

z/OS용 CICS Transaction
Server버전 5 릴리스 6

CICSplex SM CMCI REST API 참조서



참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [제품 법적 주의사항](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한 IBM® z/OS®용 CICS® Transaction Server, 버전 5 릴리스 6 (제품 번호 5655-Y305655-BTA) 및 모든 후속 릴리스와 수정에 적용됩니다.

© Copyright International Business Machines Corporation 1974, 2020.

목차

이 PDF 정보.....	v
제 1 장 CMCI REST API 개요.....	1
CMCI 요청.....	3
CMCI DELETE 요청.....	3
CMCI DELETE 요청 URI.....	4
CMCI GET 요청.....	5
CMCI GET 요청 URI.....	6
보유된 결과 세트를 사용하여 GET 요청 성능 향상.....	9
CMCI POST 요청.....	10
CMCI POST 요청 URI.....	12
CMCI PUT 요청.....	12
CMCI PUT 요청 URI.....	14
CMCI 자원 이름.....	17
CMCI XML.....	27
<request> 요소.....	27
<create> 요소.....	27
<update> 요소.....	28
<action> 요소.....	29
<delete> 요소.....	30
<parameter> 요소.....	30
<attributes> 요소.....	31
<response> 요소.....	31
<resultsummary> 요소.....	32
<records> 요소.....	33
<errors> 요소.....	34
<feedback> 요소.....	34
<installerror> 요소.....	36
<inconsistentscope> 요소.....	36
<inconsistentset> 요소.....	37
CMCI 이스케이프 시퀀스.....	39
CMCI 문제점 판별.....	41
예제: <resultsummary> 정보를 사용하여 CICS 관리 클라이언트 인터페이스 오류 진단.....	41
예제: <feedback> 정보를 사용하여 CICS 관리 클라이언트 인터페이스 오류 진단.....	42
예제: CICS 관리 클라이언트 인터페이스 설치 오류 진단.....	43
CICS 관리 클라이언트 인터페이스 오류 메시지.....	44
CMCI를 사용하여 시스템 초기화 매개변수 조회.....	47
자원 정의와 상호작용.....	51
CMCI를 사용하여 자원 정의 작성.....	51
CMCI를 사용하여 자원 정의 업데이트.....	52
CMCI를 사용하여 자원 정의에서 조치 수행.....	53
CMCI를 사용하여 자원 정의 삭제.....	54
주의사항	57
색인.....	61

이 PDF 정보

이 PDF는 z/OS용 CICS Transaction Server의 CICSplex SM 요소를 위한 CMCI REST API(Application Programming Interface)의 참조입니다. 이 문서는 CICSplex SM과 상호작용하기 위해 HTTP 기반 시스템 관리 클라이언트를 작성하는 시스템 프로그래머를 위해 제작되었습니다.

이 문서에서 사용되는 용어와 표기법에 대한 세부사항은 IBM Knowledge Center에서 [CICS 문서에서 사용하는 규칙과 용어](#)의 내용을 참조하십시오.

이 PDF의 날짜

이 PDF는 2020년 5월 28일에 작성되었습니다.

제 1 장 CMCI REST API 개요

CICS 관리 클라이언트 인터페이스(CMCI)는 IBM CICS Explorer® 등의 시스템 관리 클라이언트용 REST API(Application Programming Interface)를 제공합니다.

클라이언트는 CMCI에 대한 HTTP 요청을 시작합니다. 인터페이스에서 요청이 올바르다고 판단하는 경우 이는 CICSplex® SM API 명령을 생성합니다. 또는 독립형 CICS 리전의 경우에는 CICS 시스템 명령을 생성합니다. 명령 실행 후 CMCI는 HTTP 응답을 작성합니다. 요청에 성공하면, 이는 결과 세트가 포함된 XML 피드 및 HTTP 200(OK) 응답의 양식을 취하며 이는 다시 클라이언트에 전달됩니다. 요청에 실패하면, 응답은 실패에 대한 세부 사항과 함께 OK 아닌 HTTP 응답 코드로 구성됩니다.

CMCI HTTP 요청과 응답의 형식은 HTTP/1.1 프로토콜을 기반으로 합니다. 이 프로토콜에 대한 자세한 정보는 [HTTP 프로토콜](#)의 내용을 참조하십시오.

CMCI 요청의 요소

CMCI 요청은 HTTP 헤더, 뒤이어 URI(Universal Resource Identifier), 그리고 해당되는 경우 CICS 또는 CICSplex SM 자원에 대한 변경사항의 세부사항이 포함된 XML 본문이 차례로 나오는 양식을 취합니다.

HTTP 헤더

헤더에는 다음 HTTP 메소드 중 하나가 포함됩니다.

HTTP 메소드	설명
DELETE	CICSplex 데이터 저장소에서 자원을 제거하거나, CSD에서 자원을 제거하거나, 설치된 자원을 버립니다.
GET	CICSplex 데이터 저장소에서 자원에 대한 정보를 검색하거나, CSD에서 자원에 대한 정보를 검색하거나, 설치된 자원에 대한 정보를 검색합니다.
POST	CSD의 자원 또는 CICSplex 데이터 저장소의 자원을 작성합니다.
PUT	CICSplex 데이터 저장소의 기존 자원을 업데이트하거나, CSD의 기존 자원을 업데이트하거나, 속성을 설정하고 설치된 자원에 대한 조치를 수행합니다. 또한 CICSplex SM 및 CSD 자원에 대한 조치를 수행합니다.

URI

URI는 CICS 또는 CICSplex SM 자원의 이름을 포함하며, 지정된 자원 중 하나 이상의 인스턴스를 식별하기 위한 조회의 범위 및 속성을 세분화하는 일련의 매개변수를 지정합니다. GET 요청에서, URI는 API가 결과 세트를 보유하는지 또는 이를 버리는지 여부도 지정합니다. API가 결과를 보유하는 경우, 새 요청은 검색 오버레이션을 반복하지 않고도 보유된 결과에 영향을 줄 수 있습니다. 사용자는 후속 요청을 사용함으로써 한 번에 하나 이상의 레코드를 선택하여 보유된 결과를 페이지 단위로 넘겨 볼 수도 있습니다. 결과 세트 보유에 대한 자세한 정보는 [9 페이지의 『보유된 결과를 사용하여 GET 요청 성능 향상』](#)의 내용을 참조하십시오.

XML 본문

POST 및 PUT 요청에는 XML 본문이 포함됩니다. PUT 요청의 본문에는 자원 속성에 대한 세부 변경사항 또는 대상 자원에 대해 수행되는 조치가 포함됩니다. POST 요청의 본문에는 새 자원 인스턴스에 제공하고자 하는 속성 값이 포함됩니다.

GET 및 DELETE 요청에서는 XML 본문이 필요하지 않습니다. DELETE 요청에 대해 추가 매개변수가 필요한 경우, 해당 매개변수는 URI에 포함되어야 하며 이는 선택적으로 XML 본문에 추가될 수 있습니다. 예를 들어, CSD 자원 정의를 삭제하는 경우 URI에 `PARAMETER=CSDGROUP(csdgrp)`을 포함해야 하며, 선택적으로 XML 본문에 `<parameter name="CSD"/>`를 추가할 수 있습니다.

CMCI 요청

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 설치된 정의 CICS 및 CICSplex SM 자원을 관리하는 HTTP 클라이언트 애플리케이션을 개발할 수 있습니다. CICS 관리 클라이언트 인터페이스 요청은 URI(Universal Resource Identifier) 및 해당되는 경우 XML 본문이 뒤에 나오는 HTTP 헤더의 양식을 취합니다. 4개의 CICS 관리 클라이언트 인터페이스 요청은 DELETE, GET, POST, PUT입니다.

CMCI DELETE 요청

CICS 관리 클라이언트 인터페이스(CMCI)는 자원을 데이터 저장소에서 제거하거나 CICS 또는 CICSplex SM에서 설치된 자원을 버리기 위해 HTTP DELETE 메소드를 사용합니다.

클라이언트는 다음 파트로 구성되는 HTTP 헤더를 사용하여 DELETE 요청을 형성합니다.

- 메소드 이름(이 경우 DELETE)
- 삭제될 자원을 식별하는 URI.
- HTTP 버전
- 필요한 경우 권한 부여 인증서

DELETE 요청은 설치된 자원이나 정의의 자원에서 작동할 수 있습니다. 인터페이스는 작동 중인 자원에 대해 작동할 때 CICS DISCARD 명령을 구성하고, 정의의 자원에 대해 작동할 때 CICSplex SM REMOVE 명령을 구성합니다.

예를 들어, CICSplex PLEX1에서 TR로 시작하는 이름과 P로 시작하는 프로그램이 있는 모든 트랜잭션 정의를 삭제하려는 경우 클라이언트는 다음 HTTP 헤더를 구성합니다.

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

초기 행에는 세 개의 파트가 공백으로 구분되어 있습니다.

- 메소드 이름
- 요청 URI. 이 URI는 /CICSSystemManagement/로 시작하고 바로 뒤에 삭제할 해당 자원 유형의 인스턴스를 판별하는 필터 및 매개변수와 함께 자원을 식별하는 외부 이름이 오는 절대 경로로 지정됩니다. CICS 관리 클라이언트 인터페이스가 처리하기에 적합하도록 URI를 작성하기 위해, 클라이언트는 특정의 제한된 문자(예: 공백 및 별표)를 이스케이프 순서로 바꿉니다.
- CICS 관리 클라이언트 인터페이스에 연결 시 HTTP 버전으로, 항상 HTTP/1.1입니다.

두 번째 행은 URI에 지정된 대상의 호스트 이름 및 포트 번호(콜론으로 구분)를 식별합니다. 이 행은 뒤에 줄 바꿈기가 있는 캐리지 리턴으로 끝납니다.

선택적 세 번째 행에는 권한 부여 인증서가 있습니다. CICS 시스템 초기화 매개변수 **SEC**을 YES로 지정하여 시스템이 실행 중인 경우 기본 인증 헤더에 Base-64로 사용자 ID와 비밀번호를 제공해야 합니다. 이 예제에서 사용자 ID는 *FRED*이고 비밀번호는 *PASSWORD*입니다.

HTTP 헤더는 개별 행에서 최종 캐리지 리턴 및 줄 바꿈기로 종료됩니다.

DELETE 요청에 대한 응답

DELETE 요청 완료 시, 클라이언트는 삭제된 자원의 세부사항 및 결과 요약을 포함하는 XML 피드와 HTTP 헤더로 구성되는 CMCI에서 응답을 수신합니다.

헤더는 HTTP 응답 코드, 날짜 및 시간, 서버와 데이터 유형의 세부사항으로 구성됩니다. 다음 예제는 성공적인 DELETE 요청에 대한 헤더를 표시합니다.

```
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

응답 본문은 요청에 대한 요약 정보를 표시하는 <resultsummary> 요소를 포함하는 <response> 루트 요소로 구성됩니다. 예를 들어, 다음과 같습니다.

```
<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>
```

보안 고려사항

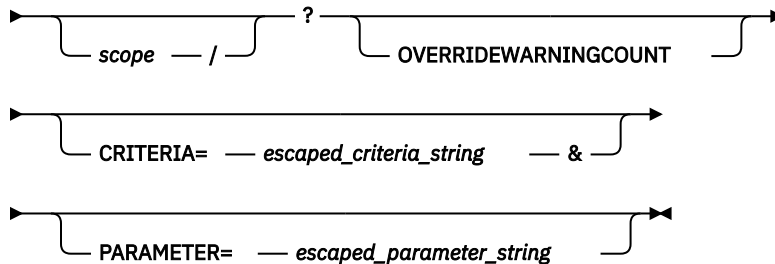
HTTP 메소드를 사용하는 경우 일부 방화벽은 보안 고려사항 때문에 방화벽을 통하여 HTTP PUT 또는 DELETE 트래픽을 허용하지 않는다는 점에 유의하십시오. 이 제한사항을 수용하려면 POST 요청을 통해 DELETE 요청을 터널링하기 위해 CMCI POST 요청을 사용할 수 있습니다.

CMCI DELETE 요청 URI

삭제 요청의 URI는 조작에 의해 삭제될 자원의 인스턴스를 식별합니다.

다음 다이어그램은 DELETE 요청에 대한 URI 구문을 보여줍니다.

➡ / — *CICSSystemManagement* — / — *resource_name* — / — *context* — / ➡



옵션

/CICSSystemManagement

요청이 CICS 관리 클라이언트 인터페이스에 액세스 중임을 표시합니다. CICSSystemManagement는 대소문자를 구분합니다.

resource_name

삭제하는 CICS 또는 CICSplex SM 자원과 연관되는 외부 자원 이름을 지정합니다. 예를 들어, 자원 이름 CICSLocalFile을 지정하면 요청이 CICSplex SM LOCFIL 자원 테이블과 연관되고, CICSResourceAssignmentDefinition을 지정하면 요청이 RASGNDEF 자원 테이블과 연관됩니다. 외부 자원 이름의 전체 목록은 17 페이지의 『CMCI 자원 이름』의 내용을 참조하십시오.

HTTP DELETE 메소드는 모든 자원에 올바르지 않습니다. 이 HTTP 메소드가 특정 자원에 대해 올바른지 판별하려면 CICSplex SM 자원 테이블에 있는 자원 설명을 참조하십시오.

자원 이름은 대소문자를 구분하지 않습니다. 클라이언트가 올바르지 않은 자원 이름을 지정하는 경우, HTTP 404(찾을 수 없음) 응답이 리턴됩니다.

context

CMCI가 CICSplex SM 환경에 설치된 경우 *context*는 요청과 연관되는 CICSplex 또는 CMAS의 이름입니다 (예: PLEX1).

CMCI가 단일 서버로 설치된 경우 *context*는 요청과 연관된 CICS 리전의 애플리케이션 ID입니다.

컨텍스트 값은 공백을 포함하면 안됩니다. 컨텍스트는 대소문자가 구분되지 않습니다.

scope

요청과 연관되는 CICSplex, CICS 그룹, CICS 리전 또는 논리 범위의 이름을 지정합니다. 범위는 컨텍스트의 서브세트로, 요청을 특정 CICS 시스템 또는 자원으로 제한합니다. 범위는 필수가 아닙니다. 범위가 없으면 요청은 컨텍스트의 값만으로 제한됩니다. 컨텍스트는 공백을 포함하면 안 됩니다. 대소문자는 구분되지 않습니다.

OVERRIDEWARNINGCOUNT

경고 개수 한계 메커니즘을 무시하고 기본 경고 개수 한계가 지정되지 않은 경우와 같이 요청을 실행할 수 있습니다. 이 옵션은 CICSplex SM 웹 사용자 인터페이스 초기화 매개변수 **RESOURCELIMIT**가 WARNING으로 설정된 경우에만 유효합니다.

조회 매개변수

요청의 범위와 속성을 세분화합니다. 조회 섹션의 구성 파트는 어떤 순서로도 발생할 수 있지만 각각 URI에서 한 번만 발생할 수 있습니다. 조회 매개변수는 앰퍼샌드(&)로 구분됩니다. 조회 매개변수 값에서 대소문자가 구분되지 않지만, 일부 속성(예: TRANID 및 DESC)이 대소문자 혼용 값을 보유할 수 있기 때문에 특정 속성 값의 대소문자는 올바르게 지정해야 합니다. 조회 매개변수는 다음과 같습니다.

CRITERIA=escaped_criteria_string

요청 시 리턴되는 데이터를 필터하는 논리식 문자열입니다. CRITERIA 매개변수의 값을 구성하는 문자열은 CICSplex SM API(application programming interface)에서 필터 표현식과 같은 규칙을 따릅니다. CICSplex SM API를 사용하는 필터 표현식을 지정하는 방법에 대한 자세한 지침은 [필터 표현식 빌드 방법의 내용을 참조하십시오](#).

특정의 제한된 문자는 이스케이프 시퀀스로 대체하여 서버가 올바르게 해석할 수 있도록 해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

PARAMETER=escaped_parameter_string

요청을 세분화하는 **parameter_name(data_value)** 양식의 하나 이상의 매개변수 및 값 문자열입니다. 이 매개변수를 지정하는 규칙은 CICSplex SM API(Application Programming Interface)에서와 같습니다. CICSplex SM API를 사용하는 매개변수 표현식을 지정하는 방법에 대한 자세한 지침은 [자원에 대한 조치 수행의 내용을 참조하십시오](#).

기준 문자열에서와 같이, 특정 문자열은 이스케이프 문자를 사용하여 인코딩해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

CMCI GET 요청

CICS 관리 클라이언트 인터페이스(CMCI)는 CICS 또는 CICSplex SM에서 자원을 검색하기 위해 HTTP GET 메소드를 사용합니다.

클라이언트는 다음 파트로부터 GET 요청을 형성합니다.

- HTTP 메소드(이 경우 GET)
- 검색할 자원을 식별하고 나중에 사용하거나 버리기 위해 결과가 저장되는지 여부를 표시하는 URI.
- HTTP 버전
- 필요한 경우 권한 부여 인증서

예를 들어, CICSplex PLEX1에서 모든 CICS 로컬 파일을 검색하려면 클라이언트가 다음 요청을 구성할 수 있습니다.

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

HTTP 헤더의 초기 요청 행에는 세 개의 파트가 있으며, 각각은 공백으로 구분되고 뒤에 줄 바꾸기가 있는 캐리지 리턴으로 끝납니다.

- HTTP 메소드 이름
- 요청 URI. 이 URI는 /CICSSystemManagement/로 시작하고 바로 뒤에 검색할 해당 자원 유형의 인스턴스를 판별하는 필터 및 매개변수와 함께 자원을 식별하는 외부 이름이 오는 절대 경로로 지정됩니다.

GET 요청은 CICS 또는 CICSplex SM 자원에서 직접 또는 이전 GET 요청에서 보유된 결과에서 작동할 수 있습니다. 일반적으로 결과는 각 요청의 마지막에서 버리지만, GET 요청 URI에 NODISCARD 옵션을 지정하여 결과 세트를 보유할 수 있습니다. 이 옵션을 사용하는 경우 각 요청 끝에서 CMCI는 모든 결과 세트 상태를 확인하고 15분 이상 사용되지 않은 항목만 버립니다.

요청이 자원에서 직접 작동하는 경우 자원 이름을 지정하십시오. 이는 운영 또는 정의의 자원일 수 있습니다. 그리고 컨텍스트 및 선택적으로 범위를 추가하고, 추가적으로 세분화한 후 하나 이상의 조회 매개변수를 포함하여 결과를 정렬 및 필터링하십시오. GET 요청 URI에서 ORDERBY 매개변수를 지정하여 하나 이상의 속성으로 결과를 정렬하십시오.

요청이 보유된 결과 세트에서 작동하는 경우 자원 이름을 CICSResultCache로 대체하고 캐시 토큰을 지정하여 결과를 식별합니다. **index** 및 **count** 옵션에 대한 값을 추가하여 보유된 결과에서 하나 이상의 레코드로 요청 범위를 좁힐 수 있습니다.

- CICS 관리 클라이언트 인터페이스에 연결 시 HTTP 버전으로, 항상 HTTP/1.1입니다.

두 번째 행은 URI에 지정된 대상의 호스트 이름 및 포트 번호(콜론으로 구분)를 식별합니다. 이 행은 뒤에 줄 바꿈기가 있는 캐리지 리턴으로 끝납니다.

선택적 세 번째 행에는 권한 부여 인증서가 있습니다. CICS 시스템 초기화 매개변수 **SEC**을 YES로 지정하여 시스템이 실행 중인 경우 기본 인증 헤더에 Base-64로 사용자 ID와 비밀번호를 제공해야 합니다. 이 예제에서 사용자 ID는 *FRED*이고 비밀번호는 *PASSWORD*입니다.

HTTP 헤더는 개별 행에서 최종 캐리지 리턴 및 줄 바꿈기로 종료됩니다.

GET 요청에 대한 응답

GET 요청 완료 시, 클라이언트는 HTTP 헤더로 구성되는 CICS 관리 클라이언트 인터페이스로부터의 응답, 결과 요약 포함하는 XML 피드, 그리고 검색된 자원의 세부사항을 수신합니다. 그러나 URI에 SUMMONLY 매개변수가 포함된 경우, 응답은 HTTP 헤더와 결과 요약으로만 구성됩니다.

헤더는 HTTP 응답, 날짜 및 시간, 서버와 데이터 유형의 세부사항으로 구성됩니다. 다음 예제는 성공적인 GET 요청에 대한 헤더를 보여줍니다.

