詳細レコードは含まれません。

CMCI PUT 要求数

CICS 管理クライアント・インターフェース (CMCI) では、HTTP PUT メソッドを使用して、CICS または CICSplex SM リソースの属性を変更したり、これらのリソースに対するアクション (INSTALL など) を実行したりします。

クライアントは、以下の構成要素から PUT 要求を形成します。

- HTTP メソッド (この場合は PUT)
- 変更対象のリソースを特定する URI
- HTTP バージョン
- 許可証明書 (必要な場合)
- 変更内容または実行するアクションを XML で指定した HTTP 本体

例えば、TR で始まる名前および P で始まるプログラムが含まれる、CICSplex PLEX1 内のトランザクション定義の属性を更新する場合、クライアントは以下の要求を作成します。

```
PUT /CICSSystemManagement/CICSDefinitionTransaction/PLEX1?CRITERIA=NAME%3DTR%2A%20
AND%20PROGRAM%3DP%2A
HTTP/1.1
Host: example.com:23792
Authorization: Basic R1JFRDpQQVNTVzBSRA==

<request>
  <update>
    <attributes STATUS="ENABLED" />
```

```
</update>  
</request>
```

最初の行には、スペースで分離された以下の 3 つの部分があります。

- メソッド名
- 要求 URI。これは絶対パスで指定され、先頭が `/CICSSystemManagement/` で、そのすぐ後にリソースを識別する外部名、ならびにそのリソース・タイプの選択対象となるインスタンスを特定するパラメータおよびフィルターが続きます。URI を CICS 管理クライアント・インターフェースによる処理に適したものとするために、クライアントはスペースやアスタリスクなどの特定の制限付き文字をエスケープ・シーケンスに置換します。
- HTTP バージョン。CICS 管理クライアント・インターフェースへの接続時は、常に HTTP/1.1 です。

2 行目では、ターゲット・システムのホスト名およびポート番号を、URI を指定するのと同様にコロンので分離して特定します。この行の最後は、復帰、続いて改行 (LF) となります。

オプションの 3 行目には、許可証明書を含めます。CICS システム初期設定パラメーター **SEC** が YES に指定されて稼働しているシステムの場合、ユーザー ID およびパスワードを Base64 方式で基本認証ヘッダーに設定する必要があります。この例では、ユーザー ID は *FRED*、パスワードは *PASSWORD* です。

それから、別の行に最後の復帰および改行が置かれて、HTTP ヘッダーが完結します。

要求の残りの部分は、特定したリソース・インスタンスに加える変更内容を指定する XML 本体です。このケースでは、要求により、選択したトランザクション定義の STATUS 属性を **ENABLED** に変更します。この本体は、1 つの `<attributes>` エレメントまたは 1 つの `<action>` エレメントを収容する、XML `<request>` エレメントで構成されています。

PUT 要求は、インストール済みリソース、定義 CICS リソース、または定義 CICSplex SM リソースのいずれかに対して作用します。単一の PUT 要求により、属性の変更を指定するか、あるいはアクションを実行することができます。単一の要求に、属性の変更とアクションを組み合わせることはできません。

PUT メソッドで、すべてのアクションがサポートされるわけではありません。次のアクションが POST メソッドでサポートされています。

- 定義リソースの **CREATE**。
- **SYSDUMP** および **TRANDUMP** リソース・タイプの **ADD**。

次のアクションが **DELETE** メソッドでサポートされています。

- 操作リソースの **DISCARD**。
- 定義リソースの **REMOVE**。

PUT 要求に対する応答

クライアントは PUT 要求の完了時に、CICS 管理クライアント・インターフェースからの応答を受信します。この応答は、HTTP ヘッダー、結果の要約が入った XML フィード、および変更されたリソースの詳細 (要求が成功した場合) で構成されます。

ヘッダーは、HTTP 応答、日時、ならびにサーバーおよびコンテンツ・タイプの詳細で構成されています。例えば次のようになります。

```
HTTP/1.1 200 OK  
Cache-Control: no-store  
Date: Wed, 10 Aug 2008 12:56:00 GMT  
Server: IBM_CICS_Transaction_Server (zOS)  
Content-Type: application/xml; charset=UTF-8  
Transfer-Encoding: chunked
```

PUT 要求が失敗の場合、サーバーは、OK ではない HTTP 応答を発行します。例えば、要求の本体が有効ではない場合、以下の 400 応答がヘッダーに組み込まれます。

```
400 Bad request. The body of the request is invalid.
```

この応答が発生するのは、クライアントが 1 つの PUT 要求で多数のタグを送信したり、要求の本体に不明なタグが含まれていたり、あるいは URI 内のモデル名とタグ名が合致しないような場合です。

応答の本体は、要求に関する要約情報を表示する <resultsummary> エLEMENTが収容された <response> ルート・ELEMENT、および選択されたリソースの詳細が収容された <records> ELEMENT (通常は含まれる) で構成されます。しかし、URI で **SUMMONLY** パラメーターを指定すると、応答の <records> ELEMENT を省くことができます。要求がエラーで完了すると、診断情報が <errors> ELEMENT に格納されます。

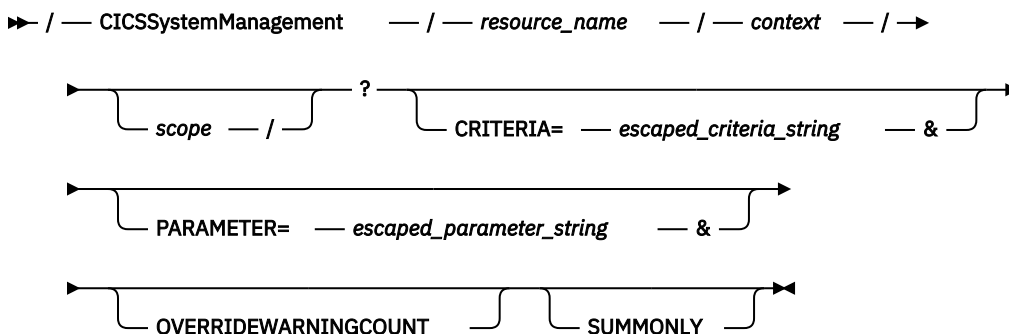
セキュリティの考慮事項

HTTP メソッドを使用する場合、一部のファイアウォールで、セキュリティ上の考慮事項が原因でファイアウォール経由の HTTP PUT トラフィックまたは DELETE トラフィックが許可されないことに注意してください。この制限に対応するために、CMCI POST 要求を使用して、PUT 要求を POST 要求によってトンネル伝送することができます。

CMCI PUT 要求 URI

PUT 要求の URI により、操作の対象となるリソースのインスタンスを特定します。

以下のダイアグラムでは、PUT 要求 URI の構文を示します。



オプション

/CICSSystemManagement

要求が CICS 管理クライアント・インターフェースにアクセスすることを示します。
CICSSystemManagement は大/小文字が区別されます。

resource_name

更新する CICS または CICSplex SM リソースに関連付けられる外部リソース名を指定します。例えば、リソース名 CICSRegion を指定すると、要求を CICSplex SM CICSRRGN リソース・テーブルに関連付け、CICSResourceAssignmentDefinition を指定すると、要求を RASGNDEF リソース・テーブルに関連付けます。外部リソース名の完全なリストについては、17 ページの『CMCI リソース名』を参照してください。

HTTP PUT メソッドは、すべてのリソースに対して有効であるわけではありません。特定のリソースにおいてこの HTTP メソッドが有効であるかどうかを判別するには、CICSplex SM リソース・テーブルにあるリソースの説明を参照してください。

リソース名は大/小文字が区別されません。正しくないリソース名がクライアントによって指定されると、HTTP 404 (Not Found) 応答が返されます。

context

CMCI が CICSplex SM 環境にインストールされている場合、context は、要求に関連付けられる CICSplex または CMAS の名前 (例えば PLEX1) です。

CMCI が単一サーバーとしてインストールされている場合、context は、要求に関連付けられる CICS 領域のアプリケーション ID です。

コンテキストの値にスペースを含めることはできません。大/小文字の区別はありません。

scope

要求に関連付けられた CICSplex、CICS グループ、CICS 領域、または論理的スコープの名前を指定します。スコープはコンテキストのサブセットで、これにより、要求を特定の CICS システムまたはリソースに限定します。スコープは必須ではありません。これが指定されない場合、要求はコンテキストの

値によってのみ制限されます。スコープにスペースを含めることはできません。大/小文字の区別はありません。

照会パラメーター

要求のスコープおよび特性を絞り込みます。照会セクションの各構成要素を任意の順序で配置できますが、1つの URI 内でそれぞれ 1 回だけ使用できます。各照会パラメーターは、アンパーサンド (&) で分離されます。照会パラメーター値には大/小文字の区別はありませんが、特定の属性値では、大文字を適正に使用する必要があります。なぜなら、TRANID および DESC など、属性によっては大/小文字混合の値を保持できるためです。照会パラメーターは以下のとおりです。

CRITERIA=escaped_criteria_string

要求で返されるデータをフィルタリングする論理式のストリングです。CRITERIA パラメーターの値を構成するストリングは、CICSplex SM アプリケーション・プログラミング・インターフェースのフィルター式と同じ規則に従います。CICSplex SM API を使用してフィルター式を指定する方法については、[フィルター式の作成方法を参照してください](#)。

特定の制限付き文字については、エスケープ・シーケンスで置換して、サーバーで正しく解釈されるようにする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

PARAMETER=escaped_parameter_string

parameter_name(data_value) 形式の 1 つ以上のパラメーターおよび値からなるストリングで、要求を絞り込みます。これらのパラメーターを指定する規則は、CICSplex SM アプリケーション・プログラミング・インターフェースのものと 同じです。CICSplex SM API を使用してパラメーター式を指定する方法については、[リソースに対するアクションの実行を参照してください](#)。

基準ストリングと同様、特定の文字をエスケープ文字でエンコードする必要があります。CICS 管理クライアント・インターフェース URI でのエスケープ・シーケンスの使用については、[41 ページの『CMCI エスケープ・シーケンス』を参照してください](#)。

OVERRIDEWARNINGCOUNT

警告カウント制限メカニズムをバイパスして、デフォルトの警告カウント制限が指定されていない場合と同様に要求を実行できるようにします。このオプションが有効になるのは、CICSplex SM Web ユーザー・インターフェースのサーバー 初期設定パラメーター **RESOURCELIMIT** が WARNING に設定されている場合だけです。

SUMMONLY

要求で結果サマリーのみが返されることを意味します。応答に詳細レコードは含まれません。

SUMMONLY オプションを指定することで、要求に対してレコードは表示されなくなりますが、それによって要求されるレコードの数が増えることはなく、またそれが警告カウント・メカニズムに影響することはありません。**SUMMONLY** を含む要求は、要求されるレコードの数が警告カウント値を超えると失敗します。

CMCI リソース名

CICS 管理クライアント・インターフェース (CMCI) では、CICSplex SM および CICS リソースにマッピングされる外部リソース名を使用します。

以下の表に、CMCI 外部リソース名と、それらに関連付けられた CICSplex SM リソース名をリストします。リソースを、CICS 運用リソース、CICS 定義リソース、CICSplex SM 定義リソース、および CICSplex SM 非定義リソースに分類しています。

各リソースで有効な CMCI メソッドなどの CICSplex SM リソースについては、[CICSplex SM リソース・テーブル](#)を参照してください。

CICS 運用リソース

表 1. CICS 運用リソースのリソース名		
外部リソース名	CICSplex SM リソース名	説明
CICSAllTaskSubpools	TSKSPOLS	CICS タスク・ストレージ・サブプール
CICSAtomService	ATOMSERV	CICS Atom サービス、フィード、コレクション、またはカテゴリー文書
CICSAutoInstallModel	AIMODEL	CICS 自動インストール端末モデル
CICSBridgeFacility	BRFACIL	3270 ブリッジ機構で使用される仮想端末 (ブリッジ機能)
CICSBundle	BUNDLE	CICS バンドル
CICSBundlePart	BUNDPART	CICS バンドルの一部
CICSCaptureSpecification	EVCSPEC	CICS イベント・キャプチャー仕様
CICSCaptureSpecificationDataPredicate	EVCSDATA	CICS イベント・キャプチャー仕様のデータ述部
CICSCaptureSpecificationInformationSource	EVCSINFO	CICS イベント・キャプチャー仕様の情報源
CICSCaptureSpecificationOptionPredicate	EVCSOPT	CICS イベント・キャプチャー仕様のコマンド・オプション述部
CICSCFDTPool	CFDTPOOL	カップリング・ファシリティー・データ・テーブル・プール
CICSDataSetName	DSNAME	アクティブな CICS システムで使用中のデータ・セット
CICSDataTable	CMDT	CICS データ・テーブルまたはユーザー保守データ・テーブル、あるいはそれらに関連付けられたカップリング・ファシリティー・データ・テーブルを含むファイル
CICSDB2Connection	DB2CONN	Db2® 接続
CICSDB2Entry	DB2ENTRY	Db2 エントリー
CICSDB2Transaction	DB2TRN	Db2 トランザクション

