Version 1 Release 1

IBM Db2 DevOps Experience for z/OS Administrator's Guide

SC27-9138-00

Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from the Db2 Tools Product Documentation page.

2020-03-27 edition

This edition applies to Version 1 Release 1 of IBM Db2 DevOps Experience for z/OS (product number 5698-DEX) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2019, 2020.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
© Rocket Software 2019, 2020.

Contents

About this information v

Chapter 1. Overview 1
- New and changed functions 1
- Features and benefits 8
- Components and architecture 9
- System requirements 10
- Service updates and support information 11
- Product documentation and updates 11

Chapter 2. Installation 13
- Installation roadmap 13
- Installation skills and authorities 14
- Configuring security 16
  - Granting catalog and bufferpool access 16
  - Configuring RACF privileges 17
  - Configuring Top Secret 18
  - Authorizing libraries and programs in APF 19
- Installing Db2 DevOps Experience 20
- Upgrading Db2 DevOps Experience 23

Chapter 3. Using 27
- Roles and responsibilities 27
- Managing 27
  - Registering subsystems 28
  - Creating teams and environments 29
  - Assigning users to teams 30
  - Setting soft storage limits 30
  - Defining applications 31
  - Creating site rules 31
- Provisioning instances 33
  - Editing objects 34
- Querying subsystems 34

Chapter 4. Troubleshooting 35
- Connection timeout 35
- zss server log error when starting Zowe 35

Chapter 5. Reference 37
- Sample RACF security setup job card 37
- Configuration file reference 39
- Complex site rule variables 42
- Installing Swagger UI on z/OS 43

Notices 45

Index 49

About this information

IBM Db2 DevOps Experience for z/OS is a web application on the Zowe Desktop and a server with a rich set of REST API-driven microservices that work with Db2 for z/OS. Developers, administrators, and development operations engineers can use IBM Db2 DevOps Experience for z/OS to discover and define database applications, provision database application objects, and edit and manage those objects.

Administrators can manage the associated teams, environments, and users.

Team administrators and team members can manage, review, and approve changes, provision application instances, and interact directly with registered Db2 systems with the built-in SQL processor.

Tip: To find the most current version of this information, always use IBM Knowledge Center, which is updated more frequently than PDF books.

© Copyright IBM Corp. 2019, 2020

Chapter 1. Overview

IBM Db2 DevOps Experience for z/OS speeds up mainframe application development release cycles by enabling developers to provision and work with their own application instances to drive needed changes for their sprint.

A web interface makes it easy for administrators to register subsystems, define applications, create teams and environments, and set rules and limits around provisioning. Developers can then quickly provision application instances, manipulate objects using data definition language (DDL), and create pull requests so that approvers can review, approve, and merge database object changes.

New and changed functions

This topic summarizes the recent enhancements and changes in IBM Db2 DevOps Experience for z/OS.

V1.1.0.6 (PTF7): March 15, 2020

Description	Related APARs
Site rules: • Fixed: Incorrect validation messages were displayed when editing site rules created in PTF5. • Fixed: Environment deletion was not cascaded to complex site rules. • Fixed: The `db2complexsiterules.yaml` file was incorrectly stored in the `var/conf` directory.	PH22352
Db2 support: • Fixed: Added support for FL levels lower than 500.	PH22073
Subsystem registration: • Fixed: GOCCM generated during subsystem registration was incorrect.	PH22070

V1.1.0.6 (PTF6): December 13, 2019

Description	Related APARs
Site rules: • Extended site rules to all objects • Support for complex site rules	PH19180
Subsystem registration: • Added "Copy From" option for registering new subsystems using the definition of an existing subsystem	PH19180

© Copyright IBM Corp. 2019, 2020

Description	Related APARs
Application and instance management: • API support for provisioning into a specific schema, database, and collection • Performance fixes for instance management • Fixed: team administrator deprovisioning issues • Fixed: failing to provision an instance containing a Java stored procedure with a JAR object in the Db2 catalog • Fixed: Not ignoring collid when comparing the DDL of stored procedures • Fixed: Allocating datasets as NEW,KEEP could cause errors. Now allocating datasets as NEW,CATLG. • Fixed: Added support for providing collid/schema name as input when provisioning an instance	PH19180, PH12677, PH19167, PH19179, PH19160
Security and authorization: • Support for CA Top Secret® security manager • Support for secondary authorization for dbaUser	PH19180
Installation: • Enhanced installation script	PH19180
Job Administration:	
Miscellaneous: • Support for open source Zowe as a prerequisite • Support for providing a custom HLQ (high level qualifier) for temporary files • Fully documented Swagger request JSON	PH19180, PH19110

V1.1.0.5 (PTF5): August 30, 2019

Description	Related APARs
Application discovery: • Added a “Select All” option for selecting objects while creating an application • Added “Expand All” and “Collapse All” options for object selection when creating an application • The Application Builder now requires parent applications to have a Db2 Object	PH15869
Application and Instance Management: • Enhanced capabilities for deploying changes into higher level environments (i.e. integration testing, staging, pre-production, production) • Created an API to keep track of the number of times an application has been provisioned, by whom, and when • Fixed an error in the Swagger documentation of the POST body that was causing an internal server error	PH15869, PH16034

Description	Related APARs
DDL management: • Editing DDL and adding objects are now prohibited after a pull request is issued against an instance • New objects do not appear in the difference section after the pull request is approved and merged.	PH15869, PH16036
Security/Authorization: • Added support for CSRF_SWITCH in z/OSMF • Made API changes to support secondary authorization for dbaUser	PH15869
Installation: • Created a sample JCL procedure to start the server • Made all SMP/E-managed directories read-only and all user-managed directories read/write • The setup wizard now recognizes the lower cased “dbaUser” • Users only enter a port number in one location now, instead of two, to prevent problems that arose when users did not enter the same port number in each location.	PH15869, PH16035
Job Administration: • You can now configure a prefix for submitted jobs	PH15869
Miscellaneous: • Added site rules for missing data types and storage groups • Added support for subsystem location names with more than 8 characters or underscores (_). • Stopped unnecessary regeneration of the DDL on application color change • Resolved the "Error searching Db2 objects" that occurred when searching for table spaces • Resolved a problem that caused the DOE Server to stop and leave objects in uncertain states • Resolved a problem with SYSAFF discovery and validation	PH15869, PH15925, PH12365, PH09306

V1.1.0.4 (PTF4): July 3, 2019

Description	Related APARs
Application discovery: • When you create an application and select an object, its child objects are selected automatically. • You cannot add implicit sequences to applications.	PH13718

Description	Related APARs
Application and Instance Management: • Fixed an issue handling user instance limits when working with multiple environments. • The Admin team can now access the Manage tab. • You can search for teams while provisioning instances. • The Admin team can de-provision team instances. • You can no longer include cast functions in application definitions. • Added support for applying site rules to databases, tablespaces, tables, indices, and stored procedures. • Added API support to allow deployment of an application's master DDL to a target subsystem.	PH13718
API: • Fixed issues with /policy/pull-requests POST API where lower case reviewer names generated an error and required manually removing the related yaml file and restarting the DOE server. The API now converts reviewer names to upper case, and makes sure that reviewers are on a team associated with the pull-request application. • Fixed an issue with the `swagger.json` API document where the __yamlClass in the requiredApps section of the /policy/applications POST API did not display clearly.	PH12194, PH09308
DDL management: • Fixed an issue where, when you edited DDL in a provisioned instance and the browser session times out, after refreshing the screen the **Apply object changes** button was incorrectly grayed out. • On the "Edit DDL & SQL" page, the **Revert** and **Revert All** buttons are disabled by default.	PH13718, PH09302
Installation: • Added a field to the DevOps set up wizard for users to enter a name for a data set that will contain artifacts created when a DevOps user registers a Db2 subsystem. The value is stored in an editable **`cmBatchDsn`** parameter in the `configuration.yaml` file. • Added four ISPF library parameters to the `configuration.yaml` file. Previously, these values were hard-coded. In `configuration.yaml`, they are editable.	PH13718
Pull requests: • Fixed issue where the IDs of de-provisioned instances were displaying in the Past Activity UI.	PH13718, PH12194

Description	Related APARs
Cosmetic changes: • Fixed a UI issue with overlapping objects in the "Edit DDL & SQL" page.	PH13718
Documentation: • Improved the Db2 Set Up wizard field labels and hover help. • In the appendix, added a JCL script that you can run to configure DevOps security. • In a new Granting catalog and bufferpool access topic, added a DCL script that you can run to grant to PUBLIC the use of all bufferpools and also SELECT access on Db2 catalog tables (except SYSIBM.USERNAMES). This is required before you install DevOps. • Simplified and clarified some installation topics. • In the DevOps installation topic, clarified when and how the user should restart Zowe during installation. • In the DevOps installation topic, clarified how to specify values for DOE_HOME and LIB_DIR in the doeserver.sh file.	PH13718

V1.1.0.3 (PTF3): May 14, 2019

Description	Related APARs
Application discovery: • Added views, synonyms, aliases, stored procedures, functions, and triggers to the list of object types you can add to applications. • Added a feature in the object discovery UI to exclude dependent objects. By default, when you select objects their dependent objects are included in the application. • Fixed an issue with searching for objects using the underscore(_) wildcard.	PH11593

