SERAM Setup Guide

Sustainable Enterprise Reporting And Management

by Sirius Technologies AG

Notices

Disclaimer
The information in this manual is provided on an “as is” basis, without warranty. While every effort has been taken by the author in the preparation of this manual, the author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this manual.

This manual may contains links to third-party web sites that are not under the control of the author. The author is not responsible for the content of any linked site. Inclusion of a link in this manual does not imply that the author endorses or accepts any responsibility for the content of that third-party site.

Trademarks
All terms mentioned in this manual that are known to be trademarks or service marks have been capitalized as appropriate. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark.

Versions
Software release: Banker (Build 2472)
Document revision: 495

Contents

Chapter 1

Software Installation

The following environment/software is required to run SERAM:

Web Server

- *Microsoft Windows Server* (2019 preferred, 2012 R2 or newer supported)
  - Web Server (IIS) Role
    - *ASP.NET* 4.x (including dependencies)
    - If Integrated Authentication shall be used, Windows Authentication
  - *Microsoft .NET Framework* 4.7.2 or 4.8

Note: Multiple *Web Servers* may be used in order to create a redundant environment.

Database Server

- *Microsoft Windows Server* (2019 preferred, 2012 R2 or newer supported)
  - *Microsoft SQL Server* (2014 or newer, any edition including Express is supported)

Note: For small-scale or test environments, the *Web Server* and *Database Server* may be installed on the same *Windows* machine.

Databases

SERAM uses multiple SQL databases: one for the global scope and one per tenant. See the *Multitenancy* chapter for more details.

Database Authentication

While any valid connection string may be provided to SERAM, and thus also any authentication type may be specified, the suggested configuration is to use Integrated Authentication and set the *SQL Server* to **Windows Authentication mode** only.

As a best practice, the login used should be specific to SERAM. When running both SQL Server and IIS on one machine, the virtual user `IIS APPPOOL\SeramAppPool` may be used, otherwise a dedicated service account or the machine account `DOMAIN\SERAM-WEBSERVER$` may be configured as *Windows* login on the *SQL Server*.

Database Roles

In its default configuration, SERAM manages its database schemas automatically. This is true both for creating tenant databases as well as when the software is updated. Therefore the `dbcreator` Server Role should be assigned to the SERAM login(s). Details about this role are available in the SQL Server documentation: *https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/server-level-roles*.

On the global database (by default called `SeramTenantManager`), the login should have the `db_owner` role in order to perform the schema updates.

Note: If corporate policies disallow the use of these roles, the configuration may be changed to disable the automatic schema management and SQL setup scripts may be obtained from Sirius Technologies AG.

Every software update which contains DB updates will require manually upgrading the database schema with scripts. Without the `dbcreator` role, creating new tenants from within the application will not work.

Global Database Setup

The global database (by default called `SeramTenantManager`) is always required.

- Administrative access to the database server is required.
- A tool for executing SQL statements is required, for instance *SQL Server Management Studio*.
- Make sure that the configured server protocols and firewall rules will allow the application on the *Web Server* to access the *SQL Server* instance.

The following steps describe how to prepare the global (aka *Tenant Manager*) database which is always required.

1. Connect to the SQL server instance.
2. Configure the `model` database to match the settings for your environment, e.g. recovery model, growth settings, etc.. When creating new tenant databases, the settings from the `model` will be used since SERAM does not specify options when creating the new database.

```
ALTER DATABASE [model] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [model] SET ALLOW_SNAPSHOT_ISOLATION ON
GO
ALTER DATABASE [model] SET READ_COMMITTED_SNAPSHOT ON
GO
```

Note: Snapshot isolation is supported by SERAM and should be enabled.

3. Create the new global database. The default name is `SeramTenantManager`.

```
CREATE DATABASE [SeramTenantManager];
```

4. Create the login to be used by the application, and grant the required roles.

```
USE [master]
GO
CREATE LOGIN [DOMAIN\USERNAME] FROM WINDOWS WITH
 DEFAULT_DATABASE=[master]
GO
ALTER SERVER ROLE [dbcreator] ADD MEMBER [DOMAIN\USERNAME]
GO
USE [SeramTenantManager]
GO
CREATE USER [DOMAIN\USERNAME] FOR LOGIN [DOMAIN\USERNAME]
GO
ALTER ROLE [db_owner] ADD MEMBER [DOMAIN\USERNAME]
GO
```

Note: Replace all `DOMAIN\USERNAME` entries to match your environment.

