Xurrent Universal Database Connector

Introduction

The Xurrent Universal Database Connector application enables seamless integration of Xurrent data with Microsoft SQL databases. It supports data aggregation, customized reporting, and integration with external systems. The connector operates in all regions where Xurrent is available and is hosted on AWS.

By writing Xurrent data to a Microsoft SQL database, organizations can:

- Create aggregated reports combining Xurrent data with other systems.
- Generate customized reports not available natively within Xurrent.
- Enable seamless integration with third-party applications

Pre-requisites

Infrastructure Setup

For the application to function, the following components must be available:

- Microsoft SQL Database: A pre-existing Microsoft SQL database.
- Schema: A schema within the database for Xurrent data.
- SQL Username and Password: Credentials to be used as part of the SQL connection string.

The following details must be provided during installation:

- Database connection string (including server, port, username, password, and schema).
- Necessary network configurations for the application to access the database.

Region-Specific IP Addresses

The application uses the following IP addresses for outbound communication, depending on the region. Ensure that these IPs are allowed to access the database server in your network firewall:

Environment	Region	IP Address
Quality Assurance	Global	52.31.6.198 52.49.91.162 176.34.236.232
Quality Assurance	Australia	54.79.100.65 52.62.190.79 13.237.143.250
Quality Assurance	United Kingdom	3.9.93.231 18.175.21.201 13.42.99.72
Quality Assurance	Switzerland	51.96.43.10 16.63.78.35 16.62.69.80
Quality Assurance	United States	54.184.54.22 54.191.89.142 44.229.200.71
Production	Global	54.195.115.207 63.35.2.106 176.34.186.84
Production	Australia	3.24.5.124 52.62.0.253 52.64.140.204
Production	United Kingdom	18.133.96.188 13.43.168.192 13.42.121.172
Production	Switzerland	51.96.61.151 51.96.117.34 16.63.178.120
Production	United States	52.41.89.225 44.245.49.36 44.227.91.25

Port Information:
Any port can be used on the SQL server, as the port number is included in the SQL connection string. Ensure that the port is open and accessible from these IP addresses.

Microsoft SQL Configuration

The application automatically manages the SQL database structure. It will:

- Create tables.
- Add or remove columns.
- Create indexes.

Note: modifying, adding or removing columns, creating or updating indexes may result in unexpected behavior.

To enable this functionality, the SQL user in the connection string requires the following permissions:

Permission	**SQL Command**
View schema definition	`GRANT VIEW DEFINITION ON SCHEMA::[schema_name] TO [sql_user];`
Data manipulation (CRUD)	`GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::[schema_name] TO [sql_user];`
Create tables	`GRANT CREATE TABLE TO [sql_user];`
Alter schema objects	`GRANT ALTER ON SCHEMA::[schema_name] TO [sql_user];`

Database Creation Commands

The following SQL script can be used to create the database, schema, SQL user, and grant the required permissions:

```
-- Switch to the master database
USE [master];
GO

-- Create the Xurrent database
CREATE DATABASE [Xurrent]
GO

-- Create a login for Xurrent
CREATE LOGIN [Xurrent] WITH PASSWORD = 'YourSecurePasswordHere';
GO

-- Switch to the Xurrent database
USE [Xurrent];
GO

-- Create the schema if it doesn't exist
EXEC ('CREATE SCHEMA [Xurrent_schema];');
GO

-- Create the user in the database
CREATE USER [Xurrent] FOR LOGIN [Xurrent];
GO

-- Grant permissions
GRANT VIEW DEFINITION ON SCHEMA::[Xurrent_schema] TO [Xurrent];
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::[Xurrent_schema] TO [Xurrent];
GRANT CREATE TABLE TO [Xurrent];
GRANT ALTER ON SCHEMA::[Xurrent_schema] TO [Xurrent];
GO
```

Schemas

The use of SQL schemas enables flexibility in how data is stored:

- Single Schema Setup: Multiple accounts can store data in the same schema, simplifying access and management.
- Multiple Schema Setup: Data for different accounts can be segregated into different schemas for enhanced security and isolation.

Consider your organization's requirements for data segregation and select the appropriate schema configuration.

Connection String Example

To connect the Xurrent application to the database, use the following format for your SQL connection string:

```
Server=<your-server-address>,<port>;Database=Xurrent;
User Id=Xurrent;Password=YourSecurePasswordHere;
```

Replace <your-server-address>, <port>, and other placeholders with your specific configuration details.

Special considerations

Date and Time Exceptions

Xurrent contains a data type ISO8601Timestamp which includes three non-datetime values: `best_effort`, `no_target`, and `clock_stopped` that cannot be converted to a standard date and time format.

Those values three values are converted to the following values:

Name	Value
Best Effort	1970-01-01T01:01:01.100Z
No Target	1970-01-01T02:02:02.200Z
Clocked Stopped	1970-01-01T03:03:03.300Z

Custom Fields Case Sensitive

Custom Field identifiers are case sensitive which means that a single object, a request for example, can contain 2 custom fields with the same ID, for example `reference` and `Reference`. If your SQL database is configured with a case insensitive collation both values will be returned when filtering on the custom field identifier. The following query return will return the two rows when the database collation is case insensitive:

```
SELECT * FROM [demo].[RequestCustomFields] WHERE [ID] = 'reference'
```

To make the “where” statement case sensitive you can use `COLLATE Latin1_General_100_BIN2` after the column name:

```
SELECT * FROM [demo].[RequestCustomFields]
WHERE [ID] COLLATE Latin1_General_100_BIN2 = 'reference'
```

System Status and Log tables

The SQL database includes two key tables for tracking system processes:

Table name	Description
SystemLogs	Stores logs of the last 200 synchronization runs. Each synchronization run is grouped by the GroupID column.
SystemStatus	Tracks the synchronization state of each data entity. This table helps monitor synchronization status for each data entity and account.

Note: Avoid manually altering the values in these tables. Changes may disrupt the synchronization process, leading to failures. Recovery from such a failure requires deleting the affected rows and restarting the synchronization process from scratch.

Troubleshooting

The application logs critical information in two places: the SQL database and the Xurrent app. These logs assist in diagnosing and resolving issues during synchronization.

Database Logs

The SQL database maintains a log table containing detailed execution information, including warnings and errors. If the system encounters issues, such as an inability to connect to or write to the database, this log serves as a primary diagnostic tool.

App in Xurrent

By default, the Xurrent app does not display detailed synchronization state information. However, you can enable the “Suspend on Failure” option within the app configuration to aid troubleshooting during the initial setup:

- Suspend on Failure: When enabled, the app enters suspend mode if an error occurs, halting synchronization and displaying the latest error message as a suspend note. This option is particularly helpful during initial configuration and testing phases.
- After successful synchronization, you can disable this option to allow the process to continue running, even in cases of network or database unavailability. You can re-enable this feature later for error capture and diagnosis if necessary.