Description	Related APARs
Application and Instance Management: • Fixed an issue with objects that were added to an instance and then merged not propagating to other instances of the application after a pull. • Fixed issue with merge conflicts occurring when creating application instances after new objects were added to the application. • Fixed an issue with provisioning instances with source data if a table in the application name contained an underscore(_). • Fixed an issue with creating applications containing system temporal tables. • Fixed an issue with creating applications containing archive tables. • Fixed an issue with UDF type objects that were added to an instance and then merged not propagating to other instances of the application after a pull. • Fixed an issue with creating applications containing large objects (LOBs). • Fixed an issue with creating applications during DDL generation.	PH11593, PH11598
DDL management: • Only members of an application's owning team can modify its instance DDLs. (All users can still provision instances and view instance DDLs.) • Fixed an issue with storing auxiliary table DDLs in two places. • Fixed an issue with DDL changes not reverting after an attempt to apply object changes failed.	PH11593
Support update: • Support for Db2v12 function level 504	None
Cosmetic changes: • Fixed an issue with search on the Instances page being case-sensitive. It is no longer case-sensitive. • Fixed an issue with the "Required Applications" dropdown field displaying incorrectly in the UI. • Fixed an issue with allowing users to type lower-case characters in subsystem registration page fields that should only allow upper-case values.	PH11593

Description	Related APARs
Documentation: • Updated the system requirements with more detail. • Updated the DOE installation steps with several small improvements. • Updated a security topic to include providing APF authorization for Db2 DevOps Experience programs in addition to libraries.	None

V1.1.0.2 (PTF2): April 3, 2019

Description	Related APARs
Application discovery: • Re-factored application discovery functionality for enhanced performance and usability. • Redesigned the graph display and selection mechanism with a tabular format populated from user-specified search criteria. • Users can select individual objects and add them to a list of objects to be included in an application. • Upon submission, the system automatically adds any required object(s) to ensure the application definition is valid. • Objects that depend upon selected objects are also automatically added.	None
Site rules: • Optionally, site rules can now be applied to future applications by default. • Fixed an issue with the “INDEX UNIQUE must_be_absent/present” rule.	None
DDL management: • Made cosmetic usability enhancement and minor content corrections to the DDL Diff page. • Fixed an issue with adding objects to existing applications. • Fixed an issue with the state of the "Apply these DDL changes" button. • Fixed an issue with nested views when generating DDL. • Fixed an issue with aliases when generating DDL. • Fixed issues with the DDL Change Report content.	PH09305

Description	Related APARs
Application and instance management: • Users can now refresh page content for Application Management and Instance Management. • Fixed issues with application deprovisioning. • Fixed an issue with provisioning applications without a database or tablespace. • Fixed an issue with stored procedure qualifiers during instance provisioning.	PH09307
Pull request management: • Fixed an issue with the list of pull requests to approve. • Fixed a problem with viewing stored procedure changes in pull requests.	None
Cosmetic changes: • Enhanced hover behavior to ensure all data is visible. • Fixed the application tile header color behavior. • Fixed selected application highlight behavior. • Fixed the position of the status bar.	None
User and team management: • Fixed a performance issue with the user and team management pages.	None
Documentation: • Improved Swagger documentation.	PH09303, PH09304, PH09308

Features and benefits

IBM Db2 DevOps Experience for z/OS allows organizations to bring mainframe Db2 applications to market more rapidly, at lower costs, and with less risk.

Db2 DevOps Experience offers several unique and significant features that you can use to improve your DevOps practices.

Simple self-service provisioning on demand

A modern web-based UI makes it easy for application developers without mainframe experience to work in a z/OS environment. Application developers can provision and deprovision their own Db2 for z/OS environments with very little training.

Role-based administration

Create teams, environments, and permissions with review and approval of change requests.

Management directives

Integrate management directives into your organization's existing continuous delivery process. Generate and enforce rules for attributes.

Infrastructure as a service

Extend infrastructure as service by representing database as code.

Platform as a service

Bring platform as a service practices to Db2 for z/OS, empowering faster development, testing, and integration of application changes through automation and standardization.

API support for CI/CD applications

Use APIs to automate your Db2 processes using continuous integration or continuous delivery tools such as Urban Code Deploy, Jenkins, or Bamboo.

Source-control using Git

Manage your Db2 application objects with source-controlled storage, retrieval, review, and versioning using Git.

Components and architecture

Db2 DevOps Experience is a web-based application that runs in the Zowe environment.

Components

Zowe open source environment
The Db2 DevOps Experience UI runs as a Zowe plug-in application. For more information about Zowe, see https://zowe.org.

DevOps Server
The DevOps server contains all of the necessary backend services to do the following:

- Provide information to the UI
- Integrate with Git for application DDL management as code
- Write instrumentation logs
- Manage product configuration data
- Discovery services for application definition
- Communicate with z/OSMF for job submission and management
- Communicate with the virtual desktop and ZSS Cross Memory server to call z/OS functions

z/OSMF
Db2 DevOps Experience uses z/OSMF to submit z/OS jobs. These jobs perform many functions, such as merging DDL into the master Git branch and source Db2 objects, object and data provisioning into environments, and deployment of changes to new instances.

Web Browser
The Db2 DevOps Experience graphical user interface is a Zowe web application. Zowe works with Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

REST APIs
REST APIs are hosted by the DevOps server to provide many database-as-a-service functions.

System requirements

Before you install and configure Db2 DevOps Experience, make sure that your environment meets the following minimum hardware and software requirements.

System requirements

- A DASD footprint of approximately 2GB
- Physical memory heap of 1GB.
- z/OS version 2.2 with PTF UI46658 or above
- IBM z/OS Management Facility (z/OSMF) version 2.2 or above
  - All Zowe and Db2 DevOps Experience users must be members of either the z/OSMF user or administrator security group. For more information about z/OSMF security configuration, see the IBM Knowledge Center.
  - For z/OS version 2.2 only, the z/OSMF parameter `CSRF_SWITCH=OFF` must be set in your `IZUPRMxx` PARMLIB member. For more information about displaying and modifying your z/OSMF server settings, see the IBM Knowledge Center.
- IBM Db2 for z/OS version 11 or above. Participating Db2 databases must have the T4 Dataserver Driver bound, and DSNTEP2 bound as Plan Name “DSNTEP2." The Db2 instance must contain the Db2-supplied ADMIN_COMMAND_DB2 stored procedure.
- IBM Resource Access Control Facility (RACF) or Top Secret
- Zowe version 1.5.x or above. If using Top Secret, you must install Zowe 1.7.1. For installation instructions, see the Zowe documentation.
- The following open source tools on z/OS. You can download them from the Rocket Software Community web site with a Rocket Software Community account:
  - Git for z/OS version 2.14.4_b08.181016 or above, Gzip version 6.0 or above, and Bash version 4.3 or above. For more information and to download open source z/OS tools, see the Rocket Open Source Tools web site.
  - Perl version 5.24. For more information and to download Perl, see the Rocket Perl for z/OS web site.
  - Optional: OpenSSL for z/OS. Only required if you are using a self-signed certificate for z/OSMF. For more information and to download open source z/OS tools, see the Rocket Open Source Tools web site.
- IBM 64-bit SDK for z/OS, Java Technology Edition Version 8.0. For information, see the IBM Java SDK downloads web site.
- IBM SDK for Node.js - z/OS, version 6.14.4.0 and above. For information, see the IBM SDK for Node.js web site.
- Curl, version 7.x or higher. To download, see the curl archive web site.
- findutils 4.4.2. For information, see the Findutils web site.
- Optional: Swagger. After you install Db2 DevOps Experience, the API documentation is available in OpenAPI format in the following file:

  ```
  <unpax_filepath>/doeserver/var/opt/doc/swagger.json
  ```

  Swagger is an open source specification for defining REST APIs. A Swagger document is the REST API equivalent of a WSDL document for a SOAP-based web service. You can use Swagger tools to visualize and test the API, and generate a Software Development Kit (SDK) in different code languages. To see an example of a Swagger document, see Swagger petstore. You can also download and install Swagger as a Node.js application on z/OS. For information about installing Swagger UI, see “Installing Swagger UI on z/OS” on page 43

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z~OS

Product documentation and updates

Db2 Tools information is available at multiple places on the web. You can receive updates to Db2 Tools information automatically by registering with the IBM® My Notifications service.

Information on the web

The most current version of this information is available on IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter

A PDF version of this information is available on the Db2 Tools Product Documentation web page; however, IBM Knowledge Center is updated more frequently than PDF books. The Db2 Tools Product Documentation web page is located at:

http://www.ibm.com/support/docview.wss?uid=swg27020910

IBM Redbooks® publications that cover Db2 Tools are available from the following web page:

http://www.redbooks.ibm.com

The IBM Information Management System website shows how IT organizations can maximize their investment in Db2 databases while staying ahead of today's top data management challenges:

https://www.ibm.com/analytics/us/en/db2/db2-for-zos/

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM product documentation, use one of the following options:

- Use the IBM Knowledge Center **Contact Us** link.
- Use the online reader comment form, which is located at http://www.ibm.com/software/data/rcf/.
- Send your comments by email to comments@us.ibm.com. Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.

Chapter 2. Installation

To install IBM Db2 DevOps Experience for z/OS, follow the installation roadmap.

Installation roadmap

Follow these steps to prepare your environment and install and configure Db2 DevOps Experience.

1. Make sure your environment meets the “System requirements” on page 10.
2. Review the “Installation skills and authorities” on page 14 required to install Db2 DevOps Experience components. Find people in your enterprise who can perform any tasks that you cannot perform.
3. Perform SMP/E installation of the product by following instructions in the IBM Db2 DevOps Experience for z/OS Program Directory.

