

How IDMS[®] DBAs Leverage Open Source and APIs for Monitoring

Introducing the IDMS REST API

The IDMS[®] Representational State Transfer (REST) Application Programming Interface (API) provides database administrators (DBAs) the ability to quickly and easily access data to analyze system issues, automate processes and alerts to create intelligent recommendations for tuning. By leveraging the IDMS REST API modern tooling can be used, offering flexibility of choice for users to work in familiar environments. The data from a REST invocation is returned in JavaScript Object Notation (JSON) which is simple to consume and parse with modern programming languages. This white paper will outline the importance of APIs, share insights on how the IDMS REST API can be used, along with real-world examples and benefits.

Data is the most valuable asset for businesses and IDMS houses this data, fueling our customers' mission critical systems. It is crucial that they have the ability to leverage their existing investments, which includes skills, to manage and maintain core systems to support their end users and customers in order to compete and succeed in their respective industries.

Why IDMS REST APIs and What Problems Do They Solve?

In many cases, organizations who have seasoned users maintaining and managing their business essential systems face upcoming retirements and a need to backfill these critical roles, often with users who are unfamiliar with the mainframe or IDMS. Opening up and making our databases more extensible through the use of APIs and services is key in simplifying and automating every day user experiences. The IDMS REST API makes it easier to maintain existing applications, build new systems, and manage the ongoing health of mission critical systems. APIs provide modern, scalable access to z/OS services, tools, and data.

With the IDMS REST API, you can use modern development tools to maintain your IDMS applications and leverage your mainframe investment.

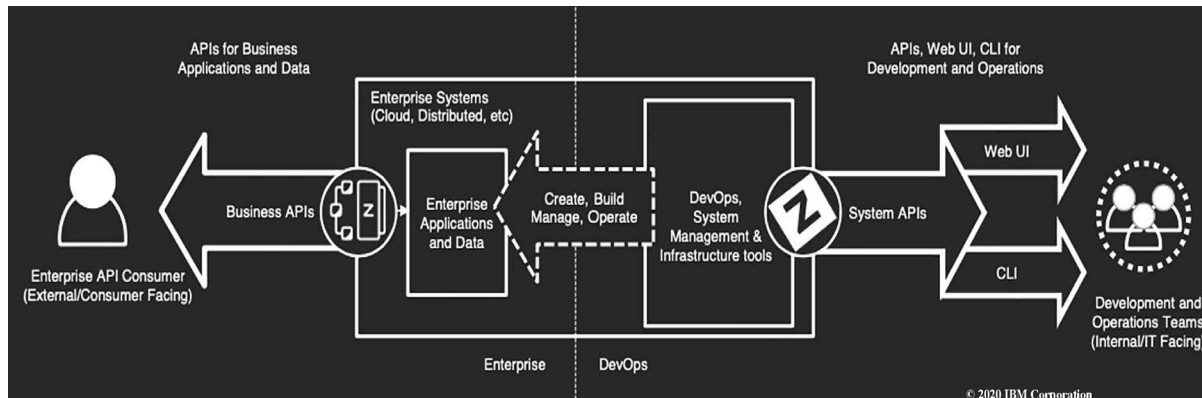
REST Assured, Your Business Data Is Not Exposed

There are two flavors of APIs:

- Business APIs for business applications and data
- System APIs that are focused on DevOps, systems management, and infrastructure

The IDMS REST API is in the category of System APIs, in which the data (JSON) returned represents IDMS system data, and performance statistics. The IDMS REST API contains functions for viewing and maintaining your IDMS databases, while keeping your data protected. Your unique business and application data stored within IDMS is not exposed or compromised.

Figure 1: The IDMS REST API exposes your IDMS system data and not your application or personal database data.



As part of the Broadcom Database modernization strategy, APIs play a role in transforming the mainframe for hybrid IT by making the mainframe a more integrated, agile platform. The IDMS REST API can automate database application maintenance and administration. The IDMS REST API allows administrators of online systems to monitor operations to see trends and recognize unusual conditions that can indicate problems in order to be proactive rather than reactive when system issues occur.