```
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
```

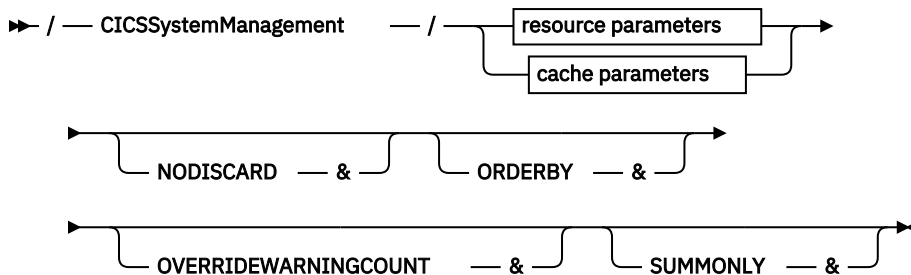
응답 본문은 요청에 대한 요약된 정보를 표시하는 <resultsummary> 요소를 포함하는 <response> 루트 요소와, 일반적으로 검색된 자원의 세부사항을 포함하는 <records> 요소로 구성됩니다. URI에서 **SUMMONLY** 매개변수를 지정하여 응답의 <records> 요소를 비활성화할 수 있습니다. 요청이 오류로 완료되면, 진단 정보는 <errors> 요소에서 제공됩니다.

CMCI GET 요청 URI

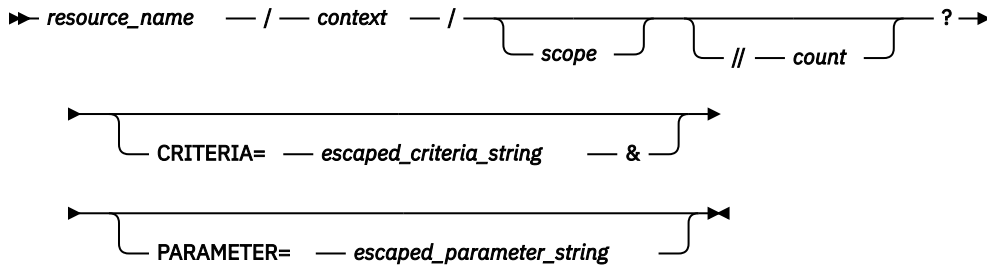
GET 요청의 URI는 조작으로 검색할 자원의 인스턴스를 식별합니다.

다음 다이어그램은 GET 요청에 대한 URI 구문을 보여줍니다.

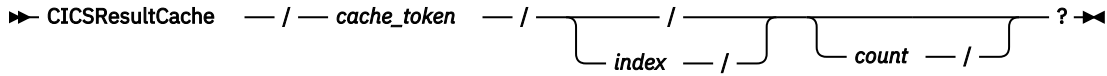
URI 구문



자원 매개변수:



캐시 매개변수:



옵션

/CICSSystemManagement

요청이 CICS 관리 클라이언트 인터페이스에 액세스 중임을 표시합니다. CICSSystemManagement는 대소문자를 구분합니다.

resource_name

조회하는 CICS 또는 CICSplex SM 자원과 연관되는 외부 자원 이름을 지정합니다. 예를 들어, 자원 이름 CICSRegion을 지정하면 요청이 CICSplex SM CICSRRGN 자원과 연관되고, CICSResourceAssignmentDefinition을 지정하면 요청이 RASGNDEF 자원과 연관됩니다. CICS 관리 클라이언트 인터페이스 요청에서 외부 자원 이름의 전체 목록은 [17 페이지의 『CMCI 자원 이름』](#)의 내용을 참조하십시오.

HTTP GET 메소드는 모든 자원에 올바르지 않습니다. 이 HTTP 메소드가 특정 자원에 대해 올바르지 판별하려면 [CICSplex SM 자원 테이블](#)에 있는 자원 설명을 참조하십시오.

자원 이름은 대소문자를 구분하지 않습니다. 클라이언트가 올바르지 않은 자원 이름을 지정하는 경우, HTTP 404(찾을 수 없음) 응답이 리턴됩니다.

context

CMCI가 CICSplex SM 환경에 설치된 경우 context는 요청과 연관되는 CICSplex 또는 CMAS의 이름입니다 (예: PLEX1). [CICSplex SM 자원 테이블](#)의 관련 자원 테이블을 참조하여 CICSplex를 지정할지 CMAS를 지정할지 결정하십시오.

CMCI가 단일 서버로 설치된 경우 context는 요청과 연관된 CICS 리전의 애플리케이션 ID입니다.

컨텍스트 값은 공백을 포함하면 안됩니다. 컨텍스트는 대소문자가 구분되지 않습니다.

scope

조회와 연관되는 CICSplex, CICS 그룹, CICS 리전 또는 논리 범위의 이름을 지정합니다. 범위는 컨텍스트의 서브세트로, 요청을 특정 CICS 시스템 또는 자원으로 제한합니다. 범위는 필수가 아닙니다. 범위가 없으면 요청은 컨텍스트의 값만으로 제한됩니다. 범위 값은 공백을 포함하면 안 됩니다. 범위는 대소문자를 구분하지 않습니다.

CICSResultCache/cache_token

보유된 결과 세트를 포함하는 결과 캐시를 식별합니다. 캐시 토큰을 지정하면 요청이 이전 GET 요청에서 보유된 레코드로 제한됩니다. 모든 CMCI 요청 끝에서 CMCI는 보유된 결과 세트 상태를 확인합니다. 15분 이상 사용되지 않은 모든 결과 세트는 해당 cache_token과 함께 버립니다.

색인

보유된 결과에서 캐시 토큰 매개변수에 지정된 레코드를 식별합니다. 레코드는 보유된 결과 세트에서 유지하고 있는 위치에 의해 식별됩니다. index는 1 - 2 147 483 647 범위(경계 포함)의 정수여야 합니다. 영(0) 값은 허용되지 않습니다. index를 지정하지 않는 경우 이 위치에 슬래시를 삽입해야 합니다.

index 옵션을 지정해도 요청된 레코드 수는 변경되지 않으며 경고 개수 메커니즘에 영향을 주지 않습니다.

count

결과 캐시에서, 결과 캐시의 첫 번째 레코드나 index 매개변수에 지정된 레코드에서 시작하는 레코드 서브세트를 식별합니다. 음수는 마지막 레코드에서 뒤로 계수함을 나타냅니다. 예를 들어, -1은 마지막 레코드를

의미하고 -2는 마지막 레코드의 다음 레코드를 의미합니다. **count**는 -2 147 483 647에서 2 147 483 647 까지(경계값 포함)의 범위에 있는 정수여야 합니다. 영(0) 값은 허용되지 않습니다. 자원 매개변수의 경우 두 개의 슬래시를 **count** 앞에 붙여야 합니다.

count 옵션을 지정하면 요청에 표시되는 레코드 수를 제한할 수 있으며, 요청된 레코드 수를 변경하지 않으며 경고 개수 메커니즘에 영향을 주지 않습니다.

조회 매개변수

요청의 범위와 속성을 세분화합니다. 조회 섹션의 구성 파트는 어떤 순서로도 발생할 수 있지만 각각 URI에서 한 번만 발생할 수 있습니다. 조회 매개변수는 앰퍼샌드(&)로 구분됩니다. 조회 매개변수 값에서 대소문자가 구분되지 않지만, 일부 속성(예: TRANID 및 DESC)이 대소문자 혼용 값을 보유할 수 있기 때문에 특정 속성 값의 대소문자는 올바르게 지정해야 합니다. 조회 매개변수는 다음과 같습니다.

CRITERIA=escaped_criteria_string

요청 시 리턴되는 데이터를 필터하는 논리식 문자열입니다. CRITERIA 매개변수의 값을 구성하는 문자열은 CICSplex SM API(application programming interface)에서 필터 표현식과 같은 규칙을 따릅니다. CICSplex SM API를 사용하는 필터 표현식을 지정하는 방법에 대한 자세한 지침은 [필터 표현식 빌드 방법](#)의 내용을 참조하십시오.

특정의 제한된 문자는 이스케이프 시퀀스로 대체하여 서버가 올바르게 해석할 수 있도록 해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

PARAMETER=escaped_parameter_string

요청을 세분화하는 **parameter_name(data_value)** 양식의 하나 이상의 매개변수 및 값 문자열입니다. 이 매개변수를 지정하는 규칙은 CICSplex SM API(Application Programming Interface)에서와 같습니다. CICSplex SM API를 사용하는 매개변수 표현식을 지정하는 방법에 대한 자세한 지침은 [자원에 대한 조치 수행](#)의 내용을 참조하십시오.

특정 문자를 이스케이프 문자로 인코딩해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

NODISCARD

요청 다음에 결과 세트가 보유됨을 표시합니다. 이 속성은 값 없이 지정됩니다. NODISCARD를 지정하지 않는 경우 결과 세트는 버려집니다. 자원 이름이나 보유된 캐시에 대해 작동하는 요청에 NODISCARD를 사용할 수 있습니다. 각 요청 끝에서 CMCI는 보존된 결과 세트 상태를 확인하고 15분 이상 사용되지 않은 항목을 버립니다.

ORDERBY

결과 세트가 지정된 필드로 정렬됨을 표시합니다. 이 속성은 논리식 문자열로 지정됩니다. **ORDERBY** 매개변수의 값을 구성하는 문자열은 CICSplex SM API(Application Programming Interface)에서와 동일한 규칙을 따릅니다. EYUDA 및 CVDA 값의 경우, 필드는 표시되는 문자 값이 아닌 숫자 값으로 정렬됩니다. CICSplex SM API를 사용하는 레코드 정렬 방법 대한 자세한 지침은 [결과 세트에 레코드 정렬](#)의 내용을 참조하십시오.

특정 문자를 이스케이프 문자로 인코딩해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

이 옵션을 사용하는 각 요청은 L8 및 L9 모드 개방형 TCB의 풀에서 L8 TCB를 요청하며, 이는 CICS에 의해 자동으로 관리됩니다.

OVERRIDEWARNINGCOUNT

경고 개수 한계 메커니즘을 무시하고 기본 경고 개수 한계가 지정되지 않은 경우와 같이 요청을 실행할 수 있습니다. 이 옵션은 CICSplex SM 웹 사용자 인터페이스 초기화 매개변수 **RESOURCELIMIT**가 WARNING으로 설정된 경우에만 유효합니다.

SUMMONLY

요청에서 결과 요약만 리턴됨을 표시합니다. 자세한 레코드는 응답에 포함되지 않습니다.

SUMMONLY 옵션을 지정하여 요청에 레코드를 표시하지 않아도 요청된 레코드 수를 변경하지 않으며 경고 개수 메커니즘에 영향을 주지 않습니다. 요청된 레코드 수가 경고 개수 값을 초과하면 **SUMMONLY**를 포함하는 요청에 실패합니다.

보유된 결과 세트를 사용하여 GET 요청 성능 향상

초기 요청이 완료된 후 서버가 메모리에 결과 세트를 보유하도록 지시하기 위해 NODISCARD 옵션을 사용하여 HTTP GET 요청의 성능을 향상시킬 수 있습니다. CMAS를 조회하는 대신, 보유 결과 세트에 대해 후속 요청을 작성할 수 있습니다. 그러면 서버에 대한 로드가 감소합니다.

이 태스크 정보

일반적으로 CMCI 가비지 컬렉션은 각 GET 요청 이후 결과 세트를 버립니다. NODISCARD 옵션을 사용하는 경우 결과 세트가 보유됩니다. 각 요청 끝에서 CMCI는 모든 결과 세트 상태를 확인하고 15분 이상 사용되지 않은 항목만 버립니다.

보유된 결과 세트를 사용하면 클라이언트가 하나의 큰 문서 대신, 작은 여러 개의 XML 응답 문서에서 조회 결과를 요청할 수 있습니다. 더 작은 XML 문서에 대해 작동하면 많은 결과를 렌더링, 전송 및 구문 분석해야 하는 필요성이 없어져서 서버, 네트워크, 클라이언트에 대한 로드가 감소됩니다. 예를 들어, 클라이언트는 보유 결과에 대해 작동하여 조회된 자원을 페이지 단위로 표시할 수 있습니다. 이때 각 페이지는 수천 개의 레코드를 포함하는 원래 결과 세트에서 단 20개 정도의 결과만 표시합니다. 그러나 보유 결과 세트의 스냅샷 속성상, 빠르게 변하는 데이터를 조회하거나 클라이언트가 가능한 최신 상태를 확보해야 하는 경우 보유된 결과 세트는 조회에 적합하지 않게 됩니다.

서버에 대해 보안이 사용 가능하면, 사용자가 작성한 보유된 결과 세트에 대한 액세스만 갖습니다. 다른 사용자가 작성한 보유된 결과 세트에 액세스하려고 하면 HTTP 403 오류가 발생합니다.

보유된 결과 세트로 작업하려면 다음을 수행하십시오.

프로시저

1. 요청 URI에서 선택적 NODISCARD 및 SUMMONLY 옵션을 지정하는 초기 HTTP GET 요청을 실행하십시오.

요청이 성공한 것으로 가정하면 CICS 관리 클라이언트 인터페이스는 조회 결과를 GCDSA 동적 저장 영역에 저장하고, 보유 결과 세트에 액세스하기 위한 후속 요청에 사용할 수 있는 토큰을 클라이언트에 리턴합니다. 토큰은 GET 요청 응답의 <resultsummary> 요소에 있는 **cachetoken** 속성의 값으로 리턴됩니다. SUMMONLY를 지정하면 서버는 응답 본문에서 레코드를 리턴하지 않습니다.

2. 외부 자원 이름 대신, CICSResultCache를 지정하고 뒤에 초기 요청에 대한 응답의 **cachetoken** 속성 값을 지정하여 두 번째 GET 요청을 발행하십시오.

인식되지 않거나 삭제된 캐시 토큰을 지정하는 경우, 그 토큰은 자원을 찾을 수 없음 조건으로 처리되고 서버는 HTTP 404 오류를 리턴합니다.

index, count, , orderby 옵션을 지정하여 요청을 더 세분화할 수 있습니다. **index**는 보유된 결과 세트에서 해당 위치로 결과 세트의 한 레코드를 식별합니다. **count**는 **index**에서 식별된 레코드 위치에서 또는 세트의 첫 번째 레코드에서 보유된 결과에 있는 레코드의 서브세트를 식별합니다. **orderby**는 결과 세트가 정렬되는 하나 이상의 필드를 식별합니다. **orderby** 옵션에 지정할 수 있는 최대 필드 개수는 32개입니다.

후속 요청에서 NODISCARD 옵션을 지정하여, 이 요청 다음에 결과 세트가 계속 사용 가능하도록 만들 수 있습니다. NODISCARD를 지정하지 않은 경우 서버는 요청의 마지막에 결과 세트를 삭제합니다.

클라이언트는 단일 캐시 토큰에 대해 서버에 여러 개의 동시 요청을 발행할 수도 있습니다. 예를 들어, 멀티스레드된 클라이언트 애플리케이션은 레코드 1 - 10, 11 - 20, 21 - 30에 대한 정보를 동시에 가져올 수 있습니다. 그러나 동시 요청 중 하나 이상에서 NODISCARD 옵션을 지정하지 못하면 예측할 수 없는 결과가 발생합니다.

예

시스템에 설치된 모든 프로그램을 보기 위한 GET 요청은 수천 개의 레코드가 클라이언트에 리턴되도록 할 수 있지만, 보유 결과 세트를 사용하여 결과에서 더 빠르게 페이지 지정할 수 있습니다.

다음 요청은 시스템에 설치된 모든 PROGRAM 자원을 식별하지만 SUMMONLY 옵션은 레코드가 리턴되지 않고 식별된 자원 요약만 리턴됨을 표시합니다. NODISCARD 옵션은 이 결과 세트를 식별할 캐시 토큰을 리턴합니다.

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

서버는 다음 응답을 리턴합니다.