   Note: You must not install Db2 DevOps Experience in the same CSI global zone as an installed DB2 Object Comparison Tool for z/OS. Db2 DevOps Experience installs its own version of the comparison tool and installing two versions in the same zone causes conflicts.
4. Install Zowe by following instructions in the Zowe documentation.
5. Configure security privileges and authorizations.
6. Install and set up Db2 DevOps Experience

© Copyright IBM Corp. 2019, 2020

Installation skills and authorities

Before you install Db2 DevOps Experience components, find and organize the system programmers and other stakeholders with the skills and access to grant required authorities to users and complete the installation.

Table 1. Required authorities, skills, and access by role

Role	TSO ID Authorities
Installer and administrator	• On z/OS: Create users and groups, grant authorities, create CDT facilities, define and work with started tasks, make PARMLIB updates, grant APF authorizations, and update started task JCL processes. • On UNIX System Services (USS): Run shell scripts and modify files. • For troubleshooting, basic z/OS and USS skills, such as viewing log files and job outputs. • Connected to the IZUADMIN or IZUUSER z/OSMF groups. • Connected to the Db2 DevOps Experience RACF users group. • Connected to the Db2 DevOps Experience RACF administrators group with the SPECIAL attribute. • Class authorization for Zowe. After you install Zowe, you can grant class authorization by entering the following command in z/OS: `ALU <userid> CLAUTH(ZOWE)` • Read authority on OMVSAPPL in the APPL class (only required if the APPL class is defined). • Read access to USS directories containing the SMP/E installed files. • Write access to USS directories where Zowe and Db2 DevOps Experience will be installed. • To log in to the Zowe desktop and configure Db2 DevOps Experience in the setup wizard, basic Db2 knowledge and skills, and write access to the `doeserver/var/conf` directory in USS. • Authority to create Db2 objects, such as databases, tables spaces, and tables, and the ability to bind plans and packages.

Table 1. Required authorities, skills, and access by role (continued)

Role	TSO ID Authorities
Team administrator	• Connected to the following z/OSMF groups: IZUUSER • Connected to the Db2 DevOps Experience RACF users group. • ALTER authority on the ZOWE.DOE.<*teamuuid*> group. (Typically granted when administrators assign team administrators in Db2 DevOps Experience.) • Read authority on OMVSAPPL in the APPL class (only required if the APPL class is defined). • Authority to select Db2 catalog tables, and use storage groups and buffer pools. To grant this authority, you can use the DCL script provided in the topic "Granting catalog and bufferpool access" on page 16.
Developer	• Connected to the following z/OSMF groups: IZUUSER • The Db2 DevOps Experience users group. • READ authority on ZOWE.DOE.<*teamuuid*> group. (Typically granted when administrators assign users to teams in Db2 DevOps Experience.) • Read authority on OMVSAPPL in the APPL class (only required if the APPL class is defined). • Authority to select Db2 catalog tables, and use storage groups and buffer pools. To grant this authority, you can use the DCL script provided in the topic "Granting catalog and bufferpool access" on page 16.
System DBA user. (When a developer creates an instance, Db2 DevOps Experience acts as this user to create the instance database and grant the developer authority to it.)	• z/OSMF groups: IZUUSER • A non-expiring password (recommended) • SYSADM authority in Db2 (If SYSADM is granted through secondary auth ID you must supply the secondary auth ID in the installation wizard or `configuration.yaml` file.) • Specify this user's ID and password in the Db2 Setup Experience wizard.

Table 2. Required USS TCP/IP port numbers		
Component	**Use**	**Port**
Db2 DevOps Experience	Internal communication	• 3444 • 3446 • 8182 (Graph database). If necessary, you can change this value in the `<active>/doeserver/var/conf/gremlin-server-doe.yaml` file. • 12023 (serves the REST APIs). If necessary, you can change this value in the `<active>/doesamplib/install.properties` file. You must then rerun the doe-install.sh and Zowe deploy.sh scripts.
Swagger (optional)		• 8080

Configuring security

Before installing Db2 DevOps Experience, you must first grant PUBLIC access to bufferpools and Db2 catalog tables, configure RACF privileges, and APF authorize certain libraries on your system.

Granting catalog and bufferpool access

Before you install and configure Db2 DevOps Experience, make sure that you grant users of Db2 DevOps Experience access on following Db2 catalog tables:

Run the following DCL script to grant the required access:

```
GRANT SELECT ON TABLE SYSIBM.SYSAUXRELS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKS2 TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDIST TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDISTSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCOLSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCOLUMNS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCONSTDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCONTEXT TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCONTEXTAUTHIDS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCONTROLS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCOPY TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSCTXTTRUSTATTRS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDATATYPES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDEPENDENCIES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMY1 TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMYA TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMYE TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDYNQRY TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSDYNQRYDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSENVIRONMENT TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSFIELDS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSFOREIGNKEYS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXPART TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXSPACESTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXSTATS TO PUBLIC;
```

```
GRANT SELECT ON TABLE SYSIBM.SYSJARCONTENTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSJAROBJECTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSJAVAOPTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSJAVAPATHS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYCOLUSE TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTARGETS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTARGETSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTGTDIST TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTGTDISTSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSLEVELUPDATES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSLOBSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSOBDS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSOBJROLEDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPACKAGE TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPACKCOPY TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPACKDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPACKLIST TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPACKSTMT TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPARMS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPENDINGDDL TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPENDINGOBJECTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPKSYSTEM TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPLAN TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPLANDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSPLSYSTEM TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSRELS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSROLES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_OPTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_SRC TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSEQUENCES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSEQUENCESDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION_EX TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION_STATUS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSTMT TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSTOGROUP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSTRINGS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSSYNONYMS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABCONST TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABLEPART TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACE TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACESTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES_PROFILES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTABSTATS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSTRIGGERS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSUTIL TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSUTILX TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSVARIABLES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSVIEWDEP TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSVIEWS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSVOLUMES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSXMLRELS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSXMLSTRINGS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSXMLTYPMOD TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.SYSXMLTYPMSCHEMA TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.XSRANNOTATIONINFO TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTCOMPONENTS TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTHIERARCHIES TO PUBLIC;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTS TO PUBLIC;
```

Configuring RACF privileges

Db2 DevOps Experience uses SAF groups to assign initial administrator roles.

About this task

The DevOps server runs as a started task. An example of the JCL procedure can be found in /<unpax path>/doeserver/var/bin/DOESRV.JCL. As an alternative to configuring security following tasks in this section, you can use the JCL job card provided in the appendix topic “Sample RACF security setup job card” on page 37.

Procedure

1. Create a RACF group for administrators and a RACF group for users.

Note: If you will be adding administrators during the installation procedure, RACF group names must start with one of the following prefixes: "ZOWE", "MVD", or "MXM". If you do not intend to add administrators during the installation procedure, there are no restrictions on RACF group names.

2. Connect all system administrators, team administrators, and users to the user group.
3. To give system administrators administrator status in Db2 DevOps Experience, connect all system administrators to the administrator group with the SPECIAL attribute.
4. Define the RACF class that will contain all administration profiles. The default class name is ZOWE. Define the ZOWE class to the CDT class:
   a) Ensure that the CDT class is active and RACLISTed.
   b) Issue the following TSO command:

```
RDEFINE CDT ZOWE UACC(NONE)
 CDTINFO(
    DEFAULTUACC(NONE)
    FIRST(ALPHA) OTHER(ALPHA,NATIONAL,NUMERIC,SPECIAL)
    MAXLENGTH(246)
    POSIT(607)
    RACLIST(DISALLOWED))
```

   You might receive the following message when defining your new class : "Warning: The POSIT value is not within the recommended ranges for installation use. The valid ranges are 19-56 and 128-527." You can ignore this message.
   c) Refresh the CDT class using the TSO command : **SETROPTS RACLIST(CDT) REFRESH**
   d) Ensure that the ZOWE class is active. Issue the following command to activate it: **SETROPTS CLASSACT(ZOWE)**

   For more information about the RACF security administration and administering dynamic class descriptor tables, see the IBM Knowledge Center https://www.ibm.com/support/knowledgecenter/.
5. Grant all system administrators and team administrators the CLAUTH attribute for the ZOWE class.
6. If the APPL class is active and OMVSAPPL is defined, use the following RACF command to grant all Db2 DevOps Experience users read access on the OMVSAPPL resource:

```
PERMIT OMVSAPPL CLASS(APPL) ACCESS(READ) ID(<user_name>)
```

   Some ZSS APIs require users to have READ access to OMVSAPPL. If access is restricted, you can get the following error message:

```
BPXTLS failed: rc=-1, return code=163, reason code=0x0be80820
```

7. To start the DevOps server, copy the JCL procedure to your system PROCLIB library and direct the USS path to the location where you installed DevOps Experience.

   Note: If the USS command is long, you can add a line continuation character in column 72 and continue the command on the next line starting in column 16.
8. To define DOESRV as started task, the assigned user must have an OMVS segment. For sample RACF commands to define the user ID and started task, see“Sample RACF security setup job card” on page 37.

Configuring Top Secret

To configure Top Secret to secure Db2 DevOps Experience, you creating a department with a DCA and adding Zowe to it. Then you create profiles for Db2 DevOps Experience administrators and users, add Zowe to a resource class, add teams to a resource, and then grant required privileges to the DCA.

About this task

To use Top Secret, you must install Db2 DevOps Experience on Zowe version 1.7.1.

The person performing these steps must have required authority, for example a central security administrator. For more information on configuring Top Secret, see the Top Secret tech docs portal.