The database setup is complete.

Web Site

SERAM is a ASP.NET MVC application running in IIS.

Hostnames and DNS

SERAM uses hostnames to differentiate between the global management part and each of the tenants.

The single SERAM website will handle requests to different hostnames under a common domain name.

Note: Only the first (left-most) part of the hostname is relevant for identification purposes in the application. A leading `www` will, however, be ignored.

Note: If a reverse proxy or load balancer is used, it has to pass the hostname through to SERAM.

By default, the global management part uses the `login` hostname, e.g. the web address would be in the form `https://login.seram.local`.

Note: If required, this may be changed with the `ManagementHost` setting, see *Configuration* for details.

Each tenant is assigned exactly one distinct hostname, e.g. the web address would be in the form `https://mytenant.seram.local`.

Note: In order to allow for tenant creation without additional configuration, the usage of a wildcard DNS entry, wildcard IIS website binding and (if SSL is used) wildcard certificate is suggested. In environments where new tenants are not created, single static bindings, DNS names and certificates may be used instead.

Web Site Deployment

The SERAM application does not require any registry changes or components to be installed in the system, it is therefore "Xcopy deployable". No installer is provided for this reason.

- Perform these steps on all *Web Servers* on which you want to run SERAM.
- Configuration of DNS is assumed to be completed before configuring IIS.

1. Extract the binaries archive to the location where the website shall reside on disk, for example `C:\inetpub\SERAM`. The application only requires read access to this directory.
2. Edit the `web.config` file and adjust the configuration as required.
   - Adjust the `DomainRoot` and `ManagementHost` to match your DNS names. This will be used when creating URLs and redirections. For example, if the host name was `*.seram.local`:

     ```
     <add key="DomainRoot" value="seram.local" />
     <add key="ManagementHost" value="login" />
     ```

   - Optionally configure the caching directory for the value cache (see *Settings* for details and default directory). For example, if the directory to use for the cache was `Z:\Seram-Cache`:

     ```
     <add key="Caching:directory" value="Z:\Seram-Cache" />
     ```

   - Verify that the correct `SessionUserProvider` for your environment is used. See *Authentication* chapter for details.
   - Adjust the connection string `TenantManager` for accessing your SQL global database.

     ```
     <add name="TenantManager"
         connectionString="Data Source=SQLSERVER;Initial
      Catalog=SeramTenantManager;Integrated Security=SSPI;"
         providerName="System.Data.SqlClient" />
     ```

3. In IIS Manager, create an application pool to be used by the application.
   - *.NET Framework* CLR **4.0.x**
   - **Integrated** pipeline mode
   - Enable 32-bit applications: **false**
   - Start immediately: true
4. Still in IIS Manager, create a website for the application.
   - Use the application pool created in the prevoious step as default.
   - Enter the path where the binaries were extracted in the first step.
   - Configure anonymous authentication when using the session-based user provider, or Windows Integrated authentication when using the session-less user provider.
   - Add a binding for accessing the application.

     Note: A wildcard binding is suggested, but a binding for the management host (`login...` by default) is required at least.
5. Open a web browser and navigate to the management host of the newly created website.

SERAM should now automatically configure its database schema and start up in *Setup Mode*.

Note: In the exceptional case where computations involving huge amounts of indicator values are expected (more than ~18mio distinct values in a single computation batch), support for arrays greater than 2GB needs to be enabled by editing the `machine.config` file in the system framework folder as follows:

```
<runtime>
 <gcAllowVeryLargeObjects enabled="true" />
</runtime>
```

First Steps

The following first steps will only be performed once for setting up the base configuration of the global database.

Setup Mode

On the first application launch, SERAM will be in *Setup Mode*. As long as this mode is active, any new user logging on will be created with administrator permissions.