Historically, a IDMS DBA analyzed problems by relying on green-screen access, as well as leveraging IDMS Performance Monitor. Although this was acceptable for an experienced DBA who was familiar with IDMS and the world of green-screens, this environment is not ideal for a new or next-generation user of IDMS who is unfamiliar with the product and green-screen navigation. APIs help address this challenge by providing a familiar and flexible way to interact with the traditional mainframe platform, easing new DBAs and developers into mainframe roles by allowing them to work with familiar tools and environments.

Embrace Open and Deliver at the Speed of Business

Our Database modernization strategy aligns with the overarching product strategy of Broadcom and the direction to “Embrace Open” and to deliver at the speed of business. By leveraging open source technologies and taking an API approach, our customers are able to integrate the mainframe with everything – data, apps, and services – from mobile to mainframe. This approach empowers end users to increase productivity, using tools they are familiar with, shielded from the perceived complexity of the mainframe.

With the inception of Zowe, an open source framework for mainframe, the platform enables customers to securely develop and manage applications like any other cloud-based platform. The framework and capabilities accelerate mainframe application development with modern DevOps tools, while introducing new levels of automation and management of infrastructure and systems. By arming new and emerging mainframe users with these valuable skills, Zowe helps in sustaining the mainframe environment by attracting and retaining next generation talent.



The IDMS REST API is Zowe Conformant and can be integrated with the Zowe/Brightside API Mediation Layer. Conformance with Zowe ensures that you can expect a high level of common functionality, interoperability, and user experience. Leveraging the IDMS REST API standalone or in conjunction with Zowe to integrate with open source tooling and dashboards are a powerful way to enhance and simplify the user experience for new DBAs and other users.

Supported Actions

The IDMS REST API lets you perform the following actions to manage and monitor your IDMS databases:

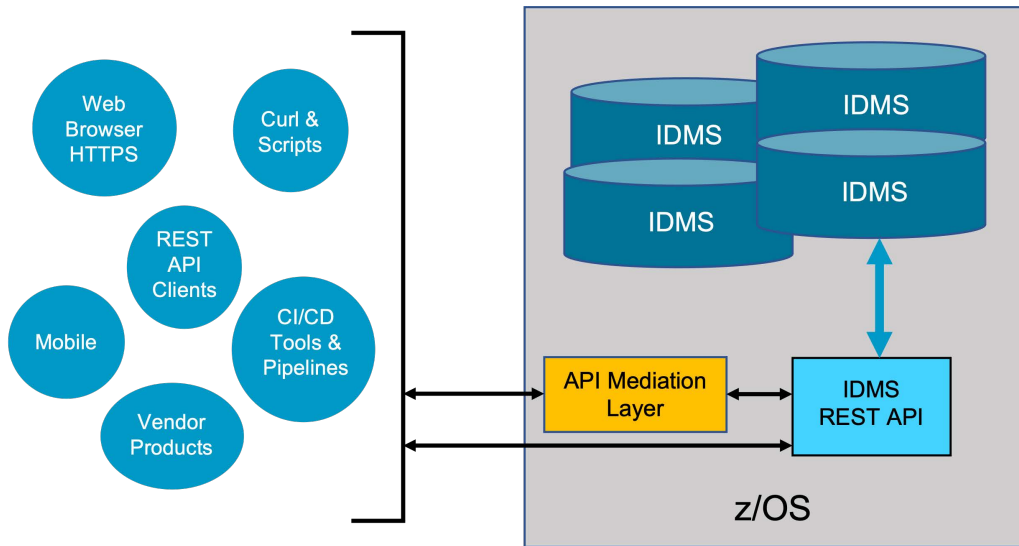
- View performance and monitor statistical data
- View DC log records
- Execute DCMT and DCUF commands

Example Use Cases

As a developer or database administrator, you can use the REST API in the following ways:

- Enable client dashboards or applications to view system and performance data.
- Enable automated warnings and recommendations to avoid critical system risks and outages.
- Obtain flexibility to leverage modern, user-friendly tooling to improve the user experience.