Procedure

1. To create a department for Zowe, use the following command:

```
TSS CREATE(ZOWEDEPT) TYPE(DEPARTMENT) NAME('ZOWE DEPARTMENT')
```

2. To create a DCA (Department Control Acid), use the following command:

```
TSS CREATE(ZOWEDCA) NAME('ZOWE DCA') TYPE(DCA) PASSWORD(HELP,,EXP) DEPARTMENT(ZOWEDEPT)
```

3. To create profiles, use the following commands:

```
TSS CREATE(MVDSPEC) TYPE(PROFILE) NAME('ZOWE ADMINISTRATORS') DEPARTMENT(ZOWEDEPT)
```

```
TSS CREATE(MVDUSER) TYPE(PROFILE) NAME('ZOWE USERS') DEPARTMENT(ZOWEDEPT)
```

4. To create a maskable resource class, use the following command:

```
TSS ADDTO(RDT) RESCLASS(ZOWE) ATTR(MASK,LONG) ACLST(READ,ALL)
```

5. To create a resource to contain teams, use the following command:

```
TSS ADDTO(ZOWEDEPT) ZOWE(ZOWE.)
```

6. To assign required privileges to DCAs, use the following commands on each DCA:

```
TSS ADMIN(DCA) MISC9(GENERIC)
```

```
TSS ADMIN(DCA) ZOWE(INFO,XAUTH)
```

```
TSS ADMIN(DCA) ACID(CREATE,MAINTAIN)
```

7. To assign required privileges to users, use the following commands on each user:

```
TSS PERMIT(USER) ZOWE(ZOWE.) ACTION(ADMIN)
```

```
TSS ADMIN(USER) ZOWE(INFO)
```

Authorizing libraries and programs in APF

You must provide APF authorization for Db2 DevOps Experience libraries and programs.

Before you begin

At a minimum, you must complete the SMP/E installation before you can provide APF authorization. For more information about installing Db2 DevOps Experience, see “Installation roadmap” on page 13.

About this task

In the authorized program facility (APF) of the z/OS system on which you will be running Db2 DevOps Experience, take the following steps:

Procedure

1. Authorize the following libraries:
   - <hlq>.SDOELINK
   - <hlq>.SDOELLIB
2. To authorize the required programs, take the following steps:
   a) Modify the SYS1.PARMLIB(IKJTSOxx) data set.
   b) Add the ADB2ATH and ADB2UTIL programs to both the AUTHPGM section, and the AUTHTSF section.

c) Activate the changes at the next IPL, or to activate them immediately enter the following TSO/E command:

```
PARMLIB UPDATE(xx)
```

Installing Db2 DevOps Experience

To install Db2 DevOps Experience you configure Zowe plug-ins, run Db2 DevOps Experience Setup, then finally set up the Db2 DevOps Experience server.

Before you begin

- You can install or upgrade Db2 DevOps Experience by granting permissions to edit files and run scripts from the directory where the files were extracted during SMP/E installation. If that is how you installed the previous PTF or plan to install the current PTF, see the PTF HOLD DATA for information. This task assumes that you installed the previous version PTF and will install the current PTF by first copying required files from the extracted location to a different USS directory.
- You can install or upgrade Db2 DevOps Experience by granting permissions to edit files and run scripts from the directory where the files were unpaxed during SMP/E installation. If you installed the previous PTF or plan to install the current PTF using that process, see the PTF HOLD DATA for information. This task assumes that you installed the previous version PTF and will install the current PTF by first copying required files from the unpaxed location to a different USS directory.
- Make sure that you find and organize the system programmers and other stakeholders with the skills and access to grant required authorities to users and complete the installation. For information, see “Installation skills and authorities” on page 14.
- When you view referenced Zowe documentation, you must view the 1.5.x version of that documentation.

About this task

This task uses the following variables to represent file locations that are specific to each user.

- <unpax_filepath> is the location where you extracted the Db2 DevOps Experience files during SMP/E installation.
- <zowe_directory> is the directory where you installed Zowe.
- <activeLibLocation> is the location where you will install and run the Db2 DevOps Experience server and UI.
- <copied_doesamplib_filepath> is the location where you will copy the doesamplib folder.

Procedure

1. In your UNIX System Services user profile, specify the following variables:

```
export _BPXK_AUTOCVT=ON
export _CEE_RUNOPTS='FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)'
export _TAG_REDIR_IN=txt
export _TAG_REDIR_OUT=txt
export _TAG_REDIR_ERR=txt
export NODE_HOME=<node_server_path>/node-v6.14.4-os390-s390x
export NODE_LIB_PATH=$NODE_HOME/lib/node_modules
export PATH=$NODE_HOME/bin:$PATH
```

2. Copy the <unpax_filepath>/doesamplib directory to a directory other than <unpax_filepath>. Do not change the directory name, it must be doesamplib.
3. To grant read/write access to the copied doesamplib directory, enter the following command:

```
chmod -R +w <copied_doesamplib_filepath>
```

4. Open the <copied_doesamplib_filepath>/install.properties file and specify values for the following properties:
   - SMPELocation (required) - The location of the unpaxed SMP/E files.
   - currentMaintenanceWriteLibLocation (optional) - Leave this field blank for first-time installation. The installation script copies the policy directory, configuration.yaml file, and cacerts file to this location, and searches for a /var directory in this location.
   - newMaintenanceReadLibLocation (required) - A directory for Db2 DevOps Experience to copy read-only files to. (If this location is the same as the SMPELocation, no files are copied since the source and target locations are the same.)
   - newMaintenanceWriteLibLocation (required) - A directory for Db2 DevOps Experience to copy read/write files to.
   - activeLibLocation (required) - The active workspace path that the installation scripts use to create links to files copied to the directories specified in the newMaintenanceReadLibLocation and newMaintenanceWriteLibLocation properties. Also used to start the Db2 DevOps Experience server and user interface.
   - JAVA_HOME (required) - The directory where Java is installed.
   - LEIBNIZ_HTTP_PORT (required) - The port that the doe-server.sh script uses to start the server.
   - OPENSSL_LOCATION (required) - The location where the installation script can find OpenSSL files.
   - MEMORY_SIZE (required) - The maximum amount of physical memory allocated to the Db2 DevOps Experience server. The default value of 1024 (approximately 1GB) is the minimum required heap size.
5. To run the installation script, navigate to <copied_doesamplib_filepath> directory and enter the following command:

```
bash install.sh
```

6. To configure Zowe to work with Db2 DevOps Experience, navigate to the <activeLibLocation>/installer/install/ directory and enter the following command:

```
bash doe-install.sh <zowe_directory>
```

7. Stop and restart Zowe by following steps in the Zowe documentation topic Starting and stopping the Zowe runtime on z/OS.

   If you get a zss log error related to permissions, see the troubleshooting topic, “zss server log error when starting Zowe” on page 35.
8. To verify that the ZSS Cross Memory Server is running in the same LPAR as Db2 DevOps Experience, look for the active process ZWESIS01.
9. To configure the Git environment for ZSS, update the following fields in the <activeLibLocation>/doeserver/var/conf/configuration.yaml file:

```
gitDir: /<gitdir>/bin
gitEnv:
- _BPXK_AUTOCVT=ON
- _CEE_RUNOPTS=FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)
- _TAG_REDIR_IN=txt
- _TAG_REDIR_OUT=txt
- _TAG_REDIR_ERR=txt
- LIBPATH=/<gitdir>/lib/perl5/5.24.0/os390/CORE
- PERL5LIB=/<gitdir>/lib/perl5
- MANPATH=/<gitdir>/man
- GIT_SHELL=/<gitdir>/bin/bash
- GIT_EXEC_PATH=/<gitdir>/libexec/git-core
- GIT_TEMPLATE_DIR=/<gitdir>/share/git-core/templates
- GIT_MAN_PATH=/<gitdir>/man
zssHost: <hostname>
zssPort: <portnumber>
teamPrefix: ZOWE.DOE
```

Note:

- The value of the `zssHost` field must be "127.0.0.1" or "localhost". By Zowe default, the ZSS server only listens on 127.0.0.1.
- <gitDir> is the installed Git directory and <portnumber> is the port that ZSS is using on that computer.
- To see documentation of the entire `configuration.yaml` file, see "Configuration file reference" on page 39.

10. Optional: To specify job codes for Db2 DevOps Experience jobs so that they are easier to find in z/OS interfaces such as SDSF, perform the following steps:
    a) Edit the `configuration.yaml` file.
    b) Add a **`jobPrefix`** key.
    c) In the **`jobPrefix`** key, add a value that is between 1 and 7 characters as a prefix for Db2 DevOps Experience job codes. Job codes are 8 characters long, so z/OS will generate characters not specified in the prefix. For example, if you specify the 6-character prefix value "IBMDOE", z/OS will generate the remaining two characters to complete the job code. Standard z/OS job code naming rules apply, so the prefix must start with an alphabetical or a national ($, #, @) character, and the remaining characters can be alphanumeric or national. If the prefix includes national characters, you must enclose the entire prefix value in single quotations, for example: `jobPrefix: 'AB@$'`

    When you create or edit a team in the user interface, you can specify a team prefix for job codes. If a team prefix is specified, Db2 DevOps Experience uses that value instead of the value that you specify in `configuration.yaml`. If you do not specify a prefix, Db2 DevOps Experience adds the default prefix "DOE."

    Note: To see complete reference information on the `configuration.yaml` file, see "Configuration file reference" on page 39.
11. Optional: By default, Db2 DevOps Experience writes temporary data sets under the TSO user IDs that submit jobs. You can specify a different location for these temporary data sets by specifying their high-level qualifier (HLQ). To specify the HLQ, perform the following steps:
    a) Edit the `configuration.yaml` file.
    b) Add a **`tempDatasetHLQ`** key with a value of "<*HLQ.XYZ*>" where *HLQ.XYZ* is the location.

    In the data set HLQ, each qualifier must be between 1 and 8 characters. Each qualifier must start with an alphabetic character (A-Z) or one of the special characters @, #, or $, and the remaining characters can be alphabetic, special, or numeric (0 to 9). The total length of the HLQ cannot exceed 17 characters.

