DIBBs Query Connector

Query Connector Maintenance Guide

Created Date: Feb 10, 2026

Last Updated: June 4, 2026

Prepared by:

U.S. Department of Health and Human Services

U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES
CENTERS FOR DISEASE CONTROL AND PREVENTION

This document has been developed by Skylight under a contract with the U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

Version History

Version	Author	Reviewed By	Revision Date	Revision Notes
1.0	Matthew Goldberg	Nick Clyde	03/30/2026	
1.1	Matthew Goldberg	Maya Mascarenhas	06/04/2026	

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

Table of Contents

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

To contact the Query Connector team, email dibbs@cdc.gov

1 Maintenance Guide overview

This guide helps partner jurisdictions keep Query Connector deployed, reachable, and functioning as expected. It explains how to update to new releases, maintain forked repos, and troubleshoot potential issues.

2 Updating to new release versions

This guide assumes customers run Query Connector in one of two ways:

- Docker container
- Virtual Machine (VM)

Important: Update steps differ depending on which deployment type is in use.

To address issues or improve stability, the DIBBs team periodically publishes new releases of Query Connector. Updating typically requires redeploying Query Connector with an updated artifact (Docker image or VM image).

Before updating, make sure to review the release notes for any breaking changes and test the upgrade in a non-production environment first. Always back up your data before upgrading.

2.1 Docker container updates

2.1.1 Required steps

Step 1: Check for new versions. Visit the Query Connector GitHub Container Registry at `ghcr.io/cdcgov/dibbs-query-connector/query-connector` to see published versions. You can also monitor the Releases page on the GitHub repository for release notes and changelogs.

Step 2: Update your deployment. Follow the instructions for your deployment type below.

Option A: Docker running directly on a VM (e.g., EC2 instance, Azure VM)

1. SSH into your VM and pull the latest image:

```
docker pull ghcr.io/cdcgov/dibbs-query-connector/query-connector:latest
```


This document has been developed by Skylight under a contract with the U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

Or pull a specific version:

```
docker pull ghcr.io/cdcgov/dibbs-query-connector/query-connector:<version>
```

2. Stop and remove the existing container:

```
docker stop query-connector && docker rm query-connector
```

3. Redeploy with the new image using your existing .env file:

```
docker run -d --name query-connector -p 3000:3000 --env-file .env ghcr.io/cdcgov/dibbs-query-connector/query-connector:latest
```

4. Verify the deployment by navigating to your Query Connector URL and reviewing logs:

```
docker logs query-connector
```

Rollback: If issues arise, stop the new container and redeploy using the previous image tag. Keep a record of previously used tags for quick rollback.

Option B: AWS Elastic Container Service (ECS)

1. If your ECS task definition references the latest tag, you can force a new deployment to pull the updated image. In the AWS Console, navigate to your ECS cluster and service, then click "Update service" and check "Force new deployment." Alternatively, using the CLI:

```
aws ecs update-service --cluster <your-cluster> --service <your-service> --force-new-deployment
```

2. If your task definition references a specific version tag, update the task definition with the new image tag:
    a. Create a new revision of your task definition with the updated image URI (e.g., `ghcr.io/cdcgov/dibbs-query-connector/query-connector:1.1.0`).
    b. Update your ECS service to use the new task definition revision.

```
aws ecs update-service --cluster <your-cluster> --service <your-service> --task-definition <new-task-def-revision>
```

3. Monitor the deployment in the ECS console. ECS will perform a rolling update, draining connections from old tasks before stopping them. Watch the "Events" tab and "Tasks" tab to confirm new tasks reach a RUNNING state.

This document has been developed by Skylight under a contract with the U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

Rollback: Update the service to point back to the previous task definition revision. ECS will perform another rolling update to revert. If you have circuit breaker enabled on the service, ECS will automatically roll back if the new tasks fail health checks.

Option C: Azure Container Apps

1. Update the container image for your Container App. In the Azure Portal, navigate to your Container App, go to "Containers" under the Application section, and click "Edit and deploy." Update the image tag to the new version, then click "Create" to deploy. Alternatively, using the Azure CLI:

```
az containerapp update --name <your-app> --resource-group <your-rg> --image ghcr.io/cdcgov/dibbs-query-connector/query-connector:latest
```

2. If using a specific version tag:

```
az containerapp update --name <your-app> --resource-group <your-rg> --image ghcr.io/cdcgov/dibbs-query-connector/query-connector:<version>
```

3. Monitor the deployment under "Revision management" in the Azure Portal. Azure Container Apps will create a new revision and route traffic to it once it passes health checks.

Rollback: Azure Container Apps retains previous revisions. Navigate to "Revision management," activate the previous revision, and route traffic back to it. You can also use multi-revision mode to do traffic splitting for canary-style rollouts.

Step 3: Verify after update. Confirm the application loads correctly, check that existing queries still work, and review logs for any migration errors or startup issues.

2.2 VM updates

2.1.2 Required steps

1. **Check for new VM images:** Visit the DIBBs VM repository at https://github.com/CDCgov/dibbs-vm for updated VM images.
2. **Download the updated VM image:** Download the latest OVA or VHDX file from the repository.
3. **Back up your current environment:** Before replacing the VM, take a snapshot of your current virtual machine and ensure your database is backed up.