Figure 2: Your IDMS databases remain safely secured as the system of record.



The IDMS REST API, deployed on z/OS UNIX System Services (USS), utilizes Transport Layer Security (TLS) certificates, secured HTTP connection (HTTPS) protocol and Secure Sockets Layer (SSL). In addition, API functions can be secured with either DCMT security or External Security Managers (ESMs).

Simplifying Performance

Maintaining the high level of performance of your IDMS databases is essential. By lowering CPU usage and response times, you can reduce your total cost of ownership (TCO) and obtain the highest possible throughput, while enabling as much work to get done as possible. While the vast flexibility and options within IDMS enable superior reliability and continuity for your essential core business, they may be overwhelming for new DBAs who might not know where to start.

How can years of experienced DBA knowledge be retained and utilized to continuously tune and improve your databases? IDMS stores an abundant amount of performance related metrics that can be viewed within IDMS through the following ways:

- DCMT, DCUF, and OPER commands
- Culprit PMA and PMI reports
- SREPORTS reports
- Performance Monitor Product screens (PMRM, PMIM, and PMAM)

However, if you do not know what you are looking for, this data may be indistinguishable and confusing. Report outputs span several pages and contain rows and columns of numerical data that require detailed analysis and correlation to form meaningful conclusions.

Figure 3: Display command, DCMT DISPLAY ACTIVE TASKS, to view active tasks within the IDMS system. Command output typically spans multiple pages, showing both system and user tasks.

```

D AC TAS
      Current max tasks      133
      Times at max tasks    3504
      Allocated DCE/TCE     241
      Number of tasks abended 2
      Number of tasks processed 4176
      Number of tasks active 137

Taskid Taskcd  Prog   LTERM   Pri Stat Stim  A(ECB)  ECB Type
000000000 *SYSTEM* *MASTER* 255 WAIT NOST 00383BF0 TCAECB
                                00052D4C PLESECB
                                363227A0 LTTMSECB
                                00383C0C Service Task ECB
000000001 *SYSTEM* *DBRC* 255 WAIT NOST 00086A08 DBRC WTOR ECB
                                3517F108 ESEECB
                                00027FA4 CCEECB
                                000280C4 CCEECB
                                00028188 CCEECB
                                00028214 CCEECB
                                000281E0 CCEECB
000000009 *DRIVER* UCFLINE 254 WAIT NOST 0005284C PLESECB
000000010 *DRIVER* UTAM 254 WAIT NOST 00052ACC PLESECB
                                006256D4 UTAM READ INIT ECB
                                00625854 UTAM RPL ECB
000000011 *DRIVER* SYSOUTL 254 WAIT NOST 00052E4C PLESECB
000000013 *DRIVER* UTAMLIN1 254 WAIT NOST 0005394C PLESECB
      PAGE 0001 - NEXT PAGE:

```

Would you like to simplify this user experience?

Scripting or coding languages can be used to parse and retrieve relevant data to compare values, determine performance issues, produce alerts, and therefore greatly reduce the length of time spent analyzing outputs. Through the use of the IDMS REST API calls, the values are returned in JSON format, which can be easily handled by many modern programming languages.

The following code example shows the JSON response from the IDMS REST call to return a list of active user tasks. The experienced DBA knows that seeing tasks waiting on the Program Definition Table (PDTECB) indicates a potential problem with available space in a storage pool, which can cause task delays while programs wait for pool availability.

```
{
  "taskNumber": 2581,
  "taskCode": "DCRTMAIN",
  "taskStatus": "WAIT",
  "taskPriority": 100,
  "currentProgram": "",
  "currentLinkLevel": 0,
  "lteName": "VL500125",
  "userId": "",
  "storageRceCount": 2,
  "programRceCount": 75,
  "runUnitRceCount": 0,
  "miscRceCount": 1,
  "privateNonXaStorage": 640,
  "privateXaStorage": 40320,
  "sharedNonXaStorage": 0,
  "sharedXaStorage": 3968,
  "programNonXaStorage": 0,
  "programXaStorage": 44576,
  "allocatedStorage": 40960,
  "systemModeTime": 0,
  "userModeTime": 0,
  "ecbAddress": "3B229858",
  "ecb1Type": "PDTECB",
  "ecb2Type": "",
  "ecb3Type": "",
  "dbkeyHolder": "0",
  "radixOfDbkeyHolder": "0",
  "dbkeyHolderTask": "",
  "defaultDictnode": "",
  "defaultDictname": "",
  "defaultDbnode": "",
  "defaultDbname": "",
  "defaultVersion": 0
}
```

