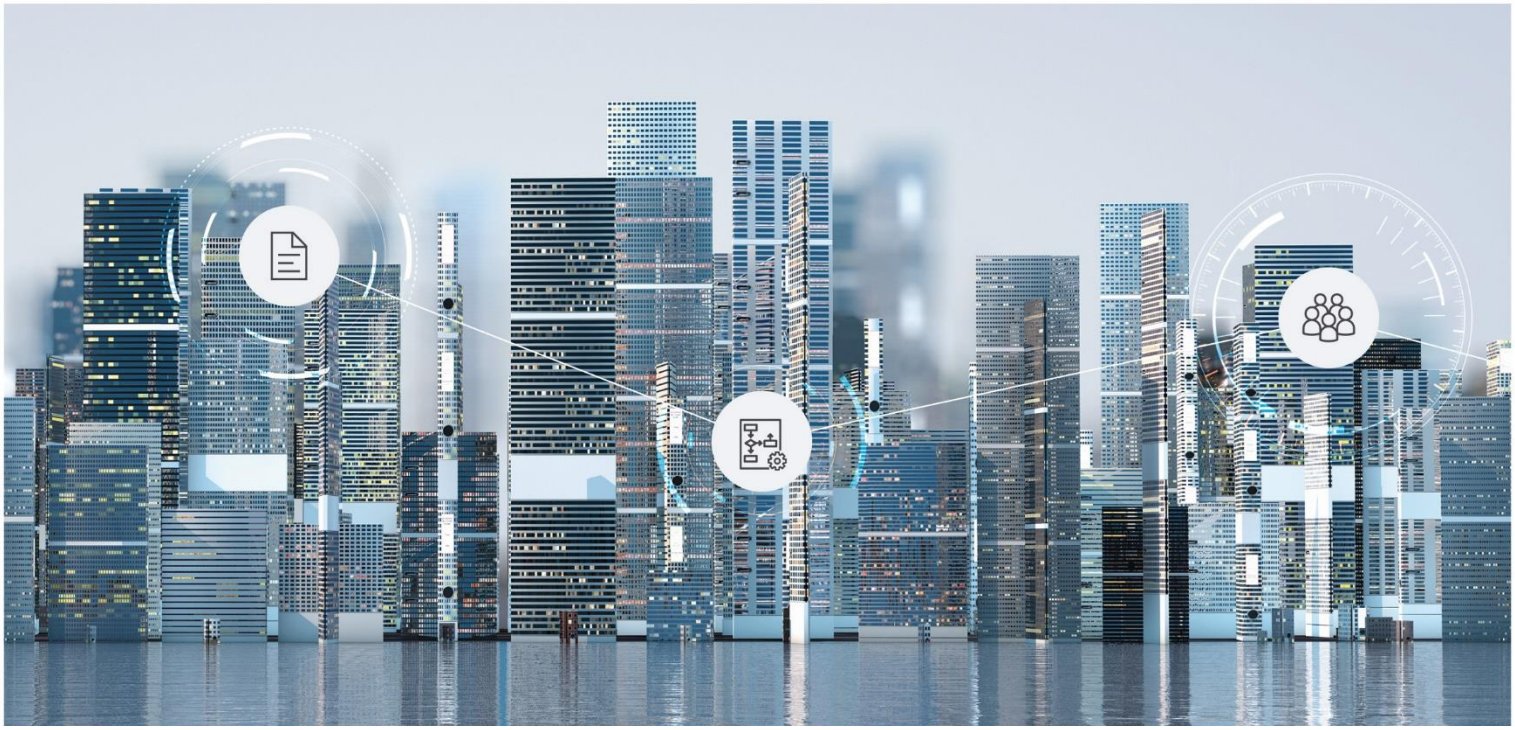
Fabasoft



White Paper

Configuration of Audit Logs

Fabasoft Folio 2019

Contents

1 Introduction 4

2 Software Requirements 4

3 Audit Data Sources 4

4 Audit Log Configuration in the Object Class 6

5 Security Aspects of the Audit 7

5.1 Reading Audit logs 7

5.2 Data Source 7

6 Example 7

1 Introduction

This document describes the configuration and use of audit logs. Using audit logs, it is possible to record access to properties, calls to actions or applications and the review of access rights.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment or Linux environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