表 1. CICS 運用リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSDBCTLSubsystem	DBCTLSS	アクティブな CICS システムから DBCTL サブシステムへの接続
CICSDFHRPLDataSet	RPLLIST	CICS DFHRPL 連結シーケンス内のデータ・セット
CICSDocumentTemplate	DOCTEMP	文書テンプレート
CICSDomainSubpool	DOMSPOOL	CICS ストレージ・ドメイン・サブプール
CICSDynamicStorageArea	CICSDSA	動的ストレージ域
CICSEnqueue	ENQUEUE	CICS エンキュー
CICSEnqueueModel	ENQMODEL	ENQ/DEQ モデル
CICSEPADapter	EPADAPT	CICS イベント処理アダプター
CICSEPADapterSet	EPADSET	CICS イベント処理アダプター・セット
CICSEPADapterInSet	EPAINSET	イベント処理アダプター・セットに含まれる CICS イベント処理アダプター
CICSEventBinding	EVNTBIND	CICS イベント・バインディング
CICSEventProcessing	EVNTGBL	CICS イベント処理
CICSEXCIRequest	EXCI	外部 CICS インターフェース API を使用するクライアント・プログラムから発生するタスク
CICSExtrapartitionTDQueue	EXTRATDQ	区画外一時データ・キュー
CICSFEPIConnection	FEPICONN	FEPI 接続
CICSFEPINode	FEPINODE	FEPI ノード
CICSFEPIPool	FEPPOOL	FEPI プール
CICSFEPIPropertySet	FEPIPROP	FEPI プロパティ・セット
CICSFEPITarget	FEPITRGT	FEPI ターゲット
CICSGlobalDispatcher	DSPGBL	CICS システムでのグローバル CICS ディスパッチャー情報
CICSGlobalDynamicStorageArea	CICSSTOR	CICS 動的ストレージ域
CICSGlobalMVSTCBStatistics	MVSTCBGL	CICS アドレス・スペース内のグローバル MVS™ TCB
CICSGlobalTDQueueStatistics	TDQGBL	区画内一時データ・キューの使用量
CICSGlobalTSQueueStatistics	TSQGBL	一時記憶域キューの使用量
CICSGlobalURIMapStatistics	URIMPGBL	URIMAP リソース用に CICS 抽出の統計から戻されるグローバル統計
CICSGlobalUserExit	EXITGLUE	インストールされた CICS TS グローバル・ユーザー出口
CICSIndirectTDQueue	INDTDQ	間接一時データ・キュー

表 1. CICS 運用リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSIntervalControlRequest	REQID	未解決のインターバル制御要求
CICSIntrapartitionTDQueue	INTRATDQ	区画内一時データ・キュー
CICSIPFacility	IPFACIL	アクティブな CICS タスクと、それらのタスクが使用中の IPIC 接続との関連
CICSIPIConnection	IPCONN	IP 相互通信接続
CICSIscMROConnection	CONNECT	ISC または MRO 接続
CICSJournalModel	JRNLMODL	インストールされたジャーナル・モデルおよび対応するログ・ストリーム名
CICSJournalName	JRNLMODL	システム・ログおよび一般ログについての情報
CICSJVM	JVM	Java™ 仮想マシン
CICSJVMClassCache	CLCACHE	共用クラス・キャッシュ
CICSJVMPool	JVMPPOOL	JVM プール
CICSJVMProfile	JVMPROF	JVM プロファイル
CICSJVMServer	JVMSERV	インストール済み JVM サーバー
CICSLIBRARY	LIBRARY	LIBRARY
CICSLIBRARYDataSetName	LIBDSN	LIBRARY データ・セット名
CICSLoader	LOADER	CICS ロダー情報
CICSLoaderByDSA	LOADACT	動的ストレージ域による CICS ロダー・アクティビティ
CICSLocalFile	LOCFILE	CICS ローカル・ファイル
CICSLocalTransaction	LOCTRAN	CICS ローカル・トランザクション
CICSLSRPool	LSRPOOL	ローカル共用リソース・プール
CICSLSRPoolBuffer	LSRPBUF	VSAM LSR プール・バッファー
CICSLU62ModeName	MODENAME	LU 6.2 モード名
CICSMonitoringAndStatistics	MONITOR	CICS モニターおよび統計情報
CICSMQConnection	MQCON	IBM MQ 接続
CICSMQConnectionStatistics	MQCONN	IBM MQ 接続統計
CICSMQInitiationQueue	MQINI	IBM MQ 開始キュー
CICSMVSLogStream	STREAMNM	MVS ログ・ストリーム名
CICSMVStorageArea	MVSESTG	MVS ストレージ域
CICSMVSTCB	MVSTCB	MVS TCB
CICSMVSWLM	MVSWLM	MVS ワークロード管理コンポーネント情報
CICSOSGIBundle	OSGIBUND	OSGi バンドル

表 1. CICS 運用リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSOSGIService	OSGISERV	OSGi サービス
CICSPagePool	CICSPAGP	CICS ページ (DSA) プール
CICSPartner	PARTNER	CPI-C パートナー・テーブル
CICSPipeline	PIPELINE	CICS Web サービス処理ノード
CICSProcessType	PROCTYP	BTS プロセス・タイプ
CICSProfile	PROFILE	CICS プロファイル
CICSProgram	PROGRAM	CICS プログラム
CICSRecoveryManager	RECOVERY	リカバリー・マネージャー情報
CICSRegion	CICSRGN	CICS 領域
CICSRemoteFile	REMFIL	リモート CICS ファイル
CICSRemoteTDQueue	REMTDQ	リモート一時データ・キュー
CICSRemoteTransaction	REMTRAN	リモート・トランザクション
CICSSharedTSQueue	TSQSHR	一時記憶域キュー
CICSSystemDump	SYSDUMP	CICS システム・ダンプ・コード
CICSSystemParameter	SYSPARM	CICS システム・パラメーター
CICSTask	TASK	CICS タスク
CICSTaskAssociation	TASKASSC	タスク関連データ
CICSTaskFileUsage	TASKFILE	タスクと、タスクで使った CICS ファイル
CICSTaskRelatedExit	EXITTRUE	タスク関連のユーザー出口
CICSTaskRMIUsage	TASKRMI	タスクごとの CICS リソース・マネージャー・インターフェース 使用量
CICSTaskStorage	TASKSTG	タスクの CICS ストレージ・エレメント
CICSTaskSubpool	TSKSPOL	CICS タスク・ストレージ・サブプール
CICSTaskTSQueueUsage	TASKTSQ	タスクと、タスクで使った CICS 一時記憶域キュー
CICSTCBMode	DSPMODE	CICS ディスパッチャー TCB モード
CICSTCBPool	DSPPOOL	CICS ディスパッチャー TCB プール
CICSTCIPService	TCPIPS	TCP/IP サービス
CICSTCIPStatistics	TCPIPGBL	CICS 内部 TCP/IP ソケット・サポート
CICSTerminal	TERMNL	CICS 端末
CICSTransactionClass	TRANCLAS	CICS トランザクション・クラス
CICSTransactionDump	TRANDUMP	CICS トランザクション・ダンプ・コード

表 1. CICS 運用リソースのリソース名 (続き)

外部リソース名	CICSplex SM リソース名	説明
CICSTSMModel	TSMODEL	CICS 一時記憶域モデル
CICSTSPool	TSPOOL	CICS 一時記憶域プール
CICSTSQueue	TSQNAME	CICS 非共用一時記憶域キュー
CICSUOW	UOW	CICS 作業単位
CICSUOWEnqueue	UOWENQ	作業単位で保留されたエンキュー
CICSUOWLink	UOWLINK	作業単位と CICS システム間のリンク
CICSUOWShuntedAndHoldingLocks	UOWDSNF	中断された作業単位
CICSURIHost	HOST	ローカル・システム内の仮想ホスト
CICSURIMap	URIMAP	URI (Universal Resource Identifier) 要求
CICSWebService	WEBSERV	Web サービス環境にデプロイされた CICS アプリケーション・プログラムの実行時環境に関する情報
CICSWorkRequest	WORKREQ	EJB 作業要求
CICSXmltransform	XMLTRANS	XML 変換

CICS 定義リソース

表 2. CICS 定義リソースのリソース名

外部リソース名	CICSplex SM リソース名	説明
CICSCSDGroup	CSDGROUP	CSD グループ定義
CICSCSDGroupInList	CSDINLST	リスト内の CSD グループ
CICSCSDList	CSDLIST	CSD リスト定義
CICSCSDResource	CSDINGRP	グループ内の CSD CICS リソース
CICSDefinitionAtomService	ATOMDEF	Atom サービス定義
CICSDefinitionBundle	BUNDDEF	バンドル定義
CICSDefinitionCorbaServer	EJCODEF	CorbaServer 定義
CICSDefinitionDb2Connection	DB2CDEF	Db2 接続定義
CICSDefinitionDb2Entry	DB2EDEF	Db2 エントリーの定義
CICSDefinitionDb2Transaction	DB2TDEF	Db2 トランザクションの定義
CICSDefinitionDeployedJARFile	EJDJDEF	CICS 配置 JAR ファイルの定義
CICSDefinitionDocumentTemplate	DOCDEF	文書テンプレートの定義
CICSDefinitionEnqueueModel	ENQMDEF	グローバル・エンキューの定義
CICSDefinitionFEPINode	FENODDEF	FEPI ノードの定義
CICSDefinitionFEPIPool	FEPOODEF	FEPI プールの定義

表 2. CICS 定義リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSDefinitionFEPIPropertySet	FEPRODEF	FEPI プロパティ・セットの定義
CICSDefinitionFEPITarget	FETRDEF	FEPI ターゲットの定義
CICSDefinitionFile	FILEDEF	CICS ファイル定義
CICSDefinitionIPICConnection	IPCONDEF	IPIC 接続定義
CICSDefinitionISCMROConnection	CONNDEF	ISC/MRO 接続定義
CICSDefinitionJournalModel	JRNMDEF	ジャーナル・モデルの定義
CICSDefinitionLIBRARY	LIBDEF	LIBRARY 定義
CICSDefinitionLSRPool	LSRPOOL	ローカル共用リソース・プールの定義
CICSDefinitionMapSet	MAPDEF	マップ・セットの定義
CICSDefinitionMQConnection	MQCONDEF	IBM MQ 接続定義
CICSDefinitionPartitionSet	PRTNDEF	区分セット定義
CICSDefinitionPartner	PARTDEF	パートナー定義
CICSDefinitionPipeline	PIPEDEF	パイプライン定義
CICSDefinitionProcessType	PROCDEF	プロセス・タイプの定義
CICSDefinitionProfile	PROFDEF	プロファイル定義
CICSDefinitionProgram	PROGDEF	プログラム定義
CICSDefinitionRequestModel	RQMDEF	要求モデルの定義
CICSDefinitionSession	SESSDEF	セッション定義
CICSDefinitionTCPIPService	TCPDEF	TCP/IP サービス定義
CICSDefinitionTDQueue	TDQDEF	一時データ・キューの定義
CICSDefinitionTerminal	TERMDEF	端末の定義
CICSDefinitionTransaction	TRANDEF	トランザクション定義
CICSDefinitionTransactionClass	TRNCLDEF	トランザクション・クラス定義
CICSDefinitionTSMModel	TSMDEF	一時記憶域モデルの定義
CICSDefinitionTypeterm	TYPTMDEF	TYPETERM 定義
CICSDefinitionURIMap	URIMPDEF	URI マッピング定義
CICSDefinitionWebService	WEBSVDEF	Web サービス定義

CICSplex SM 定義リソース

表 3. CICSplex SM 定義リソースのリソース名		
外部リソース名	CICSplex SM リソース名	説明
CICSActiveAnalysisPointSpecification	APSPEC	RTA 分析点仕様
CICSApplicationDefinition	APPLDEF	アプリケーション定義

表 3. CICSplex SM 定義リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSBatchrepRequest	BATCHREP	バッチ・リポジトリ更新ジョブ
CICSCICSplexDefinition	CPLEXDEF	CICSplex 定義
CICSCMASToCMASLinkDefinition	CMTCMDEF	CMAS から CMAS へのリンク定義
CICSMonitorDefinition	MONDEF	モニター定義
CICSMonitorGroup	MONGROUP	モニター・グループ
CICSMonitorGroupInSpecification	MONINSPC	モニター仕様のモニター・グループ
CICSMonitorResourceInGroup	MONINGRP	グループのモニター定義
CICSMonitorSpecification	MONSPEC	モニター仕様
CICSMonitorSpecificationsToSystem	LNKSMSCS	CICS システム・リンクへのモニター仕様
CICSMonitorSpecificationsToSystemGroup	LNKSMSCG	モニター仕様とシステム・グループとの間のリンク
CICSPeriodDefinition	PERIODEF	時間枠定義
CICSPlatformDefinition	PLATDEF	プラットフォーム定義
CICSPrimaryAnalysisPointCMAS	CMDMPAPS	1 次 CMAS 分析点仕様
CICSRegionDefinition	CSYSDEF	CICS 領域の定義
CICSRegionGroup	CSYSGRP	CICS システム・グループの定義
CICSResourceAssignmentDefinition	RASGNDEF	リソース割り当て定義
CICSResourceAssignmentInDescription	RASINDSC	リソース記述内のリソース割り当て
CICSResourceDescription	RESDISC	リソース記述の定義
CICSResourceGroup	RESGROUP	リソース・グループの定義
CICSResourceInGroup	RESINGRP	リソース・グループに関連付けられたリソース
CICSResourceGroupInDescription	RESINDSC	リソース記述内のリソース・グループ
CICSRTAActionDefinition	ACTION	RTA アクションの定義
CICSRTADefinition	RTADEF	RTA 定義
CICSRTAEvaluationDefinition	EVALDEF	評価定義
CICSRTAGroup	RTAGROUP	RTA グループの定義
CICSRTAResourceInGroup	RTAINGRP	RTA グループ内の RTA 定義
CICSRTASpecification	RTASPEC	RTA 仕様定義
CICSRTAStatusDefinition	STATDEF	RTA 状況定義
CICSRTAStatusDefinitionInGroup	STAINGRP	RTA グループ内の状況定義
CICSSecondaryAnalysisPointCMAS	CMDMSAPS	2 次 CMAS 分析点仕様
CICSSystemGroupToSystemGroup	CSGLCGCS	CICS システムとグループのリンク
CICSSystemLink	SYSLINK	CICS システム・リンク定義