```
<?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" recordcount="1662"/>
</response>
```

api_response1 및 api_response2 속성이 각각 1024 및 0인지 확인하여 조회가 성공했는지 검증할 수 있습니다.

<resultsummary> 요소는 cachetoken 값 C3D526A2F224FFE1로 결과 세트를 식별합니다. recordcount 속성은 총 1662개의 레코드가 결과 세트에 있음을 나타냅니다.

이 보유 결과 세트에서 처음 10개의 레코드를 조사하려면 다음 요청을 발행하면 됩니다.

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

서버는 다음 응답을 리턴합니다.

```
<?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 속성은 <records> 요소가 <cicsprogram> 하위 요소를 포함한다는 것을 표시합니다. cachetoken 속성은 결과 세트가 나중에 사용하도록 서버에 보유되었음을 확인합니다.

클라이언트는 다음 요청을 발행하여 레코드 11에서 시작하여 10개의 다음 레코드를 검색할 수 있습니다.

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

보유된결과 세트 조사를 완료하면, NODISCARD 옵션을 지정하지 않고 결과 세트에 액세스하여 서버에 알립니다. 다음 요청은 다시 SUMMONLY 매개변수를 사용하여 서버가 이 요청에 대해 레코드를 리턴하지 않도록 합니다.

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

서버는 다음 응답을 리턴합니다.

```
<?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> 요소에 cachetoken 속성이 없으면 결과 세트가 더 이상 서버에 저장되지 않습니다. 보유 결과 세트를 사용하기 위해 추가 요청을 발행하려고 하면 서버는 HTTP 404 찾을 수 없음 오류를 리턴합니다.

CMCI POST 요청

CICS 관리 클라이언트 인터페이스(CMCI)는 CICS 또는 CICSplex SM에서 자원을 작성하기 위해 HTTP POST 메소드를 사용합니다.

클라이언트는 다음 파트에서 POST 요청을 형성합니다.

- HTTP 메소드(이 경우 POST)
- 검색할 자원을 식별하고 나중에 사용하거나 버리기 위해 결과가 캐시되는지 여부를 표시하는 URI.
- HTTP 버전
- 필요한 경우 권한 부여 인증서
- 작성할 자원의 세부사항을 포함하는 XML 본문.

예를 들어, 다음과 같습니다.

```
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>
```

초기 행에는 세 개의 파트가 공백으로 구분되어 있습니다.

- HTTP 메소드 이름
- 요청 URI. /CICSSystemManagement/로 시작하고 바로 뒤에 자원을 식별하는 외부 이름과, 작성할 자원의 인스턴스를 식별하는 필터 및 매개변수가 오는 절대 경로로 지정됩니다.
- CICS 시스템 관리 클라이언트 API에 연결할 때 HTTP 버전으로, 항상 HTTP/1.1입니다.

두 번째 행은 URI에 지정된 대상의 호스트 이름 및 포트 번호(콜론으로 구분)를 식별합니다. 이 행은 뒤에 줄 바꾸기가 있는 캐리지 리턴으로 끝납니다.

선택적 세 번째 행에는 권한 부여 인증서가 있습니다. CICS 시스템 초기화 매개변수 **SEC**을 YES로 지정하여 시스템이 실행 중인 경우 기본 인증 헤더에 Base-64로 사용자 ID와 비밀번호를 제공해야 합니다. 이 예제에서 사용자 ID는 *FRED*이고 비밀번호는 *PASSWORD*입니다.

HTTP 헤더는 개별 행에서 최종 캐리지 리턴 및 줄 바꾸기로 종료됩니다.

요청의 리마인더는 새 자원의 속성을 지정하는 XML 본문입니다. 본문은 하나의 <attributes> 하위 요소를 포함하는 XML <request> 요소로 구성됩니다.

POST 요청에 대한 응답

POST 요청 완료 시, 클라이언트는 작성된 자원의 세부사항 및 결과 요약에 포함하는 XML 피드와 HTTP 헤더로 구성되는 클라이언트 API에서 응답을 수신합니다. 그러나 요청에 SUMMONLY 매개변수가 포함된 경우, 응답은 HTTP 헤더와 결과 요약으로만 구성됩니다.

헤더는 HTTP 응답, 날짜 및 시간, 서버와 데이터 유형의 세부사항으로 구성됩니다. 다음 예제는 성공적인 요청에 대한 헤더를 표시합니다.

```
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
```

응답 본문은 요청에 대한 요약된 정보를 표시하는 <resultsummary> 요소를 포함하는 <response> 루트 요소와, 일반적으로 새 자원의 세부사항을 포함하는 <records> 요소로 구성됩니다. 그러나 URI에서 SUMMONLY 매개변수를 지정하여 응답의 <records> 요소를 비활성화할 수 있습니다. 요청이 오류로 완료되면, 진단 정보는 <errors> 요소에서 제공됩니다.

보안 고려사항

HTTP 메소드를 사용하는 경우 일부 방화벽은 보안 고려사항 때문에 방화벽을 통하여 HTTP PUT 또는 DELETE 트래픽을 허용하지 않는다는 점에 유의하십시오. 이 제한사항을 수용하려면 POST 요청을 통해 PUT 또는 DELETE 요청을 터널링하기 위해 X-HTTP-Method-Override HTTP 헤더 필드를 사용할 수 있습니다. 예:

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

CMCI POST 요청 URI

POST 요청의 URI는 조작에 의해 작성될 자원을 식별합니다.

다음 다이어그램은 POST 요청에 대한 URI 구문을 보여줍니다.

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



POST 요청에서는 기준 또는 매개변수 문자열을 지정하지 않습니다. 작성될 자원에 대한 모든 정보는 요청 본문에 포함합니다.

옵션

/CICSSystemManagement

요청이 CICS 관리 클라이언트 인터페이스에 액세스 중임을 표시합니다. CICSSystemManagement는 대소문자를 구분합니다.

resource_name

작성하는 CICS 또는 CICSplex SM 자원과 연관되는 외부 자원 이름을 지정합니다. 예를 들어, 자원 이름 CICSRegion을 지정하면 요청이 CICSplex SM CICSRRGN 자원 테이블과 연관되고, CICSResourceAssignmentDefinition을 지정하면 요청이 RASGNDEF 자원 테이블과 연관됩니다. 외부 자원 이름의 전체 목록은 17 페이지의 『CMCI 자원 이름』의 내용을 참조하십시오.

HTTP POST 메소드는 모든 자원에 올바르지 않습니다. 이 HTTP 메소드가 특정 자원에 대해 올바른지 판별하려면 CICSplex SM 자원 테이블에 있는 자원 설명을 참조하십시오.

자원 이름은 대소문자를 구분하지 않습니다. 클라이언트가 올바르지 않은 자원 이름을 지정하는 경우, HTTP 404(찾을 수 없음) 응답이 리턴됩니다.

컨텍스트

요청과 연관되는 CICSplex 또는 CMAS의 이름을 지정합니다(예: PLEX1). 이 매개변수는 CICSplex SM과 상호작용하는 URI에 필요합니다. 컨텍스트는 공백을 포함하면 안 됩니다. 대소문자는 구분되지 않습니다.

SUMMONLY

요청에서 결과 요약만 리턴됨을 표시합니다. 자세한 레코드는 응답에 포함되지 않습니다.

CMCI PUT 요청

CICS 관리 클라이언트 인터페이스(CMCI)는 CICS 또는 CICSplex SM 자원의 속성을 변경하거나 해당 자원에서 INSTALL을 포함한 조치를 수행하기 위해 HTTP PUT 메소드를 사용합니다.

클라이언트는 다음 구성요소로부터 PUT 요청을 형성합니다.

- HTTP 메소드(이 경우 PUT)
- 변경할 자원을 식별하는 URI
- HTTP 버전
- 필요한 경우 권한 부여 인증서
- 변경할 사항 또는 수행할 조치를 XML로 지정하는 HTTP 본문

예를 들어, CICSplex PLEX1에서 TR로 시작하는 이름과 P로 시작하는 프로그램이 있는 트랜잭션 정의의 속성을 업데이트하려는 경우 클라이언트는 다음 요청을 구성합니다.

```

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

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

```

초기 행에는 세 개의 파트가 공백으로 구분되어 있습니다.

- 메소드 이름
- 요청 URI입니다. 이 URI는 /CICSSystemManagement/로 시작하고 바로 뒤에 선택할 해당 자원 유형의 인스턴스를 판별하는 필터 및 매개변수와 함께 자원을 식별하는 외부 이름이 오는 절대 경로로 지정됩니다. CICS 관리 클라이언트 인터페이스가 처리하기에 적합하도록 URI를 작성하기 위해, 클라이언트는 특정의 제한된 문자(예: 공백 및 별표)를 이스케이프 순서로 바꿉니다.
- CICS 관리 클라이언트 인터페이스에 연결 시 HTTP 버전으로, 항상 HTTP/1.1입니다.

두 번째 행은 URI에 지정된 대상의 호스트 이름 및 포트 번호(콜론으로 구분)를 식별합니다. 이 행은 뒤에 줄 바꿈기가 있는 캐리지 리턴으로 끝납니다.

선택적 세 번째 행에는 권한 부여 인증서가 있습니다. CICS 시스템 초기화 매개변수 **SEC**을 YES로 지정하여 시스템이 실행 중인 경우 기본 인증 헤더에 Base-64로 사용자 ID와 비밀번호를 제공해야 합니다. 이 예제에서 사용자 ID는 *FRED*이고 비밀번호는 *PASSWORD*입니다.

HTTP 헤더는 개별 행에서 최종 캐리지 리턴 및 줄 바꿈기로 종료됩니다.

요청의 나머지는 식별된 자원 인스턴스에 대해 수행할 변경사항을 지정하는 XML 본문입니다. 이 경우, 요청은 선택된 트랜잭션 정의의 STATUS 속성을 ENABLED로 변경합니다. 본문은 하나의 <attributes> 요소 또는 하나의 <action> 요소를 포함하는 XML <request> 요소로 구성됩니다.

PUT 요청은 설치된 자원이나 정의 CICS 또는 CICSplex 자원에 대해 작동할 수 있습니다. 단일 PUT 요청은 속성 변경사항을 지정하거나 조치를 수행할 수 있습니다. 속성 변경사항과 조치를 단일 요청으로 결합할 수 없습니다.

PUT 메소드는 모든 조치를 지원하지 않습니다. 다음 조치는 POST 메소드에서 지원됩니다.

- 정의 자원에 대한 CREATE.
- SYSDUMP 및 TRANDUMP 자원 유형에 대한 ADD.

다음 조치는 DELETE 메소드에서 지원됩니다.

- 운영 자원에 대한 DISCARD.
- 정의 자원에 대한 REMOVE.

PUT 요청에 대한 응답

PUT 요청 완료 시, 클라이언트는 HTTP 헤더로 구성되는 CICS 관리 클라이언트 인터페이스의 응답과, 결과 요약 을 포함하는 XML 피드, 그리고 요청이 성공하는 경우 변경된 자원의 세부사항을 수신합니다.

헤더는 HTTP 응답, 날짜 및 시간, 서버와 콘텐츠 유형의 세부사항으로 구성됩니다. 예를 들어, 다음과 같습니다.

```

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 요청에 실패하면, 서버는 HTTP 비정상 응답을 발행합니다. 예를 들어, 요청의 본문이 올바르지 않으면 헤더에는 다음 400 응답이 포함됩니다.

```

400 잘못된 요청. 요청의 응답이 올바르지 않습니다.

```

이 응답은 클라이언트가 PUT 요청에 대한 여러 태그를 전송하거나, 요청 본문에 알 수 없는 태그가 포함되거나, 태그 이름이 URI의 모델 이름과 일치하지 않은 경우에 발생할 수 있습니다.

가 구분되지 않지만, 일부 속성(예: TRANID 및 DESC)이 대소문자 혼용 값을 보유할 수 있기 때문에 특정 속성 값의 대소문자는 올바르게 지정해야 합니다. 조회 매개변수는 다음과 같습니다.

CRITERIA=escaped_criteria_string

요청 시 리턴되는 데이터를 필터하는 논리식 문자열입니다. CRITERIA 매개변수의 값을 구성하는 문자열은 CICSplex SM API(application programming interface)에서 필터 표현식과 같은 규칙을 따릅니다. CICSplex SM API를 사용하는 필터 표현식을 지정하는 방법에 대한 자세한 지침은 [필터 표현식 빌드 방법](#)의 내용을 참조하십시오.

특정의 제한된 문자는 이스케이프 시퀀스로 대체하여 서버가 올바르게 해석할 수 있도록 해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

PARAMETER=escaped_parameter_string

요청을 세분화하는 **parameter_name(data_value)** 양식의 하나 이상의 매개변수 및 값 문자열입니다. 이 매개변수를 지정하는 규칙은 CICSplex SM API(Application Programming Interface)에서와 같습니다. CICSplex SM API를 사용하는 매개변수 표현식을 지정하는 방법에 대한 자세한 지침은 [자원에 대한 조치 수행](#)의 내용을 참조하십시오.

기준 문자열에서와 같이, 특정 문자열은 이스케이프 문자를 사용하여 인코딩해야 합니다. CICS 관리 클라이언트 인터페이스 URI에서의 이스케이프 순서 사용에 대한 지침은 [39 페이지의 『CMCI 이스케이프 시퀀스』](#)의 내용을 참조하십시오.

OVERRIDEWARNINGCOUNT

경고 개수 한계 메커니즘을 무시하고 기본 경고 개수 한계가 지정되지 않은 경우와 같이 요청을 실행할 수 있습니다. 이 옵션은 CICSplex SM 웹 사용자 인터페이스 초기화 매개변수 **RESOURCELIMIT**가 WARNING으로 설정된 경우에만 유효합니다.

SUMMONLY

요청에서 결과 요약만 리턴됨을 표시합니다. 자세한 레코드는 응답에 포함되지 않습니다.

SUMMONLY 옵션을 지정하여 요청에 레코드를 표시하지 않아도 요청된 레코드 수를 변경하지 않으며 경고 개수 메커니즘에 영향을 주지 않습니다. 요청된 레코드 수가 경고 개수 값을 초과하면 **SUMMONLY**를 포함하는 요청에 실패합니다.

CMCI 자원 이름

CICS 관리 클라이언트 인터페이스(CMCI)는 CICSplex SM 및 CICS 자원에 맵핑되는 외부 자원 이름을 사용합니다.

다음 표에서는 CMCI 외부 자원 이름 및 연관된 CICSplex SM 자원 이름을 나열합니다. 자원은 CICS 운영 자원, CICS 정의 자원, CICSplex SM 정의 자원, CICSplex SM 비정의 자원으로 나뉩니다.

각 자원에 올바른 CMCI 메소드를 비롯하여 CICSplex SM 자원에 대한 정보는 [CICSplex SM 자원 테이블](#)의 내용을 참조하십시오.

CICS 운영 자원

표 1. CICS 운영 자원의 자원 이름		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSAllTaskSubpools	TSKSPOLS	CICS 태스크 스토리지 서브풀
CICSAtomService	ATOMSERV	CICS Atom 서비스, 피드, 콜렉션 또는 범주 문서
CICSAutoInstallModel	AIMODEL	CICS 자동 설치 터미널 모델
CICSBridgeFacility	BRFACIL	3270 브릿지 메커니즘에 사용되는 가상 터미널(브릿지 기능)
CICSBundle	BUNDLE	CICS 번들
CICSBundlePart	BUNDPART	CICS 번들 파트
CICSCaptureSpecification	EVCSPEC	CICS 이벤트 캡처 스펙
CICSCaptureSpecificationDataPredicate	EVCSDATA	CICS 이벤트 캡처 스펙 데이터 술어
CICSCaptureSpecificationInformationSource	EVCSINFO	CICS 이벤트 캡처 스펙 정보 소스
CICSCaptureSpecificationOptionPredicate	EVCSOPT	CICS 이벤트 캡처 스펙 명령 옵션 술어
CICSCFDTPool	CFDTPOOL	결합 기능 데이터 테이블 풀
CICSDataSetName	DSNAME	활성 CICS 시스템에서 사용 중인 데이터 세트
CICSDataTable	CMDT	CICS 또는 사용자 유지보수 데이터 테이블 또는 이와 연관된 결합 기능 데이터 테이블을 가지고 있는 파일
CICSDB2Connection	DB2CONN	Db2® 연결
CICSDB2Entry	DB2ENTRY	Db2 항목
CICSDB2Transaction	DB2TRN	Db2 트랜잭션
CICSDBCTLSubsystem	DBCTLSS	DBCTL 서브시스템에 대한 활성 CICS 시스템의 연결
CICSDFHRPLDataSet	RPLLIST	CICS DFHRPL 병합 시퀀스의 데이터 세트
CICSDocumentTemplate	DOCTEMP	문서 템플릿
CICSDomainSubpool	DOMSPOOL	CICS 스토리지 도메인 서브풀
CICSDynamicStorageArea	CICSDSA	동적 스토리지 영역