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

4. **Import the new VM image:** Import the updated image into your hypervisor (VMware, Hyper-V, VirtualBox, or cloud provider). Configure VM resources: 4+ vCPUs, 8+ GB RAM, and 50+ GB storage for production; 2+ vCPUs, 4+ GB RAM, and 20+ GB storage for testing.
5. **Run the setup wizard:** Execute the dibbs-query-connector-wizard.sh script. This will walk you through configuring the application name, node environment, database connection string, authentication settings, and API keys (eRSD and UMLS).
6. **Verify deployment:** After the wizard completes, navigate to Portainer (included in the VM) to view the container dashboard. Confirm the Query Connector container is running and check logs for errors.

Rollback: If the update fails, restore from the VM snapshot you created in step 3.

3 Maintaining Query Connector from a forked repository

Query Connector is designed to be maintainable by partner jurisdictions. If you need to make changes—such as updating dependencies, applying security fixes, or customizing functionality—you can do so by forking the repository and maintaining your own build and deployment process.

3.1 Required steps

1. **Fork the repository:** Create a fork of the Query Connector repository at https://github.com/CDCgov/dibbs-query-connector using the Fork button on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:<your-org>/dibbs-query-connector.git
```

3. **Keep your fork in sync:** Periodically pull updates from the upstream CDCgov repository to stay current with bug fixes and new features:

```
git remote add upstream https://github.com/CDCgov/dibbs-query-connector.git
git fetch upstream && git merge upstream/main
```

4. **Make and test your changes:** Install dependencies with npm install, run locally with npm run dev, and verify changes work as expected. Run the test suite before deploying.
5. **Build and deploy your custom image:** Build a Docker image from your fork and push it to your own container registry:

```
docker build -t your-registry/query-connector:custom
```

Monitor upstream releases: Watch the upstream repository for new releases and merge relevant changes into your fork as needed. Pay close attention to database migration files in the flyway directory.

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

4 Database troubleshooting

Query Connector requires a PostgreSQL database (version 13 or later) to store information on conditions, value sets, Fast Healthcare Interoperability Resources® (FHIR®)[1] server configuration, user management, and audit logs. The app automatically manages database migrations using Flyway and applies them at startup.

4.1 Common issues and steps

1. **Connection errors:** Verify that the DATABASE_URL environment variable is correctly set and points to your PostgreSQL instance. Ensure the database host is reachable from the Query Connector container or VM. Check that the database user has the necessary permissions.
2. **Migration failures:** If the application fails to start due to a Flyway migration error, check the Flyway logs for details. Ensure the flyway.conf file points to your production database. In some cases, you may need to manually resolve migration conflicts by inspecting the flyway_schema_history table.
3. **Empty value sets:** If the database has been set up but contains no value set data, navigate to /queryBuilding as an admin user. Click "Create your first query" to initiate the value set ingestion process from the eRSD and VSAC APIs. This requires valid ERSD_API_KEY and UMLS_API_KEY environment variables.
4. **Performance issues:** For production environments, use a managed database service (such as AWS RDS, Azure Database for PostgreSQL, or GCP Cloud SQL). Allocate at least 10 GB of storage. Configure automated backups and monitor database performance.
5. **Backup and recovery:** Set up regular backups of your PostgreSQL database. For managed services, enable point-in-time recovery. Test restoration procedures periodically. The application itself is stateless, so focus backup efforts on the database.

5 Setting up FHIR connections

Query Connector connects to one or more FHIR servers to retrieve patient data. These servers are typically provided by healthcare organizations or a Qualified Health Information Network (QHIN) within TEFCA.

To configure a FHIR server connection:

1. **Register Query Connector as a client:** Work with your FHIR server administrator to register Query Connector as an authorized client. Obtain client credentials (client ID and secret) or appropriate authentication tokens. Ensure your Query Connector's IP address or domain is allowlisted.

[1] HL7®, and FHIR® are the registered trademarks of Health Level Seven International and their use of these trademarks does not constitute an endorsement by HL7.

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

2. **Add the server in the admin interface:** Use the FHIR server configuration page in the Query Connector admin UI. Provide the server name, base URL (e.g., https://fhir.example.org/R4), and authentication method. Supported authentication methods include bearer token, OAuth client credentials, and SMART on FHIR with asymmetric key exchange.
3. **Test the connection:** After configuration, use the built-in test feature in the FHIR server configuration UI to verify that Query Connector can successfully authenticate and retrieve data.
4. **Configure multiple servers (optional):** Query Connector supports connections to multiple FHIR servers. Each server is configured independently. Users with appropriate permissions can select which server to query when running a search.

Mutual TLS (optional): If your FHIR server requires mutual TLS authentication, refer to the Mutual TLS Setup documentation at https://queryconnector.dev/docs/mutual-tls-setup for configuration instructions.

6 How to get help

6.1 Contact the DIBBs team

If your jurisdiction is getting active support, email your point of contact on the DIBBs team. If not, email us directly at dibbs@cdc.gov for any other support needs.

This document has been developed by Skylight under a contract with the
U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.