表 3. CICSplex SM 定義リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSSystemToSystemGroup	CSGLCGCG	CICS システム・グループとグループのリンク
CICSTransactionGroup	TRANGRP	CICS トランザクション・グループの定義
CICSTransactionInGroup	DTRINGRP	トランザクション・グループ内のトランザクション
CICSWLMActiveRouter	WLMAROUT	アクティブ・ワークロード内の CICS ルーター領域
CICSWLMDefinition	WLMDEF	WLM 定義
CICSWLMGroup	WLMGROUP	WLM グループの定義
CICSWLMResourceInGroup	WLMINGRP	ワークロード・グループ内の WLM 定義
CICSWLMSpecification	WLMSPEC	WLM 仕様

CICSplex SM 非定義リソース

表 4. CICSplex SM 非定義リソースのリソース名		
外部リソース名	CICSplex SM リソース名	説明
CICSActiveAnalysisPoint	APACTV	分析点仕様に関連付けられたインストール済み分析定義
CICSApplication	APPLCTN	デプロイ済みアプリケーション
CICSCICSManagingAddressSpace	CMAS	アクティブ CMAS
CICSCICSManagingAddressSpaceList	CMASLIST	ローカル CMAS に認識されている CMAS
CICSCICSplex	CICSplex	CMAS によって管理されている CICSplex
CICSCICSplexList	CMASplex	ローカル CMAS によって管理されているすべての CICSplex
CICSCMASInCICSplex	CPLXCMAS	CICSplex 定義中の CMAS
CICSCMASToCMASLink	CMTCLNK	CMAS 間リンク
CICSManagedRegion	MAS	CICSplex SM 管理対象 CICS 領域
CICSManagedRegionStatus	MASSTAT	CMAS による管理 CICS 領域の状況
CICSManagementPart	MGMTPART	アプリケーションまたはプラットフォームの管理パート
CICSPlatform	PLATFORM	プラットフォーム
CICSResourceByAssignment	RASPROC	リソース割り当てによって選択されたリソース
CICSResourceByDescription	RDSCPROC	リソース記述によって選択されたリソース
CICSResourceBySystem	SYSRES	CICS システムに割り当てられたリソース

表 4. CICSplex SM 非定義リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSRTAEvent	EVENT	CICSplex またはその CICS システム 1 つにおける、未処理の大きな状況の変更
CICSRTAEventDetail	EVENTDTL	イベントの生成に関係した評価
CICSRTAInstalledDefinition	RTAACTV	RTA インストール済み分析および状況定義
CICSRule	RULE	ポリシー・ルール情報
CICSTaskHistory	HTASK	完了したタスク
CICSTaskHistoryCollection	MASHIST	CICS タスク・ヒストリー・コレクション
CICSTopologyAtomService	CRESATOM	CICS システム内の Atom サービス
CICSTopologyAutoInstallModel	CRESAIMD	自動インストール・モデル用のトポロジー・データ
CICSTopologyBundle	CRESBUND	CICS システム内のバンドル
CICSTopologyBTSPProcessType	CRESPTY	CICS システムの BTS プロセス・タイプ
CICSTopologyCaptureSpecification	CRESEVCS	イベント処理キャプチャー仕様
CICSTopologyDataSet	CRESDSNM	CICS システム内のデータ・セット
CICSTopologyDb2Connection	CRESDB2C	CICS システム内の Db2 接続
CICSTopologyDb2Entry	CRESDB2E	CICS システム内の Db2 エントリー
CICSTopologyDb2Transaction	CRESDB2T	CICS システム内の Db2 トランザクション
CICSTopologyDocumentTemplate	CRESDOCT	CICS システム内の文書テンプレート
CICSTopologyEnqueueModel	CRESENQM	CICS システム内の ENQ/DEQ モデル
CICSTopologyEPAdapter	CRESEPAD	イベント処理アダプター
CICSTopologyEPAdapterSet	CRESEPAS	イベント処理アダプター・セット
CICSTopologyEventBinding	CRESEVBD	CICS システム内のイベント・バインディング
CICSTopologyFEPIConnection	CRESFECO	CICS システム内の FEPI 接続
CICSTopologyFEPINode	CRESFEND	CICS システム内の FEPI ノード
CICSTopologyFEPIPool	CRESFEPO	CICS システム内の FEPI プール
CICSTopologyFEPITarget	CRESFETR	CICS システム内の FEPI ターゲット
CICSTopologyFile	CRESFILE	CICS システム内のファイル
CICSTopologyGlobalUserExit	CRESGLUE	CICS システム内のグローバル・ユーザー出口
CICSTopologyIPICConnection	CRESIPCN	CICS システム内の IPIC 接続
CICSTopologyISCMROConnection	CRESCONN	CICS システム内の ISC/MRO 接続
CICSTopologyJournalName	CRESJRNM	CICS システム内のジャーナル名
CICSTopologyJVMServer	CRESJVMS	CICS システム内の JVM サーバー
CICSTopologyLibrary	CRESLIBR	LIBRARY 用のトポロジー・データ
CICSTopologyLU62ModeName	CRESMODE	CICS システム内の LU62 モード名

表 4. CICSplex SM 非定義リソースのリソース名 (続き)		
外部リソース名	CICSplex SM リソース名	説明
CICSTopologyPartner	CRESPART	CICS システム内のパートナー
CICSTopologyPipeline	CRESPIPE	PIPELINE 用のトポロジー・データ
CICSTopologyProfile	CRESPROF	CICS システム内のプロファイル
CICSTopologyProgram	CRESPRGM	CICS システム内のプログラム
CICSTopologyRequestModel	CRESRQMD	CICS システム内の要求モデル
CICSTopologySystemDump	CRESSDMP	CICS システム内のシステム・ダンプ
CICSTopologyTaskRelatedExit	CRESTRUE	CICS システム内のタスク関連出口
CICSTopologyTCPIPService	CRESTCPS	CICS システム内の TCP/IP サービス
CICSTopologyTDQueue	CRESTDQ	CICS システム内の一時データ・キュー
CICSTopologyTerminal	CRESTERM	CICS システム内の端末
CICSTopologyTransaction	CRESTRAN	CICS システム内のトランザクション
CICSTopologyTransactionDump	CRESTDMP	CICS システム内のトランザクション・ダンプ
CICSTopologyTSModel	CRESTSMD	CICS システム内の一時記憶域モデル
CICSTopologyUrimap	CRESURIM	URIMAP 用のトポロジー・データ
CICSTopologyWebservice	CRESWEBS	WEBSERVICE 用のトポロジー・データ
CICSTopologyXMLTransform	CRESXMLT	XML 変換用のトポロジー・データ
CICSWLMActiveAffinity	WLMATAFF	アクティブ・ワークロード・トランザクション・グループの類縁性
CICSWLMActiveAOR	WLMAWAOR	アクティブ・ワークロード・ターゲット分散係数
CICSWLMActiveDefinition	WLMAWDEF	アクティブ・ワークロードの定義
CICSWLMActiveTOR	WLMAWTOR	アクティブ・ワークロード・ルーティング領域
CICSWLMActiveTransactionGroup	WLMATGRP	アクティブ・ワークロード・トランザクション・グループ
CICSWLMActiveTransaction	WLMATRAN	アクティブ・ワークロード動的トランザクション
CICSWLMActiveWorkload	WLMAWORK	アクティブ・ワークロード
CICSWLMTarget	WLMATARG	1 つ以上のアクティブ・ワークロード用のターゲット領域

CMCI 要求および応答には、XML 形式の情報が収容されています。CMCI 内の固有 XML エlement は、CICSSystemManagement というスキーマ内で定義されます。このスキーマは、リソース・タイプごとに別個のスキーマを取り込んで構成されています。

クライアント・アプリケーションは、CICS 管理クライアント・インターフェース 要求に対する応答の本体にある情報を検証してフォーマット設定するために、CICSSystemManagement スキーマにアクセスする必要があります。クライアントは、この出力 XML が定義されたスキーマに以下の URI でアクセスできます。
`http://hostname:portnumber/CICSSystemManagement/schema/CICSSystemManagement.xsd`

注: CMCI は、EBCDIC 037 コード・ページで扱われている文字のみ解釈できます。CMCI 要求で他の文字を使用すると、予測不能な結果が生じます。

<request> エlement

<request> Element は、CICS 管理クライアント・インターフェースの POST または PUT 要求の本体 XML におけるルート・Element です。

POST 要求では、<request> Element には、1 つの子 <create> Element が入ります。PUT 要求の <request> Element には、1 つの子 Element <update> または、例に示しているように、1 つの <action> Element を含めることができます。

DELETE 要求の <request> Element は通常は空です。指定する場合、<request> Element には 1 つの子 Element <delete> を含めなければなりません。

CMCI 要求には、<request> Element を 1 つだけ含めることができます。

含まれる内容:

POST 要求内: 1 つの 27 ページの『<create> Element』

PUT 要求内: 1 つの 28 ページの『<update> Element』または 1 つの 29 ページの『<action> Element』

DELETE 要求内: 1 つの 30 ページの『<delete> Element』

例

以下の例は、名前が ABCD である BAS リソースに対するインストール・アクションを示しています。<action> Element が <request> Element に含まれていることに注意してください。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>/?CRITERIA=NAME%3DABCD

<request>
  <action name="INSTALL">
    <parameter name="TARGET" value="SYSTEM1"/>
    <parameter name="USAGE" value="LOCAL"/>
  </action>
</request>
```

<create> Element

<create> Element には、POST 要求の <parameter> Element と <attributes> Element が入ります。

参照元:

27 ページの『<request> Element』

含まれる内容:

<create> エレメントには、1 つの <attributes> エレメントを含めなければならず、また 1 つ以上の子エレメント <parameter> を含めることができます。POST 要求には 1 つの <create> エレメントだけを含めることができます。

[31 ページの『<parameter> エレメント』](#)

[31 ページの『<attributes> エレメント』](#)

例

以下の例は、名前が ABCD である BAS リソースを作成することを示しています。

```
POST http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>
```

```
<request>
  <create>
    <attributes name="ABCD" defver="1" />
  </create>
</request>
```

以下の例は、名前が ABCD である BAS リソースをグループ GROUP1 に作成することを示しています。

```
POST http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>
```

```
<request>
  <create>
    <parameter name="RESGROUP" value="GROUP1"/>
    <attributes name="ABCD" defver="1" />
  </create>
</request>
```

以下の例は、名前が ABCD である CSD リソースをグループ GROUP2 に作成することを示しています。

```
POST http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>/<SCOPE>
```

```
<request>
  <create>
    <parameter name="CSD"/>
    <attributes name="ABCD" csdgroup="GROUP2"/>
  </create>
</request>
```

<update> エレメント

<update> エレメントには、PUT 要求の <parameter> エレメントと <attributes> エレメントが入ります。

参照元:

[27 ページの『<request> エレメント』](#)

含まれる内容:

<update> エレメントには、1 つの <attributes> エレメントを含めなければならず、また 1 つ以上の子エレメント <parameter> を含めることができます。PUT 要求には 1 つの <update> エレメントだけを含めることができます。

[31 ページの『<parameter> エレメント』](#)

[31 ページの『<attributes> エレメント』](#)

例

以下の例では、名前が ABCD である BAS トランザクション定義が更新されて、値 shutdown="DISABLED"、isolate="YES"、および dynamic="NO" を持つようになります。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/CICSDefinitionalTransaction/<CONTEXT>/?
CRITERIA=NAME%3DABCD
```

```
<request>
  <update>
    <attributes shutdown="DISABLED" isolate="YES" dynamic="NO" />
  </update>
</request>
```

以下の例では、グループ GROUP2 にある、名前が ABCD である CSD リソースの description が更新されます。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>/<SCOPE>
/?CRITERIA=NAME%3DABCD&PARAMETER=CSDGROUP%28GROUP2%29
```