표 1. CICS 운영 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSEnqueue	ENQUEUE	CICS 인큐
CICSEnqueueModel	ENQMODEL	ENQ/DEQ 모델
CICSEPADapter	EPADAPT	CICS 이벤트 처리 어댑터
CICSEPADapterSet	EPADSET	CICS 이벤트 처리 어댑터 세트
CICSEPADapterInSet	EPAINSET	이벤트 처리 어댑터 세트에 있는 CICS 이벤트 처리 어댑터
CICSEventBinding	EVNTBIND	CICS 이벤트 바인딩
CICSEventProcessing	EVNTGBL	CICS 이벤트 처리
CICSEXCIRequest	EXCI	외부 CICS 인터페이스 API를 사용하여 클라이언트 프로그램에서 시작된 태스크
CICSExtrapartitionTDQueue	EXTRATDQ	리전 외부 트랜지언트 데이터 큐
CICSFEPICConnection	FEPICONN	FEPI 연결
CICSFEPINode	FEPINODE	FEPI 노드
CICSFEPIPool	FEPIPOOL	FEPI 풀
CICSFEPIPropertySet	FEPIPROP	FEPI 특성 세트
CICSFEPITarget	FEPITRGT	FEPI 대상
CICSGlobalDispatcher	DSPGBL	CICS 시스템에 대한 글로벌 CICS 디스패처 정보
CICSGlobalDynamicStorageArea	CICSSTOR	CICS 동적 스토리지 영역
CICSGlobalMVSTCBStatistics	MVSTCBGL	CICS 주소 공간의 글로벌 MVS™ TCB
CICSGlobalTDQueueStatistics	TDQGBL	리전 내부 트랜지언트 데이터 큐 사용
CICSGlobalTSQueueStatistics	TSQGBL	임시 스토리지 큐 사용
CICSGlobalURIMapStatistics	URIMPGBL	URIMAP 자원에 대해 CICS 추출 통계가 리턴한 글로벌 통계
CICSGlobalUserExit	EXITGLUE	설치된 CICS TS 글로벌 사용자 엑시트
CICSIndirectTDQueue	INDTDQ	간접 트랜지언트 데이터 큐
CICSIntervalControlRequest	REQID	미해결 간격 제어 요청
CICSIntrapartitionTDQueue	INTRATDQ	리전 내부 트랜지언트 데이터 큐
CICSIPFacility	IPFACIL	활성 CICS 태스크 및 해당 태스크가 사용 중인 IPIC 연결 간 연관
CICSIPICConnection	IPCONN	IP 상호 통신 연결
CICSISCMROConnection	CONNECT	ISC 또는 MRO 연결
CICSJournalModel	JRNLMODL	설치된 저널 모델 및 해당되는 로그 스트림 이름
CICSJournalName	JRNLNAM	시스템 로그 및 일반 로그에 대한 정보
CICSJVM	JVM	JVM(Java™ Virtual Machine)

표 1. CICS 운영 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSJVMClassCache	CLCACHE	공유 클래스 캐시
CICSJVMPool	JVMPool	JVM 풀
CICSJVMProfile	JVMPROF	JVM 프로파일
CICSJVMServer	JVMSERV	설치된 JVM 서버
CICSLIBRARY	LIBRARY	LIBRARY
CICSLIBRARYDataSetName	LIBDSN	LIBRARY 데이터 세트 이름
CICSLoader	LOADER	CICS 로더 정보
CICSLoaderByDSA	LOADACT	동적 스토리지별 CICS 로더 활동
CICSLocalFile	LOCFILE	CICS 로컬 파일
CICSLocalTransaction	LOCTAN	CICS 로컬 트랜잭션
CICSLSRPool	LSRPOOL	로컬 공유 자원 풀
CICSLSRPoolBuffer	LSRPBUF	VSAM LSR 풀 버퍼
CICSLU62ModeName	MODENAME	LU 6.2 모드 이름
CICSMonitoringAndStatistics	MONITOR	CICS 모니터링 및 통계 정보
CICSMQConnection	MQCON	IBM MQ 연결
CICSMQConnectionStatistics	MQCONN	IBM MQ 연결 통계
CICSMQInitiationQueue	MQINI	IBM MQ 시동 큐
CICSMVSLogStream	STREAMNM	MVS 로그 스트림 이름
CICSMVSStorageArea	MVSESTG	MVS 스토리지 영역
CICSMVSTCB	MVSTCB	MVS TCB
CICSMVSWLM	MVSWLM	MVS 워크로드 관리 컴포넌트 정보
CICSOSGiBundle	OSGIBUND	OSGi 번들
CICSOSGiService	OSGISERV	OSGi 서비스
CICSPagePool	CICSPAGP	CICS 페이지(DSA) 풀
CICSPartner	PARTNER	CPI-C 파트너 테이블
CICSPipeline	PIPELINE	CICS 웹 서비스 처리 노드
CICSProcessType	PROCTYP	BTS 프로세스 유형
CICSProfile	PROFILE	CICS 프로파일
CICSProgram	PROGRAM	CICS 프로그램
CICSRecoveryManager	RECOVERY	복구 관리자 정보
CICSRegion	CICSRGN	CICS 리전
CICSRemoteFile	REMFIL	원격 CICS 파일
CICSRemoteTDQueue	REMTDQ	원격 트랜지언트 데이터 큐
CICSRemoteTransaction	REMTRAN	원격 트랜잭션

표 1. CICS 운영 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSSharedTSQueue	TSQSHR	임시 스토리지 큐
CICSSystemDump	SYSDUMP	CICS 시스템 덤프 코드
CICSSystemParameter	SYSPARM	CICS 시스템 매개변수
CICSTask	TASK	CICS 태스크
CICSTaskAssociation	TASKASSC	태스크 연관 데이터
CICSTaskFileUsage	TASKFILE	태스크 및 태스크에 사용된 CICS 파일
CICSTaskRelatedExit	EXITTRUE	태스크 관련 사용자 엑시트
CICSTaskRMIUsage	TASKRMI	태스크별 CICS 자원 관리자 인터페이스 사용
CICSTaskStorage	TASKESTG	태스크에 대한 CICS 스토리지 요소
CICSTaskSubpool	TSKSPool	CICS 태스크 스토리지 서브풀
CICSTaskTSQueueUsage	TASKTSQ	태스크 및 태스크에 사용된 CICS 임시 스토리지 큐
CICSTCBMode	DSPMODE	CICS 디스패처 TCB 모드
CICSTCBPool	DSPPOOL	CICS 디스패처 TCB 풀
CICSTCIPService	TCPIPS	TCP/IP 서비스
CICSTCIPStatistics	TCPIPGBL	CICS 내부 TCP/IP 소켓 지원
CICSTerminal	TERMNL	CICS 터미널
CICSTransactionClass	TRANCLAS	CICS 트랜잭션 클래스
CICSTransactionDump	TRANDUMP	CICS 트랜잭션 덤프 코드
CICSTSMModel	TSMODEL	CICS 임시 스토리지 모델
CICSTSPool	TSPOOL	CICS 임시 스토리지 풀
CICSTSQueue	TSQNAME	CICS 비공유 임시 스토리지 큐
CICSUOW	UOW	CICS 작업 단위
CICSUOWEnqueue	UOWENQ	작업 단위별로 보유된 큐 넣기
CICSUOWLink	UOWLINK	작업 단위 및 CICS 시스템 사이의 링크
CICSUOWShuntedAndHoldingLocks	UOWDSNF	처리 지연된 작업 단위
CICSURIHost	HOST	로컬 시스템의 가상 호스트
CICSURIMap	URIMAP	범용 자원 ID 요청
CICSWebService	WEBSERV	웹 서비스 설정에 배치된 CICS 애플리케이션 프로그램에 대한 런타임 환경의 정보
CICSWorkRequest	WORKREQ	EJB 작업 요청
CICSXmltransform	XMLTRANS	XML 변환

CICS 정의 자원

표 2. CICS 정의 자원의 자원 이름		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSCSDGroup	CSDGROUP	CSD 그룹 정의
CICSCSDGroupInList	CSDINLST	목록의 CSD 그룹
CICSCSDList	CSDLIST	CSD 목록 정의
CICSCSDResource	CSDINGRP	그룹의 CSD CICS 자원
CICSDefinitionAtomService	ATOMDEF	Atom 서비스 정의
CICSDefinitionBundle	BUNDDEF	번들 정의
CICSDefinitionCorbaServer	EJCODEF	CorbaServer 정의
CICSDefinitionDb2Connection	DB2CDEF	Db2 연결 정의
CICSDefinitionDb2Entry	DB2EDEF	Db2 항목 정의
CICSDefinitionDb2Transaction	DB2TDEF	Db2 트랜잭션 정의
CICSDefinitionDeployedJARFile	EJDJDEF	CICS 배치된 JAR 파일 정의
CICSDefinitionDocumentTemplate	DOCDEF	문서 템플릿 정의
CICSDefinitionEnqueueModel	ENQMDEF	글로벌 인큐 정의
CICSDefinitionFEPINode	FENODDEF	FEPI 노드 정의
CICSDefinitionFEPIPool	FEPOODEF	FEPI 풀 정의
CICSDefinitionFEPIPropertySet	FEPRODEF	FEPI 특성 세트 정의
CICSDefinitionFEPITarget	FETRDEF	FEPI 대상 정의
CICSDefinitionFile	FILEDEF	CICS 파일 정의
CICSDefinitionIPICConnection	IPCONDEF	IPIC 연결 정의
CICSDefinitionISCMROConnection	CONNDEF	ISC/MRO 연결 정의
CICSDefinitionJournalModel	JRNMDEF	저널 모델 정의
CICSDefinitionLIBRARY	LIBDEF	LIBRARY 정의
CICSDefinitionLSRPool	LSRPOOL	로컬 공유 자원 풀 정의
CICSDefinitionMapSet	MAPDEF	맵 세트 정의
CICSDefinitionMQConnection	MQCONDEF	IBM MQ 연결 정의
CICSDefinitionPartitionSet	PRTNDEF	파티션 세트 정의
CICSDefinitionPartner	PARTDEF	참여자 정의
CICSDefinitionPipeline	PIPEDEF	파이프라인 정의
CICSDefinitionProcessType	PROCDEF	프로세스 유형 정의
CICSDefinitionProfile	PROFDEF	프로파일 정의
CICSDefinitionProgram	PROGDEF	프로그램 정의
CICSDefinitionRequestModel	RQMDEF	요청 모델 정의
CICSDefinitionSession	SESSDEF	세션 정의

표 2. CICS 정의 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSDefinitionTCIPService	TCPDEF	TCP/IP 서비스 정의
CICSDefinitionTDQueue	TDQDEF	트랜지언트 데이터 큐 정의
CICSDefinitionTerminal	TERMDEF	터미널 정의
CICSDefinitionTransaction	TRANDEF	트랜잭션 정의
CICSDefinitionTransactionClass	TRNCLDEF	트랜잭션 클래스 정의
CICSDefinitionTSMModel	TSMDEF	임시 스토리지 모델 정의
CICSDefinitionTypeterm	TYPTMDEF	Typeterm 정의
CICSDefinitionURIMap	URIMPDEF	URI 맵핑 정의
CICSDefinitionWebService	WEBSVDEF	웹 서비스 정의

CICSplex SM 정의 자원

표 3. CICSplex SM 정의 자원의 자원 이름		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSActiveAnalysisPointSpecification	APSPEC	RTA 분석 위치 스펙
CICSApplicationDefinition	APPLDEF	애플리케이션 정의
CICSBatchrepRequest	BATCHREP	일괄처리 저장소-업데이트 작업
CICSCICSplexDefinition	CPLEXDEF	CICSplex 정의
CICSCMASToCMASLinkDefinition	CMTCMDEF	CMAS 대 CMAS 링크 정의
CICSMonitorDefinition	MONDEF	모니터 정의
CICSMonitorGroup	MONGROUP	모니터 그룹
CICSMonitorGroupInSpecification	MONINSPC	모니터 스펙의 모니터 그룹
CICSMonitorResourceInGroup	MONINGRP	그룹의 모니터 정의
CICSMonitorSpecification	MONSPEC	모니터 스펙
CICSMonitorSpecificationsToSystem	LNKSMSCS	CICS 시스템 링크에 대한 모니터 스펙
CICSMonitorSpecificationsToSystemGroup	LNKSMSCG	시스템 그룹 링크에 대한 모니터 스펙
CICSPeriodDefinition	PERIODEF	기간 정의
CICSPlatformDefinition	PLATDEF	플랫폼 정의
CICSPrimaryAnalysisPointCMAS	CMDMPAPS	1차 CMAS 분석 위치 스펙
CICSRegionDefinition	CSYSDEF	CICS 리전 정의
CICSRegionGroup	CSYSGRP	CICS 시스템 그룹 정의
CICSResourceAssignmentDefinition	RASGNDEF	자원 지정 정의
CICSResourceAssignmentInDescription	RASINDSC	자원 설명의 자원 지정
CICSResourceDescription	RESDESC	자원 설명 정의
CICSResourceGroup	RESGROUP	자원 그룹 정의

표 3. CICSplex SM 정의 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSResourceInGroup	RESINGRP	자원 그룹과 연관된 자원
CICSResourceGroupInDescription	RESINDSC	자원 설명의 자원 그룹
CICSRTAActionDefinition	ACTION	RTA 조치 정의
CICSRTADefinition	RTADEF	RTA 정의
CICSRTAEvaluationDefinition	EVALDEF	평가 정의
CICSRTAGroup	RTAGROUP	RTA 그룹 정의
CICSRTAResourceInGroup	RTAINGRP	RTA 그룹의 RTA 정의
CICSRTASpecification	RTASPEC	RTA 스펙 정의
CICSRTAStatusDefinition	STATDEF	RTA 상태 정의
CICSRTAStatusDefinitionInGroup	STAINGRP	RTA 그룹의 상태 정의
CICSSecondaryAnalysisPointCMAS	CMDMSAPS	2차 CMAS 분석 위치 스펙
CICSSystemGroupToSystemGroup	CSGLCGCS	CICS 시스템 대 그룹 링크
CICSSystemLink	SYSLINK	CICS 시스템 링크 정의
CICSSystemToSystemGroup	CSGLCGCG	CICS 시스템 그룹 대 그룹 링크
CICSTransactionGroup	TRANGRP	CICS 트랜잭션 그룹 정의
CICSTransactionInGroup	DTRINGRP	트랜잭션 그룹의 트랜잭션
CICSWLMActiveRouter	WLMAROUT	활성 워크로드에서 CICS 라우터 리전
CICSWLMDefinition	WLMDEF	WLM 정의
CICSWLMGroup	WLMGROUP	WLM 그룹 정의
CICSWLMResourceInGroup	WLMINGRP	워크로드 그룹의 WLM 정의
CICSWLMSpecification	WLMSPEC	WLM 스펙

CICSplex SM 비정의의 자원

표 4. CICSplex SM 비정의 자원의 자원 이름		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSActiveAnalysisPoint	APACTV	분석 위치 스펙과 연관된 설치된 분석 정의
CICSApplication	APPLCTN	배치된 애플리케이션
CICSCICSManagingAddressSpace	CMAS	활성 CMAS
CICSCICSManagingAddressSpaceList	CMASLIST	로컬 CMAS에 알려진 CMAS
CICSCICSplex	CICSplex	CMAS에서 관리하는 CICSplex
CICSCICSplexList	CMASplex	로컬 CMAS에서 관리 중인 모든 CICSplex
CICSCMASInCICSplex	CPLXCMAS	CICSplex 정의의 CMAS
CICSCMASToCMASLink	CMTCLNK	CMAS 대 CMAS 링크
CICSManagedRegion	MAS	CICSplex SM 관리 CICS 리전

표 4. CICSplex SM 비정의 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSManagedRegionStatus	MASSTAT	CMAS별 관리 대상 CICS 리전의 상태
CICSManagementPart	MGMTPART	애플리케이션 또는 플랫폼의 관리 파트
CICSPlatform	PLATFORM	플랫폼
CICSResourceByAssignment	RASPROC	자원 지정에서 선택된 자원
CICSResourceByDescription	RDSCPROC	자원 설명에서 선택된 자원
CICSResourceBySystem	SYSRES	CICS 시스템에 지정된 자원
CICSRTAEvent	EVENT	CICSplex 또는 해당되는 CICS 시스템 중 하나의 상태에서의 중요한 미해결 변경
CICSRTAEventDetail	EVENTDTL	이벤트 생성에 참여한 평가
CICSRTAInstalledDefinition	RTAACTV	RTA 설치 분석 및 상태 정의
CICSRule	RULE	정책 규칙 정보
CICSTaskHistory	HTASK	완료 태스크
CICSTaskHistoryCollection	MASHIST	CICS 태스크 히스토리 컬렉션
CICSTopologyAtomService	CRESATOM	CICS 시스템에 있는 Atom 서비스
CICSTopologyAutoInstallModel	CRESAIMD	자동 설치 모델에 대한 토폴로지 데이터
CICSTopologyBundle	CRESBUND	CICS 시스템에 있는 번들
CICSTopologyBTSPProcessType	CRESPTY	CICS 시스템에 있는 BTS 프로세스 유형
CICSTopologyCaptureSpecification	CRESEVCS	이벤트 처리 캡처 스펙
CICSTopologyDataSet	CRESDSNM	CICS 시스템에 있는 데이터 세트
CICSTopologyDb2Connection	CRESDB2C	CICS 시스템에서 Db2 연결
CICSTopologyDb2Entry	CRESDB2E	CICS 시스템에 있는 Db2 항목
CICSTopologyDb2Transaction	CRESDB2T	CICS 시스템에 있는 Db2 트랜잭션
CICSTopologyDocumentTemplate	CRESDOCT	CICS 시스템에 있는 문서 템플릿
CICSTopologyEnqueueModel	CRESENQM	CICS 시스템에 있는 ENQ/DEQ 모델
CICSTopologyEPAdapter	CRESEPAD	이벤트 처리 어댑터
CICSTopologyEPAdapterSet	CRESEPAS	이벤트 처리 어댑터 세트
CICSTopologyEventBinding	CRESEVBD	CICS 시스템에 있는 이벤트 바인딩
CICSTopologyFEPIConnection	CRESFECO	CICS 시스템에 있는 FEPI 연결
CICSTopologyFEPINode	CRESFEND	CICS 시스템에 있는 FEPI 노드
CICSTopologyFEPIPools	CRESFEPOL	CICS 시스템에 있는 FEPI 풀
CICSTopologyFEPITarget	CRESFETR	CICS 시스템에 있는 FEPI 대상
CICSTopologyFile	CRESFIL	CICS 시스템에 있는 파일
CICSTopologyGlobalUserExit	CRESGLUE	CICS 시스템에 있는 글로벌 사용자 엑시트
CICSTopologyIPICConnection	CRESI PCN	CICS 시스템에 있는 IPIC 연결
CICSTopologyISCMROConnection	CRESCONN	CICS 시스템에 있는 ISC/MRO 연결