    Note: To see complete reference information on the `configuration.yaml` file, see "Configuration file reference" on page 39.
12. Optional: If you configured the z/OSMF server for single sign-on (SSL) with a self-signed certificate, you can follow these steps to configure Db2 DevOps Experience to work with z/OSMF.
    a) Navigate to the <activeLiblocation>/doeserver/var/bin directory and issue the following command to generate a `trustStore` file:

    ```
    bash doe-gen-cacerts.sh
    ```

    b) Open the <activeLibLocation>/doeserver/var/bin/doe-server.sh file and uncomment the following line:

    ```
    -Djavax.net.ssl-trustStore=$DOE_HOME/var/conf/cacerts \
    ```

    Move the line above the following line:

    ```
    com.rocketsoft.newton.server.NewtonServer \
    ```
13. To start the Db2 DevOps Experience server, perform either of the following steps:

- Navigate to the <activeLibLocation>/doeserver/var/bin directory and enter the following command:

```
bash doe-server.sh start
```

- Copy the JCL procedure from the <activeLibLocation>/doeserver/var/bin/DOESRV.JCL file to your system PROCLIB library. Edit the copied content and specify the USS path as the <activeLibLocation>/doeserver/var/bin/doe-server.sh script. Then enter the following command:

```
/s DOESRV
```

  The server will write log files to the <activeLibLocation>/doeserver/var/logs directory.

14. Open the Db2 DevOps Experience application from the Zowe desktop start menu. For help getting started, see the Chapter 3, “Using,” on page 27 section of this documentation.

Upgrading Db2 DevOps Experience

To upgrade Db2 DevOps Experience, you perform SMP/E installation following instructions in the Program Directory, and then follow these steps to install the upgraded version in a new location.

Before you begin

- You can install or upgrade Db2 DevOps Experience by granting permissions to edit files and run scripts from the directory where the files were extracted during SMP/E installation. If that is how you installed the previous PTF or plan to install the current PTF, see the PTF HOLD DATA for information. This task assumes that you installed the previous version PTF and will install the current PTF by first copying required files from the extracted location to a different USS directory.
- Make sure that you find and organize the system programmers and other stakeholders with the skills and access to grant required authorities to users and complete the installation. For information, see “Installation skills and authorities” on page 14.
- When you view referenced Zowe documentation, you must view the 1.5.x version of that documentation.

About this task

This task uses the following variables to represent file locations that are specific to each user.

- <unpax_filepath> is the location where you extracted the Db2 DevOps Experience files during SMP/E installation.
- <zowe_directory> is the directory where you installed Zowe.
- <activeLibLocation> is the location where you will install and run the Db2 DevOps Experience server and UI.
- <copied_doesamplib_filepath> is the location where you will copy the doesamplib folder.

Procedure

1. In your UNIX System Services user profile, specify the following variables:

```
export _BPXK_AUTOCVT=ON
export _CEE_RUNOPTS='FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)'
export _TAG_REDIR_IN=txt
export _TAG_REDIR_OUT=txt
export _TAG_REDIR_ERR=txt
export NODE_HOME=<node_server_path>/node-v6.14.4-os390-s390x
export NODE_LIB_PATH=$NODE_HOME/lib/node_modules
export PATH=$NODE_HOME/bin:$PATH
```

2. Copy the <unpax_filepath>/doesamplib directory to a directory other than <unpax_filepath>.

3. To grant read/write access to the copied doesamplib directory, enter the following command:

```
chmod -R +w <copied_doesamplib_filepath>
```

4. Open the <copied_doesamplib_filepath>/install.properties file and specify values for the following properties:
   - SMPELocation (required) - The location of the unpaxed SMP/E files.
   - currentMaintenanceWriteLibLocation (optional) - Leave this field blank for first-time installation. The installation script copies the policy directory, configuration.yaml file, and cacerts file to this location, and searches for a /var directory in this location.
   - newMaintenanceReadLibLocation (required) - A directory for Db2 DevOps Experience to copy read-only files to.
   - newMaintenanceWriteLibLocation (required) - A directory for Db2 DevOps Experience to copy read/write files to.
   - activeLibLocation (required) - The active workspace path that the installation scripts use to create links to files copied to the directories specified in the newMaintenanceReadLibLocation and newMaintenanceWriteLibLocation properties.
   - JAVA_HOME (required) - The directory where Java is installed.
   - LEIBNIZ_HTTP_PORT (required) - The port that the doe-server.sh script uses to start the server.
   - OPENSSL_LOCATION (required) - The location where the installation script can find OpenSSL files.
   - MEMORY_SIZE (required) - The maximum amount of physical memory allocated to the Db2 DevOps Experience server. The default value of 1024 (approximately 1GB) is the minimum required heap size.
5. To run the installation script, navigate to <copied_doesamplib_filepath> directory and enter the following command:

```
bash install.sh
```

6. To configure Zowe to work with Db2 DevOps Experience, navigate to the <activeLibLocation>/installer/install/ directory and enter the following command:

```
bash doe-install.sh <zowe_directory>
```

7. To apply Db2 DevOps Experience configuration values in Zowe, navigate to the <zowe_directory>/zlux-build directory and enter the following command:

```
bash deploy.sh
```

8. Stop and restart Zowe by following steps in the Zowe documentation topic Starting and stopping the Zowe runtime on z/OS.
9. To verify that the ZSS Cross Memory Server is running in the same LPAR as Db2 DevOps Experience, look for the active process ZWESIS01.
10. To start the Db2 DevOps Experience server, perform either of the following steps:
    - Navigate to the <activeLibLocation>/doeserver/var/bin directory and enter the following command:

```
bash install.sh start
```

    - Copy the JCL procedure from the <activeLibLocation>/doeserver/var/bin/DOESRV.JCL file to your system PROCLIB library. Edit the copied content and specify the USS path as the <activeLibLocation>/doeserver/var/bin/doe-server.sh script. Then enter the following command:

```
/s DOESRV
```

      The server will write log files to the <activeLibLocation>/doeserver/var/log directory.

11. Open the Db2 DevOps Experience application from the Zowe desktop start menu. For help getting started, see the Chapter 3, “Using,” on page 27 section of this documentation.

Chapter 3. Using

If you are an administrator, you can manage Db2 DevOps Experience. Users who are not administrators can provision instances and query subsystems.

Roles and responsibilities

In Db2 DevOps Experience, there are three roles: Administrator, team administrator, and user.

Table 3. User roles and responsibilities

Role	Responsibilities
Administrator	• Register subsystems • Create environments • Establish site rules and storage limits • Assign subsystems to environments • Create and manage teams • Assign teams to environments • Assign team administrators • Create applications • Assign applications to teams
Team administrator	• Update team members • Manage instances • Review and approve pull requests • Commit approved changes
User	• Provision application instances • Deprovision their own instances • Update objects • Create pull requests • Merge updates • Revert to commits

Managing

Administrators can follow the tasks in this section to register subsystems, create teams and environments, set storage monitoring limits, and create site rules.

© Copyright IBM Corp. 2019, 2020

Registering subsystems

Before you can create applications or assign provisioning environments, you must register subsystems to make their objects available. Only Administrators can register subsystems.

About this task

Registering a subsystem makes its objects available in Db2 DevOps Experience. You can then define groups of subsystem objects as applications. And you can assign registered subsystems to environments so that users can provision application instances from those environments.

When you register a subsystem, you must provide its information. To register subsystems that Db2 DevOps Experience discovered, you must specify certain subsystem objects, such as data sets, the job card template, and schema for operational objects. To register subsystems that were not discovered, you must provide connection information and subsystem object information.

To register a subsystem, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Subsystems**.
3. Take one of the following steps:
   - To register discovered subsystems, click the **Discover** tab, select the subsystem, and click **Register**. Proceed Step 8.
   - To register undiscovered subsystems, click the **Registered** tab, then click **Register Subsystem**. Proceed to Step 4.
4. Type a subsystem name using a maximum of 4 characters.
5. Select the subsystem's Db2 version.
6. Type the subsystem's host name, port number, and location.
7. Click **Save**.
8. In the **Subsystem Setup** dialog, specify values in the following fields:

Field	Description
SDSNEXIT	Type the fully-qualified names of the subsystem's Db2 SDSNEXIT data sets, for example DB2A.SDSNEXIT, DB2B.SDSNEXIT. Separate data sets with commas.
SDSNLOAD	Type the fully-qualified names of the subsystem's Db2 SDSNLOAD data sets, for example DSN.VB10.SDSNEXIT, DSN.VB11.SDSNEXIT. Separate data sets with commas.
RUNLIB	Type the fully-qualified names of the subsystem's Db2 RUNLIB.LOAD data sets, for example DSN.VB10.RUNLIB.LOAD, DSN.VB11.RUNLIB.LOAD. Separate data sets with commas.
Job card text	Type the job card template for jobs submitted on this subsystem.
Schema	Type the name of the schema to create when the subsystem is registered. This schema is for Db2 DevOps Experience operational objects.

Field	Description
Naming Rule	Type the four-character prefix to add to Db2 objects and collection names when the subsystem is registered.
Plan name	Type the name of the plan to bind to this subsystem when it is registered.
Replace Installation Artifacts	Select to replace artifacts in this subsystem, such as objects and plans, with new artifacts defined in this dialog.

9. Click **Install components**. After a few moments, the subsystem displays in the **Registered** tab.

   Note: On z/OS, the jobname of the subsystem registration job is `INIT<ssid>`.

Creating teams and environments

Create teams to develop applications.