```
<request>
  <update>
    <parameter name="CSD"/>
    <attributes description="updated"/>
  </update>
</request>
```

<action> エレメント

<action> エレメントには、URI によって指定されたリソースに対して CICS 管理クライアント・インターフェースの PUT 要求によって実行されるアクションの詳細が入ります。

参照元:

[27 ページの『<request> エレメント』](#)

属性

name="action_name"

<action> エレメントには、実行されるアクションを識別する **name** 属性を 1 つだけ含める必要があります。

action_name は、対応する単一引用符または二重引用符で囲む必要があります。XML のコーディング規則と同様、エスケープ・シーケンスを使用して &、<、>などの特定の特殊文字を置換できます。アクションにパラメーターが関連付けられている場合は、<parameter> エレメントを使用して <action> の子エレメントとして指定する必要があります。

HTTP 本体の中で <action> エレメントを指定する場合、同じ要求の中に <attributes> エレメントを組み込んではいけません。

含まれる内容:

<action> エレメントには、1 つ以上の子エレメント <parameter> を含めることができます。

[31 ページの『<parameter> エレメント』](#)

例

以下の例は、名前が ABCD である BAS リソースに対するインストールを示しています。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/<RESOURCE NAME>/<CONTEXT>/?CRITERIA=NAME%3DABCD
```

```
<request>
  <action name="INSTALL">
    <parameter name="TARGET" value="SYSTEM1"/>
    <parameter name="USAGE" value="LOCAL"/>
  </action>
</request>
```

以下の例は CSDADD アクションを示し、CSD グループ GROUP4 が CSD リスト LIST2 に、CSD グループ GROUP3 の後へ追加されます。

CSDADD アクションに対して指定されているパラメーターは、TO_CSDLIST、ADD_CSDGROUP、および ADD_LOCATION です。

TO_CSDLIST は、どの CSD リストに CSD グループを追加するかを指定します。

ADD_CSDGROUP は、指定された CSD グループを、CSD リストでどの CSD グループの隣に追加するかを指定します。

ADD_LOCATION は、指定された CSD グループを、CSD リストで既存の CSD グループの "BEFORE" つまり前に追加するか、あるいは "AFTER" つまり後に追加するかを指定します。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/CICSCSDGroup/<CONTEXT>/<SCOPE>/?CRITERIA=NAME%3DGROUP4
```

```
<request>
  <action name="CSDADD">
    <parameter name="TO_CSDLIST" value="LIST2"/>
    <parameter name="ADD_CSDGROUP" value="GROUP3"/>
    <parameter name="ADD_LOCATION" value="AFTER"/>
  </action>
</request>
```

以下の例は、CICS 操作リソース RESOURCE に対する DISABLE アクションを示しています。リソースのタイプが URI に含まれていることに注意してください。この場合、リソースのタイプは CICSAtomService です。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/CICSAtomService/<CONTEXT>/<SCOPE>/?CRITERIA=NAME%3DRESOURCE
```

```
<request>
  <action name="DISABLE"/>
</request>
```

<delete> エレメント

<delete> エレメントには、URI によって指定されたリソースに対して CICS 管理クライアント・インターフェースの DELETE 要求によって実行されるアクションの詳細が入ります。

参照元:

[27 ページの『<request> エレメント』](#)

含まれる内容:

<delete> エレメントは通常、パラメーターをまったく含まないため不要です。

[31 ページの『<parameter> エレメント』](#)

CSD リソースを削除する場合は、PARAMETER=CSDGROUP を URI に含める必要があります。<parameter name="CSD"/> を XML 本体に含めることもできますが、その必要はありません。PARAMETER=CSDGROUP を URI に指定して、<parameter name="CSD"/> を XML 本体に指定しないと、CICS は <parameter name="CSD"/> が指定されたと解釈します。

例

以下の例は、名前が ABCD である BAS リソース定義に対する削除要求を示しています。削除要求に必要なのは URI のみです。

```
DELETE /CICSSystemManagement/<RESOURCE_NAME>/<CONTEXT>/?CRITERIA=NAME%3DABCD
```

以下の例は、CSD リソース定義に対する削除要求を示しています。PARAMETER=CSDGROUP が URI に含まれていることに注意してください。

```
DELETE /CICSSystemManagement/<RESOURCE_NAME>/<CONTEXT>/<SCOPE>?CRITERIA=NAME%3DABCD&PARAMETER=CSDGROUP%28GROUP1%29
```

以下の例は前の例と等価で、異なる点はオプションの <delete> エレメントが XML 本体に含まれていることです。

```
DELETE /CICSSystemManagement/<RESOURCE_NAME>/<CONTEXT>/<SCOPE>?CRITERIA=NAME%3DABCD&PARAMETER=CSDGROUP%28GROUP1%29

<request>
  <delete>
    <parameter name="CSD"/>
  </delete>
</request>
```

<parameter> エレメント

<parameter> エレメントには、CICS 管理クライアント・インターフェースの POST 要求または PUT 要求に関連したパラメーターの名前と値が入ります。POST または PUT 要求には、1 つ以上のパラメーター・エレメントを含めることができます。

参照元:

POST 要求の場合: [27 ページの『<create> エレメント』](#)

PUT 要求内: 1 つの [29 ページの『<action> エレメント』](#) または 1 つの [28 ページの『<update> エレメント』](#)

属性

name="parameter_name"

アクションに関連したパラメーターの名前。 *parameter_name* には大/小文字の区別があり、対応する単一引用符または二重引用符で囲む必要があります。

value="parameter_value"

指定されたパラメーターの値。 *parameter_value* には大/小文字の区別があり、対応する単一引用符または二重引用符で囲む必要があります。

例

```
<parameter name="TARGET" value="SYSTEM1"/>
<parameter name="USAGE" value="LOCAL"/>
```

<attributes> エレメント

<attributes> エレメントには、CICS 管理クライアント・インターフェースの POST 要求または PUT 要求の属性の新しい値の詳細が入ります。1 つの要求には 1 つの <attributes> エレメントだけを含めることができます。

参照元:

[27 ページの『<request> エレメント』](#)

属性

attribute_name="data_value"

<attributes> エレメントには、指定された属性に提供される新しい値を指定する *attribute_name* "data_value" の組み合わせを、1 つ以上含める必要があります。各リソースの必要属性のリストについては、[CICSplex SM リソース・テーブル](#)を参照してください。

data_value 属性には大/小文字の区別があり、対応する単一引用符または二重引用符で囲む必要があります。XML のコーディング規則と同様、エスケープ・シーケンスを使用して &、<、>などの特定の特殊文字を置換できます。

HTTP 本体の中で <attributes> エレメントを指定する場合、同じ要求の中に <action> エレメントを組み込んではいけません。

例

```
<attributes name="ABCD" defver="1" status="ENABLED" taskdataloc="ANY"
taskdatakey="USER" storageclear="NO" shutdown="DISABLED" isolate="YES"
dynamic="NO" routable="NO" restart="NO" spurge="NO" tpurge="NO" dump="YES"
trace="YES" otsttimeout="NO" wait="YES" ressec="NO" cmdsec="NO"
program="ABCDEFGH" />
```

<response> エlement

<response> エlementは、CICS 管理クライアント・インターフェース 要求に対する応答内のルート・Elementです。<response> Elementには子Elementを3つまで収容できます。それらの子Elementは、必須の<resultsummary> Element、要求により返されたすべてのレコードを収容する<records> Element、および (要求によりエラーが返される場合に) 診断情報を収容する<errors> Elementです。

含まれる内容:

- [32 ページの『<resultsummary> Element』](#)
- [35 ページの『<errors> Element』](#)
- [34 ページの『<records> Element』](#)

属性

xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int"

CICS 管理クライアント・インターフェースのネーム・スペース。

xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int http://hostname:portnumber/CICSSystemManagement/schema/CICSSystemManagement.xsd"

CICS 管理クライアント・インターフェース XMLスキーマの場所。

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

XMLスキーマ・インスタンスのネーム・スペース。

version="schema_version"

CICS 管理クライアント・インターフェース・スキーマのバージョン番号。この番号は1.0から開始され、スキーマの更新ごとにインクリメントされます。

connect_version="connect_version"

EXEC CPSM CONNECT API コマンドを使用して CICSplex SM に接続したときに使用された CICSplex SM リリース番号。

例

```
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int" xsi:schemaLocation=
"http://www.ibm.com/xmlns/prod/CICS/smw2int http://example.com:27231/
CICSSystemManagement/schema/CICSSystemManagement.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
connect_version="0410">
```

```
...
</response>
```

<resultsummary> Element

<resultsummary> Elementには、CICS 管理クライアント・インターフェース 要求に関する要約情報が入ります。CICS 管理クライアント・インターフェース 要求に対する応答には、<resultsummary> Elementが含まれている必要があります。

参照元:

- [32 ページの『<response> Element』](#)

属性

api_source="value"

要求に失敗した場合、API ソースの名前 (例えば、「CICSplex SM」または「CMCI」) が入ります。要求が成功した場合には、この属性は応答に含まれません。

api_function="function"

要求に失敗した場合、失敗した CICS システム・コマンドまたは CICSplex SM API コマンドの名前が入ります (例えば PERFORM SET)。要求が成功した場合には、この属性は応答に含まれません。

api_response1="value"

数値としての CICS EIBRESP コードまたは CICSplex SM API EYUDA 応答コード。CICS RESP 値に関する詳細は、[RESP および RESP2 オプション](#)を参照してください。CICSplex SM API 応答コードに関する詳細は、[EYUDA 値](#)を参照してください。

api_response1_alt="value"

テキスト・ストリングとしての CICS EIBRESP コードまたは CICSplex SM API EYUDA 応答コード。CICS RESP 値に関する詳細は、[RESP および RESP2 オプション](#)を参照してください。CICSplex SM API 応答コードに関する詳細は、[EYUDA 値](#)を参照してください。

api_response2="value"

数値としての CICS EIBRESP2 コードまたは CICSplex SM API EYUDA 理由コード。成功した要求の値は 0 です。CICS RESP2 値に関する詳細は、[RESP および RESP2 オプション](#)を参照してください。CICSplex SM API 理由コードに関する詳細は、[EYUDA 値](#)を参照してください。

api_response2_alt="value"

テキスト・ストリングとしての CICS EIBRESP2 コードまたは CICSplex SM API EYUDA 理由コード。CICS RESP2 値に関する詳細は、[RESP および RESP2 オプション](#)を参照してください。CICSplex SM API 理由コードに関する詳細は、[EYUDA 値](#)を参照してください。

recordcount="value"

要求によって返されるレコードの総数。

recordcount 属性は常に、結果内のレコードの総数 (要求が失敗したレコードを含む) を示します。レコード範囲が指定されている場合、**recordcount** の値は、要求によって返されるレコードの数より大きくなる場合があります。

displayed_recordcount="value"

要求によって返されるレコードのカウント。

レコード範囲が指定されている場合、**displayed_recordcount** の値は、**recordcount** によって示される結果内のレコードの総数より小さくなる場合があります。

successcount="value"

DELETE 要求の場合のみ、要求が成功したレコードの数。

cachetoken="value"

保持された結果のセットが入っている結果キャッシュを識別するトークン。**cachetoken** 属性は、GET 要求で NODISCARD オプションが指定される場合のみ応答に含まれます。**cachetoken** の値は、保管クロック値を 16 文字の固定長 16 進数で表したものです。

例

以下の例は、成功した DELETE 要求の結果を示しています。

```
<resultsummary api_response1="1024" api_response1_alt="OK"
api_response2="0" api_response2_alt="" recordcount="20"
displayed_recordcount="20" successcount="18" />
```

以下の例は、正常に完了しない DELETE 要求の結果を示しています。

```
<resultsummary api_source="CICSplex SM" api_function="SET" api_response1="1038"
api_response1_alt="TABLEERROR" api_response2="1361" api_response2_alt="DATAERROR"
recordcount="0" displayed_recordcount="0" successcount="0"/>
```


以下の例は、NODISCARD オプションを含む GET 要求の結果を示します。

```
<resultsummary api_response1="1024" api_response1_alt="OK"
api_response2="0" api_response2_alt="" recordcount="2"
dispayed_recordcount="2" cachetoken="C36E880ECAC5818B" />
```

<records> エレメント

<records> エレメントには、要求によって返されるレコードの詳細が入ります。<records> エレメントには、要求によって返されるレコードごとに 1 つの子エレメントが入ります。要求がレコードを返さない場合、<records> エレメントはありません。

参照元:

[32 ページの『<response> エレメント』](#)

含まれる内容:

<records> エレメントには、要求によって変更されるリソースのインスタンスごとに 1 つの子エレメントが入ります。それらの子エレメントは、要求 URI で識別されるリソースによって指定されます。例えば、PUT 要求が CICS ローカル・ファイルに対して変更を行う場合、PUT 応答の <records> エレメントには <cicslocalfile> という名前の 1 つ以上の子エレメント、つまり、要求によって返されるレコードごとに 1 つの <cicslocalfile> エレメントが入ります。子エレメントの名前は、CICS 管理クライアント・インターフェースのリソース名から派生したものになります。これらの名前の詳しいリストについては、[17 ページの『CMCI リソース名』](#)を参照してください。

属性

<records> エレメント単体では属性を持ちません。

子エレメントの属性で、識別されたリソースの属性の値を指定します。属性の値が変更されたかどうかに関係なく、リソースの属性はすべてこのエレメントに存在します。値がブランクになっている属性は、デフォルト値を取ります。各リソースの必要属性のリストについては、[CICSplex SM リソース・テーブル](#)を参照してください。

各リソース・エレメントには、_keydata="data string" 形式の **_keydata** 属性も含まれます。ここで *data string* は、リソースのインスタンスを一意的に識別する文字のストリングです。