표 4. CICSplex SM 비정의 자원의 자원 이름 (계속)		
외부 자원 이름	CICSplex SM 자원 이름	설명
CICSTopologyJournalName	CRESJRNM	CICS 시스템에 있는 저널 이름
CICSTopologyJVMServer	CRESJVMS	CICS 시스템에 있는 JVM 서버
CICSTopologyLibrary	CRESLIBR	LIBRARY에 대한 토폴로지 데이터
CICSTopologyLU62ModeName	CRESMODE	CICS 시스템에 있는 LU62 모드 이름
CICSTopologyPartner	CRESPART	CICS 시스템에 있는 파트너
CICSTopologyPipeline	CRESPIPE	PIPELINE에 대한 토폴로지 데이터
CICSTopologyProfile	CRESPROF	CICS 시스템에 있는 프로파일
CICSTopologyProgram	CRESPRGM	CICS 시스템에 있는 프로그램
CICSTopologyRequestModel	CRESRQMD	CICS 시스템에 있는 요청 모델
CICSTopologySystemDump	CRESSDMP	CICS 시스템에 있는 시스템 덤프
CICSTopologyTaskRelatedExit	CRESTRUE	CICS 시스템에 있는 태스크 관련 엑시트
CICSTopologyTCPIPService	CRESTCPS	CICS 시스템에 있는 TCP/IP 서비스
CICSTopologyTDQueue	CRESTDQ	CICS 시스템에 있는 트랜지언트 데이터 큐
CICSTopologyTerminal	CRESTERM	CICS 시스템에 있는 터미널
CICSTopologyTransaction	CRESTRAN	CICS 시스템에 있는 트랜잭션
CICSTopologyTransactionDump	CRESTDMP	CICS 시스템에 있는 트랜잭션 덤프
CICSTopologyTSModel	CRESTSMD	CICS 시스템에 있는 임시 스토리지 모델
CICSTopologyUrimap	CRESURIM	URIMAP에 대한 토폴로지 데이터
CICSTopologyWebservice	CRESWEBS	WEBSERVICE에 대한 토폴로지 데이터
CICSTopologyXMLTransform	CRESXMLT	XML 변환에 대한 토폴로지 데이터
CICSWLMActiveAffinity	WLMATAFF	활성 워크로드 트랜잭션 그룹 연관관계
CICSWLMActiveAOR	WLMAWAOR	활성 워크로드 대상 분배 요인
CICSWLMActiveDefinition	WLMAWDEF	활성 워크로드 정의
CICSWLMActiveTOR	WLMAWTOR	활성 워크로드 라우팅 리전
CICSWLMActiveTransactionGroup	WLMATGRP	활성 워크로드 트랜잭션 그룹
CICSWLMActiveTransaction	WLMATRAN	활성 워크로드 동적 트랜잭션
CICSWLMActiveWorkload	WLMAWORK	활성 워크로드
CICSWLMTarget	WLMATARG	하나 이상의 활성 워크로드에 대한 대상 리전

CMCI 요청 및 응답은 XML 형식의 정보를 포함합니다. CMCI의 고유한 XML 요소는 이름이 CICSSystemManagement인 스키마에서 정의됩니다. 이 스키마는 각 자원 유형의 개별 스키마를 통합하여 구성됩니다.

클라이언트 애플리케이션은 CICSSystemManagement 스키마에 액세스하여 CICS 관리 클라이언트 인터페이스 요청에 대한 응답 본문에서 정보의 유효성을 검사하고 형식화해야 합니다. 클라이언트는 `http://hostname:portnumber/CICSSystemManagement/schema/CICSSystemManagement.xsd` URI에서 이 출력 XML을 정의하는 스키마에 액세스할 수 있습니다.

참고: CMCI는 EBCDIC 037 코드 페이지에 의해 처리되는 문자만 해석할 수 있습니다. CMCI 요청에 다른 문자를 사용하면 예기치 않은 결과가 발생합니다.

<request> 요소

<request> 요소는 CICS 관리 클라이언트 인터페이스 POST 또는 PUT 요청의 본문 XML에 대한 루트 요소입니다.

POST 요청에서, <request> 요소에는 하나의 하위 <create> 요소가 포함됩니다. PUT 요청에서, <request> 요소는 예제에서 표시된 대로, 하나의 하위 <update> 요소 또는 하나의 <action> 요소를 포함할 수 있습니다.

DELETE 요청에서, <request> 요소는 일반적으로 비어 있습니다. 지정되면, <request> 요소는 하나의 하위 <delete> 요소를 포함해야 합니다.

CMCI 요청은 하나의 <request> 요소만 포함할 수 있습니다.

포함 대상:

POST 요청에서: 하나의 [27 페이지의 『<create> 요소』](#)

PUT 요청에서: 하나의 [28 페이지의 『<update> 요소』](#) 또는 하나의 [29 페이지의 『<action> 요소』](#)

DELETE 요청에서: 하나의 [30 페이지의 『<delete> 요소』](#)

예

다음 예제는 이름이 ABCD인 BAS 자원에 대한 설치 조치를 표시합니다. <action> 요소는 <request> 요소에 포함된다는 점을 참고하십시오.

```
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> 요소

<create> 요소는 POST 요청에 <parameter> 및 <attributes> 요소를 포함합니다.

다음에 포함됨:

[27 페이지의 『<request> 요소』](#)

포함 대상:

<create> 요소는 하나의 <attributes> 요소를 포함해야 하며, 하나 이상의 <parameter> 요소를 포함할 수 있습니다. POST 요청은 하나의 <create> 요소만 포함할 수 있습니다.

[30 페이지의 『<parameter> 요소』](#)

예제

다음 예제에서는 이름이 ABCD인 BAS 자원 작성을 표시합니다.

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

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

다음 예제에서는 GROUP1 그룹에서 이름이 ABCD인 BAS 자원 작성을 표시합니다.

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

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

다음 예제에서는 GROUP2 그룹에서 이름이 ABCD인 CSD 자원 작성을 표시합니다.

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

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

<update> 요소

<update> 요소는 PUT 요청에 <parameter> 및 <attributes> 요소를 포함합니다.

다음에 포함됨:

27 페이지의 『<request> 요소』

포함 대상:

<update> 요소는 하나의 <attributes> 요소를 포함해야 하며, 하나 이상의 <parameter> 요소를 포함할 수 있습니다. PUT 요청은 하나의 <update> 요소만 포함할 수 있습니다.

30 페이지의 『<parameter> 요소』

31 페이지의 『<attributes> 요소』

예

다음 예제에서는 이름이 ABCD인 BAS 트랜잭션 정의를 shutdown="DISABLED", isolate="YES", dynamic="NO" 값을 보유하도록 업데이트합니다.

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

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

다음 예제에서는 GROUP2 그룹에서 이름이 ABCD인 CSD 자원의 description을 업데이트합니다.

```
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> 요소

<action> 요소는 URI에 의해 지정된 자원에서 CICS 관리 클라이언트 인터페이스 PUT 요청이 수행할 조치에 대한 세부사항을 포함합니다.

다음에 포함됨:

[27 페이지의 『<request> 요소』](#)

속성

name="action_name"

<action> 요소는 수행할 조치를 식별하는 하나의 **name** 속성만 포함해야 합니다.

*action_name*을 일치하는 작은따옴표 또는 큰따옴표로 묶어야 합니다. XML 코딩 규칙에 따라, 이스케이프 시퀀스를 사용하여 특정 특수 문자(예: &, <, >)를 대체할 수 있습니다. 조치에 연관된 매개변수가 있으면 <parameter> 요소를 사용하여 <action>의 하위 요소로 이를 지정해야 합니다.

HTTP 본문에서 <action> 요소를 지정하는 경우 <attributes> 요소를 동일한 요청에 포함시키지 않아야 합니다.

포함 대상:

<action> 요소는 하나 이상의 하위 <parameter> 요소를 포함할 수 있습니다.

[30 페이지의 『<parameter> 요소』](#)

예제

다음 예제는 이름이 ABCD인 BAS 자원에 대한 설치를 표시합니다.

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

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

다음 예제는 CSD 그룹, GROUP4가 CSD 그룹, GROUP3 이후, CSD 목록, LIST2에 추가되는 CSDADD 조치를 표시합니다.

CSDADD 조치에 대해 지정된 매개변수는 TO_CSDLIST, ADD_CSDGROUP, ADD_LOCATION과 같습니다.

TO_CSDLIST는 CSD 그룹을 추가할 CSD 목록을 지정합니다.

ADD_CSDGROUP은 CSD 목록 옆에 있는 제공된 CSD 그룹을 추가할 CSD 그룹을 지정합니다.

ADD_LOCATION은 CSD 목록의 기존 CSD 그룹 "BEFORE"(앞) 또는 "AFTER"(뒤) 위치에 제공된 CSD 그룹을 추가할지 여부를 지정합니다.

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

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

```
<parameter name="ADD_LOCATION" value="AFTER"/>
</action>
</request>
```

다음 예제는 CICS 운영 자원, RESOURCE에서 DISABLE 조치를 표시합니다. 자원의 유형이 URI에 포함된다는 점을 참고하십시오. 이 경우 자원 유형은 CICSAtomService입니다.

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

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

<delete> 요소

<delete> 요소는 URI에 의해 지정된 자원에서 CICS 관리 클라이언트 인터페이스 DELETE 요청이 수행할 삭제에 대한 세부사항을 포함합니다.

다음에 포함됨:

27 페이지의 『<request> 요소』

포함 대상:

일반적으로 <delete> 요소는 매개변수를 포함하지 않으므로 필요하지 않습니다.

30 페이지의 『<parameter> 요소』

CSD 정의의 자원을 삭제하는 경우, PARAMETER=CSDGROUP을 URI에 포함해야 합니다. 또한 XML 본문에 <parameter name="CSD"/>를 포함할 수 있지만, 이는 필수가 아닙니다. URI에 PARAMETER=CSDGROUP을 지정하고 XML 본문에 <parameter name="CSD"/>를 지정하지 않은 경우, CICS에서는 <parameter name="CSD"/>가 지정되었다고 가정합니다.

예제

다음 예제는 이름이 ABCD인 BAS 자원 정의에 대한 삭제 요청을 표시합니다. 삭제 요청에 대해 URI만 필요합니다.

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

다음 예제는 CSD 자원 정의에 대한 삭제 요청을 표시합니다. PARAMETER=CSDGROUP이 URI에 포함된다는 점을 참고하십시오.

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

다음 예제는 XML 본문에 선택적 <delete> 요소가 포함된다는 점을 제외하고 이전 예제와 같습니다.

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

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

<parameter> 요소

<parameter> 요소는 CICS 관리 클라이언트 인터페이스 POST 또는 PUT 요청과 연관된 매개변수의 이름 및 값을 포함합니다. POST 또는 PUT 요청은 하나 이상의 parameter 요소를 포함할 수 있습니다.

다음에 포함됨:

POST 요청에서: 27 페이지의 『<create> 요소』

PUT 요청에서: 하나의 29 페이지의 『<action> 요소』 또는 하나의 28 페이지의 『<update> 요소』

속성

name="parameter_name"

조치와 연관된 매개변수의 이름입니다. *parameter_name*은 대소문자를 구분하고, 일치하는 작은따옴표 또는 큰따옴표로 묶어야 합니다.

value="parameter_value"

이름 지정된 매개변수의 값입니다. *parameter_value*는 대소문자를 구분하고, 일치하는 작은따옴표 또는 큰따옴표로 묶어야 합니다.

예

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

<attributes> 요소

<attributes> 요소는 CICS 관리 클라이언트 인터페이스 POST 또는 PUT 요청에서 속성의 새 값에 대한 세부사항을 포함합니다. 요청은 하나의 <attributes> 요소만 포함할 수 있습니다.

다음에 포함됨:

27 페이지의 『<request> 요소』

속성

attribute_name="data_value"

<attributes> 요소는 이름 지정된 속성에 제공할 새 값을 지정하는 하나 이상의 *attribute_name* "data_value" 조합을 포함해야 합니다. 각 자원의 필수 속성 목록은 [CICSplex SM 자원 테이블](#)의 내용을 참조하십시오.

data_value 속성은 대소문자를 구분하고, 일치하는 작은따옴표 또는 큰따옴표로 묶어야 합니다. XML 코딩 규칙에 따라, 이스케이프 시퀀스를 사용하여 특정 특수 문자(예: &, <, >)를 대체할 수 있습니다.

HTTP 본문에서 <attributes> 요소를 지정하는 경우 <action> 요소를 동일한 요청에 포함시키지 않아야 합니다.

예

```
<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" otstimeout="NO" wait="YES" ressec="NO" cmdsec="NO"
program="ABCDEFGH" />
```

<response> 요소

<response> 요소는 CICS 관리 클라이언트 인터페이스 요청에 대한 응답에서 루트 요소입니다. <response> 요소는 세 개까지 하위 요소를 포함할 수 있습니다(필수 <resultsummary> 요소, 요청에서 리턴되는 레코드를 포함하는 <records> 요소, 요청이 오류를 리턴하는 경우 진단 정보를 포함하는 <errors> 요소).

포함 대상:

32 페이지의 『<resultsummary> 요소』

34 페이지의 『<errors> 요소』

33 페이지의 『<records> 요소』

속성

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

CICS 관리 클라이언트 인터페이스 네임스페이스입니다.

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

CICS 관리 클라이언트 인터페이스 XML 스키마의 위치입니다.

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

XML 스키마 인스턴스 네임스페이스입니다.

version="schema_version"

CICS 관리 클라이언트 인터페이스 스키마의 버전 번호입니다. 이 번호는 1.0에서 시작하고 스키마가 업데이트될 때마다 증가합니다.

connect_version="connect_version"

EXEC CPSM CONNECT API 명령을 사용하여 CICSplex SM에 연결할 때 사용된 CICSplex SM 릴리스 번호입니다.

예

```
<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> 요소

<resultsummary> 요소는 CICS 관리 클라이언트 인터페이스 요청과 연관된 요약 정보를 포함합니다. CICS 관리 클라이언트 인터페이스에 대한 응답은 <resultsummary> 요소를 포함해야 합니다.

다음에 포함됨:

[31 페이지의 『<response> 요소』](#)

속성

api_source="value"

요청에 실패하면 API 소스의 이름을 포함하십시오(예: "CICSplex SM" 또는 "CMCI"). 이 속성은 성공한 요청에 대한 응답에 없습니다.

api_function="function"

요청에 실패하면 CICS 시스템 명령 또는 실패한 CICSplex SM API 명령의 이름을 포함하십시오(예: PERFORM SET). 이 속성은 성공한 요청에 대한 응답에 없습니다.

api_response1="value"

CICS EIBRESP 코드 또는 숫자 값인 CICSplex SM API EYUDA 응답 코드입니다. CICS RESP 값에 대한 자세한 정보는 [RESP 및 RESP2 옵션](#), CICSplex SM API 응답 코드에 대한 정보는 [EYUDA 값의 내용](#)을 참조하십시오.

api_response1_alt="value"

CICS EIBRESP 코드 또는 텍스트 문자열인 CICSplex SM API EYUDA 응답 코드입니다. CICS RESP 값에 대한 자세한 정보는 [RESP 및 RESP2 옵션](#), CICSplex SM API 응답 코드에 대한 정보는 [EYUDA 값의 내용](#)을 참조하십시오.

api_response2="value"

CICS EIBRESP2 코드 또는 숫자 값인 CICSplex SM API EYUDA 이유 코드입니다. 성공적인 요청 값은 0입니다. CICS RESP2 값에 대한 자세한 정보는 [RESP 및 RESP2 옵션](#), CICSplex SM API 이유 코드에 대한 정보는 [EYUDA 값의 내용](#)을 참조하십시오.

api_response2_alt="value"

CICS EIBRESP2 코드 또는 텍스트 문자열인 CICSplex SM API EYUDA 이유 코드입니다. CICS RESP2 값에 대한 자세한 정보는 [RESP 및 RESP2 옵션](#), CICSplex SM API 이유 코드에 대한 정보는 [EYUDA 값의 내용](#)을 참조하십시오.

recordcount="value"

요청에서 리턴하는 총 레코드 수입니다.

recordcount 속성은 요청에 실패한 항목을 포함하여 항상 결과에 있는 총 레코드 수를 표시합니다. 레코드 범위를 지정하면 **recordcount** 값은 요청에서 리턴하는 레코드 수보다 많을 수 있습니다.

displayed_recordcount="value"

요청에서 리턴하는 레코드 개수입니다.

레코드 범위를 지정하면 **displayed_recordcount** 값은 **recordcount**에서 표시한 대로, 결과의 총 레코드 수보다 적을 수 있습니다.

successcount="value"

DELETE 요청의 경우에만 요청이 성공한 레코드 개수입니다.

cachetoken="value"

보유된 결과 세트를 포함하는 결과 캐시를 식별하는 토큰입니다. **cachetoken** 속성은 NODISCARD 옵션을 지정하는 GET 요청에 대한 응답에만 존재합니다. **cachetoken** 값은 저장소 클럭 값의 16자의 고정 길이 16진 표현입니다.

예제

다음 예제는 성공적인 DELETE 요청에 대한 결과를 표시합니다.

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

다음 예제는 실패한 DELETE 요청에 대한 결과를 표시합니다.

