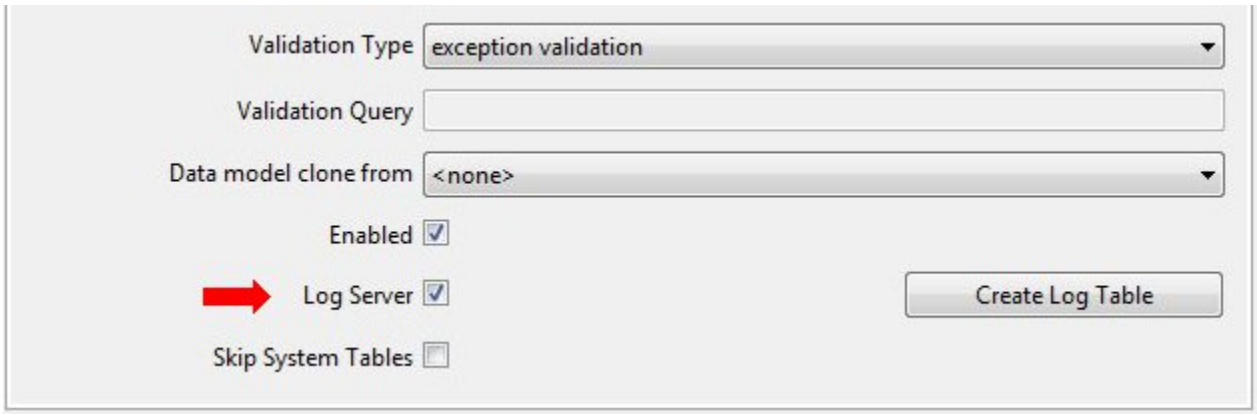
Implementing Audit Logging

Servoy provides the ability to log any data change or data read in a table in Servoy. This data is stored in a table specified in the database and can be reported, displayed, and used within a Servoy solution.

In This Chapter

Configuring Audit Logging

Configuring Servoy for audit logging is done in the database server connection pages. Any database server can contain the log table. To enable a server to be the log server, select Log Server in that database server's configuration editor.



The log table can a table with all the other data tables for your solution, or it could be in its own separate database server. Some reasons for having a separate database server for the log table include:

- As the audit log table could be very large, it can be stored separately from the other data tables.
- The log database could be optimized separately for inserting records, as audit logging is primarily inserting records into the log.
- Separate backup or archive strategy for the audit log database.

Once the database server has been decided, check the Log Server box and click the Create Log Table button. A table named "log" will be created in the database server. Now audit logging is available for any table in the resource project (or on the application server for Servoy Server).

To log any changes or views for a table, open the table in the table editor (crossreferece here), select the security tab, select the group, and enable the Tracking and/or Tracking views option. If Tracking option is enabled any adds, edits, or deletes to the table will be tracked for the selected group. If Tracking views is enabled any views of data to the table will be tracked for the selected group (Tracking views was added in Servoy 6.0.2 release).

Using Audit Log Data

As the log table is simply another database table, a developer can builds forms against the log table to allow users the ability to read and work with the data. Data within the audit log could also be used to access any changed record in the database.

Below are descriptions of every column in the log database.

- event_time - The time the change occured
- log_id - The auto incrementing ID of the log table
- log_action - The type of action that occurred. 1 = delete; 2=insert (or add); 3=update(changed record); 4=view(data read)
- server_name - Name of the DB server registered in the preferences window
- table_name - Name of the table affected by the action
- column_name - Name of the column (field) affected by the action
- pk_data - Primary key data for the record being changed/viewed. Format: x.yyyy; x=number of characters in the key. y=the actual key value. Multiples are separated by semicolons
- old_data - The old data value before the change (will be null for views)
- new_data - The new data value after the change or the data that is read from database
- user_uid - The UID value for the user making the action. This can be set during the login process or is stored in the repository if using basic authentication.