例

以下の例は、CICSLocalFile リソースのインスタンスをリストする 1 つのレコードを示しています。

```
<records>
<cicslocalfile _keydata="C4C6C8C3E2C44040" accessmethod="VSAM" add="ADDDABLE" addcnt="0"
basedsname="USER1.ALLAPPL.DFHCS" blockformat="BLOCKED" blockkeyln="N/A"
blocksize="N/A" browse="BROWSABLE" browsecnt="2115" browupdcnt="0" datasettype="K"
delete="DELETABLE" dexpcnt="0" disposition="SHARE" dsname="USER1.ALLAPPL.DFHCS"
emptystatus="NOEMPTYREQ" enablestatus="ENABLED" exclusive="NOTAPPLIC" file="DFHCS"
fwdrccstatus="NOTFWDRCVBLE" getcnt="115" getupdcnt="0" gmfilecls="00:00:00.0000"
gmfileopn="00:00:00.0000" iexpcnt="0" journalnum="0" keylength="0"
keyposition="0" locdelcnt="0" lsrpoolid="01" numactstring="0" numdatbuff="0"
numindexbuff="0" numstringwt="0" object="BASE" openstatus="CLOSED"
rbatype="NOTAPPLIC" read="READABLE" readinteg="NOTAPPLIC" recordformat="VARIABLE"
recordsize="0" recovstatus="NOTRECOVABLE" reltype="NOTAPPLIC" rlsaccess="NOTRLS"
rlsreqwtto="0" strings="6" timeclose="0:00:00.0" timeopen="0:00:00.0"
update="UPDDABLE" updatecnt="0" vsamtype="NOTAPPLIC" wstrccurcnt="0" wstrcnt="0"
eyu_cicsname="U1WUIA" eyu_cicsrel="E660" eyu_reserved="0"/>
</records>
```


<errors> エlement

<errors> Elementには、CICS 管理クライアント・インターフェース 要求に関連した診断情報が入ります。どの子Elementでも表示されるような診断情報がある場合、CICS 管理クライアント・インターフェース要求に対する応答に<errors> Elementが入ります。

<errors> Elementには、要求に関連した1つ以上の<feedback> 子Elementが入ります。<errors> Elementには、関連した属性がありません。

参照元:

[32 ページの『<response> Element』](#)

含まれる内容:

[35 ページの『<feedback> Element』](#)

例

<feedback> Element

<feedback> Elementには、CICS 管理クライアント・インターフェース 要求に関連した FEEDBACK レコードからの診断データが入ります。<errors> Elementには、要求によって返されるレコードごとに1つ以上の<feedback> Elementを含めることができます。

FEEDBACK レコードに関する詳細は、[フィードバック・レコードの取得](#)を参照してください。

CICS 管理クライアント・インターフェースの PUT インストール要求については、<feedback> Elementに以下の子Elementを含めることができます。

- BINSTERR レコードで失敗するインストール・アクションごとに1つの<installerror> Element
- BINCONSC レコードで失敗するインストール・アクションごとに1つの<inconsistentscope> Element
- BINCONRS レコードで失敗するインストール・アクションごとに1つの<inconsistentset> Element

参照元:

[35 ページの『<errors> Element』](#)

含まれる内容:

[36 ページの『<installerror> Element』](#)

[37 ページの『<inconsistentscope> Element』](#)

[38 ページの『<inconsistentset> Element』](#)

属性

action="action_name"

失敗したアクションの名前。

attribute1="attribute_name"

エラーに関連した最大6つの属性のうちの1つの名前。

attribute2="attribute_name"

エラーに関連した最大6つの属性のうちの1つの名前。

attribute3="attribute_name"

エラーに関連した最大6つの属性のうちの1つの名前。

attribute4="attribute_name"

エラーに関連した最大6つの属性のうちの1つの名前。

attribute5="attribute_name"

エラーに関連した最大 6 つの属性のうちの 1 つの名前。

attribute6="attribute_name"

エラーに関連した最大 6 つの属性のうちの 1 つの名前。

eibfn="eibfn_number"

要求に関連した機能コード。

eibfn_alt="function"

要求に関連した機能の名前。

errorcode="error_code"

リソースに関連した CICSplex SM エラー・コード。

eyu_cicsname="name"

エラーに関連した CICS 領域または CICSplex の名前。

keydata="data"

エラーに関連したリソースのインスタンスを識別するデータのストリング。

resp="resp_number"

数値としての CICS RESP コードまたは CICSplex SM API EYUDA 応答コード。

resp2="resp2_number"

数値としての CICS RESP2 コードまたは CICSplex SM API EYUDA 理由コード。

resp_alt="resp_alt_text"

resp 値に相当するテキスト。例えば、16 という resp 値に相当するテキストは INVREQ です。

例

```
<feedback action="STOP" eibfn="4C10" eibfn_alt="DISCARD FILE" eyu_cicsname="REG1"
keydata="FILEX" resp="16" resp_alt="INVREQ" resp2="2"/>
```

<installerror> エlement

<installerror> Element には、CICS 管理クライアント・インターフェースの PUT 要求に関連した BINSTERR レコードからの診断データが入ります。<installerror> Element の親は、<feedback> Element です。

BINSTERR レコードに関する詳細は、[BINSTERR リソース・テーブルのレコードの評価](#)を参照してください。

参照元:

[35 ページの『<feedback> Element』](#)

属性

eibfn="eibfn_number"

要求に関連した機能コード。

eyu_cicsname="name"

インストール・エラーに関連した CICS 領域または CICSplex の名前。

cresp1="resp_number"

数値としての CICS RESP コードまたは CICSplex SM API EYUDA 応答コード。

cresp2="resp2_number"

数値としての CICS RESP2 コードまたは CICSplex SM API EYUDA 理由コード。

errorcode="error_code"

リソースに関連した CICSplex SM エラー・コード。

ressname="resource_name"

エラーに関連したリソースの名前。

resver="resource_version"

エラーに関連したリソースのバージョン番号。

例

```
<installerror eibfn="3036" errorcode="4" eyu_cicsname="MYWUIB" resourcename="URIMAP_X"
resourceversion="1" resp="16" resp2="626"/>
```

<inconsistentscope> エlement

<inconsistentscope> Elementには、CICS 管理クライアント・インターフェースの PUT 要求に関連した BINCONSC レコードからの診断データが入ります。<inconsistentscope> Elementの親は、<feedback> Elementです。

BINCONSC レコードに関する詳細は、[BINCONSC リソース・テーブル・レコードの評価](#)を参照してください。

参照元:

[35 ページの『<feedback> Element』](#)

属性

eibfn="eibfn_number"

要求に関連した機能コード。

eyu_cicsname="name"

スコープ不整合エラーに関連した CICS 領域または CICSplex の名前。

erroroperation="value"

エラーが発生したときに実行される操作を識別する数値。

errorcode="error_code"

リソースに関連した CICSplex SM エラー・コード。

targetassignment="assignment"

ターゲット・スコープの割り当て。

targetdescription="description"

ターゲット・スコープのリソース記述。

relatedassignment="assignment"

関連スコープのリソース割り当て。

relateddescription="assignment"

関連スコープのリソース記述。

relatedscope="scope"

関連スコープの名前。

例

リソース割り当てをリソース記述に追加しようとする試みは、ターゲット・スコープとリソース割り当ての関連スコープとが同じエラー・コード 1 を BINCONSC を生成しているため失敗します。

```
<?xml version="1.0"?>
<response xmlns="http://www.example.com/xmlns/prod/CICS/smw2int"
xmlns:xsi="http://www.example.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.com/xmlns/prod/CICS/smw2int
http://example.com:27870/CICSSystemManagement/schema/CICSSystemManagement.xsd"
version="2.0" connect_version="0420">
  <resultsummary api_function="PERFORM SET" api_response1="1038" api_response2="1361"
api_response1_alt="TABLEERROR" api_response2_alt="DATAERROR" recordcount="1" />
</errors>
```

```

<feedback action="ADDTODSC">
  <inconsistentscope erroroperation="1" errorcode="1" targetassignment="ATEST1"
    targetscope="MCLMASA" relatedassignment="ATEST1" relatedscope="MCLMASA" />
</feedback>
<feedback keydata="C1E3C5E2E3F14040" action="ADDTODSC" errorcode="24"
  attribute1="RESASSGN" />
</errors>
</response>

```

<inconsistentset> エレメント

<inconsistentset> エレメントには、CICS 管理クライアント・インターフェースの PUT 要求に関連した BINCONRS レコードからの診断データが入ります。<inconsistentset> エレメントの親は、<feedback> エレメントです。

BINCONRS レコードに関する詳細は、[BINCONRS リソース・テーブル・レコードの評価](#)を参照してください。

参照元:

[35 ページの『<feedback> エレメント』](#)

属性

candidatename="resource_name"

候補リソースの名前。

candidateversion="resource_version"

候補リソースのバージョン番号。

candidategroup="group_name"

候補リソースのリソース・グループ。

candidateassignment="assignment"

候補リソースの割り当て。

candidatedescription="assignment"

候補リソースの記述。

candidateusage="usage"

候補リソースの割り当ての使用法。

candidatesystemgroup="group_name"

候補リソースのシステム・グループ。

candidatetype="system_type"

候補リソースのシステム・タイプ。

candidateoverride="assignment_override"

候補リソースの割り当ての指定変更。

eyu_cicsname="name"

インストール・エラーに関連した CICS 領域の名前。

erroroperation="value"

エラーが発生したときに実行される操作を識別する数値。

existingname="resource_name"

既存のリソースの名前。

existingversion="resource_version"

既存のリソースのバージョン番号。

existinggroup="group_name"

既存のリソースのリソース・グループ。

existingassignment="assignment"

既存のリソースの割り当て。

existingdescription="assignment"

既存のリソースの記述。

existingusage="usage"

既存のリソースの割り当ての使用法。

existingsystemgroup="group_name"

既存のリソースのシステム・グループ。

existingtype="system_type"

既存のリソースのシステム・タイプ。

existingoverride="assignment_override"

既存のリソースの割り当ての指定変更。

例

リソース割り当てをリソース記述に追加しようとする試みは、そのリソース記述とそれに関連した既存のリソース割り当てとの競合のため失敗します。

```
<?xml version="1.0"?>
<response xmlns="http://www.example.com/xmlns/prod/CICS/smw2int"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.com/xmlns/prod/CICS/smw2int
    http://example.com:27870/CICSSystemManagement/schema/CICSSystemManagement.xsd"
  version="2.0" connect_version="0420" >
  <resultsummary api_function="PERFORM SET" api_response1="1038" api_response2="1361"
    api_response1_alt="TABLEERROR" api_response2_alt="DATAERROR" recordcount="1" />
  <errors >
    <feedback action="ADDTODSC">
      <inconsistentset candidatename="ATEST1" candidateversion="1" candidategroup="ATEST1"
        candidateassignment="ATEST2" candidatedescription="ATEST" candidateusage="LOCAL"
        candidatetype="TARGET" candidateoverride="NO" eyu_cicsname="MCLMASA" erroroperation="4"
        existingname="ATEST1" existingversion="1" existinggroup="ATEST1" existingassignment="ATEST1"
        existingdescription="ATEST" existingusage="REMOTE" existingtype="TARGET"
        existingoverride="NO" />
      <inconsistentset candidatename="ATEST2" candidateversion="1" candidategroup="ATEST1"
        candidateassignment="ATEST2" candidatedescription="ATEST" candidateusage="LOCAL"
        candidatetype="TARGET" candidateoverride="NO" eyu_cicsname="MCLMASA" erroroperation="4"
        existingname="ATEST2" existingversion="1" existinggroup="ATEST1" existingassignment="ATEST1"
        existingdescription="ATEST" existingusage="REMOTE" existingtype="TARGET"
        existingoverride="NO" />
    </feedback >
    <feedback keydata="C1E3C5E2E3F24040" action="ADDTODSC" errorcode="13" attribute1="RESASSGN" />
  </errors>
</response>
```


CMCI エスケープ・シーケンス

CMCI の URI では特定の文字が制限されます。そのような文字については、パーセント文字 (%) とそれに続く 16 進値から成るエスケープ・シーケンスで置換する必要があります。

以下の表では、制限付き文字とそれに対応する UTF-8 ASCII 16 進数エスケープ・シーケンスをリストしています。

表 5. 制限付き文字とそれに対応するエスケープ・シーケンス			
文字	エスケープ・シーケンス	文字	エスケープ・シーケンス
SPACE	%20	.(ピリオド)	%2E
!	%21	/	%2F
#	%23	:(コロン)	%3A
\$	%24	;(セミコロン)	%3B
%	%25	<	%3C
&	%26	=	%3D
'(単一引用符)	%27	>	%3E
(%28	?	%3F
)	%29	@	%40
*	%2A	[%5B
+	%2B]	%5D
, (コンマ)	%2C	¬ (否定)	%AC
- (負符号)	%2D		

例

以下の例は、エスケープ文字でエンコードする前と後の、CICS 管理クライアント・インターフェースの URI からの基準ストリングを示しています。

エンコード前:

```
CRITERIA=(TRANID=P* AND PROGRAM=PAY* AND STATUS=ENABLED) AND
((USECOUNT>0 AND STGVCNT>0) OR NOT RESTARTCNT=0)
```

エンコード後:

```
CRITERIA=%28TRANID%3DP%2A%20AND%20PROGRAM%3DPAY%2A%20AND%20STATUS%3DENABLED%29%20AND%20
%28%28USECOUNT%3E0%20AND%20STGVCNT%3E0%29%20OR%20NOT%20RESTARTCNT%3D0%29
```