```
<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"/>
```

다음 예제는 NODISCARD 옵션을 포함하는 GET 요청의 결과를 표시합니다.

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

<records> 요소

<records> 요소는 요청에서 리턴하는 레코드의 세부사항을 포함합니다. <records> 요소는 요청에서 리턴하는 각 레코드에 대해 하나의 하위 요소를 포함합니다. 요청이 레코드를 리턴하지 않는 경우 <records> 요소가 없습니다.

다음에 포함됨:

[31 페이지의 『<response> 요소』](#)

포함 대상:

<records> 요소는 요청에서 변경하는 자원의 각 인스턴스에 대해 하나의 하위 요소를 포함합니다. 이러한 하위 요소는 요청 URI에서 식별된 자원에 의해 이름 지정됩니다. 예를 들어, PUT 요청이 CICS 로컬 파일을 변경하면 PUT 응답의 <records> 요소는 이름이 <cicslocalfile>인 하나 이상의 하위 요소를 포함합니다(즉, 요청에서 리턴하는 각 레코드에 대한 하나의 <cicslocalfile> 요소). 하위 요소 이름은 CICS 관리 클라이언트 인터페이스 자원 이름에서 파생됩니다. 이러한 이름의 전체 목록은 [17 페이지의 『CMCI 자원 이름』](#)의 내용을 참조하십시오.

속성

<records> 요소에는 고유한 속성이 없습니다.

하위 요소의 속성은 식별된 자원의 속성 값을 지정합니다. 속성의 값이 변했는지 여부에 상관없이, 자원의 모든 속성이 요소에 존재합니다. 값이 공백인 속성은 기본값을 사용합니다. 각 자원의 필수 속성 목록은 [CICSplex SM 자원 테이블](#)의 내용을 참조하십시오.

각 자원 `_keydata="data string"` 양식을 사용하는 **_keydata** 속성도 포함합니다. 여기서, *data string*은 자원의 인스턴스를 고유하게 식별하는 문자열입니다.

예

다음 예제는 CICSLocalFile 자원의 인스턴스를 나열하는 단일 레코드를 표시합니다.

```
<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" dexcpnt="0" disposition="SHARE" dsname="USER1.ALLAPPL.DFHCS"
emptystatus="NOEMPTYREQ" enablestatus="ENABLED" exclusive="NOTAPPLIC" file="DFHCS"
fwdrecstatus="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="UPDATABL" updatecnt="0" vsamtype="NOTAPPLIC" wstrccurcnt="0" wstricnt="0"
eyu_cicsname="U1WUIA" eyu_cicsrel="E660" eyu_reserved="0"/>
</records>
```

<errors> 요소

<errors> 요소는 CICS 관리 클라이언트 인터페이스 요청과 연관된 진단 정보를 포함합니다. CICS 관리 클라이언트 인터페이스 요청에 대한 응답은 하위 요소에 표시할 진단 정보가 있는 경우 <errors> 요소를 포함합니다.

<errors> 요소는 요청과 연관된 하나 이상의 <feedback> 하위 요소를 포함합니다. <errors> 요소에는 연관된 속성이 없습니다.

다음에 포함됨:

[31 페이지의 『<response> 요소』](#)

포함 대상:

[34 페이지의 『<feedback> 요소』](#)

예제

<feedback> 요소

<feedback> 요소는 CICS 관리 클라이언트 인터페이스 요청과 연관된 FEEDBACK 레코드의 진단 데이터를 포함합니다. <errors> 요소는 요청에서 리턴된 각 레코드에 대한 하나 이상의 <feedback> 요소를 포함할 수 있습니다.

FEEDBACK 레코드에 대한 자세한 정보는 [피드백 레코드 검색](#)의 내용을 참조하십시오.

CICS 관리 클라이언트 인터페이스 PUT 설치 요청의 경우 <feedback> 요소는 다음과 같은 하위 요소를 포함할 수 있습니다.

- BINSTERR 레코드를 포함하는, 실패한 각 설치 조치에 대한 하나의 <installerror> 요소
- BINCONSC 레코드를 포함하는, 실패한 각 설치 조치에 대한 하나의 <inconsistentscope> 요소
- BINCONRS 레코드를 포함하는, 실패한 각 설치 조치에 대한 하나의 <inconsistentset> 요소

다음에 포함됨:

34 페이지의 『<errors> 요소』

포함 대상:

36 페이지의 『<installerror> 요소』

36 페이지의 『<inconsistentscope> 요소』

37 페이지의 『<inconsistentset> 요소』

속성

action="action_name"

실패한 조치의 이름입니다.

attribute1="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

attribute2="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

attribute3="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

attribute4="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

attribute5="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

attribute6="attribute_name"

오류와 연관된 최대 6개 속성 중 하나의 이름입니다.

eibfn="eibfn_number"

요청과 연관된 기능 코드입니다.

eibfn_alt="function"

요청과 연관된 기능의 이름입니다.

errorcode="error_code"

자원과 연관된 CICSplex SM 오류 코드입니다.

eyu_cicsname="name"

오류와 연관된 CICS 리전 또는 CICSplex의 이름입니다.

keydata="data"

오류와 연관된 자원의 인스턴스를 식별하는 데이터 문자열입니다.

resp="resp_number"

CICS RESP 코드 또는 숫자 값인 CICSplex SM API EYUDA 응답 코드입니다.

resp2="resp2_number"

CICS RESP2 코드 또는 숫자 값인 CICSplex SM API EYUDA 이유 코드입니다.

resp_alt="resp_alt_text"

resp 값과 동등한 텍스트입니다. 예를 들어, resp 값, 16과 동등한 텍스트는 INVREQ입니다.

예

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

<installerror> 요소

<installerror> 요소는 CICS 관리 클라이언트 인터페이스 PUT 요청과 연관된 BINSTERR 레코드의 진단 데이터를 포함합니다. <installerror> 요소는 <feedback> 요소에 의해 포함됩니다.

BINSTERR 레코드에 대한 자세한 정보는 [BINSTERR 자원 테이블 레코드 평가](#)의 내용을 참조하십시오.

다음에 포함됨:

[34 페이지의 『<feedback> 요소』](#)

속성

eibfn="eibfn_number"

요청과 연관된 기능 코드입니다.

eyu_cicsname="name"

설치 오류와 연관된 CICS 리전 또는 CICSplex의 이름입니다.

cresp1="resp_number"

CICS RESP 코드 또는 숫자 값인 CICSplex SM API EYUDA 응답 코드입니다.

cresp2="resp2_number"

CICS RESP2 코드 또는 숫자 값인 CICSplex SM API EYUDA 이유 코드입니다.

errorcode="error_code"

자원과 연관된 CICSplex SM 오류 코드입니다.

ressname="resource_name"

오류와 연관된 자원의 이름입니다.

resver="resource_version"

오류와 연관된 자원의 버전 번호입니다.

예

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

<inconsistentscope> 요소

<inconsistentscope> 요소는 CICS 관리 클라이언트 인터페이스 PUT 요청과 연관된 BINCONSC 레코드의 진단 데이터를 포함합니다. <inconsistentscope> 요소는 <feedback> 요소에 의해 포함됩니다.

BINCONSC 레코드에 대한 자세한 정보는 [BINCONSC 자원 테이블 레코드 평가](#)의 내용을 참조하십시오.

다음에 포함됨:

[34 페이지의 『<feedback> 요소』](#)

속성

eibfn="eibfn_number"

요청과 연관된 기능 코드입니다.

eyu_cicsname="name"

일관되지 않은 범위 오류와 연관된 CICS 리전의 이름입니다.

erroroperation="value"

오류가 발생한 경우 수행할 오퍼레이션을 식별하는 숫자 값

errorcode="error_code"

자원과 연관된 CICSplex SM 오류 코드입니다.

targetassignment="assignment"

대상 범위의 지정입니다.

targetdescription="description"

대상 범위에 대한 자원 설명입니다.

relatedassignment="assignment"

관련 범위에 대한 자원 지정입니다.

relateddescription="assignment"

관련 범위에 대한 자원 설명입니다.

relatedscope="scope"

관련 범위의 이름입니다.

예

BINCONSC의 경우해 자원의 대상 범위 및 관련 범위에 대해 생성되는 오류 코드는 동일하게 1이므로 자원 설명에 자원 지정을 추가하려는 시도에 실패합니다.

```
<?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> 요소

<inconsistentset> 요소는 CICS 관리 클라이언트 인터페이스 PUT 요청과 연관된 BINCONRS 레코드의 진단 데이터를 포함합니다. <inconsistentset> 요소는 <feedback> 요소에 의해 포함됩니다.

BINCONRS 레코드에 대한 자세한 정보는 [BINCONRS 자원 테이블 레코드 평가](#)의 내용을 참조하십시오.

다음에 포함됨:

34 페이지의 『<feedback> 요소』

속성

candidatename="resource_name"

후보 자원의 이름입니다.

candidateversion="resource_version"

후보 자원의 버전 번호입니다.

candidategroup="group_name"

후보 자원의 자원 그룹입니다.

candidateassignment="assignment"

후보 자원의 지정입니다.

candidatedescription="assignment"

후보 자원에 대한 설명입니다.

candidateusage="usage"

후보 자원의 지정 사용법입니다.

candidatesystemgroup="group_name"

후보 자원의 시스템 그룹입니다.

candidatetype="system_type"

후보 자원의 시스템 유형입니다.

candidateoverride="assignment_override"

후보 자원의 지정 대체입니다.

eyu_cicsname="name"

설치 오류와 연관된 CICS 리전의 이름입니다.

erroroperation="value"

오류가 발생한 경우 수행할 오퍼레이션을 식별하는 숫자 값

existingname="resource_name"

기존 자원의 이름입니다.

existingversion="resource_version"

기존 자원의 버전 번호입니다.

existinggroup="group_name"

기존 자원의 자원 그룹입니다.

existingassignment="assignment"

기존 자원의 지정입니다.

existingdescription="assignment"

기존 자원에 대한 설명입니다.

existingusage="usage"

기존 자원의 지정 사용법입니다.

existingsystemgroup="group_name"

기존 자원의 시스템 그룹입니다.

existingtype="system_type"

기존 자원의 시스템 유형입니다.

existingoverride="assignment_override"

기존 자원의 지정 대체입니다.

예

자원 설명과 연관된 기존 자원 지정의 충돌로 인해 자원 설명에 자원 지정을 추가하려는 시도에 실패합니다.

```
<?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 이스케이프 시퀀스

특정 문자는 CMCI URI에서 제한됩니다. 이러한 문자의 경우 %(퍼센트) 문자 및 16진수 값의 순서로 구성된 이스케이프 시퀀스를 대체해야 합니다.

다음 표에서는 제한된 문자 및 해당 UTF-8 ASCII 16진 이스케이프 시퀀스를 나열합니다.

표 5. 제한된 문자 및 해당 이스케이프 시퀀스			
문자	이스케이프 순서	문자	이스케이프 순서
SPACE	%20	. (마침표)	%2E
!	%21	/	%2F
#	%23	:(콜론)	%3A
\$	%24	;(세미콜론)	%3B
%	%25	<	%3C
&	%26	=	%3D
'(작은따옴표)	%27	>	%3E
(%28	?	%3F
)	%29	@	%40
*	%2A	[%5B
+	%2B]	%5D
, (쉼표)	%2C	~ (부정)	%AC
-(빼기 부호)	%2D		

예제

다음 예제는 CICS 관리 클라이언트 인터페이스 URI에서 이스케이프 문자로 인코딩하기 전과 이후의 특정 문자열을 표시합니다.

이전:

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

이후:

```
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
```

다음 예제는 CICS 관리 클라이언트 인터페이스 URI에서 이스케이프 문자로 인코딩하기 전과 이후의 매개변수 문자열을 표시합니다.

이전:

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

이후:

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

또한 인터페이스는 기준 및 매개변수 문자열의 끝에서 선택적 기간(%2E)도 지원합니다.

CMCI 문제점 판별

CICS 관리 클라이언트 인터페이스(CMCI)는 오류의 원인을 판별하는 데 유용한 리턴코드 및 몇 가지 유형의 진단 정보를 제공합니다.

CICS 관리 클라이언트 인터페이스가 유효하지 않는 클라이언트로부터 요청을 수신하면, 요청에 실패하고 API는 응답의 HTTP 헤더에서 비정상 코드를 리턴합니다. CICS 관리 클라이언트 인터페이스는 요청을 처리하지 않고, 다른 정보도 제공하지 않습니다. 이러한 응답 코드의 설명은 [44 페이지의 『CICS 관리 클라이언트 인터페이스 오류 메시지』](#)의 내용을 참조하십시오.

CICS 관리 클라이언트 인터페이스가 요청이 유효함을 판별하면, HTTP 200 정상 응답 코드를 리턴하고, EXEC CICS 또는 EXEC CPSM 호출을 생성합니다. 이 호출에서 오류가 발생하면 API는 응답의 XML 본문에서 진단 정보를 리턴합니다. 이 정보는 XML 응답의 <resultsummary> 요소에서 응답 및 이유 코드, 그리고 특정 경우에 추가 정보를 제공하는 피드백 레코드의 양식을 취합니다.

<resultsummary> 요소는 실패의 원인을 판별하는 데 사용할 수 있는 4개의 속성을 포함합니다.

- api_function
- api_response1
- api_response1_alt
- api_response2
- api_response2_alt

해당 속성에 대한 설명은 [32 페이지의 『<resultsummary> 요소』](#)의 내용을 참조하십시오.

CICS 관리 클라이언트 인터페이스에서 예외 조건을 수신하면 XML 응답의 <errors> 요소에서 클라이언트로 전달됩니다. 요청이 완료하면 <errors> 요소가 없습니다. <errors> 요소는 각각 FEEDBACK 레코드를 포함하는, 하나 이상의 <feedback> 요소를 포함합니다. 특정 BAS 오류의 경우, <feedback> 요소는 각각 BINSTERR, BINCONSC 또는 BINCONRS 레코드를 포함하는 추가 <installerror>, <inconsistentscope> 또는 <inconsistentset> 요소를 포함합니다.

예제: <resultsummary> 정보를 사용하여 CICS 관리 클라이언트 인터페이스 오류 진단

CICS 관리 클라이언트 인터페이스, <resultsummary> 요소는 오류의 원인을 판별하는 데 유용한 리턴 코드를 포함합니다.

이 태스크 정보

다음 URI 및 XML 본문에서 CICS 관리 클라이언트 인터페이스 PUT 요청을 사용하여 CICSManagedRegion 자원(즉, CICSplex SM에서 관리하는 CICS 리전 또는 MAS)에서 START 조치를 수행하려고 시도했습니다.

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

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

HTTP/1.1 200 정상 응답을 CICS 관리 클라이언트 인터페이스에서 수신해도 레코드는 리턴되지 않으며 응답은 다음 <resultsummary> 정보를 포함합니다.

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

이 문제점에 대한 이유를 이해하기 위해 다음을 수행하십시오.

프로시저

1. **api_function** 속성에서 오류 소스를 판별하십시오.

이 속성은 CICS 관리 클라이언트 인터페이스가 PERFORM SET 기능을 호출하려고 시도했지만 호출에 실패했음을 나타냅니다.

2. **api_response1_alt** 및 **api_response2_alt** 속성에서 실패 이유를 판별하십시오.

이 값은 CICSplex SM EYUDA 응답 및 이유 코드입니다.

EYUDA 응답에는 기호 이름 NOTFOUND이 있고, EYUDA 이유 값에는 기호 이름 ACTION이 있습니다. 이 값은 CICSplex SM MAS 자원 테이블에서 시도한 조치를 찾을 수 없음을 나타냅니다. MAS 자원 테이블에 대해 사용 가능한 유일한 조치는 STOP, STOPUNCON, FORCEDISCON입니다. START 조치는 허용되지 않습니다. MAS 자원에 대해 사용 가능한 조치를 확인하기 위해 [MAS 테이블](#)을 참조하십시오.

예제: <feedback> 정보를 사용하여 CICS 관리 클라이언트 인터페이스 오류 진단

CICS 관리 클라이언트 인터페이스는 CICS 관리 클라이언트 인터페이스 응답의 <feedback> 요소에서 FEEDBACK 레코드를 제공합니다. CICS 관리 클라이언트 인터페이스 요청의 문제점을 진단하기 위해 FEEDBACK 레코드에서 제공된 추가 정보를 사용할 수 있습니다.

이 태스크 정보

다음 URI에서 클라이언트 API DELETE 요청을 사용하여 CICS 로컬 파일 자원 MYFILE_A를 삭제하려고 시도했습니다.

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

요청이 올바르게 구성되었음을 나타내는 HTTP/1.1 200 정상 응답을 CICS 관리 클라이언트 인터페이스에서 수신해도 레코드는 리턴되지 않으며 응답은 다음 정보를 포함합니다.

```
<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>
```

이 문제점에 대한 이유를 이해하기 위해 다음을 수행하십시오.

프로시저

1. **api_function** 속성에서 오류 소스를 판별하십시오.

이 속성은 CICS 관리 클라이언트 인터페이스가 PERFORM SET 기능을 호출하려고 시도했지만 실패했음을 나타냅니다.

2. **api_response1_alt** 및 **api_response2_alt** 속성에서 실패 이유를 판별하십시오.

이 값은 CICSplex SM EYUDA 응답 및 이유 코드입니다.

EYUDA 응답 값은 TABLEERROR이고 EYUDA 이유 값은 DATAERROR입니다.

3. <feedback> 요소에서 **조치** 속성을 포함하는 <resultsummary> 태그의 **api_function** 속성에 있는 정보를 결합하여 실패한 API 호출을 판별하십시오.

이 정보를 사용하여 CICS 관리 클라이언트 인터페이스가 CICSplex SM API 명령 **EXEC CPSM PERFORM SET ACTION('DISCARD')**을 사용하여 파일을 삭제하려는 경우 실패했는지를 판별할 수 있습니다.

4. <feedback> 요소의 **eibfn_alt** 속성에서 연관된 EXEC CICS 명령을 식별하십시오.

eibfn_alt 값은 명령 **EXEC CICS DISCARD FILE**을 나타냅니다.

5. 실패의 정확한 원인을 판별하기 위해 <feedback> 요소의 **resp** 및 **resp2** 속성을 사용하십시오.

두 값은 10진수 형식으로 표시됩니다.

- a) CICS **resp** 값 16에 대한 기호 이름을 찾아보십시오.

EIBREP 값의 목록에 대해서는 [EIB 필드](#)의 내용을 참조하십시오.

이 목록에서 값이 16인 EIBRESP 필드에서 기호 이름이 INVREQ인지 판별할 수 있습니다.

- b) 마지막으로 **EXEC CICS DISCARD FILE** 명령에 대해 **resp2** 값이 2인 INVREQ 응답의 의미를 찾아보십시오.

INVREQ 조건에 대한 RESP2 값의 목록에 대해서는 DISCARD FILE의 내용을 참조하십시오.

이 목록에서 RESP2 값이 2인 INVREQ 조건을 리턴하는 **EXEC CICS DISCARD FILE** 명령의 실패 원인이 "파일이 닫히지 않음"인지를 판별할 수 있습니다.

예제: CICS 관리 클라이언트 인터페이스 설치 오류 진단

CICS 관리 클라이언트 인터페이스를 사용하여 BAS 자원을 설치하려면 INSTALL 조치에서 PUT 요청을 사용합니다. 설치에 실패하면 CICSplex SM은 진단 정보를 포함하는 BINSTERR 레코드를 생성합니다. 이 레코드는 <installerror> 요소에서 리턴되며, 이 요소는 <feedback>의 하위 요소입니다.

이 태스크 정보

다음 URI 및 XML 본문에서 PUT 요청을 사용하여 리전 DEVRG1에 이름이 URIMAP_A인 URIMAP 자원을 설치하려고 시도했습니다.

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

```
<request>
  <action name="INSTALL">
    <parameter name="TARGET" value="DEVRG1"/>
    <parameter name="USAGE" value="LOCAL"/>
  </action>
</request>
```

요청이 올바르게 구성되었음을 알리는 HTTP/1.1 200 정상 응답을 CICS 관리 클라이언트 인터페이스에서 수신해도 URIMAP이 설치되지 않습니다. 응답은 다음 정보를 제공합니다.