3 Audit Data Sources

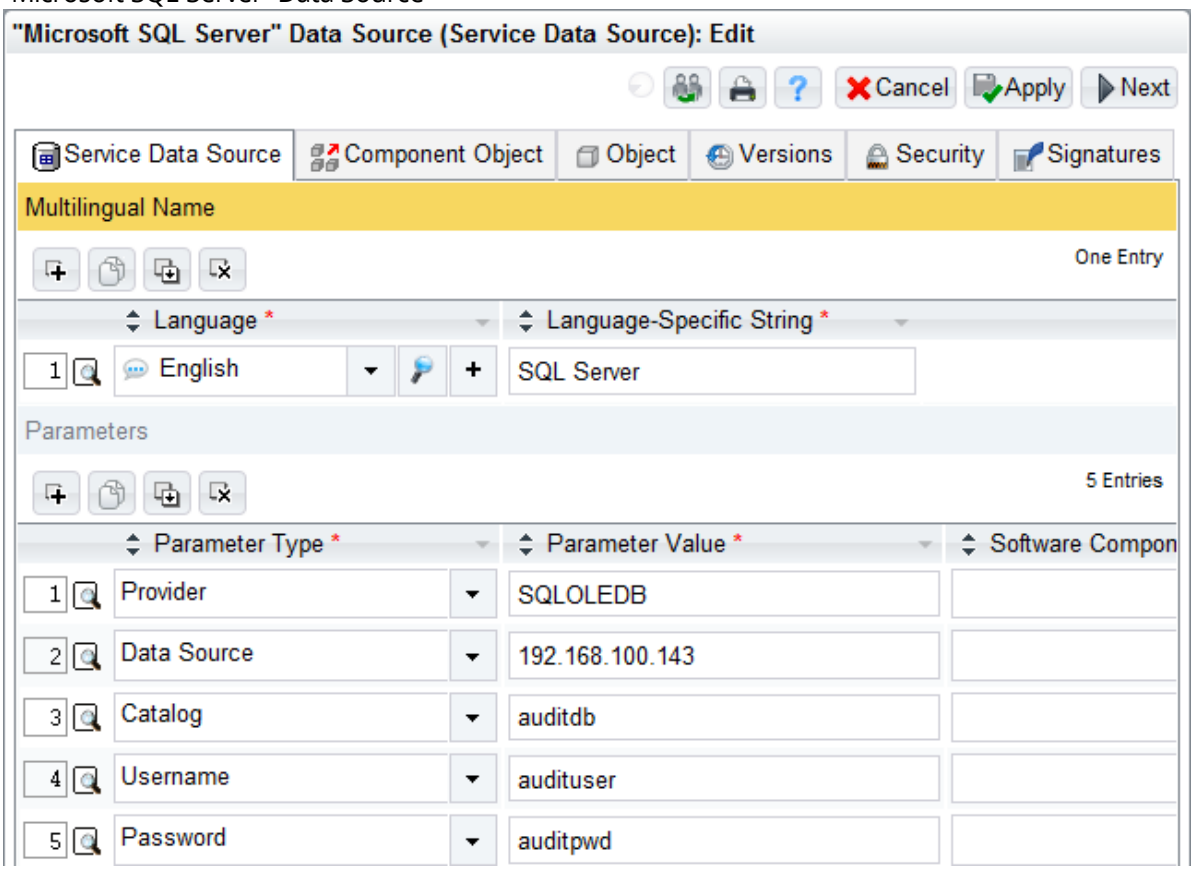
In the Fabasoft Folio object model, audit information is stored in objects of the object class *Audit Log* (`COOSYSTEM@1.1:AuditLog`) per default.

Alternatively, own data sources can be configured for storing audit information. Using dedicated audit databases, audit information is written directly to a database by the Fabasoft Folio Kernel. So the Fabasoft Folio Kernel has to be able to connect to the audit database via network.