About this task

When you create a team, you define team administrators and environments into which the team can provision application instances. When you define environments, you assign subsystems to host application objects that are provisioned to the environment. And you set limits on the number of instances that the team, and individual team members, can provision. You assign users to teams in the **Users** section of Db2 DevOps Experience.

Team administrators can add or remove team members, manage team applications, review pull requests, and commit changes. If you add only one team administrator to a team, that administrator can approve their own pull requests. If you add more than one team administrator, they cannot approve their own pull requests; another team administrator must approve them. It is recommended that you create at least two team administrators to back each other up, and review each others' pull requests if necessary.

By default, Db2 DevOps Experience enforces subsystems' existing database and schema rules in environments. In each environment, you can create rules for changing the names of provisioned schemas and databases. Provisioned objects must have unique names if they will be stored in the same subsystems as their original versions, because two objects in the same subsystem cannot have the same name.

If you use Top Secret for security, note the following issues:

- Users who create teams are automatically made team administrators. If they should not be a team administrator, you must edit the team and remove them.
- Team administrators who are members of the MVDUSER group (typically developers) cannot add or remove people from the team. At least some team administrators must be members of the MVDSPEC group so that they add and remove team members.

To create a team and its environments, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Teams & Environments**.
3. Click **Add team**.
4. Type the team name.
5. Optional: Type a job prefix to identify Db2 DevOps Experience jobs in z/OS interfaces such as System Display and Search Facility (SDSF).

6. Select team administrators. If you use Top Secret, make sure at least one team administrator is a member of the MVDSPEC group.
7. In the environment section, type an environment name.
8. In the **Instance limit** field, specify the maximum number of application instances that the team can provision at one time.
9. In the **User instance** field, specify the maximum number of application instances that each member can provision at one time.
10. To assign subsystems to the environment, click **Assign subsystems**. Then search for and select registered subsystems and click **Create assignment**.
11. Optional: To create a schema or database rule, click **Create rule**. Then specify the conditions and click **Create rule**.
12. Optional: To create another environment, click **Add environment**.
13. Optional: If you use Top Secret, you are automatically added as a team administrator. You can edit the team to remove yourself.

Assigning users to teams

After you create teams, you can add groups and users with TSO IDs to those teams.

About this task

After you assign a user to a team, the user can provision application instances to environments that were assigned to the team.

To assign users to teams, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Users**.
3. Find and select users, and then click **Edit teams**.
4. Find and select teams and click **Edit team(s)**. The selected users are added to the selected teams.

Setting soft storage limits

When users provision applications, those applications occupy storage. To manage this storage, you can set limits by team, environment, user, and application. The limits are soft, meaning when they are exceeded Db2 DevOps Experience displays alerts but does not prevent continued activity.

About this task

To set limits, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Storage**.
3. Click the tab that represents how you want to limit storage. For example, to limit storage by environment, click **By Environment**.
4. Find and select the objects that you want to limit, and then click **Create limit**.
5. Specify the storage limit for that object. For example, for an environment specify 15 GB to specify that provisioned instances of applications in that environment that occupy more than 15 GB in the Db2 database will generate warnings.

Defining applications

Define applications from which users can provision application instances.

About this task

An application is a set of objects that are grouped together so that they can be managed and provisioned as a single unit. To register an application, first compile a list of the objects to include in the application.

Application objects are logical, meaning that they are only references to the objects. When users provision instances of an application, the referenced objects are copied to create the instance.

To define an application, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.
3. Click **Start new discovery plan**.
4. Click **Select source subsystem** and select the subsystem that contains the application objects. Then click **Next**.
5. To find objects, click **Search object by type** and repeat the following steps until you have added all objects that you want in the application:
   a) Select the type of object you are searching for, for example **Database**.
   b) Enter search terms in the fields, and then click **Search**.

      Note: You can substitute wildcard characters for characters in your search terms. Use "%" (percent) to represent zero or more characters. Use "_" (underscore) to represent a single character. The search is case-sensitive.
   c) To include objects in the application, expand results and select objects. Selected objects display in the **Selected objects** panel.
6. By default, objects that are dependent on the selected objects are included in the application. To exclude dependent objects, disable **Include all objects**.
7. Select any required applications from the **Required application(s)** menu.
8. Click **Continue**.
9. Type a name for your new application in the **Name** field.
10. Click the **Color** swatch to select a color that will identify this application when it appears on your **Applications** screen.
11. Briefly describe the application in the **Description** field.
12. Select a team from the **Owning Team** menu. The owning team is responsible for the application, and owning team administrators approve pull requests for the application.
13. Click the **Commit application** button to save and commit the application.

Creating site rules

Administrators can create site rules to guide how developers should change object definitions in provisioned application instances. Administrators can create multiple site rules. Site rules can be associated with applications and environments.

About this task

You can create simple and complex site rules. Simple rules are built using three segments: object, attribute, and verification type. For example, you can create a site rule that specifies that table (object) names (attribute) must start with (verification type) the string "TAB". When applied to an application or environment, developers creating a table in a provisioned application instance are notified if they violate the rule.

Complex site rules are more flexible and powerful. You define them using a small domain-specific programming language that is composed of a subset of the Python syntax. Every valid complex site rule is a Python expression that is evaluated and resolved to either True or False according to normal Python rules. For example, an empty string is False, or the number 7 is True. Expressions can use the following features:

- Python ternary expressions (x if y else z)
- Boolean and/or/not
- Parentheses
- <
- >
- <=
- >=
- ==
- !=
- +
- -
- String subscription (stride not supported)
- The functions: `len`, `startswith`, and `endswith`

Supported data types are integers, strings (delimited by double or single quotes), and booleans.

Complex site rules have one non-standard Python feature: a pre-defined variable for every object attribute. For example, to specify that “table names must start with the letter 'T'" you would select **Table** create the following site rule:

```
name.startswith(“T”)
```

`name` is a predefined string variable that holds the value of the table name. The rest is standard Python syntax. Here is another valid way to write that rule:

```
name[:1] == “T”
```

For a list of objects and their attribute variables, see the reference topic “Complex site rule variables” on page 42.

Rules can be combined to make standard Python expressions. For example, here is one way to create the rule “if a tablespace name is between 4 and 6 characters, then tables in that tablespace must end their names with their tablespace’s last four characters”:

```
name.endswith(TSNAME[-4:]) if 4 <= len(TSNAME) <= 6 else True
```

Note that the Python ternary expression is different from ternary expressions in C/C++/Java, and is different from a Python if/else block.

Important: Values must be uppercase. The rule `log=="LOGGED"` works; the rule `log=="logged"` does not work.

The objects and attributes supported by complex site rules are the same as those supported by simple site rules.

To create and apply a site rule, complete the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Site Rules**.
3. Click **Add new rule**.

4. Perform one of the following tasks:
   - To create a simple rule, follow these steps:
     a. Select the type of object to specify the rule on. For example, select **Database** to specify the rule on database objects.
     b. Select conditions, and then click **Next**. Conditions are different for each object. For example, for database objects you can select **NAME** to specify that the rule applies to the object name, and then **starts with**, and then type characters to specify what characters the database name must start with.
     c. Type a name for the rule, and then click **Create**.
   - To create a complex rule, follow these steps:
     a. Click the **Complex rule** switch.
     b. Select the object type.
     c. Create the rule, and then click **Next**. Make sure that values are uppercase.
     d. Type a name for the rule, and then click **Create**.
5. To apply a rule, select it and click **Assign to Applications** or **Assign to Environments**.
6. Find the applications or environments to apply the rule to, and then click **Assign**.
7. To edit, delete, or duplicate a rule to assign to different applications or environments, click the menu icon in the rule row.

Provisioning instances

You can provision an instance of an application to develop and test code that works with the database. A provisioned application is a copy of the objects in that application. You can safely work on the instance objects without affecting the original objects.

About this task

Before you can provision an instance of an application, an administrator must register subsystems that contain the application objects, create a team and environment, assign you to the team, and create the application.

To provision an instance, complete the following steps:

Procedure

1. Click the navigation menu and select **Applications**.
2. Open the application.
3. Click **Provision new instance**.
4. Type a name for the instance.
5. Select an application from which to provision the instance.
6. Optional: Select **Include data from source application**.
7. Select the team under which to provision the instance. If you are a member of multiple teams, you must specify which team you are provisioning the instance under. This can be relevant, since storage limits can be placed on teams.
8. Specify the environment to which you will provision the application.
9. Click **Provision**.

Editing objects

You can edit the objects in an application instance.

About this task

Object changes must be reviewed by a Team Administrator of the team under which you provisioned the instance.

To edit application instance objects, complete the following steps:

Procedure

1. Click the navigation menu and select **Instances**.
2. Find the instance that you want to edit and click **Details**.
3. In the **DDL** pane, click an object and edit it.
4. When you are finished editing the object, take one of the following steps:
   - To apply changes to the current object, click **Apply object changes**.
   - Click other objects and edit them. To apply all of your changes, click **Apply all changes**.

Querying subsystems

You can create and send SQL queries directly to subsystems.

About this task

To query a subsystem, complete the following steps:

Procedure

1. Click the navigation menu and select **SQL Processor**.
2. Select the subsystem connection to query.
3. Type or paste your query into the editor.
4. Click **Send query**.

Chapter 4. Troubleshooting

Use this section to diagnose and correct problems that you experience with Db2 DevOps Experience.

Connection timeout

You can change the length of time that Db2 DevOps Experience will try to connect to z/OSMF before it times out.

Symptoms
The connection between Db2 DevOps Experience and z/OSMF keeps timing out.