```
<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" resourcenname="URIMAP_A"
    resourceversion="1" resp="16" resp2="500"/>
  </feedback>
</errors>
```

이 문제점에 대한 이유를 이해하기 위해 다음 단계를 수행하십시오.

프로시저

1. **api_function** 속성에서 오류의 원인을 판별하십시오.

이 속성은 CICS 관리 클라이언트 인터페이스가 PERFORM SET 기능을 호출하려고 시도했지만 실패했음을 나타냅니다.

2. **api_response1_alt** 및 **api_response2_alt** 속성에서 실패 이유를 판별하십시오.

이 값은 CICSplex SM EYUDA 응답 및 이유 코드입니다.

EYUDA 응답 값은 TABLEERROR이고 EYUDA 이유 값은 DATAERROR입니다.

3. <feedback> 요소에서 **조치** 속성을 포함하는 <resultsummary> 태그의 **api_function** 속성에 있는 정보를 결합하여 실패한 API 호출을 판별하십시오.

이 정보를 사용하여 CICS 관리 클라이언트 인터페이스가 CICSplex SM API 명령 **EXEC CPSM PERFORM SET ACTION('INSTALL')**을 수행하려는 경우 실패했는지를 판별할 수 있습니다.

4. <installerror> 요소의 **eibfn_alt** 속성에서 연관된 EXEC CICS 명령을 식별하십시오.

eibfn_alt은 명령 **EXEC CICS CREATE URIMAP**을 나타냅니다

5. CICSplex SM BINSTERR 오류 코드를 판별하려면 <installerror> 요소의 **오류 코드** 속성의 값을 사용하십시오.

오류 코드와 의미의 목록에 대해서는 [BINSTERR](#) 자원 테이블을 참조하십시오. 이 경우 **errorcode** 값 4는 (INSTFAIL) - Install failure입니다.

6. 실패의 정확한 원인을 판별하기 위해 <installerror> 요소의 **resp** 및 **resp2** 속성을 사용하십시오. 두 값은 10진수 형식으로 표시됩니다.

- a) CICS **resp** 값 16에 대한 기호 이름을 찾아보십시오.

[EIBRESP](#) 값의 목록에 대해서는 [EIB](#) 필드의 내용을 참조하십시오.

이 목록에서 값이 16인 EIBRESP 필드에서 기호 이름이 INVREQ인지 판별할 수 있습니다.

- b) 마지막으로 **EXEC CICS CREATE URIMAP** 명령에 대해 **resp2** 값이 500인 INVREQ 응답의 의미를 찾아보십시오.

EXEC CICS CREATE 명령의 목록 RESP2 값에 대해서는 [CREATE](#) 및 [CSD](#) 명령의 RESP2 값의 내용을 참조하십시오.

CICS 관리 클라이언트 인터페이스 오류 메시지

CICS 관리 클라이언트 인터페이스(CMCI)는 메시지 ID 범위가 DFHWU4001에서 DFHWU5002까지인 XML 메시지를 발행합니다. 이 메시지는 HTTP 응답 코드와 연관되며 실패 원인을 이해하는 데 도움이 되도록 설명 및 오류 정보를 동반합니다.

CMCI 메시지의 대표적인 예는 다음과 같습니다.

```
<?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>
```

메시지는 메시지 ID, 제목, 간단한 설명과 자세한 설명, 특정 오류 정보로 구성됩니다. 이 경우 <errorInfo> 요소의 정보는 URI에서 알 수 없는 조회 매개변수를 식별합니다.

DFHWU4001 - DFHWU4029, HTTP 400 오류:

HTTP 400 응답 코드는 서버가 요청을 이해할 수 없음을 표시합니다. 모든 HTTP 400 응답에는 다음 정보가 포함됩니다.

400 CICS 관리 클라이언트 인터페이스 오류

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. 요청을 처리할 수 없습니다. *full explanation*

*full explanation*은 다음 중 하나입니다.

- URI에서 자원 이름이 지정되지 않았습니다.
- 지정된 URI는 허용 가능한 최대 길이, 256바이트를 초과하는 PATH를 포함합니다.
- URI에서 알 수 없는 조회 매개변수가 지정되었습니다.
- URI에서 컨텍스트가 누락되었습니다.
- URI에서 결과 캐시 토큰이 누락되었습니다.
- HTTP 요청의 본문이 누락되었습니다.
- HTTP 요청의 본문을 올바르게 지정하지 않았습니다.
- URI에 지정한 레코드 색인이 올바르지 않습니다.
- URI에 지정한 레코드 수가 올바르지 않습니다.
- 캐시되지 않은 결과에 대해 레코드 색인을 지정했습니다.
- URI 끝에서 관계없는 데이터가 발견되었습니다.
- URI에서 여러 개의 CRITERIA 표현식이 발견되었습니다.
- URI에서 여러 개의 PARAMETER 표현식이 발견되었습니다.

- URI에서 여러 개의 NODISCARD 표현식이 발견되었습니다.
- URI에서 여러 개의 SUMMONLY 표현식이 발견되었습니다.
- 결과 캐시 조작에 대해 CRITERIA가 올바르지 않습니다.
- 결과 캐시 조작에 대해 PARAMETER가 올바르지 않습니다.
- URI에서 자원 이름이 누락되었습니다.
- 지정된 속성이 이 자원에 대해 올바르지 않습니다.
- DEFVER 속성을 지정하지 않았거나 영(0) 값으로 지정했습니다.
- 지정된 속성의 값이 범위를 벗어났거나 올바르지 않습니다.
- NODISCARD는 HTTP GET 요청에 대해서만 올바릅니다.
- CRITERIA는 HTTP POST 요청에 대해서만 올바릅니다.
- PARAMETER는 HTTP POST 요청에 대해서만 올바릅니다.
- 지정된 결과 캐시 토큰이 최대 허용 가능 길이를 초과했습니다.
- 올바르지 않은 HTTP 본문에 조치를 지정했습니다.

DFHWU4030, HTTP 401 오류:

401 기본 인증 오류

서버에 대한 인증에 실패했습니다.

사용자 이름이나 비밀번호 또는 모두 필수이지만 누락되거나 올바르지 않습니다.

DFHWU4300, HTTP 403 오류:

403 CICS 관리 클라이언트 인터페이스 액세스 금지

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. 지정된 결과 캐시에 대한 액세스가 거부되었습니다.

URI에 지정된 결과 캐시 토큰이 요청을 작성한 사용자에게 속하지 않습니다.

DFHWU4400 - DFHWU4402, HTTP 404 오류:

404 CICS 관리 클라이언트 인터페이스 자원을 찾을 수 없음

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. URI에 지정된 자원을 찾을 수 없습니다.

자원 유형이 현재 버전의 CICS 관리 클라이언트 인터페이스에서 지원되지 않습니다.

404 CICS 관리 클라이언트 인터페이스 결과 캐시 레코드를 찾을 수 없음

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. 지정된 결과 캐시 레코드를 찾을 수 없습니다.

결과 캐시 레코드 색인이 범위를 벗어났습니다.

404 CICS 관리 클라이언트 인터페이스 결과 캐시를 찾을 수 없음

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. 지정된 결과 캐시를 찾을 수 없습니다.

결과 캐시 토큰을 찾을 수 없습니다.

DFHWU4500, HTTP 405 오류:

405 CICS 관리 클라이언트 인터페이스 메소드(DELETE|GET|POST|PUT|HEAD|Unknown)가 허용되지 않음

CICS 관리 클라이언트 인터페이스에서 오류가 발생했습니다. 지정된 HTTP 메소드가 URI에 대해 허용되지 않습니다.

CICS 관리 클라이언트 인터페이스에 전송된 URI에 대해 올바르지 않은 메소드가 지정되었습니다.

DFHWU5000 - DFHWU5002, HTTP 500 오류:

500 시스템 관리 클라이언트 내부 서버 오류

CICS 관리 클라이언트 인터페이스에서 내부 오류가 발생했습니다.

시스템 관리자에게 문의하십시오.

500 CICS 관리 클라이언트 인터페이스 스토리지 부족

CICS 관리 클라이언트 인터페이스 서버에서 막대 아래의 스토리지 부족이 발생했습니다.
시스템 관리자에게 문의하십시오.

500 CICS 관리 클라이언트 인터페이스 스토리지 부족

요청을 완료하기에는 사용 가능한 GCDSA 스토리지가 부족합니다.
시스템 관리자에게 문의하십시오.

CMCI를 사용하여 시스템 초기화 매개변수 조회

시스템 초기화 매개변수 및 해당 대체에 대한 정보를 검색하기 위해 CICSSystemParameter 외부 자원에서 작동하는 CMCI GET 요청 오퍼레이션을 사용합니다.

시작하기 전에

CMCI에 연결해야 합니다. 자세한 정보는 [CMCI 설정의 내용](#)을 참조하십시오.

검색할 시스템 초기화 매개변수가 CICS 시동 시 CICS 리전에 대해 올바른지 확인해야 합니다. 시동 시 콘솔에서 올바른 매개변수 값을 설정한 경우 검색 조작이 일관되지 않게 작동할 수 있습니다. 일부 매개변수는 정정된 값을 표시하는 반면, 다른 매개변수는 원래 SIT 값을 표시합니다.

이 태스크 정보

CMCI 요청은 시스템 초기화 테이블 매개변수, 시스템 초기화 테이블 대체 또는 이 둘의 조합을 검색할 수 있습니다.

여러 다른 CICSplex SM 조작과 마찬가지로 컨텍스트와 범위를 지정하여 검색이 작동할 CICS 리전을 제어할 수 있습니다.

시스템 초기화 매개변수 및 해당 대체에 대한 정보를 검색하려면 다음 단계를 완료하십시오.

프로시저

1. HTTP GET 메소드를 사용하고 자원 이름으로 CICSSystemParameter를 지정하는 CMCI 요청을 구성하십시오.

a) 요청이 작동하는 CICS 리전을 지정하려면 **context** 및 **scope** 매개변수를 사용하십시오.

- b) URI에서 검색할 매개변수 또는 대체를 지정하기 위해 매개변수 **PARMSRCE** 및 **PARMTYPE**을 통합하는 매개변수 표현식을 사용하십시오.

이 매개변수 모두 필수입니다. **PARMTYPE**의 경우 SIT 값을 지정해야 합니다. **PARMSRCE**의 경우 다음 옵션 중 하나를 지정하십시오.

COMBINED

원본 시스템 초기화 테이블 정의와 적용된 매개변수 대체의 조합

CONSOLE

시스템 콘솔에서 시동 시 지정된 덮어쓰기 매개변수

JCL

JCL EXEC PGM 명령문을 통해 제공된 덮어쓰기 매개변수

SYSIN

SYSIN 데이터 세트에 정의된 시동 작업 스트림의 덮어쓰기 매개변수

TABLE

xx 로드 모듈에서 추출된 원본 시스템 초기화 테이블 값

GET 요청에 대한 자세한 정보는 [CICS 관리 클라이언트 인터페이스 GET 요청](#)을 참조하십시오.

2. CMCI 요청을 제출하십시오.

매개변수와 대체 값은 소스(예: SYSIN 데이터 세트)의 순서와 상관없이 SYSPARM 자원 테이블의 KEYWORD 속성에 따라 리턴됩니다.

이 동작으로 대체 매개변수 조합에 대해 예상치 못한 결과가 생성될 수 있습니다(**SPCTR** 및 **SPCTRxx**, 그리고 **STNTR** 및 **STNTRxx**) 이러한 쌍은 지정되는 순서가 중요하기 때문입니다. 예를 들어, SPCTR=ALL 및 SPCTR=OFF의 순서로 지정하는 경우 SPCTRAP 추적을 사용할 수 없지만, SPCTR=OFF 및 SPCTR=ALL의 순서로 지정하면 SPCTRAP 추적을 사용할 수 있습니다. 그러나 SYSPARM 자원에 대한 요청은 항상 **SPCTRxx** 전에 **SPCTR**을 리턴합니다.

- 이 요청은 CICSplex MYPLEX의 REGION 리전용으로 설정된 제어 테이블 접미부 값을 검색합니다.

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

요청은 결과 요약 및 일치하는 여러 레코드를 리턴합니다. 예를 들어, 다음과 같습니다.

[illegible]

- 다음 요청은 CICsplex MYPLEX에 있는 모든 리전에 대한 **MXT**(MAX TASKS) 시스템 초기화 매개변수에 설정된 값을 검색합니다.

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

- 다음은 CICSplex MYPLEX의 REGION 리전용으로 설정된 제어 테이블 접미부 값을 검색합니다.

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

- 다음 요청은 기본 런어웨이 태스크 시간 간격이 꺼진, CICSplex MYPLEX에 있는 리전 이름을 검색합니다.

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

- 다음 요청은 SYSIN 대체에 의해 CICSplex MYPLEX의 리전 REGION에 적용된 대체를 검색합니다.

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

- 다음 요청은 JCL 대체에 의해 CICSplex MYPLEX의 리전 REGION에 적용된 대체를 검색합니다.

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


자원 정의와 상호작용

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 BAS 및 CSD 자원 정의에서 조치를 작성하고, 업데이트하며, 삭제하고, 수행합니다.

이 태스크 정보

다음과 같은 방법으로 CICSplex SM에서 자원 정의와 상호작용할 수 있습니다.

- 애플리케이션에서 HTTP 요청을 승인하는 CMCI 사용
- CMCI를 활용하는 CICS Explorer 사용
- CICSplex SM 웹 사용자 인터페이스(WUI) 사용
- CICSplex SM API 명령을 애플리케이션에 직접 추가

CMCI를 사용하여 자원 정의 작성

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 BAS 또는 CSD 자원 정의를 작성합니다.

시작하기 전에

CMCI에 연결해야 합니다. 자세한 정보는 [CMCI 설정](#)의 내용을 참조하십시오.

이 태스크 정보

CMCI를 사용하여 자원 정의를 작성하려면 다음 단계를 완료하십시오.

프로시저

1. 자원 정의를 작성하려면 HTTP POST 메소드로 요청을 사용하십시오.
POST 요청에 대한 자세한 정보는 [CICS\(r\) 관리 클라이언트 인터페이스 POST 요청](#)을 참조하십시오.
2. 요청에 대한 URI를 지정하십시오.
 - BAS 자원을 작성하며 URI에서 *SCOPE*를 지정하면, 이는 무시됩니다.
 - CSD 자원을 작성하는 경우 URI에서 *SCOPE*를 지정하십시오. *SCOPE*는 자원을 작성하는 CICS 시스템이어야 합니다.

POST 요청의 URI를 지정하는 방법에 대한 자세한 정보는 [CICS 관리 클라이언트 인터페이스 POST 요청 URI](#)를 참조하십시오.
3. 자원에 대한 매개변수 및 속성을 포함하는 XML 본문을 추가하십시오.
XML 본문은 <request> 요소 내 <create> 요소를 구성합니다.
 - BAS 자원을 작성하는 경우 *defver* 속성을 XML 본문에 포함해야 합니다.
 - CSD 자원을 작성하는 경우 XML 본문에서 CSD 매개변수 및 CSD group 속성을 지정해야 합니다. CSD 자원을 작성하는 경우 *defver* 속성을 포함하지 마십시오.

예

다음 예제는 BAS 트랜잭션 정의를 작성하기 위해 CMCI에 전송할 내용을 표시합니다.

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

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

이 예제는 CSD 트랜잭션 정의를 작성하기 위해 CMCI에 전송할 내용을 표시합니다. CSD 매개변수 및 csdgroup 속성이 XML에 추가되고 defver 속성이 제외된다는 점을 참고하십시오.

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

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

두 예제(BAS 자원 정의 작성 및 CSD 자원 정의 작성)는 성공적으로 완료되면 비슷한 출력을 리턴합니다. 성공한 작성은 다음과 같이 api_response1="1024" 및 api_response1_alt="OK"를 포함하는 CICSplex SM API 응답 메시지를 리턴합니다.

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

CMCI를 사용하여 자원 정의 업데이트

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 BAS 또는 CSD 자원 정의를 업데이트합니다.

시작하기 전에

CMCI에 연결해야 합니다. 자세한 정보는 [CMCI 설정](#)의 내용을 참조하십시오.

이 태스크 정보

CMCI를 사용하여 자원 정의를 업데이트하려면 다음 단계를 완료하십시오.

프로시저

1. 자원 정의를 업데이트하려면 PUT 요청을 사용하십시오.
PUT 요청에 대한 자세한 정보는 [12 페이지의 『CMCI PUT 요청』](#)의 내용을 참조하십시오.
2. 요청에 대한 URI를 지정하십시오.
 - BAS 자원을 업데이트하며 URI에서 SCOPE를 지정하면, 이는 무시됩니다.
 - CSD 자원을 업데이트하는 경우 URI에서 SCOPE 및 PARAMETER=CSDGROUP(value)을 지정하십시오.PUT 요청에 대한 URI를 지정하는 방법에 대한 자세한 정보는 [14 페이지의 『CMCI PUT 요청 URI』](#)의 내용을 참조하십시오.
3. 자원에 대한 매개변수 및 속성을 포함하는 XML 본문을 추가하십시오.
XML 본문은 <request> 요소 내 <update> 요소를 구성합니다.
 - BAS 자원을 업데이트하는 경우 defver 속성을 XML 본문에 포함해야 합니다.
 - CSD 자원을 업데이트하는 경우 XML 본문에서 CSD group 속성을 지정해야 합니다. CSD 자원을 작성하는 경우 defver 속성을 포함하지 마십시오.

예

다음 예제는 BAS 트랜잭션 정의를 업데이트하기 위해 지정할 내용을 표시합니다. 이 예제에서 트랜잭션 정의, TRN1에 대한 프로그램 이름 값은 PRGM2로 업데이트됩니다.

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

<request>
```

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

다음 예제는 CSD 트랜잭션 정의에 대한 업데이트 요청을 표시합니다. 이 예제에서 GRP1 자원 그룹에 있는 TRN1이라는 자원의 프로그램 이름은 PRGM2로 업데이트됩니다. PARAMETER=CSDGROUP이 URI에 추가되고 defver 속성이 제외된다는 점을 참고하십시오.