1. Access SERAM via its management URL, for example `login.seram.local`. If you have already configured access through a SAML2 IdP portal (see *SAML2 IdP Configuration*), then you should also be able to log on using the portal.
   You should see the SERAM application with a red banner on the top of the page stating **SETUP MODE**. If you don't see the banner, setup mode has been disabled already.
2. Log on to SERAM.

   Note: If you don't have an existing OpenID account for your first login, you may use the **Log on with username and password** option. Unless configured otherwise, this will redirect you to the *SERAM OpenID* provider hosted by Sirius Technologies AG.

   You may create an account there in order to have an account for creating your administrator account in your new SERAM installation in order to proceed with the configuration.
3. When prompted, enter your full name and e-mail address, and **Register with SERAM**.
   You should now be logged on and on the **Tenant Management** page.
4. Disable **Setup Mode** by blicking the **Disable** button in the red banner.

You have now created your administrative user and disabled **Setup Mode**, so that new users loggin on to the application will no longer be made administrators.

Chapter 2

Authentication

SERAM relies on external authentication systems for identifying users.

Currently, SERAM supports the following protocols for authentication:

- Session-based user provider: `Seram.Web.Authentication.OpenIdUserProvider`
  - OpenID
  - Google OAuth
  - SAML2 (IdP-Initiated)
- Session-less user provider: `Seram.Web.Authentication.WindowsUserProvider`
  - Windows Integrated

Independently of the chosen user provider, SERAM uses *Claims* for identifying users. Each user may have any number of claims, therefore it is also possible to enable multiple means of authentication for specific users, or to seamlessly migrate existing users to a new authentication system.

Claims

The *Claim* is used to identify the user based on information (claims) provided by a ticket of a trusted external authentication provider.

Each *Claim* consis of three parts:

Issuer

The issuer identifies the issuing party of the ticket/claim. In *OpenID* and *SAML2* protocols this is a URI, for Windows Authentication this is `LOCAL AUTHORITY`.

Type

The data type of the claim. This usually has the form of a URI, common values are `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress` for e-mail addresses, `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier` for identifiers, or `http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid` for *Windows* groups.

Value

The value of the claim, such as the e-mail address.

Note: All parts of the *Claim* are case-sensitive when being matched against the information of an authentication ticket.

Add or Edit Claim Issuer Metadata

Claim Issuers may be configured with specific metadata. This advanced configuration can only be performed directly on the database level.

- Write access to the *Tenant Manager* database (by default called `SeramTenantManager`) is required.
- A tool for executing SQL statements is required, for instance *SQL Server Management Studio*.
- For good measure you should have a backup of the database in the case something goes wrong.

The following steps describe how to access and modify the metadata of *Claim Issuers* used by SERAM. The *Claim Issuers* are identified by their URI in the database.

1. Connect to the SQL server instance.
2. Change the active database to the *Tenant Manager* database.

```
USE SeramTenantManager;
```

3. Optional: Get the current metadata value for a *Claim Issuer*, for instance as basis for modification.

```
SELECT iPriority, xMetadata
FROM Shared.tblClaimIssuer
WHERE sClaimIssuer=N'https://sso.acme.com/SAML2';
```

4. Add or update the *Claim Issuer* metadata and priority.

```
MERGE INTO Shared.tblClaimIssuer AS t
USING (VALUES (
N'https://sso.acme.com/SAML2', -- Claim Issuer URI
75, -- Priority, lower number comes first, default is 100
N'<xml>
  <samlCertificate>MIIDSjCCAjKgAwIBAg...</samlCertificate>
</xml>' -- Metadata as XML
)) AS s(sClaimIssuer, iPriority, xMetadata)
    ON s.sClaimIssuer=t.sClaimIssuer
WHEN NOT MATCHED THEN INSERT (sClaimIssuer, iPriority, xMetadata)
    VALUES (s.sClaimIssuer, s.iPriority, s.xMetadata)
WHEN MATCHED THEN UPDATE SET iPriority=s.iPriority,
 xMetadata=s.xMetadata;
```

The metadata settings of the *Claim Issuer* are now used for all claims with the corresponding URI.

Claim Issuer XML Metadata

The following *Claim Issuer* metadata is supported by SERAM.