以下の例は、エスケープ文字でエンコードする前と後の、CICS 管理クライアント・インターフェースの URI からのパラメーター・ストリングを示しています。

エンコード前:

```
PARAMETER=STARTDATE(07/17/2006) STARTTIME(17:00) INTERVAL(300)
```

エンコード後:

```
PARAMETER=STARTDATE%2807%2F17%2F2006%29%20STARTTIME%2817%3A00%29%20INTERVAL%28300%29
```

インターフェースは、基準ストリングとパラメーター・ストリングの最後にある任意指定のピリオド (%2E) もサポートしています。

CMCI 問題判別

CICS システム 管理クライアント・インターフェース (CMCI) は、エラーの原因を判別する上で役立つ戻りコードおよびさまざまなタイプの診断情報を提供します。

CICS 管理クライアント・インターフェースが無効なクライアントからの要求を受け取った場合、要求は失敗し、API は OK 以外のコードを応答の HTTP ヘッダーで返します。CICS 管理クライアント・インターフェースは要求の処理を行わず、それ以外の情報を提供しません。これらの応答コードの説明については、46 ページの『CICS 管理クライアント・インターフェースのエラー・メッセージ』を参照してください。

CICS 管理クライアント・インターフェースは、要求が有効であることを判別すると、HTTP 200 OK 応答コードを返し、EXEC CICS または EXEC CPSM 呼び出しを生成します。この呼び出しでエラーが検出されると、API は応答の XML 本体で診断情報を返します。この情報は、XML 応答の <resultsummary> エレメントにおける応答コードと理由コードの形式を取ります。場合によっては、追加情報を提供するフィードバック・レコードの形式を取ることもあります。

<resultsummary> エレメントには、失敗の原因の判別に使用できる次の 4 つの属性が入ります。

- api_function
- api_response1
- api_response1_alt
- api_response2
- api_response2_alt

これらの属性の説明については、32 ページの『<resultsummary> エレメント』を参照してください。

CICS 管理クライアント・インターフェースは、例外条件を受け取った場合、それを XML 応答の <errors> エレメントでクライアントに渡します。要求が正常に完了した場合には <errors> エレメントはありません。<errors> エレメントには 1 つ以上の <feedback> エレメントが含まれ、それぞれに FEEDBACK レコードが入ります。BAS エラーによっては、<feedback> エレメントに <installerror>、<inconsistentscope>、または <inconsistentset> エレメントが追加で含まれることがあり、エレメントにはそれぞれ BINSTERR、BINCONSC、または BINCONRS レコードが含まれます。

例: <resultsummary> 情報を使用した CICS 管理クライアント・インターフェース・エラーの診断

CICS 管理クライアント・インターフェースの <resultsummary> エレメントには、エラーの原因を判別する上で役立つ戻りコードが入ります。

このタスクについて

CICSManagedRegion リソース (つまり、CICSplex SM 管理対象 CICS 領域または MAS) に対して START アクションの実行を試みました。その際、以下の URI および XML 本体を持つ CICS 管理クライアント・インターフェースの PUT 要求を使用しました。

```
http://wmvs2c.mycorp.com:27650/CICSSystemManagement/CICSManagedRegion/IBEUR912?
CRITERIA=CICSNAME%3DDEVGRN1
```

```
<request>
  <action name="START"/>
</request>
```

CICS 管理クライアント・インターフェースからは HTTP/1.1 200 OK 応答を受け取りますが、レコードは返されず、応答には以下の <resultsummary> 情報が含まれています。

```
<resultsummary api_function="PERFORM SET" api_response1="1026" api_response1_alt="NOTFOUND"
api_response2="1301" api_response2_alt="ACTION" recordcount="1"/>
```

この問題が発生した理由を調べるには、以下のステップを実行します。

手順

1. **api_function** 属性からエラーの原因を判別します。
この属性は、CICS 管理クライアント・インターフェースが **PERFORM SET** 関数の呼び出しを試みたものの、呼び出しが失敗したことを示しています。

2. **api_response1_alt** 属性と **api_response2_alt** 属性から失敗の理由を判別します。

これらの値は CICSplex SM EYUDA の応答コードと理由コードです。

EYUDA 応答値のシンボル名は NOTFOUND であり、EYUDA 理由値のシンボル名は ACTION です。これらの値は、試行されたアクションが CICSplex SM MAS リソース・テーブルで見つからなかったことを意味します。MAS リソース・テーブルから選択できるアクションは、STOP、STOPUNCON、および FORCEDISCON だけです。START アクションは許可されていません。MAS リソースで選択可能なアクションを確認するには、[MAS テーブル](#)を参照してください。

例: <feedback> 情報を使用した CICS 管理クライアント・インターフェース・エラーの診断

CICS 管理クライアント・インターフェースは、CICS 管理クライアント・インターフェース 応答の <feedback> エレメントに FEEDBACK レコードを提供します。FEEDBACK レコードで提供される追加情報は、CICS 管理クライアント・インターフェース 要求に関する問題の診断に使用できます。

このタスクについて

以下の URI を持つクライアント API の DELETE 要求を使用して、CICS ローカル・ファイル・リソース MYFILE_A の削除を試みました。

```
http://mvs2c.mycorp.com:27850/CICSSystemManagement/CICSLocalFile/IBEUR912?
CRITERIA=file%3DMYFILE_A
```

CICS 管理クライアント・インターフェースからは要求が正しく形成されたことを示す HTTP/1.1 200 OK 応答を受け取りますが、レコードは返されず、応答には以下の情報が含まれています。

```
<resultsummary api_function="PERFORM SET" api_response1="1038" api_response1_alt="TABLEERROR"
api_response2="1361" api_response2_alt="DATAERROR" recordcount="1"/>
<errors>
  <feedback action="DISCARD" eibfn="4C10" eibfn_alt="DISCARD FILE" eyu_cicsname="MAS1"
keydata="MYFILE_A"
  resp="16" resp2="2"/>
</errors>
```

この問題が発生した理由を調べるには、以下のステップを実行します。

手順

1. **api_function** 属性からエラーの原因を判別します。
この属性は、CICS 管理クライアント・インターフェースが **PERFORM SET** 関数の呼び出しを試みたものの、失敗したことを示しています。
2. **api_response1_alt** 属性と **api_response2_alt** 属性から失敗の理由を判別します。
これらの値は CICSplex SM EYUDA の応答コードと理由コードです。
EYUDA 応答値は TABLEERROR であり、EYUDA 理由値は DATAERROR です。
3. <resultsummary> タグの **api_function** 属性の情報と <feedback> エレメントの **action** 属性の情報を組み合わせることにより、失敗した API 呼び出しを判別します。
この情報を使用すると、CICS 管理クライアント・インターフェースが CICSplex SM API コマンド **EXEC CPSM PERFORM SET ACTION('DISCARD')** を使用してファイルを削除しようとしたときに失敗したことが分かります。
4. <feedback> エレメントの **eibfn_alt** 属性から、関連する EXEC CICS コマンドを識別します。
eibfn_alt 値は、コマンド **EXEC CICS DISCARD FILE** を表します。
5. <feedback> エレメントの **resp** 属性と **resp2** 属性を使用して、失敗の正確な原因を判別します。

どちらの値も 10 進数形式で示されます。

- a) CICS **resp** 値 16 のシンボル名を調べます。

EIBREP 値のリストについては、[EIB フィールド](#)を参照してください。

このリストから、値 16 を持つ EIBRESP フィールドのシンボル名が INVREQ であることが分かります。

- b) 最後に、**EXEC CICS DISCARD FILE** コマンドの、**resp2** 値が 2 である INVREQ 応答の意味を調べます。

INVREQ 条件の RESP2 値のリストについては、[DISCARD FILE](#) を参照してください。

このリストから、RESP2 値が 2 である INVREQ 状態を返す **EXEC CICS DISCARD FILE** コマンドの失敗の原因が「ファイルが閉じられていない」ことであることが分かります。

例: CICS 管理クライアント・インターフェースのインストール・エラーの診断

CICS 管理クライアント・インターフェースを使用して BAS リソースをインストールするには、INSTALL アクションで PUT 要求を使用します。インストールが失敗すると、CICSplex SM は、診断情報を含む BINSTERR レコードを生成します。このレコードは <installerror> エレメントで返されます。このエレメントは <feedback> の子エレメントです。

このタスクについて

URIMAP_A という名前の URIMAP リソースを、以下の URI および XML 本体を持つ PUT 要求を使用して領域 DEVRGN1 にインストールしようとしてしました。

```
http://wmvs2c.mycorp.com:27640/CICSSystemManagement/CICSDefinitionURIMap/IBEUR912?
CRITERIA=name%3DURIMAP_A
```

```
<request>
  <action name="INSTALL">
    <parameter name="TARGET" value="DEVRGN1"/>
    <parameter name="USAGE" value="LOCAL"/>
  </action>
</request>
```

CICS 管理クライアント・インターフェースからは要求が正しく形成されたことを示す HTTP/1.1 200 OK 応答を受け取りますが、URIMAP はインストールされていません。応答には、以下の情報が含まれます。

```
<resultsummary api_function="PERFORM SET" api_response1="1038" api_response1_alt="TABLEERROR"
api_response2="1361" api_response2_alt="DATAERROR" recordcount="1"/>
<errors>
  <feedback action="INSTALL" attribute1="NAME" errorcode="31"> <installerror eibfn="3036"
    eibfn_alt="CREATE URIMAP" errorcode="4" eyu_cicsname="IBWUIA" resourcename="URIMAP_A"
    resourceversion="1" resp="16" resp2="500"/>
  </feedback>
</errors>
```

この問題が発生した理由を調べるには、以下のステップを実行します。

手順

1. **api_function** 属性からエラーの原因を判別します。
この属性は、CICS 管理クライアント・インターフェースが PERFORM SET 関数の呼び出しを試みたものの、失敗したことを示しています。
2. **api_response1_alt** 属性と **api_response2_alt** 属性から失敗の理由を判別します。
これらの値は CICSplex SM EYUDA の応答コードと理由コードです。
EYUDA 応答値は TABLEERROR であり、EYUDA 理由値は DATAERROR です。
3. <resultsummary> タグの **api_function** 属性の情報と <feedback> エレメントの **action** 属性の情報を組み合わせることにより、失敗した API 呼び出しを判別します。
この情報を使用すると、CICS 管理クライアント・インターフェースが CICSplex SM API コマンド **EXEC CICS PERFORM SET ACTION('INSTALL')** を試行したときに失敗したことが分かります。
4. <installerror> エレメントの **eibfn_alt** 属性から、関連する EXEC CICS コマンドを識別します。

eibfn_alt は、コマンド **EXEC CICS CREATE URIMAP** を表します。

5. <installerror> エレメントの **errorcode** 属性の値を使用して、CICSplex SM BINSTERR エラー・コードを判別します。

エラー・コードとその意味のリストについては、[BINSTERR Resource Table](#) を参照してください。この場合、**errorcode** 値 4 は (INSTFAIL) - Install failure です。

6. <installerror> エレメントの **resp** 属性と **resp2** 属性を使用して、失敗の正確な原因を判別します。どちらの値も 10 進数形式で示されます。

- a) CICS **resp** 値 16 のシンボル名を調べます。

EIBRESP 値のリストについては、[EIB フィールド](#) を参照してください。

このリストから、値 16 を持つ EIBRESP フィールドのシンボル名が INVREQ であることが分かります。

- b) 最後に、**EXEC CICS CREATE URIMAP** コマンドの、**resp2** 値が 500 である INVREQ 応答の意味を調べます。

EXEC CICS CREATE コマンドの RESP2 値のリストについては、[CREATE コマンド](#) と [CSD コマンド](#) の RESP2 値を参照してください。

CICS 管理クライアント・インターフェースのエラー・メッセージ

CICS 管理クライアント・インターフェース (CMCI) は、CMCI 要求の状況を示すため、DFHWU4001 から DFHWU5002 までの範囲のメッセージ ID を持つ XML メッセージを発行します。これらのメッセージは、HTTP 応答コードと関連しており、失敗の原因を理解する上で役立つ説明およびエラー情報が伴います。

