OLTP transaction looping, you can take care of it! (and other enhancements)

Aleksa Medakovic

Agenda

1. Why Transaction Cancellation?
2. Transaction Loops & the Timeout Problem
3. Feature Overview & Architecture
4. REST API, List Feature
5. Security: Auth & Access Control
6. Java GUI & Custom Integrations
7. Live Demo
8. Q&A

Why Transaction Cancellation?

Looping Transactions

Hung transactions tie up resources and block other work from completing

Manual Intervention

Operators must log in and start a new CEMT transaction

No Automation

External systems cannot programmatically resolve transaction issues

Transaction Loops in OLTP

Tight Loop

Timeout CAN detect

A single program executes the same instructions repeatedly and never returns control to OLTP.

Non-yielding Loop

Timeout CAN detect

Control returns to OLTP temporarily

Yielding Loop

Timeout CANNOT detect

Control returns to an OLTP

The Timeout Problem

How the OLTP Timer Works (ICVR)

OLTP tracks how long each transaction has been dispatched.

If a single dispatch exceeds the configured time limit, the system purges the transaction.

Why Yielding Breaks It

A yielding transaction gives up the thread before the timer expires, then gets redispatched.

Each new dispatch restarts the timer from zero

Examples of Yielding Loops

1. Loop with reading and writing

```
WHILE 1 {
  EXEC CICS READ FILE(...)
  EXEC CICS WRITE FILE(...)
}
```

2. Loop with SUSPEND

```
WHILE 1 {
  EXEC CICS SUSPEND
}
```

IESCANTR Overview

REST Endpoint

Standard HTTP interface. Any language or tool can integrate

Secured by Default

Basic authentication with facility based access control

Java GUI Included

Ready to use graphical interface

Extend Your Operations

Build custom operator tools and automation scripts

OLTP Web Support

Before the cancellation REST endpoint can accept requests, OLTP web support must be configured.

Setup Checklist

- Enable OLTP web support in the SIT (TCPIP=YES)
- Submit DFHWBEP, DFHCNV (ICCF 59 lib)
- Define the hostname in TCP/IP
  - CSI - DEFINE NAME,NAME=XX,IPaddr=YY
  - BSI - HOST XX YY
- Identify the TCP/IP stack ID in your OLTP startup job using // OPTION SYSPARM=nn
- Define a TCPIPSERVICE with the desired port and protocol (HTTP or HTTPS)

1. Client sends a request with operation, task number, and Basic auth credentials
2. REST endpoint validates the request format and routes to the IESCANTR
3. IESCANTR verifies credentials and checks facility-level access permissions and cancels the target transaction

REST API

https://[VSEn-System-IP]:[Port]/cics/CWBA/IESRESTS/IESCANTR

Request Parameters

Parameter	Location	Required	Description
`taskNumber`	Query	Yes	Task number of the task/ transaction to cancel
`operation[list/ purge]`	Query	Yes	Whether to list active transactions or attempt a cancellation
`Authorization`	Header	Yes	Basic auth credentials (Base64-encoded user:password)

Authentication

1

Credentials Encoded

Client encodes user:password in Base64 format (colon separator)

2

Attach to Request

Include credentials as Authorization: Basic <credentials>

3

Server Validates

IESCANTR validates via VSEn (RACR) or LDAP

4

Access Granted

If valid, the request proceeds to facility authorization

Facility-Based Access Control

Beyond authentication, the system checks whether the authenticated user has ALTER privilege on the CANCELTR facility.

How It Works

A check is done for CANCELTR facility.

Cancellation is only allowed if the user has READ , UPDATE or ALTER access to the CANCELTR facility.

The List Command

By using list command, the API returns a JSON array of all currently active transactions instead of performing a cancellation.

Request

```
GET /IESCANTR?operation=list

Headers:
  Authorization: Basic dXNl...
```

Response (200 OK)

```
[
  {
    "task": 1688,
    “tran": “CWBA"
  },
{
    "task": 1689,
    “tran": “CEMT"
  },
  ...
]
```

Cancel — Request & Response

Request

```
GET /IESCANTR?operation=purge&taskNumber=[Task-id]

Headers:
  Authorization: Basic dXNl...
```

Response (200 OK)

```
 text/plain

   Task [Task-ID] purged successfully
```

Response (Failure)

```
text/plain

  Purge failed for task [Task-ID] resp=XX resp2=YY
```

HTTP Status Codes

Code	Status	Meaning
200	OK	Transaction cancelled, or active list returned
400	Bad Request	Invalid parameters or operation failed (details in response body)
401	Unauthorized	Authentication failed, invalid credentials or missing Authorization header
500	Server Error	System error

Java GUI

A ready to use Java GUI.

Search & Filter

List all the active transactions and filter by name

One-Click Cancel

Select a transaction and cancel with a single action

Demo

Build Your Own Integration

Production Readiness

- Use HTTPS.
- Grant CANCELTR ALTER only to authorized operators or service IDs.
- Never store or log passwords

Getting Started

- Encode user:password in Base64 for the Authorization header
- Call ?operation=list first to verify active tasks before purging

Custom Dashboards

Build cancellation into your existing ops dashboards or admin portals

Custom Automation

Use any language that supports HTTP requests to call the API from the console or batch streams

Python and Java

1 Python — list then purge

```
import requests
URL  = "https://vse:8080/.../IESCANTR"
auth = ("user", "pwd")
r = requests.get(URL, params={"operation":"list"}, auth=auth)
for t in r.json():
    requests.get(URL, auth=auth,
        params={"operation":"purge","taskNumber":t["task"]})
```

2 Java — list then purge

```
String cred = Base64.getEncoder()
    .encodeToString("user:pwd".getBytes());
HttpClient c = HttpClient.newHttpClient();
// 1. list active tasks
c.send(build(URL + "?operation=list", cred), ofString());
// 2. purge a specific task
c.send(build(URL + "?operation=purge&taskNumber=1688", cred), ofString());
```

Documentation

- Check OLTP Enhancements guide for detailed explanations and details.

- IESCANTR provided in VA00319 / VP00313

Other Enhancements

1 New Short on Storage Warnings VA00277

DFHSM0137 Close to short on storage below 16 MB

DFHSM0138 Storage critical below 16 MB

DFHSM0139 Close to short on storage above 16 MB

DFHSM0140 Storage critical above 16 MB

2 Lower monitoring interval VA00278

1 min

new minimum monitoring interval

The minimum value for the monitoring frequency has been lowered to one minute

Questions?

Thank you for your time