Building the Future

Time-consuming and manual performance diagnostic activities performed by experienced DBAs can be turned into reusable scripts or programs, capturing the knowledge and expertise for the next generation of mainframe users. Problem detection and root cause analysis is simplified, saving valuable time to address issues before business impacts occur. By leveraging the IDMS REST API to invoke repeatable processes, you have increased flexibility to drill down into the data, to form insights, and to determine meaningfulness by understanding the relationships for corrective actions.

The following example shows how programming languages contain built-in methods (such as `JSON.parse`) to easily handle REST API response data. Key/Value pairs can be referenced by name and then used for subsequent analysis and steps.

```
// Produce an alert if active user tasks are waiting on the Program Definition Table
// (The JSON response is in variable "myJSON")
var myObj = JSON.parse(myJSON);
var string1 = 'PDTECB'
var string2 = 'WAIT'
if ((string1.localeCompare(myObj.ecb1Type) == 0) && (string2.localeCompare(myObj.taskStatus) == 0)
{
    performAlert();
}
```

If the user task is waiting for the Program Definition Table, then a warning is automatically generated to alert the DBA of a potential issue.

Executing REST API calls can also be performed easily within scripts and programs.

The following example shows an example of how a IDMS REST API call is used to display storage pools for a IDMS system. The following examples shows a Python code snippet for a GET call to retrieve a list of storage pools for a given IDMS system:

```
import requests

url = "https://myhost:myport/api/v1/storagePools/SYSDemo"
headers = {
    "cache-control": "no-cache",
    "accept": "application/json; charset=UTF-8",
    "cookie": "XSRF-TOKEN mytokenvalue",
    "Authorization": "Basic myauth"
}

response = requests.request("GET", url, headers=headers, verify=False)
```

DevOps Automation and Continuous Delivery

Using repeatable scripts or by capturing the logic of rules, performance impacts can be obtained and compared before and after IDMS system or application changes to ensure quality assurance and stability. Expected or target values for performance statistics become part of the DevOps process, where tests are run on an automation pipeline within the development cycle. The IDMS REST API enables you to retrieve data to help analyze accuracy and performance differences within your CI/CD pipelines for continuous deployment best practices. The open access to the various combinations of data provides you with countless opportunities, as well as more freedom and control.

IDMS system and database benchmarks can be automated with the IDMS REST API by utilizing scripting logic to validate performance metric ranges during the development cycle.

The example on the following page displays various statistics that without automation, would need to be manually analyzed and compared.

Figure 4: Display for the command, DCMT DISPLAY STATISTICS SYSTEM, to receive an overview of system statistics.

```

D AC TAS
  Current max tasks      133
  Times at max tasks    3504
  Allocated DCE/TCE     241
  Number of tasks abended 2
  Number of tasks processed 4176
  Number of tasks active 137

Taskid Taskcd Prog LTERM Pri Stat Stim A(ECB) ECB Type
000000000 *SYSTEM* *MASTER* 255 WAIT NOST 00383BF0 TCAECB
                                00052D4C PLESECB
                                363227A0 LTTMSECB
                                00383C0C Service Task ECB
000000001 *SYSTEM* *DBRC* 255 WAIT NOST 00086A08 DBRC WTOR ECB
                                3517F108 ESEECB
                                00027FA4 CCEECB
                                000280C4 CCEECB
                                00028188 CCEECB
                                00028214 CCEECB
                                000281E0 CCEECB
000000009 *DRIVER* UCFLINE 254 WAIT NOST 0005284C PLESECB
000000010 *DRIVER* UTAM 254 WAIT NOST 00052ACC PLESECB
                                006256D4 UTAM READ INIT ECB
                                00625854 UTAM RPL ECB
000000011 *DRIVER* SYSOUTL 254 WAIT NOST 00052E4C PLESECB
000000013 *DRIVER* UTAMLIN1 254 WAIT NOST 0005394C PLESECB
PAGE 00001 - NEXT PAGE:

```

With end-to-end mainframe DevOps, customers can achieve greater business agility to deliver innovations and to keep pace with business-centric demands. Existing DevOps solutions and products that are popular today leverage REST APIs and can be used to accelerate and automate mainframe development across hybrid and cross-platform applications and ecosystems.

The following workflow is an example of a DevOps workflow for updating a IDMS application:

1. A change request is issued and an alert is sent to the DBA.
2. System analysis and performance baselines are established for the IDMS application.
3. Changes are now ready to be promoted from development to QA.
4. Performance metrics are captured and compared to baselines.
5. If performance metrics meet or exceed the expected results, changes are promoted to production.

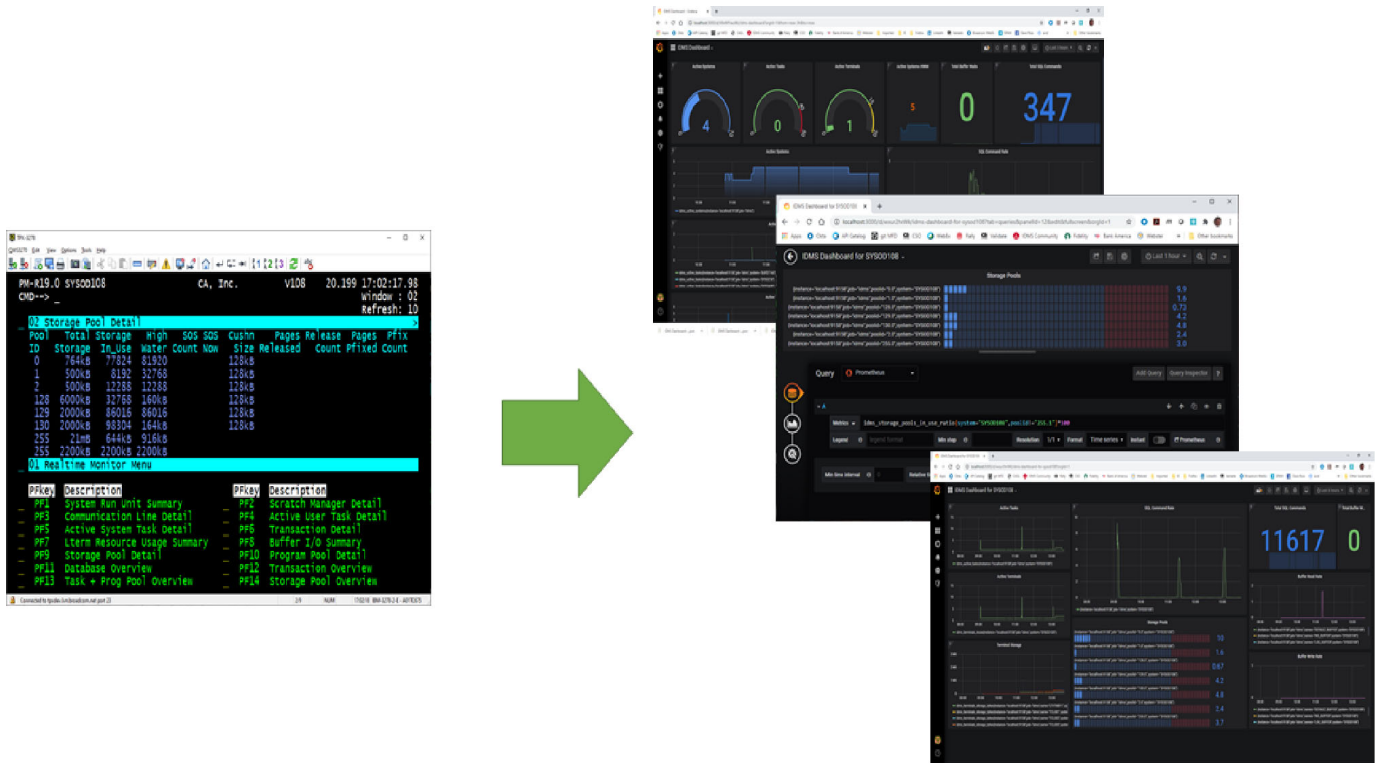
Modern and Graphical Real-Time Monitoring