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

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

두 예제(자원 정의 업데이트 및 CSD 자원 정의 업데이트)는 성공적으로 완료되면 비슷한 출력을 리턴합니다. 성공한 업데이트는 다음과 같이 api_response1="1024" 및 api_response1_alt="OK"를 포함하는 CICSplex SM API 응답 메시지를 리턴합니다.

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

CMCI를 사용하여 자원 정의에서 조치 수행

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 BAS 또는 CSD 자원 정의에 조치를 수행합니다.

시작하기 전에

CMCI에 연결해야 합니다. 자세한 정보는 [CMCI 설정](#)의 내용을 참조하십시오.

이 태스크 정보

CMCI를 사용하여 자원 정의에서 조치를 수행하려면 다음 단계를 완료하십시오.

프로시저

1. 자원 정의에서 조치를 수행하려면 PUT 요청을 사용하십시오.
HTTP PUT 요청을 사용하는 방법에 대한 자세한 정보는 [12 페이지의 『CMCI PUT 요청』](#)의 내용을 참조하십시오.
2. 요청에 대한 URI를 지정하십시오.
CSD 자원에서 조치를 수행하는 경우, URI에서 SCOPE 및 PARAMETER=CSDGROUP을 지정하십시오.
PUT 요청에 대한 URI를 지정하는 방법에 대한 자세한 정보는 [14 페이지의 『CMCI PUT 요청 URI』](#)의 내용을 참조하십시오.
3. 자원에 대한 매개변수 및 속성을 포함하는 XML 본문을 추가하십시오.
XML 본문은 <request> 요소 내 <action name="action_name"> 요소를 구성합니다.
조치에 매개변수가 필요하면, <action> 요소 내 <parameter> 요소에서 이를 지정해야 합니다.
4. 수행하려는 조치를 지정하십시오.
각 자원 테이블은 수행할 수 있는 조치의 목록을 포함합니다.

예

다음 예제는 이름이 AA로 시작하는 범위 내 모든 BAS 트랜잭션 정의에 적용되는 INSTALL 요청을 표시합니다.

```
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>
```

다음 예제는 CSDINSTALL 동작 요청에 대해 지정할 내용을 표시합니다. CSD 자원에서 조치에 대해 지정할 수 있는 매개변수는 명령 설명에 나열됩니다.

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

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

CMCI를 사용하여 자원 정의 삭제

CICS 관리 클라이언트 인터페이스(CMCI)를 사용하여 BAS 또는 CSD 자원 정의를 삭제합니다.

시작하기 전에

CMCI에 연결해야 합니다. 자세한 정보는 [CMCI 설정](#)의 내용을 참조하십시오.

이 태스크 정보

CMCI를 사용하여 자원 정의를 삭제하려면 다음 단계를 완료하십시오.

프로시저

1. 자원 정의를 삭제하려면 DELETE 요청을 사용하십시오.
DELETE 요청에 대한 자세한 정보는 3 페이지의 『CMCI DELETE 요청』의 내용을 참조하십시오.
2. 요청에 대한 URI를 지정하십시오.
CSD 자원을 삭제하는 경우, 독립형 CICS 리전에서 작업하는 경우가 아니라면 URI에 *SCOPE*를 지정해야 합니다.
DELETE 요청에 대한 URI를 지정하는 방법에 대한 자세한 정보는 4 페이지의 『CMCI DELETE 요청 URI』의 내용을 참조하십시오.
3. 매개변수가 있는 자원을 삭제하려면 URI에서 자원에 대한 매개변수를 지정해야 합니다.
예를 들어, CSD 자원을 삭제하려면 URI `PARAMETER=CSDGROUP(group)`에서 CSDGROUP 매개변수를 지정해야 합니다.
4. 옵션: XML 본문에서 요청에 대한 세부사항을 지정할 수도 있습니다.
삭제 조치를 수행하기 위해 이러한 세부사항은 XML 본문에 필요하지 않습니다.
XML 본문은 <request> 요소 내 <delete> 요소를 구성합니다.
일부 클라이언트는 XML 본문을 구문 분석할 수 없습니다. 그런 이유로, 삭제할 자원을 지정하려면 매개변수를 URI에 추가해야 합니다.

예

다음 예제는 BAS 트랜잭션 정의를 삭제하기 위해 지정할 내용을 표시합니다. 이 예제에서 이름이 "AAAA"인 모든 자원이 삭제됩니다.

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

CSD 자원에 대한 삭제 요청에는 XML 본문이 필요하지 않습니다.

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

XML 본문을 포함하는 CSD 자원에 대한 이 삭제 요청은 이전 요청과 동일합니다.

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

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


주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 본 자료는 다른 언어로도 제공될 수 있습니다. 그러나 자료에 접근하기 위해서는 해당 언어로 된 제품 또는 제품 버전의 사본이 필요할 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 3IFC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 3IFC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 본 샘플 프로그램은 일체의 보증 없이 "현상태대로" 제공됩니다. IBM은 귀하의 샘플 프로그램 사용과 관련되는 손해에 대해 책임을 지지 않습니다.

프로그래밍 인터페이스 정보

CICS에서는 프로그래밍 인터페이스로 간주될 수 있는 몇 가지 문서와 프로그래밍 인터페이스로 간주될 수 없는 몇 가지 문서를 제공합니다.

고객이 z/OS용 CICS Transaction Server, 버전 5 릴리스 6 의 서비스를 얻는 프로그램을 작성하는 데 사용할 수 있는 프로그래밍 인터페이스는 온라인 제품 문서의 다음 절에 포함되어 있습니다.

- [애플리케이션 개발](#)
- [시스템 프로그램 개발](#)
- [CICS TS 보안](#)
- [외부 인터페이스용으로 개발](#)
- [애플리케이션 개발 참조서](#)
- [참조: 시스템 프로그래밍](#)
- [참조: 연결성](#)

z/OS용 CICS Transaction Server, 버전 5 릴리스 6 의 프로그래밍 인터페이스로 사용하지 않아야 하지만 프로그래밍 인터페이스로 오해할 수 있는 정보는 온라인 제품 문서의 다음 섹션에 포함되어 있습니다.

- [문제점 해결 및 지원](#)
- [CICS TS 진단 참조](#)

PDF 형식 매뉴얼의 CICS 문서에 액세스하려는 경우 고객이 z/OS용 CICS Transaction Server, 버전 5 릴리스 6 의 서비스를 얻는 프로그램을 작성하는 데 사용할 수 있는 프로그래밍 인터페이스는 다음 매뉴얼에 포함되어 있습니다.

- 애플리케이션 프로그래밍 안내서 및 애플리케이션 프로그래밍 참조서
- Business Transaction Services
- 사용자 정의 안내서
- C++ OO 클래스 라이브러리
- 디버깅 도구 인터페이스 참조
- 분산 트랜잭션 프로그래밍 안내서
- 외부 인터페이스 안내서
- Front End Programming Interface 안내서
- IMS 데이터베이스 제어 안내서
- 설치 안내서
- 보안 안내서
- 제공 트랜잭션
- CICSplex SM 워크로드 관리
- CICSplex SM 관리 자원 사용법
- CICSplex SM 애플리케이션 프로그래밍 안내서 및 애플리케이션 프로그래밍 참조서

- CICS의 Java 애플리케이션

PDF 형식 매뉴얼의 CICS 문서에 액세스하는 경우 z/OS용 CICS Transaction Server, 버전 5 릴리스 6 의 프로그래밍 인터페이스로 사용하지 않아야 하지만 프로그래밍 인터페이스로 오해할 수 있는 정보는 다음 매뉴얼에 포함되어 있습니다.

- 데이터 영역
- 진단 참조
- 문제점 판별 안내서
- CICSplex SM 문제점 판별 안내서

상표

IBM, IBM 로고 및 ibm.com®은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹 [저작권 및 상표 정보](http://www.ibm.com/legal/copytrade.shtml)(www.ibm.com/legal/copytrade.shtml)에 있습니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

Apache, Apache Axis2, Apache Maven, Apache Ivy, Apache Software Foundation(ASF) 로고 및 ASF 기능 로고는 Apache Software Foundation의 상표입니다.

Gradle 및 Gradlephant 로고는 미국 또는 기타 국가에서 사용되는 Gradle, Inc.의 등록상표입니다.

Intel, Intel 로고, Intel Inside Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Red Hat® 및 Hibernate®는 미국 또는 기타 국가에서 사용되는 Red Hat, Inc.의 상표 또는 등록상표입니다.

Spring Boot는 미국 또는 기타 국가에서 사용되는Pivotal Software, Inc.의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Zowe™, Zowe 로고 및 Open Mainframe Project™는 The Linux Foundation의 상표입니다.

제품 문서의 이용 약관

다음 이용 약관에 따라 이 책을 사용할 수 있습니다.

적용성

본 이용 약관은 IBM 웹 사이트의 모든 이용 약관에 추가됩니다.

개인적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 발행물 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 귀하의 기업집단 외에서는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

권한

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 발행물의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용과 관련하여 아무런 보장을 하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

IBM 온라인 개인정보처리방침

서비스 솔루션 소프트웨어를 비롯한 IBM 소프트웨어 제품(소프트웨어 오퍼링)은 제품 사용 정보 수집, 일반 사용자 편의성 향상, 일반 사용자와의 상호작용 조정 및 기타 목적을 위해 쿠키 또는 기타 기술을 사용할 수 있습니다. 많은 경우에 있어서, 소프트웨어 오퍼링은 개인 식별 정보를 수집하지 않습니다. IBM의 일부 소프트웨어 오퍼링은 귀하가 개인 식별 정보를 수집하도록 도울 수 있습니다. 본 소프트웨어 오퍼링이 쿠키를 사용하여 개인 식별 정보를 수집할 경우, 본 오퍼링의 쿠키 사용에 대한 특정 정보가 다음에 규정되어 있습니다:

CICSplex SM 웹 사용자 인터페이스(기본 인터페이스)의 경우:

이 소프트웨어 오퍼링은 배치된 구성에 따라 세션 관리, 인증, 사용자 편리성 개선, 기타 사용량의 추적이나 기능적인 용도로 각 사용자의 이름과 개인 정보를 수집하는 세션 및 지속적 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICSplex SM 웹 사용자 인터페이스(데이터 인터페이스)의 경우:

본 소프트웨어 오퍼링은 배치된 구성에 따라 세션 관리, 인증 또는 기타 사용량 추적이나 기능의 용도로 각 사용자의 사용자 이름 및 개인 식별 정보를 수집하는 세션 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICSplex SM 웹 사용자 인터페이스("hello world" 페이지)의 경우:

배치된 구성에 따라 이 소프트웨어 오퍼링은 개인 식별 정보를 수집하지 않는 세션 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICS Explorer의 경우:

배치된 구성에 따라 이 소프트웨어 오퍼링은 세션 관리, 인증 및 싱글 사인온 구성을 위해 사용자의 사용자 이름 및 비밀번호를 수집하는 세션 및 지속적 환경 설정을 사용할 수 있습니다. 사인온 중에 선택란을 선택하여 사용자 비밀번호를 암호화된 양식으로 저장하면 사용자의 명시적인 조치에 의해서만 사용으로 설정할 수 있지만 이러한 환경 설정은 사용 안함으로 설정할 수 없습니다.

이 소프트웨어 오퍼링에 대해 배치된 구성이 고객님의 귀하에게 쿠키 및 기타 기술을 통해 일반 사용자로부터 개인적으로 식별 가능한 정보를 수집하는 기능을 제공하는 경우에는 공지사항 및 동의에 대한 요구사항을 포함하여 해당 데이터 콜렉션에 적용할 수 있는 법률에 대한 자체 법률 자문을 구해야 합니다.

해당 용도로 쿠키를 비롯한 다양한 기술을 사용하는 데 관한 자세한 정보는 [IBM 개인정보처리방침](#) 및 [IBM 온라인 개인정보처리방침](#), 쿠키, 웹 비콘 및 기타 기술이라는 절과 [IBM 소프트웨어 제품 및 Software-as-a-Service 개인정보처리방침](#)을 참조하십시오.

색인

특수 문자

<action>
 자원 정의
 <action> [53](#)
 CMCI
 <action> [53](#)
<create>
 자원 정의
 <create> [51](#)
 CMCI
 <create> [51](#)
<delete>
 자원 정의
 <delete> [54](#)
 CMCI
 <delete> [54](#)
<update>
 자원 정의
 <update> [52](#)
 CMCI
 <update> [52](#)

A

API(Application Programming Interface)
 CICS 관리 클라이언트 인터페이스 [1](#)

C

CICS 관리 클라이언트 인터페이스
 개요 [1](#)
 문제점 판별 [41-43](#)
 사용공간 정리 [9](#)
 요청 [3](#)
 이스케이프 시퀀스 [39](#)
 제한된 문자 [39](#)
 DELETE 메소드 [3](#)
 DELETE 요청 URI [4](#)
 errors XML 요소 [34](#)
 FEEDBACK 레코드 [42, 43](#)
 feedback XML 요소 [34](#)
 GET 메소드 [5, 9](#)
 GET 요청 URI [6](#)
 inconsistentscope XML 요소 [36](#)
 inconsistentset XML 요소 [37](#)
 installerror XML 요소 [36](#)
 POST 메소드 [10](#)
 POST 요청 URI [12](#)
 PUT 메소드 [12](#)
 PUT 요청 URI [14](#)
 put XML 요소 [28](#)
 request XML 요소 [27](#)
 response XML 요소 [31-33](#)
 resultsummary 오류 [41](#)
 URI 자원 이름
 CICS 관리 클라이언트 인터페이스 [17](#)

CICS 관리 클라이언트 인터페이스 (계속)
 XML 스키마 [27](#)
CICSSystemParameter 외부 자원 [47](#)
CMCI
 사용공간 정리 [9](#)
 SIT 매개변수 [47](#)
 XML [27](#)

D

DELETE 요청
 CICS 관리 클라이언트 인터페이스에서 [3](#)
DELETE 요청 URI
 CICS 관리 클라이언트 인터페이스에서 [4, 6](#)
DFHWU4001 [44](#)
DFHWU4002 [44](#)
DFHWU4003 [44](#)
DFHWU4004 [44](#)
DFHWU4005 [44](#)
DFHWU4006 [44](#)
DFHWU4007 [44](#)
DFHWU4008 [44](#)
DFHWU4009 [44](#)
DFHWU4010 [44](#)
DFHWU4011 [44](#)
DFHWU4012 [44](#)
DFHWU4013 [44](#)
DFHWU4014 [44](#)
DFHWU4015 [44](#)
DFHWU4016 [44](#)
DFHWU4017 [44](#)
DFHWU4018 [44](#)
DFHWU4019 [44](#)
DFHWU4020 [44](#)
DFHWU4021 [44](#)
DFHWU4022 [44](#)
DFHWU4023 [44](#)
DFHWU4024 [44](#)
DFHWU4025 [44](#)
DFHWU4026 [44](#)
DFHWU4027 [44](#)
DFHWU4028 [44](#)
DFHWU4029 [44](#)
DFHWU4030 [44](#)
DFHWU4401 [44](#)
DFHWU5001 [44](#)

E

errors XML 요소
 CICS 관리 클라이언트 인터페이스에서 [34](#)

F

FEEDBACK 레코드
 CICS 관리 클라이언트 인터페이스 [42, 43](#)

feedback XML 요소
CICS 관리 클라이언트 인터페이스에서 [34](#)

G

GET 요청
예제 [9](#)
CICS 관리 클라이언트 인터페이스에서 [5](#)

I

inconsistentscope XML 요소
CICS 관리 클라이언트 인터페이스에서 [36](#)
inconsistentset XML 요소
CICS 관리 클라이언트 인터페이스에서 [37](#)
installerror XML 요소
CICS 관리 클라이언트 인터페이스에서 [36](#)

N

NODISCARD 옵션
GET 요청에서 [9](#)

P

POST 요청
CICS 관리 클라이언트 인터페이스에서 [10](#)
POST 요청 URI
CICS 관리 클라이언트 인터페이스에서 [12](#)
PUT 요청
CICS 관리 클라이언트 인터페이스에서 [12](#)
PUT 요청 URI
CICS 관리 클라이언트 인터페이스에서 [14](#)
put XML 요소
CICS 관리 클라이언트 인터페이스에서 [28](#)

R

request XML 요소
CICS 관리 클라이언트 인터페이스에서 [27](#)
response XML 요소
CICS 관리 클라이언트 인터페이스에서 [31–33](#)
REST API [1](#)
REST(Representational State Transfer) [1](#)

S

SIT 매개변수
CMCI [47](#)

X

XML
CICS 관리 클라이언트 인터페이스에서 [27](#)

사

사용공간 정리
CMCI에서 [9](#)

아

오류

CICS 관리 클라이언트 인터페이스에서 [41](#)
이스케이프 시퀀스
CICS 관리 클라이언트 인터페이스에서 [39](#)