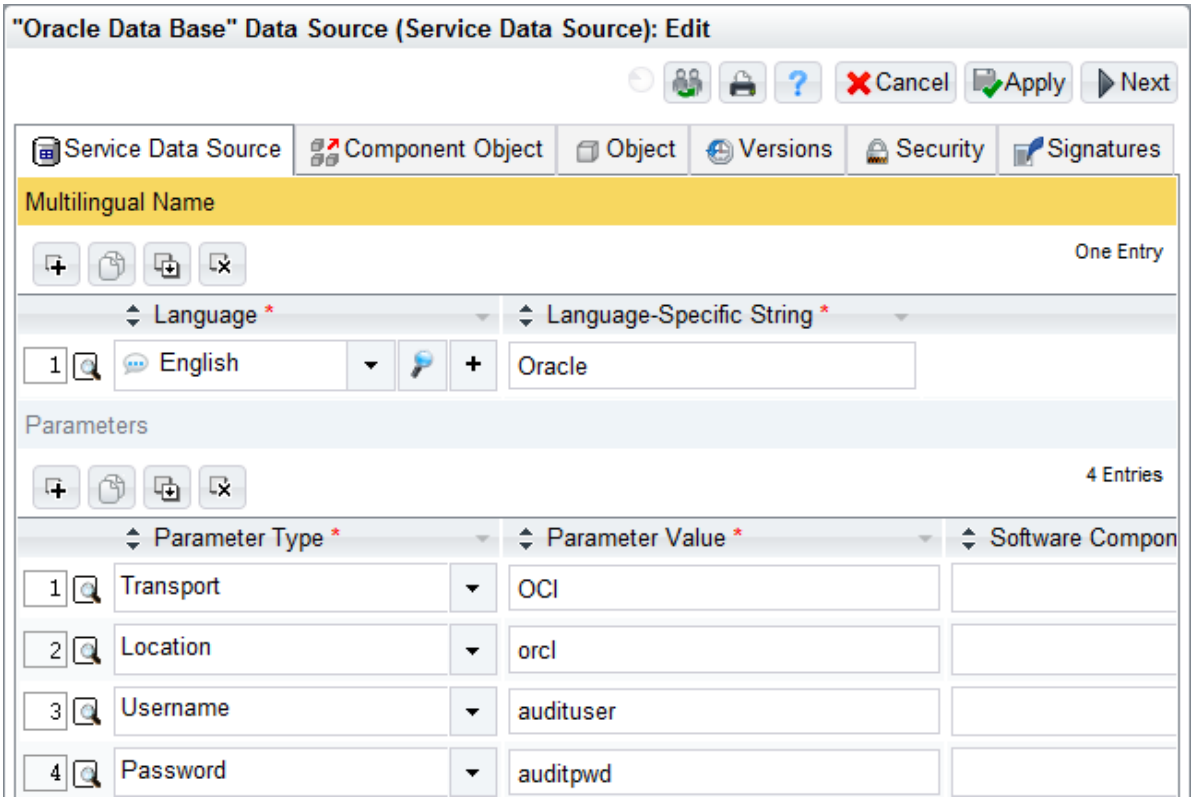
The data sources to be used (*Service Data Source*, `COOSYSTEM@1.1:ServiceDataSource`) have to be entered in the *Audit Data Sources* (`COOSYSTEM@1.1:domainauditdatasource`) property of the *Current Domain*.

Examples of Data Sources:

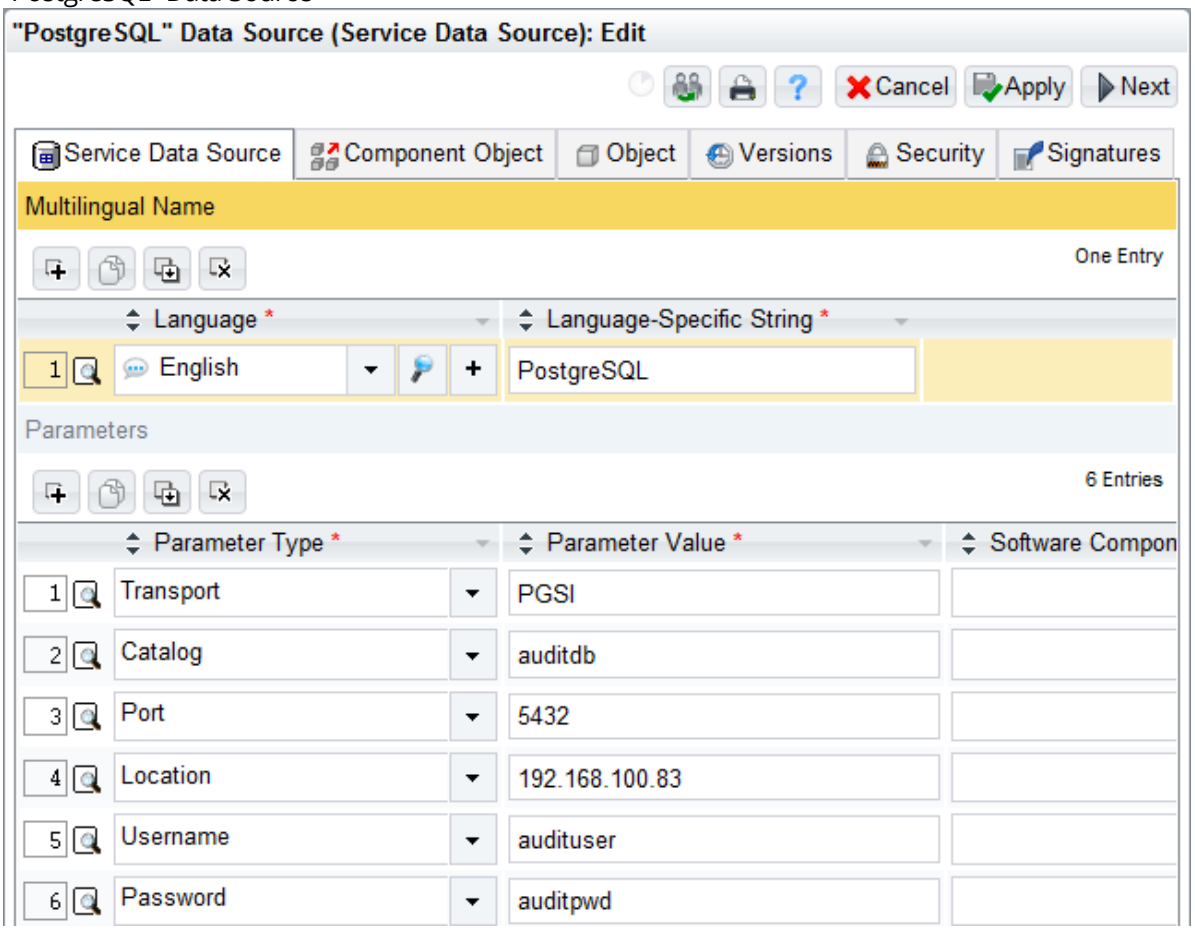
- "Microsoft SQL Server" Data Source



- “Oracle Database“ Data Source



- “PostgreSQL“ Data Source



Note:

- The use of a dedicated audit database should be preferred due to performance benefits.
- To make the audit log highly available it is necessary to configure several data bases in the *Current Domain*. If writing audit information fails on one data source, an automatic failover to another data source is performed.
- If no audit log data source is configured, all audit log entries are written to an audit log object. When the first audit log entry is written to an audit log object, this audit log object is automatically assigned to the `COOSYSTEM@1.1:objauditlogobj` property of the object. For this procedure, the object has to be locked. If it is not possible, to lock the object, the transaction fails. To avoid this problem, make sure that for each created object an audit log object is created and assigned. This can be done using an audit log configuration for `COOSYSTEM@1.1:objcreatedat`.

4 Audit Log Configuration in the Object Class

For each object class can be configured which access and calls to objects of this object class should be logged. The configuration is done on the "Advanced" tab of an object class in the aggregate list *Audit Log Configuration* (`COOSYSTEM@1.1:classauditconf`).