Open source dashboard tools are familiar to the next-generation workforce and are generally more visually appealing and attractive, providing an alternative option for database monitoring without displacing existing tools and monitoring methods. Embracing open source solutions helps you to save costs and improves the overall return on investment (ROI) for integrating your mainframe and empowering enterprise innovation.

Would you like to deliver a better user experience, improve slow manual processes, and ease resource constraints?

The IDMS REST API enables you to move beyond traditional approaches, and to move from reactive mode to proactive techniques. Data is more sharable, easy to access, and supports self-service.

Figure 5: Sample use case: Transition from the mainframe terminal displays, such as PMRM screens with the IDMS Performance Monitor, to graphical dashboards with open source products.



In the above figure, a JSON exporter invokes the IDMS REST API every five minutes, populates a time series database called Prometheus, and is represented graphically by a web-enabled analytics product called Grafana. These tools not only aid displays, but are configurable for alerts and rules logic.

With enhanced visualization to various performance metrics, DBAs can more easily monitor operations to see trends and recognize unusual conditions that can indicate problems. Statistics exposed by the IDMS REST API can be collected at regular intervals for further analytical and historical diagnostics and queries. Conditions, such as Short on Storage (SOS), can be monitored by setting thresholds in the dashboard tool or generating alerts and notification emails. REST API calls and JSON data formats can even be consumed by familiar tools, such as Microsoft Excel, so that you can feed data into Excel reports or click a button that invokes and retrieves a metric value. Embracing open source and APIs can help you to deliver at the speed of business, securely and predictively, so that you can leverage systems data and services from mobile to mainframe.

Leveraging Your Investments for Future Growth

The IDMS REST API builds upon the existing value that customers have already invested in their IDMS mission-critical systems that have grown with and supported their businesses for decades. IDMS is a proven, reliable, scalable, and highly performant database management system. Our product strategy of “Embrace Open” positions us to be more agile in serving our customers’ modernization needs to optimize, simplify, and automate, and to help further exploit the benefits of the platform while helping address the skills challenge. The end goal is to enable our customers to leverage their existing investment and to transform user experiences by opening up IDMS through APIs.

How to Get Started?

- If you are a licensed IDMS/DB customer running the latest version (19.0), the IDMS REST API is available to you as part of the base license.
- You can download your copy of the IDMS Rest API online at [Support](#). To install your product, follow the installation procedures for your product on [Tech Docs](#). If you have any questions or require assistance, contact your [local technical support group](#) or visit the [Support Portal](#).
- Please visit the [IDMS IUA EIUA Community](#) site and participate in conversations regarding the IDMS REST API. Let us know what you think we should do next.
- Learn more by watching the webcast replay, [IDMS REST API: Overview and Demo](#).
- Learn more by watching the video, [APIs and Integration Using Open Source Monitoring Tools](#), delivered by David Ross, Database Architect (David.Ross@broadcom.com).
- For more information, contact Product Manager, Sheila Miller (Sheila.Miller2@broadcom.com) or Product Owner, Nakesha Newbury (Nakesha.Newbury@broadcom.com).

Broadcom, the pulse logo, Connecting everything, and Automic are among the trademarks of Broadcom and/or its affiliates in the United States, certain other countries, and/or the EU.

Copyright © 2020 Broadcom. All Rights Reserved.

The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, please visit www.broadcom.com.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.