Causes
By default, if Db2 DevOps Experience tries to connect to z/OSMF for 180 seconds without a response, the connection times out.

Resolving the problem

Operator response: To change the timeout parameters

1. Open `<unpax_filepath>/doeserver/var/bin/doe-server.sh`.
2. Locate `-Dcom.rocketsoft.nm.discovery.defaultHttpRequestTimeout=180`.
3. Change the value from `180` to the desired length of time, in seconds.

zss server log error when starting Zowe

When you stop and restart Zowe while installing Db2 DevOps Experience, you might get an error related to permissions.

Symptoms
You get the following zss log error while starting Zowe:

```
zssServer startup, version 1.5.0+20190917
Server config file=../deploy/instance/ZLUX/serverConfig/zluxserver.json
Config path=../deploy/instance/ZLUX/serverConfig has group & other
permissions that are too open! Refusing to start.
Ensure group has no write (or execute, if file) permission. Ensure
other has no permissions. Then, restart zssServer to retry.
```

Resolving the problem
Run the following commands, and then restart Zowe:

```
cd <zowe_directory>/zlux-app-server/deploy/instance/ZLUX
```

```
chmod -R 740 serverConfig
```

© Copyright IBM Corp. 2019, 2020

Chapter 5. Reference

This section contains material referenced from elsewhere in this guide.

Sample RACF security setup job card

This script configures security for Db2 DevOps Experience. Where necessary, replace the provided values with values that reflect your environment.

```
//JOBCARD
//*
//* Dynamic APF authorization of hlq.SDOELINK
//* Add the datasets to your PROGxx system parmlib member for
//* APF authorization to persist across IPLs.
//*
//* Before you change the format of the APF list to dynamic, contact the system
//* programmer to validate that all programs and vendor products are converted to
//* use dynamic APF services and that the proper program products are installed.
//*
//* For non-SMS managed datasets use the following syntax instead:
//* SETPROG APF,ADD,DSNAME=hlq.SDOELINK,VOL=volser
//*
  SETPROG APF,FORMAT=DYN
  SETPROG APF,ADD,
    DSNAME=hlq.SDOELINK,SMS
//*
//*
//*
//STEP1  EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *

/*                                              */
/* User IDs for started tasks                   */
/*                                              */
/* #stcgroup# : group for started tasks         */
/* #doestc#   : STC user ID for the DOE server  */
/* #dbauser#  : Functional user ID used by DOE  */
/*              for certain provisioning and    */
/*              merge operations. Need SYSADM auth. */
/*              You need the user ID and password */
/*              when completing the Setup wizard */
/* #grpowner# : Owner of groups for STCs        */
/*                                              */
/* Note: AUTOGID requires SHARED.IDS to be      */
/*       implemented                            */
/* Note: HOME and OWNER are commented out for   */
/*       the user ID creation                   */

ADDGROUP #stcgroup#                          -
   /* OWNER(#grpowner) SUPGROUP(xxx) */     -
   OMVS(AUTOGID)

ADDUSER (#doestc#)                           -
   OMVS(AUTOUID                              -
        /* HOME(xxx) */                      -
        PROGRAM(/bin/sh) )                   -
   NOPASSWORD                                -
   /* OWNER(xxx) */                         -
   DFLTGRP(#stcgroup#)

ADDUSER (#dbauser#)                          -
   OMVS(AUTOUID                              -
        /* HOME(xxx) */                      -
        PROGRAM(/bin/sh)                     -
    PASSWORD(password) )                -
   TSO( ACCTNUM(xxx) COMMAND(ISPF)           -
        PROC(xxx) SIZE(1024000) SYS(x) )     -
   /* OWNER(xxx) */                         -
   DFLTGRP(#stcgroup#)
PASSWORD USER(#dbauser#) NOINTERVAL

/*                                              */
```

© Copyright IBM Corp. 2019, 2020

```
/* Activate required classes                    */
/*                                              */

SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)

SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS GENERIC(STARTED)

/*                                              */
/* DevOps server started task                   */
/* There are currenly no JCL procedure sample   */
/* supplied for DevOps server started tasks     */
/* The RACF commands below assume you create    */
/* a JCL proc DOESRV                            */
/*                                              */

RDEFINE STARTED DOESRV.* UACC(NONE) -
   STDATA(USER(#doestc#) GROUP(#stcgroup#) TRUSTED(NO))

/*                                              */
/* Refresh RACF FACILITY and STARTED class */
/*                                              */

SETROPTS RACLIST(FACILITY) REFRESH
SETROPTS RACLIST(STARTED) REFRESH

/*                                              */
/* Define the RACF class that will contain all  */
/* administration profiles.                     */
/* The default class name is ZOWE               */
/*                                              */

SETROPTS RACLIST(CDT)

RDEFINE CDT ZOWE UACC(NONE)    -
  CDTINFO(                      -
   DEFAULTUACC(NONE)            -
   FIRST(ALPHA)                 -
   OTHER(ALPHA,NATIONAL,NUMERIC,SPECIAL) -
   MAXLENGTH(246)               -
   POSIT(607)                   -
   RACLIST(DISALLOWED))

SETROPTS RACLIST(CDT) REFRESH
SETROPTS CLASSACT(ZOWE)

/*                                              */
/* Create groups for Db2 DevOps Experience      */
/* Replace #doeusergrp# and #doeadmingrp# with  */
/* group names of your choice                   */
/*                                              */

ADDGROUP #doeusergrp# OWNER(#OWNER#)
ADDGROUP #doeadmingrp# OWNER(#OWNER#)


/*                                              */
/* Authorizations needed for DevOps Experience  */
/* administrators                               */
/* - connected to DevOps Experience user group  */
/* - connected to DevOps Experience admin group */
/* - connected to z/OSMF admin group            */
/* - class authorization for ZOWE               */
/* - permit read to OMVSAPPL in class APPL if   */
/*   defined                                    */

CONNECT adminuser GROUP(#doeadmingrp#) SPECIAL
CONNECT adminuser GROUP(#doeusergrp#)
CONNECT adminuser GROUP(IZUADMIN)
ALU (adminuser) CLAUTH(ZOWE)
/* PERMIT OMVSAPPL CLASS(APPL) -
   ID(adminuser) ACCESS(READ) */

/*                                              */
/* Authorizations needed for DevOps Experience  */
/* team administrators and developers           */
/* - connected to DevOps Experience user group  */
/* - connected to z/OSMF user group             */
/* - permit read to OMVSAPPL in class APPL if   */
/*   defined                                    */
/*                                              */
```

```
CONNECT developer GROUP(#doeusergrp#)
CONNECT developer GROUP(IZUUSER)
/* PERMIT OMVSAPPL CLASS(APPL) -
   ID(developer) ACCESS(READ) */

/*                                              */
/* Authorizations needed for DevOps Experience  */
/* functional user id                           */
/* - must hold SYSADM or equivalent Db2 auth    */
/* - connected to z/OSMF user group             */
/* - permit read to OMVSAPPL in class APPL if   */
/*   defined                                    */
/*                                              */

CONNECT #dbauser# GROUP(IZUUSER)
/* PERMIT OMVSAPPL CLASS(APPL) -
   ID(#dbauser#) ACCESS(READ) */
```

Configuration file reference

Use these configuration files to administer Db2 DevOps Experience.

configuration.yaml

The `configuration.yaml` file is located in the `<unpax_filepath>/doeserver/var/conf` directory. The values shown below are either default values or variables. You must specify values that reflect your environment:

```
appDir: ./applications
instancesDir: ./instances
gitDir: /rsusr/ported/bin
gitEnv:
- _BPXK_AUTOCVT=ON
- _CEE_RUNOPTS=FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)
- _TAG_REDIR_IN=txt
- _TAG_REDIR_OUT=txt
- _TAG_REDIR_ERR=txt
- LIBPATH=/rsusr/ported/lib/perl5/5.24.0/os390/CORE
- PERL5LIB=/rsusr/ported/lib/perl5
- MANPATH=/rsusr/ported/man
- GIT_SHELL=/rsusr/ported/bin/bash
- GIT_EXEC_PATH=/rsusr/ported/libexec/git-core
- GIT_TEMPLATE_DIR=/rsusr/ported/share/git-core/templates
- GIT_MAN_PATH=/rsusr/ported/
ISPFMessageLibrary: ISP.SISPMENU
ISPFSkeletonLibrary: ISP.SISPSENU
ISPFTableLibrary: ISP.SISPTENU
ISPFLoadLibrary: ISP.SISPLOAD
ISPFLPALibrary: ISP.SISPLPA
cmbatchDsn: <hlq>
failsafeTimeout: 100
adminGroup: MVDSPEC
userGroup: MVDUSER
teamPrefix: <team_prefix>
hlq: <libs_hlq>
zssHost: <zss_host>
zssPort: <zss_port>
zosmfHost: <zosmf_host>
zosmfPort: <zosmf_port>
dbaUsername: <admin_un>
dbaPassword: <admin_id>
jobPrefix: <jobcode_value>
tempDatasetHLQ: <hql>
dbaSqlid: <sqlid>
```

Table 4. configuration.yaml keys

key	Description
appDir	Directory that contains user-defined applications.
instancesDir	Directory that contains user-provisioned application instances.