In the *Audit Log Configuration* the following settings can be made:

- *Context* (`COOSYSTEM@1.1:auditactionattrdef`)
  Definiert den Kontext für das Audit. Erlaubte Objektklassen sind:
  - *Action* (`COOSYSTEM@1.1:Action`)
  - *Property* (`COOSYSTEM@1.1:AttributeDefinition`)
    z.B.: objcreatedat
  - *Access Type* (`COOSYSTEM@1.1:AccessType`)
  - *Application* (`FSCVAPP@1.1001:Application`)
- *Audit Type* (`COOSYSTEM@1.1:audittype`)
  - "Read Property"
    Allowed context: Property.
  - Change Property"
    Allowed context: Property.
  - "Change Property (Values Saved)"
    Allowed context: Property.
  - "Change Property (Version Saved)"
    Allowed context: Property.
  - "Call Action"
    Allowed context: Action, Application.
  - "Call Action (Version saved)"
    Allowed context: Action, Application.
  - "Call Action With Success"
    Allowed context: Action, Application.
  - "Call Action With Error"
    Allowed context: Action, Application.
  - "Read Content"

- „"Access allowed "
  Allowed context: Access Type.
- "Access Denied "
  Allowed context: Access Type.
- *Condition* (`COOSYSTEM@1.1:auditcondexpr`)
  Additional condition in form of a Fabasoft DUCX expression.

- *Software Component* (`COOSYSTEM@1.1:auditcomponent`)

Note: An object class inherits the audit settings of her base class.

5 Security Aspects of the Audit

5.1 Reading Audit logs

To be able to read the audit log of an object class the *Read Audit Log* permission (`COOSYSTEM@1.1:AccTypeReadAuditLog`) is needed. The audit log can be read using the *Audit Log* property (`COOSYSTEM@1.1:objauditlog`) of an object.

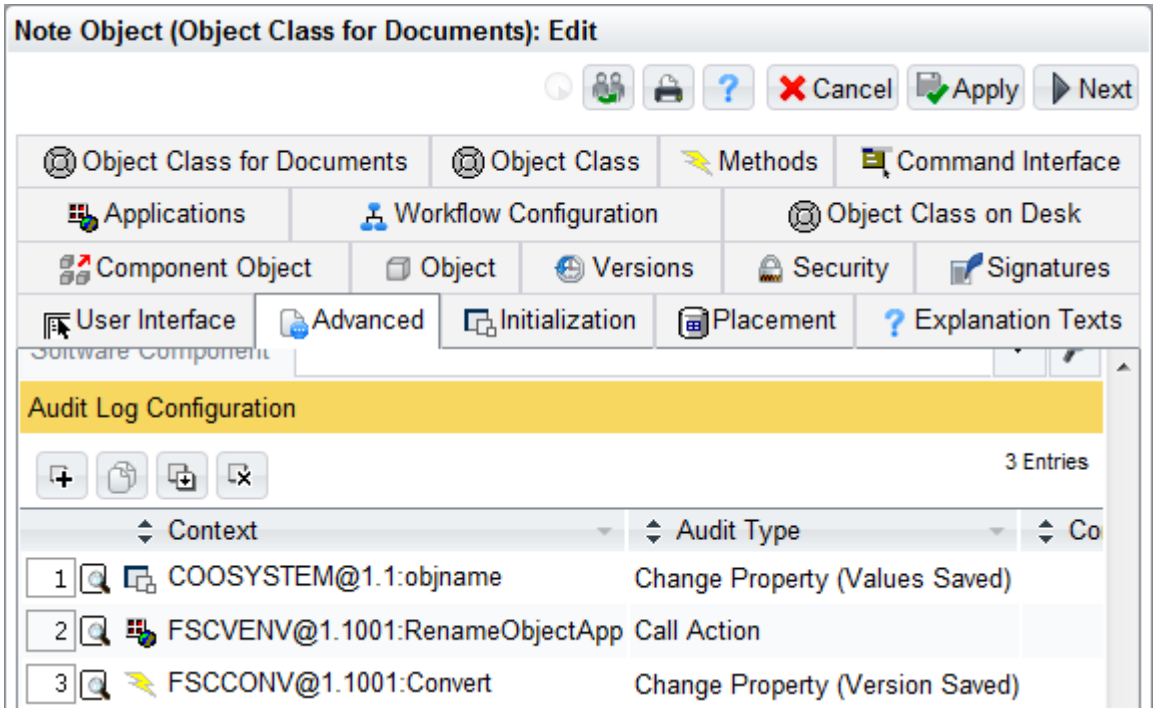
Note: If dedicated audit data sources are used, successful reading of data is required for each data source to be able to read audit information.

5.2 Data Source

If a user accesses a data source, for the first write access he or she needs the permission for creating tables. The table (`fscauditlogentrylist`) is created automatically and audit information is written to this table. Further on, only write and read access for this table is required.