CMCI メッセージに標準的な例を以下に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<error message_id="DFHWU4003" connect_version="0410">
<title>400 CICS management client interface HTTP Error</title>
<short>An error has occurred in the CICS management client interface. The request cannot be
processed.</short>
<full> An unknown query parameter was specified in the URI.</full>
<errorInfo name="ccc">ccc</errorInfo>
</error>
```

このメッセージは、メッセージ ID、タイトル、短い説明と長い説明、および特定のエラー情報で構成されています。この場合、<errorInfo> エレメントの情報は URI の不明な照会パラメーターを識別します。

DFHWU4001 - DFHWU4029、HTTP 400 エラー

HTTP 400 応答コードは、サーバーで要求を解釈できなかったことを表します。すべての HTTP 400 応答には、以下の情報が含まれます。

400 CICS 管理クライアント・インターフェースのエラー

CICS 管理クライアント・インターフェースでエラーが発生しました。要求を処理できません。
full explanation

ここで、*full explanation* は以下のいずれかです。

- URI でリソース名が指定されていませんでした。
- 指定された URI に、許容最大長の 256 バイトを超えるパスが含まれています。
- URI で不明な照会パラメーターが指定されていました。
- URI からコンテキストが欠落していました。
- URI から結果キャッシュ・トークンが欠落していました。
- HTTP 要求の本体が欠落していました。
- HTTP 要求の本体が正しく指定されていませんでした。
- URI で指定されたレコード・インデックスが、無効です。
- URI で指定されたレコード・カウントが、無効です。

- キャッシュされていない結果に対して、レコード・インデックスが指定されていました。
- 無関係なデータが URI の末尾で検出されました。
- 複数の CRITERIA 式が URI で検出されました。
- 複数の PARAMETER 式が URI で検出されました。
- 複数の NODISCARD 式が URI で検出されました。
- 複数の SUMMONLY 式が URI で検出されました。
- CRITERIA は、結果キャッシュ操作で有効ではありません。
- PARAMETER は、結果キャッシュ操作で有効ではありません。
- URI からリソース名が欠落していました。
- 指定された属性は、このリソースで有効ではありませんでした。
- DEFVER 属性が、指定されていなかったか、ゼロの値に指定されていました。
- 指定された属性の値が、範囲外または無効でした。
- NODISCARD は、HTTP GET 要求でのみ有効です。
- CRITERIA は、HTTP POST 要求で有効ではありません。
- PARAMETER は、HTTP POST 要求で有効ではありません。
- 指定された結果キャッシュ・トークンは、許容最大長を超えていました。
- 無効な HTTP 本体で、アクションが指定されました。

DFHWU4030、HTTP 401 エラー

401 Basic Authentication Error

サーバーでの認証に失敗しました。

ユーザー名またはパスワード、あるいはその両方が必要ですが、欠落しているか、または正しくありません。

DFHWU4300、HTTP 403 エラー

403 CICS management client interface Access Forbidden (403 CICS 管理クライアント・インターフェースのアクセス禁止)

CICS 管理クライアント・インターフェースでエラーが発生しました。指定された結果キャッシュへのアクセスが拒否されました。

URI で指定された結果キャッシュ・トークンは、要求をしたユーザーに属していません。

DFHWU4400 - DFHWU4402、HTTP 404 エラー

404 CICS 管理クライアント・インターフェース・リソース未検出

CICS 管理クライアント・インターフェースでエラーが発生しました。URI で指定されたリソースを検出できませんでした。

リソース・タイプが、このバージョンの CICS 管理クライアント・インターフェースでサポートされていません。

404 CICS 管理クライアント・インターフェース 結果キャッシュ・レコード未検出

CICS 管理クライアント・インターフェースでエラーが発生しました。指定された結果キャッシュ・レコードを検出できませんでした。

結果キャッシュ・レコードのインデックスが範囲外でした。

404 CICS 管理クライアント・インターフェース 結果キャッシュ未検出

CICS 管理クライアント・インターフェースでエラーが発生しました。指定された結果キャッシュを検出できませんでした。

結果キャッシュ・トークンを検出できませんでした。

DFHWU4500、HTTP 405 エラー

405 CICS management client interface Method (DELETE|GET|POST|PUT|HEAD|Unknown) Not Allowed (405 CICS 管理クライアント・インターフェース・メソッド (DELETE|GET|POST|PUT|HEAD|Unknown) 不可)

CICS 管理クライアント・インターフェースでエラーが発生しました。指定された HTTP メソッドはその URI に対しては許可されていません。

CICS 管理クライアント・インターフェースに送信された URI に無効なメソッドが指定されました。

DFHWU5000 - DFHWU5002、HTTP 500 エラー

500 システム管理クライアント内部サーバー・エラー

CICS 管理クライアント・インターフェースで内部エラーが発生しました。
システム管理者に連絡してください。

500 CICS management client interface Short On Storage (500 CICS 管理クライアント・インターフェースのストレージ不足)

CICS 管理クライアント・インターフェース・サーバーが、ストレージ不足 (2 GB 境界より下) になりました。

システム管理者に連絡してください。

500 CICS management client interface Short On Storage (500 CICS 管理クライアント・インターフェースのストレージ不足)

要求を完了するために使用できる GCDSA ストレージが不足しています。

システム管理者に連絡してください。

CMCI を使用したシステム初期設定パラメーターの照会

システム初期設定パラメーターとその指定変更に関する情報を見つけるには、CICSSystemParameter 外部リソースで動作する CMCI GET 要求を使用できます。

始める前に

CMCI との接続が必要です。詳しくは、[CMCI のセットアップ](#)を参照してください。

取り出すシステム初期設定パラメーターが CICS 始動時に CICS 領域に有効であったことを確認します。取り出す操作は、無効なパラメーター値が始動時にコンソールから修正されていれば、矛盾することなく実行可能です。一部のパラメーターでは修正された値が表示されますが、元のシステム初期設定テーブル値が表示されるものもあります。

このタスクについて

CMCI 要求は、システム初期設定テーブルのパラメーターまたはシステム初期設定テーブルの指定変更、あるいはその両方を組み合わせて取り出すことができます。

他の多くの CICSplex SM 操作の場合と同様に、コンテキストおよびスコープを指定することによって、どの CICS 領域で取り出し操作を機能させるかを制御できます。

システム初期設定パラメーターとその指定変更に関する情報を取り出すには、以下のステップを実行します。

手順

1. HTTP GET メソッドを使用し、またリソース名として CICSSystemParameter を使用して CMCI 要求を構成します。

- a) **context** パラメーターおよび **scope** パラメーターを使用して、要求をどの CICS 領域で機能させる予定であるかを指定します。
- b) URI で、パラメーター **PARMSRCE** および **PARMTYPE** を組み入れたパラメーター式を使用して、どのパラメーターまたは指定変更を取り出すかを指定します。

これらのパラメーターは両方とも必須です。**PARMTYPE** には、システム初期設定テーブルの値を指定する必要があります。**PARMSRCE** には、以下のオプションの 1 つを指定します。

COMBINED

元のシステム初期設定テーブル定義と適用されたパラメーター指定変更の組み合わせ

CONSOLE

システム・コンソールで始動に指定されたとおりの指定変更パラメーター

JCL

JCL EXEC PGM ステートメントにより提供される指定変更パラメーター

SYSIN

SYSIN データ・セットに定義されている始動ジョブ・ストリームからの指定変更パラメーター

TABLE

DFHSITxx ロード・モジュールから抽出される元のシステム初期設定テーブル値

GET 要求について詳しくは、[CICS 管理クライアント・インターフェース GET 要求](#)を参照してください。

2. CMCI 要求を実行依頼します。

パラメーターおよび指定変更値は、ソース、例えば、SYSIN データ・セットにおける順序に関係なく、SYSPARM リソース・テーブルの KEYWORD 属性の順序で返されます。

この動作のため、指定変更パラメーターの組み合わせ **SPCTR** と **SPCTRxx**、また **STNTR** と **STNTRxx** の結果が予想しないものになる場合があります。その理由は、これらの対の指定順序が重要であるためです。例えば、SPCTR=ALL の後に SPCTR=OFF を指定した場合、SPCTR トレースは使用不可です。

が、先に SPCTR=OFF を指定してその後に SPCTRAP=ALL を指定した場合、SPCTRAP トレースは使用可能です。ただし、SYSPARM リソースに対する要求は常に **SPCTR** を **SPCTR_{xx}** より先に返します。

例

- この要求は、CICSplex MYPLEX における領域 REGION に対して設定されている制御テーブル接尾部の値を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>/<REGION>?PARAMETER=PARMSRCE (COMBINED)
%20PARMTYPE (SIT) & CRITERIA=KEYWORD%3D++T%20AND%20NOT%20KEYWORD%3DMXT
```

この要求によって返されるのは、例えば、以下のような結果の要約といくつかの一致レコードです。

```
<resultsummary api_response1="1024" api_response2="0" api_response1_alt="OK" api_response2_alt=""
recordcount="10" displayed_recordcount="10" />
<records>
<cicssystemparameter_keydata="C3D3E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="CLT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
<cicssystemparameter
_keydata="C6C3E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="FCT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
<cicssystemparameter_keydata="D4C3E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="MCT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
<cicssystemparameter_keydata="D4E7E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="MXT" segnum="1" segtot="1"
source="SYSIN"
totallen="3" type="SIT" value="100" valuelen="3" />
<cicssystemparameter_keydata="D9E2E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="RST" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
<cicssystemparameter
_keydata="E2C9E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="SIT" segnum="1" segtot="1"
source="SYSIN"
totallen="2" type="SIT" value="EU" valuelen="2" />
<cicssystemparameter_keydata="E2D9E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="SRT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="1$" valuelen="2" />
<cicssystemparameter_keydata="E3C3E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="TCT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="DY" valuelen="2" />
<cicssystemparameter_keydata="E3E2E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="TST" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
<cicssystemparameter_keydata="E7D3E3404040404040404040404040404040404040404040404040404040404040404040404040404040404040404040404000000001"
eyu_cicsname="WLW26W1" eyu_cicsrel="E660" eyu_reserved="0" keyword="XLT" segnum="1" segtot="1"
source="TABLE"
totallen="2" type="SIT" value="N0" valuelen="2" />
</records>
</response>
```

- 以下の要求は、CICsplex MYPLEX におけるすべての領域について、**MXT** (MAX TASKS) システム 初期設定パラメーターに対して設定されている値を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>?PARAMETER=PARMSRCE (COMBINED)
%20PARMTYPE (SIT) &CRITERIA=KEYWORD%3DMXT
```

- 以下の要求は、CICSplex MYPLEX における領域 REGION に対して設定されている制御テーブル接尾部の値を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>/<REGION>?PARAMETER=PARMSRCE (COMBINED)
%20PARMTYPE (SIT) & CRITERIA=KEYWORD%3D++T%20AND%20NOT%20KEYWORD%3DMXT
```


- 以下の要求は、デフォルトのランナウェイ・タスク時間間隔をオフにした CICSplex MYPLEX における領域の名前を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>?PARAMETER=PARMSRCE(COMBINED)%20PARMTYPE(SIT)&CRITERIA=KEYWORD%3DICVR%20AND%20VALUE%3D0
```

- 以下の要求は、SYSIN 指定変更によって CICSplex MYPLEX における領域 REGION に対して適用された指定変更を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>/<REGION>?PARAMETER=PARMSRCE(SYSIN)%20PARMTYPE(SIT)
```

- 以下の要求は、JCL 指定変更によって CICSplex MYPLEX における領域 REGION に対して適用された指定変更を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>/<REGION>?PARAMETER=PARMSRCE(JCL)%20PARMTYPE(SIT)
```


リソース定義との相互作用

CICS 管理クライアント・インターフェース (CMCI) を使用して BAS リソース定義と CSD リソース定義に対して作成、更新、削除、およびアクションの実行を行います。

このタスクについて

リソース定義との相互作用は、以下の方法で CICSplex SM から行えます。

- CMCI の使用 (アプリケーションからの HTTP 要求を受け入れます)
- CICS Explorer の使用 (CMCI を利用します)
- CICSplex SM Web ユーザー・インターフェース (WUI) の使用
- アプリケーションに対する CICSplex SM API コマンドの直接の追加

CMCI を使用したリソース定義の作成

CICS 管理クライアント・インターフェース (CMCI) を使用して BAS リソースまたは CSD リソース定義を作成します。

始める前に

CMCI との接続が必要です。詳しくは、[CMCI のセットアップ](#)を参照してください。

このタスクについて

CMCI を使用してリソース定義を作成するには、以下のステップを実行します。

手順

1. HTTP POST メソッドによる要求を使用して、リソース定義を作成します。
POST 要求について詳しくは、[CICS\(r\) 管理クライアント・インターフェース POST 要求](#)を参照してください。
2. 要求の URI を指定します。
 - BAS リソースを作成する場合に *SCOPE* を URI に指定すると、それは無視されます。
 - CSD リソースを作成する場合には、*SCOPE* を URI に指定します。*SCOPE* は、その中にリソースを作成する CICS システムでなければなりません。POST 要求の URI の指定について詳しくは、[CICS 管理クライアント・インターフェース POST 要求 URI](#)を参照してください。
3. リソースに対するパラメーターおよび属性が入った XML 本体を追加します。
XML 本体は、<request> エレメントの中に <create> エレメントが入って構成されます。
 - BAS リソースを作成する場合は、*defver* 属性を XML 本体に含める必要があります。
 - CSD リソースを作成する場合は、CSD パラメーターおよび CSD グループ属性を XML 本体に含める必要があります。CSD リソースを作成するときには *defver* 属性を含めないでください。

例

以下の例は、BAS トランザクション定義を作成するために何を CMCI に送信するかを示しています。

```
POST http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/

<request>
  <create>
    <attributes name="NAME" defver="1" program="test"/>
```

```
</create>
</request>
```

次の例は、CSD トランザクション定義を作成するために何を CMCI に送信するかを示しています。CSD パラメーターおよび csdgroup 属性が XML に追加されていること、および defver 属性が除外されていることを注意してください。

```
POST http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/<SCOPE>/

<request>
  <create>
    <parameter name="CSD"/>
    <attributes name="resourceName" program="test" csdgroup="CSDGR"/>
  </create>
</request>
```

両方の例とも、つまり BAS リソース定義の作成でも CSD リソース定義の作成でも、正常に完了すると似たような出力が返されます。正常に作成されると、api_response1="1024" および api_response1_alt="OK" を含む以下のような CICSplex SM API 応答メッセージが返されます。