Table 4. configuration.yaml keys (continued)	
key	**Description**
gitDir	Directory of the git instance used with Db2 DevOps Experience.
gitEnv	Parameters that define the git environment.
ISPFMessageLibrary	High-level qualifier (HLQ) of the ISPF message library that Db2 DevOps Experience uses to register subsystems.
ISPFSkeletonLibrary	High-level qualifier (HLQ) of the ISPF skeleton library that Db2 DevOps Experience uses to register subsystems.
ISPFTableLibrary	High-level qualifier (HLQ) of the ISPF table library that Db2 DevOps Experience uses to register subsystems.
ISPFLoadLibrary	High-level qualifier (HLQ) of the ISPF load library that Db2 DevOps Experience uses to register subsystems.
ISPFLPALibrary	High-level qualifier (HLQ) of the ISPF LPA library that Db2 DevOps Experience uses to register subsystems.
cmbatchDsn	High-level qualifier (HLQ) of the data set that contains user-registered Db2 subsystem information. This is the global value. Administrators can specify a value in the **cmbatchDsn** key in the `subsystem.yaml` file for individual registered subsystems to override this value. Specified in the Db2 DevOps Experience set up wizard.
failsafeTimeout	Length of time, in minutes, to wait for submitted jobs to complete before ending them as failed. This is the backup setting. The default time is 60 minutes, but users can override that by using a TIME parameter in their JCL. If the default or JCL-specified time passes without a response from z/OS, the **failsafeTimeout** time is used.
adminGroup	SAF group that contains Db2 DevOps Experience administrators. This value is specified in the set up wizard.
userGroup	SAF group that contains Db2 DevOps Experience users. This value is specified in the set up wizard.
teamPrefix	Prefix for the SAF CDT facilities created when a Db2 DevOps Experience user creates a team. For example, a value of ZOWE.DOE means that the facilities will look like: ZOWE.DOE.*<team_uuid>*.
hlq	High-level qualifier for the Db2 DevOps Experience libraries, for example the SMP/E target libraries. This value is specified in the set up wizard.
zssHost	IP address of the Zowe ZSS host server.

Table 4. configuration.yaml keys (continued)	
key	**Description**
zssPort	Port number of the Zowe ZSS host server.
zosmfHost	IP address of the z/OSMF host server.
zosmfPort	Port number of the z/OSMF host server.
dbaUsername	TSO name of a Db2 administrator that Db2 DevOps Experience can use to perform tasks that require a high level of authority. This value is specified in the set up wizard.
dbaPassword	Password of the Db2 administrator specified in the **dbaUsername** parameter. This value is specified in the set up wizard.
jobPrefix	Job code for Db2 DevOps Experience jobs so that they are easier to find in z/OS interfaces such as SDSF. Db2 DevOps Experience adds the default prefix "DOE". Codes are 8 characters long in total, so z/OS will generate random characters not specified in the prefix. For example, if you specify the 6-character prefix value "IBMDOE", z/OS will generate the remaining two characters to complete the job code. Standard z/OS job code naming rules apply, so the prefix must start with an alphabetical or a national ($, #, @) character, and the remaining characters can be alphanumeric or national. If the prefix includes national characters, you must enclose the entire prefix value in single quotations, for example: `jobPrefix: 'AB@$'`. Prefixes specified for teams supersede values specified here.
tempDatasetHLQ	By default, Db2 DevOps Experience writes temporary data sets under the TSO user IDs that submit jobs. You can specify a different location for these temporary data sets by specifying their high-level qualifier (HLQ), for example "<*HLQ.XYZ*>". In the data set HLQ, each qualifier must be between 1 and 8 characters. Each qualifier must start with an alphabetic character (A-Z) or one of the special characters @, #, or $ and the remaining characters can be alphabetic, special, or numeric (0 to 9). The total length of the HLQ cannot exceed 17 characters.

Table 4. configuration.yaml keys (continued)	
key	**Description**
dbaSqlid	Optional. Specifies whether SET CURRENT SQLID statements are generated and, if so, what SQLID value to use. Values: **A SQLID** The specified Run SQLID is the owner of databases and table spaces. If the specified Run SQLID is different from the current owner, the databases, table spaces, and all dependent objects are dropped and re-created to accomplish the change of owner. **<NONE>** No SET CURRENT SQLID statements are generated. **blank** SET CURRENT SQLID statements are generated when necessary. Default: blank

Complex site rule variables

When you create complex site rules, you can use the following variables in the Python expressions to represent object attributes. For example, when writing an expression that applies to database objects you use the variable `encoding_scheme` to represent the encoding scheme of that database. The table below describes the objects and attribute variables. You can also do a GET call to the following API address `<server address>:<port>/ws/ddl/site-rules/allowed-rules/` and it will return the same list.

Table 5. Object attribute variables	
Object	**Attributes**
alias	name, schema
column	name, coltype
database	encoding_scheme, bpool, stgroup, name, indexbp
index	name, creator, type, tbname, storname, vcatname, bpool, partitioned, padded, gbpcache, pctfree, freepage, closerrule, copy, compress, clustering, priqty, secqty, erase, uniquerule, defer, define
procedure	name, creator, language, asutime
sequence	name, schema, datatype, start, increment, cycle, order, cache, maxvalue, minvalue
stogroup	name, volumes, vcatname, dataclass, mgmtclas, storclas
synonym	name, creator

Table 5. Object attribute variables (continued)	
Object	**Attributes**
table	auditing, encoding_scheme, column, tsname, valproc, append, datacapture, name, edproc, clustertype, dbname, creator, partitioning, primary_key, foreign_key, volatile, pagenum, trackmod, member_cluster, dssize, log, organizationtype
tablespace	encoding_scheme, priqty, secqty, erase, gbpcache, maxrows, maxpartitions, freepage, pctfree, pctfree_update, dbname, log, bpool, storname, closerule, name, trackmod, dssize, segsize, locksize, lockmax, compress, define
trigger	name, schema, granularity, sqlpl, instead_of, secure
userDefinedFunction	name, schema, external
view	name, creator

Installing Swagger UI on z/OS

Before you begin

Install Node.js and run `npm install http-server –g`.

Procedure

1. Download https://github.com/swagger-api/swagger-ui/archive/master.zip.
2. Transfer the zip file to z/OS UNIX System Services and unzip.
3. Copy `<unpax_filepath>/doeserver/opt/doc/swagger.json` to `swagger-ui/dist/doe.json`.
4. For the Zowe API documentation, copy `<zowe_directory>/zlux-app-server/doc/swagger/*.yaml` to `swagger-ui/dist`.
5. In `swagger-ui/dist/index.html` change `url: "https://petstore.swagger.io/v2/swagger.json"`, to `url: “doe.json",`.
6. Start Swagger UI.

Example

Sample JCL procedure

```
//DOESWAG  PROC
//*
//DOESWAG EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//  PARM='PGM /bin/sh /prefix/swagger-ui/dist/swagger.sh'
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
```

Sample shell script:

```
export _BPXK_AUTOCVT=ON
export _CEE_RUNOPTS='FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)'
export _TAG_REDIR_IN=txt
export _TAG_REDIR_OUT=txt
export _TAG_REDIR_ERR=txt

export NODE_HOME=set to your node home
```

```
export PATH=$NODE_HOME/bin:$PATH

cd /prefix/swagger-ui/dist
echo "Starting Swagger Server"
http-server -p 12081 -c-1
```

Note: Modify –p if you want a different port number. `-c-1` disables caching. For more information about `http-server`, see https://www.npmjs.com/package/http-server.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive

© Copyright IBM Corp. 2019, 2020

Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 DevOps Experience.

This publication documents information that is NOT intended to be used as Programming Interfaces of Db2 DevOps Experience.

This publication primarily documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 DevOps Experience.

This publication also documents information that is NOT intended to be used as Programming Interfaces of Db2 DevOps Experience. This information is identified where it occurs by an introductory statement to a topic or section.

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of Db2 DevOps Experience.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 DevOps Experience. This information is identified where it occurs by an introductory statement to a topic or section.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at https://www.ibm.com/legal/copytrade.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the Unites States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to

collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

A

access
  catalog and bufferpool 16
administration
  configuration files 39
APF
  authorization 19
appendix 37
applications
  defining 31
  discovering 31
architecture
  configuration files 39
  diagram 9

C

components
  DevOps server 9
  REST APIs 9
  web browser 9
  z/OSMF 9
  Zowe 9
configuring
  RACF 17
  Top Secret 18
cookie policy 45, 47

D

Db2 Devops Experience
  benefits 8
  features 8
  new and changed functions 1
Db2 DevOps Experience
  overview 1
documentation
  accessing 11
  sending feedback 11

E

editing
  objects 34
environments
  creating 29

H

hardware requirements 10

I

installation
  Db2 DevOps Experience 13
installation *(continued)*
  grant access 16
  overview 13
  prerequisites 10
  procedure 20
  required skills and authorities 14
  SMPE 13
  ZSS Cross Memory Server 13
installing
  Swagger UI on z/OS 43
  Zowe 20
instances
  provisioning 33

L

legal notices
  cookie policy 45, 47
  notices 45
  programming interface information 45, 46
  trademarks 45, 47

M

managing
  using Db2 DevOps Experience 27

N

notices 45, 46

O

objects
  editing 34
operating systems
  supported 10

P

prerequisites
  installation 10, 13
programming interface information 45, 46

R

RACF
  privileges 17
reader comment form 11
requirements
  hardware and software 10

S

SAF

SAF *(continued)*
groups 17
security
APF authorization 19
overview 16
service information 11
site rules
creating 31
object attribute variables 42
software requirements 10
storage
limiting 30
subsystems
querying 34
registering 28
support information 11
Swagger
installing on z/OS 43

T

teams
assigning users 30
creating 29
technotes 11
timeout
z/OSMF connection 35
Top Secret
privileges 18
trademarks 45, 47
troubleshooting 35

U

upgrade
procedure 23
users
assigning to teams 30
Using
roles and responsibilities 27
using Db2 DevOps Experience
managing 27

Product Number: 5698-DEX

SC27-9138-00