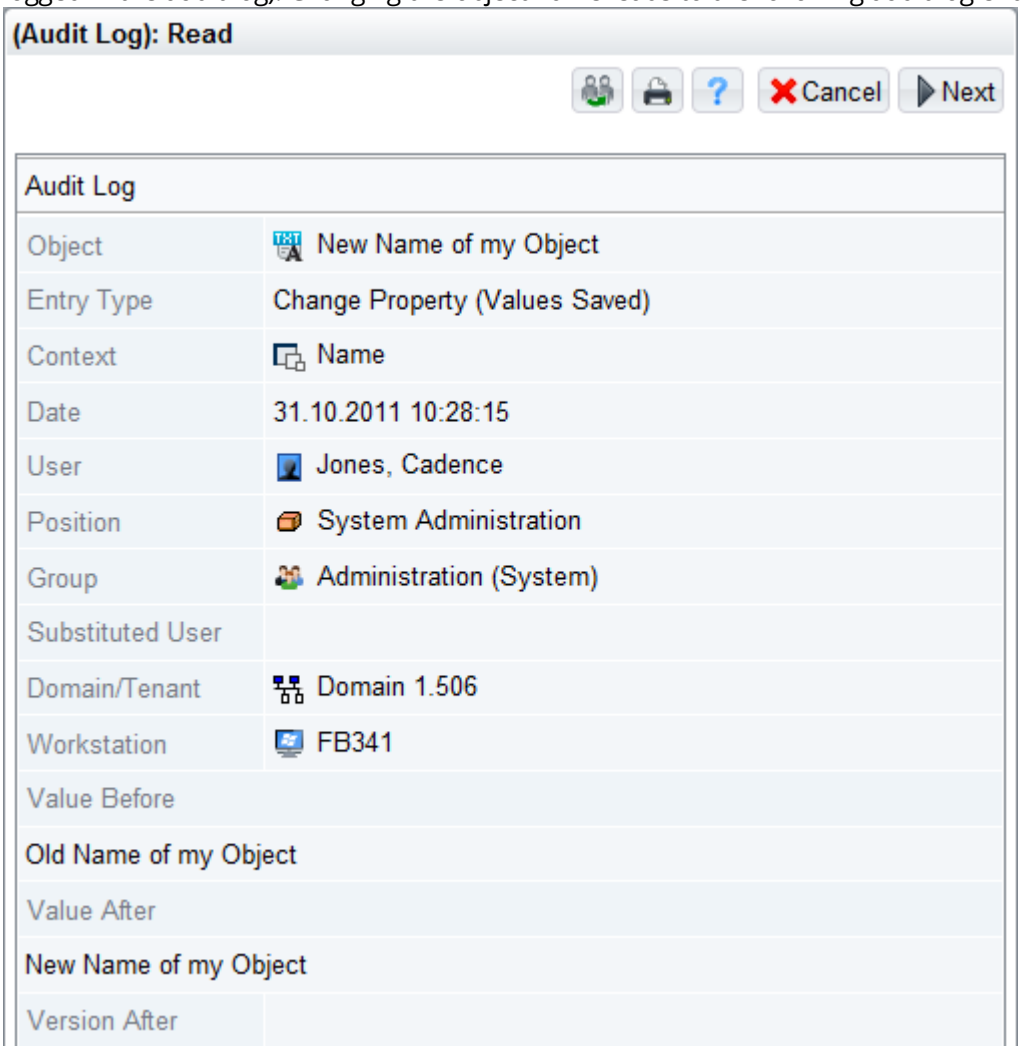
6 Example

Configuration of the object class (e.g. *Note-Object*, "Advanced" tab):



- In the first configuration line is determined, that all changes of the *Name* property (`COOSYSTEM@1.1:objname`) are logged and the values are saved (the old and the new value are

logged in the audit log). Changing the object name leads to the following audit log entry:



- In the second configuration line is determined, that all calls of the *Rename Object* application (`FSCVENV@1.1001:RenameObjectApp`) are logged.
- In the third configuration line is determined, that all calls of the *Convert content* action (`FSCCONV@1.1001:Convert`) are logged. Additionally with each action call a new version is created.