```
<response xmlns:...>
  <resultsummary api_response1="1024" api_response1_alt="OK".. />
  <records>
    ...
  </records>
</response>
```

CMCI を使用したリソース定義の更新

CICS 管理クライアント・インターフェース (CMCI) を使用して BAS リソースまたは CSD リソース定義を更新します。

始める前に

CMCI との接続が必要です。詳しくは、[CMCI のセットアップ](#)を参照してください。

このタスクについて

CMCI を使用してリソース定義を更新するには、以下のステップを実行します。

手順

1. PUT 要求を使用してリソース定義を更新します。
PUT 要求について詳しくは、[13 ページの『CMCI PUT 要求数』](#)を参照してください。
2. 要求の URI を指定します。
 - BAS リソースを更新する場合に *SCOPE* を URI に指定すると、それは無視されます。
 - CSD リソースを更新する場合には、*SCOPE* および *PARAMETER=CSDGROUP(value)* を URI に指定します。

PUT 要求の URI の指定について詳しくは、[15 ページの『CMCI PUT 要求 URI』](#)を参照してください。
3. リソースに対するパラメーターおよび属性が入った XML 本体を追加します。
XML 本体は、<request> エレメントの中に <update> エレメントが入って構成されます。
 - BAS リソースを更新する場合は、defver 属性を XML 本体に含める必要があります。
 - CSD リソースを更新する場合は、CSD グループ属性を XML 本体に含める必要があります。CSD リソースを更新するときには defver 属性を含めないでください。

例

以下の例は、BAS トランザクション定義を更新するために何を指定するかを示しています。この例では、トランザクション定義 TRN1 のプログラム名の値が PRGM2 に更新されます。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/
?CRITERIA=NAME%3TRN1

<request>
  <update>
    <attributes program="PRGM2" />
  </update>
</request>
```

以下の例は、CSD トランザクション定義に対する更新要求を示しています。この例では、リソース・グループ GRP1 の中の TRN1 というプログラム名が PRGM2 に更新されます。PARAMETER=CSDGROUP が URI に追加されていること、および defver 属性が除外されていることに注意してください。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/<SCOPE>/
?CRITERIA=NAME%3DAAAA&PARAMETER=CSDGROUP%28GRP1%29

<request>
  <update>
    <attributes program="PRGM2" />
  </update>
</request>
```

両方の例とも、つまりリソース定義の更新でも CSD リソース定義の更新でも、正常に完了すると似たような出力が返されます。正常に更新されると、api_response1="1024" および api_response1_alt="OK" を含む以下のような CICSplex SM API 応答メッセージが返されます。

```
<response xmlns:...>
  <resultsummary api_response1="1024" api_response1_alt="OK".. />
  <records>
    ...
  </records>
</response>
```

CMCI を使用したリソース定義に対するアクションの実行

CICS 管理クライアント・インターフェース (CMCI) を使用して BAS リソースまたは CSD リソース定義に対してアクションを実行します。

始める前に

CMCI との接続が必要です。詳しくは、[CMCI のセットアップ](#)を参照してください。

このタスクについて

CMCI を使用してリソース定義に対してアクションを実行するには、以下のステップを実行します。

手順

1. PUT 要求を使用してリソース定義に対してアクションを実行します。
HTTP PUT 要求の使用について詳しくは、[13 ページの『CMCI PUT 要求数』](#)を参照してください。
2. 要求の URI を指定します。
CSD リソースに対してアクションを実行する場合には、SCOPE および PARAMETER=CSDGROUP を URI に指定します。
PUT 要求の URI の指定について詳しくは、[15 ページの『CMCI PUT 要求 URI』](#)を参照してください。
3. リソースに対するパラメーターおよび属性が入った XML 本体を追加します。
XML 本体は、<request> エレメントの中に <action name="action_name"> エレメントが入って構成されます。

アクションにパラメーターが必要な場合には、それらを <action> エレメントの中の <parameter> エレメントに指定する必要があります。

4. 実行するアクションを指定します。

各リソース・テーブルに、実行できるアクションのリストが含まれます。

例

以下の例は、名前が AA で始まる、スコープ中のすべての BAS トランザクション定義に適用される INSTALL 要求を示しています。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/
?CRITERIA=NAME%3DAA*

<request>
  <action name="INSTALL">
    <parameter name="FORCEINS" value="NO"/>
    <parameter name="USAGE" value="LOCAL"/>
  </action>
</request>
```

以下の例は、CSDINSTALL アクション要求に対して何を指定するかを示しています。CSD リソースに対するアクションに指定できるパラメーターについては、コマンド説明にリストされています。

```
PUT http://exampledomain.com:12345/CICSSystemManagement/
CICSDefinitionTransaction/<CONTEXT>/<SCOPE>/
?CRITERIA=NAME%3DAA*&PARAMETER=CSDGROUP%28group%29

<request>
  <action name="CSDINSTALL">
  </action>
</request>
```

CMCI を使用したリソース定義の削除

CICS 管理クライアント・インターフェース (CMCI) を使用して BAS リソースまたは CSD リソース定義を削除します。

始める前に

CMCI との接続が必要です。詳しくは、[CMCI のセットアップ](#)を参照してください。

このタスクについて

CMCI を使用してリソース定義を削除するには、以下のステップを実行します。

手順

1. DELETE 要求を使用してリソース定義を削除します。

DELETE 要求について詳しくは、[3 ページの『CMCI DELETE 要求数』](#)を参照してください。

2. 要求の URI を指定します。

CSD リソースを削除する場合は、スタンドアロン CICS 領域での処理でなければ、SCOPE を URI に指定する必要があります。

DELETE 要求の URI の指定について詳しくは、[4 ページの『CMCI DELETE 要求 URI』](#)を参照してください。

3. パラメーターを持つリソースを削除する場合には、そのリソースに対するパラメーターを URI に指定する必要があります。

例えば、CSD リソースを削除したければ、CSDGROUP パラメーターを URI PARAMETER=CSDGROUP(*group*) に指定しなければなりません。

4. オプション: 要求の詳細を XML 本体に指定することもできます。

これらの詳細は削除要求を実行するために XML 本体に必要なものでないことに注意してください。

XML 本体は、<request> エlement の中に <delete> Element が入って構成されます。

すべてのクライアントが XML 本体を解析できるわけではないことに注意してください。そのため、どのリソースを削除するかを指定するためのパラメーターを URI に追加しなければなりません。

例

以下の例は、BAS トランザクション定義を削除するために何を指定するかを示しています。この例では、"AAAA" という名前のすべてのリソースが削除されます。

```
DELETE http://exampledomain.com:12345/CICSSystemManagement/  
CICSDefinitionTransaction/?CRITERIA=NAME%3DAAAA
```

CSD リソースに対する削除要求に XML 本体は必要ありません。

```
DELETE http://exampledomain.com:12345/CICSSystemManagement/  
<RESOURCE_NAME>/<CONTEXT>/<SCOPE>/  
?CRITERIA=NAME%3Dname&PARAMETER=CSDGROUP%28group%29
```

XML 本体を含む CSD リソースに対する次の削除要求は、前の要求と等価です。

```
DELETE http://exampledomain.com:12345/CICSSystemManagement/  
<RESOURCE_NAME>/<CONTEXT>/<SCOPE>/  
?CRITERIA=NAME%3Dname&PARAMETER=CSDGROUP%28group%29
```

```
<request>  
  <delete>  
    <parameter name="CSD"/>  
  </delete>  
</request>
```


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渉外

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](#)
- [Developing system programs](#)
- [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。

- [Troubleshooting and support](#)
- [CICS TS 診断リファレンス](#)

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
- Java Applications in CICS

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[®] 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。

Apache、Apache Axis2、Apache Maven、Apache Ivy、Apache Software Foundation (ASF) ロゴ、および ASF feather ロゴは、Apache Software Foundation の商標です。

Gradle および Gradlephant ロゴは、Gradle, Inc. およびその子会社の米国およびその他の国における登録商標です。

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。

登録商標 Linux[®] は、世界中で商標の所有者である Linux Torvalds の独占的ライセンシーである Linux Foundation のサブライセンスに従って使用されています。

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。

Red Hat[®]、および Hibernate[®] は、Red Hat, Inc. またはその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。

Spring Boot は、Pivotal Software, Inc. 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。

Zowe[™]、Zowe ロゴ、および Open Mainframe Project[™] は、Linux Foundation の商標です。

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を提供され、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 プライバシー・ポリシー』](#) および [『IBM オンラインでのプライバシー・ステートメント』](#) の『クッキー、Web ビーコン、その他のテクノロジー』および [『IBM ソフトウェア製品および Software as a Service のプライバシー・ステートメント』](#) 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]

アプリケーション・プログラミング・インターフェース
CICS 管理クライアント・インターフェース [1](#)
エスケープ・シーケンス
CICS 管理クライアント・インターフェースにおける [41](#)
エラー
CICS 管理クライアント・インターフェースにおける [43](#)

[カ行]

ガーベッジ・コレクション
CMCI [9](#)

C

CICS 管理クライアント・インターフェース
エスケープ・シーケンス [41](#)
ガーベッジ・コレクション [9](#)
概要 [1](#)
制限付き文字 [41](#)
問題判別 [43-45](#)
要求 [3](#)
DELETE メソッド [3](#)
DELETE 要求 URI [4](#)
errors XML エlement [35](#)
feedback XML エlement [35](#)
FEEDBACK レコード [44, 45](#)
GET メソッド [5, 9](#)
GET 要求 URI [7](#)
inconsistentscope XML エlement [37](#)
inconsistentset XML エlement [38](#)
installerror XML エlement [36](#)
POST メソッド [11](#)
POST 要求 URI [12](#)
put XML エlement [28](#)
PUT メソッド [13](#)
PUT 要求 URI [15](#)
request XML エlement [27](#)
response XML エlement [32, 34](#)
resultsummary エラー [43](#)
URI リソース名
CICS 管理クライアント・インターフェース [17](#)
XML スキーマ [27](#)
CICSSystemParameter 外部リソース [49](#)
CMCI
ガーベッジ・コレクション [9](#)
SIT パラメーター [49](#)
XML [27](#)

D

DELETE 要求
CICS 管理クライアント・インターフェースにおける [3](#)

DELETE 要求 URI
CICS 管理クライアント・インターフェースにおける [4, 7](#)
DFHWU4001 [46](#)
DFHWU4002 [46](#)
DFHWU4003 [46](#)
DFHWU4004 [46](#)
DFHWU4005 [46](#)
DFHWU4006 [46](#)
DFHWU4007 [46](#)
DFHWU4008 [46](#)
DFHWU4009 [46](#)
DFHWU4010 [46](#)
DFHWU4011 [46](#)
DFHWU4012 [46](#)
DFHWU4013 [46](#)
DFHWU4014 [46](#)
DFHWU4015 [46](#)
DFHWU4016 [46](#)
DFHWU4017 [46](#)
DFHWU4018 [46](#)
DFHWU4019 [46](#)
DFHWU4020 [46](#)
DFHWU4021 [46](#)
DFHWU4022 [46](#)
DFHWU4023 [46](#)
DFHWU4024 [46](#)
DFHWU4025 [46](#)
DFHWU4026 [46](#)
DFHWU4027 [46](#)
DFHWU4028 [46](#)
DFHWU4029 [46](#)
DFHWU4030 [46](#)
DFHWU4401 [46](#)
DFHWU5001 [46](#)

E

errors XML エlement
CICS 管理クライアント・インターフェースにおける [35](#)

F

feedback XML エlement
CICS 管理クライアント・インターフェースにおける [35](#)
FEEDBACK レコード
CICS 管理クライアント・インターフェース [44, 45](#)

G

GET requests (GET 要求)
例 [9](#)
CICS 管理クライアント・インターフェースにおける [5](#)

I

inconsistentscope XML エlement
CICS 管理クライアント・インターフェースにおける [37](#)
inconsistentset XML エlement
CICS 管理クライアント・インターフェースにおける [38](#)
installerror XML エlement
CICS 管理クライアント・インターフェースにおける [36](#)

<delete> (続き)
CMCI
 <delete> [56](#)
<update>
 リソース定義
 <update> [54](#)
CMCI
 <update> [54](#)

N

NODISCARD オプション
GET 要求での [9](#)

P

POST 要求
CICS 管理クライアント・インターフェースにおける [11](#)
POST 要求 URI
CICS 管理クライアント・インターフェースにおける [12](#)
put XML エlement
CICS 管理クライアント・インターフェースにおける [28](#)
PUT 要求
CICS 管理クライアント・インターフェースにおける [13](#)
PUT 要求 URI
CICS 管理クライアント・インターフェースにおける [15](#)

R

Representational State Transfer [1](#)
request XML エlement
CICS 管理クライアント・インターフェースにおける [27](#)
response XML エlement
CICS 管理クライアント・インターフェースにおける [32](#),
[34](#)
REST API [1](#)

S

SIT パラメーター
CMCI [49](#)

X

XML
CICS 管理クライアント・インターフェースにおける [27](#)

[特殊文字]

<action>
 リソース定義
 <action> [55](#)
CMCI
 <action> [55](#)
<create>
 リソース定義
 <create> [53](#)
CMCI
 <create> [53](#)
<delete>
 リソース定義
 <delete> [56](#)