`samlCertificate`

Specifies the certificate used to verify the SAML2 assertion. If the certificate does not match, the SAML2 call to the SERAM *Assertion Consumer Service* will fail.

The X.509 certificate has to be provided as BASE64 (same as in the SAML assertion).

`logoutJavaScript`

Specifies a JavaScript snippet which will be executed when the user logs out of SERAM.

The snippet can for instance be used to notify the *Claim Issuer* that the session of the user shall be invalidated.

Common Claim Types

SERAM allows any *Claim Type* to be specified, but usually they are well-known.

Claim Type URI	Description
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/denyonlysid`	The deny-only group SID of the user
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`	The e-mail address of the user
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name`	The unique name of the user

Claim Type URI	**Description**
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier`	The SAML name identifier of the user
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier`	The private identifier of the user
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn`	The user principal name (UPN) of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/denyonlyprimarygroupsid`	The deny-only primary group SID of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/denyonlyprimarysid`	The deny-only primary SID of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid`	The group SID of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/name`	Name as passed in by ACS
`http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid`	The primary group SID of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid`	The primary SID of the user
`http://schemas.microsoft.com/ws/2008/06/identity/claims/role`	Role that the user has based on group membership
`http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname`	The domain account name of the user in the form of `DOMAIN\USER`
`http://schemas.xmlsoap.org/claims/CommonName`	The common name of the user when interoperating with AD FS 1.1 or ADFS 1.0
`http://schemas.xmlsoap.org/claims/EmailAddress`	The e-mail address of the user when interoperating with AD FS 1.1 or ADFS 1.0
`http://schemas.xmlsoap.org/claims/Group`	A group that the user is a member of when interoperating with AD FS 1.1 or ADFS 1.0
`http://schemas.xmlsoap.org/claims/UPN`	The UPN of the user when interoperating with AD FS 1.1 or ADFS 1.0

SAML2 IdP Configuration

SAML2 IdP providers have to be configured as follows to interoperate with the SERAM *Assertion Consumer Service* (ACS).

Single Sign On URL (Recipient URL and Destination URL)

The address of the SERAM ACS. Replace the hostname `seram.local` with your host.

`https://login.`*`seram.local`*`/Authentication/Saml2/AssertionConsumerService`

Audience URI / SP Entity ID

The address of the SERAM ACS. Replace the hostname `seram.local` with your host.

`https://login.seram.local`

Name ID Format

Email Address is suggested (*Claim Type* `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`).

You may be able to use other formats with other *Claim Types*.

Assertion Signature

Signed.

Signature Algorithm

RSA-SHA-256.

Digest Algorithm

SHA256.

Assertion Encryption

Unencrypted.

SAML Issuer ID

Must match the *Claim Issuer* used in SERAM claims.

Add SERAM as Microsoft Entra (Azure AD) application

Follow these steps to add SERAM as application for Microsoft Entra ID (Azure AD) users.

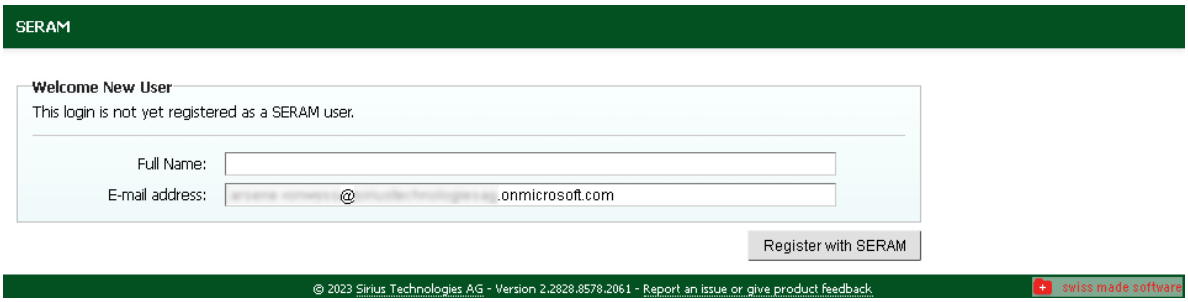
1. Open the *Microsoft Entra admin center*
2. Go to **Applications** > **Enterprise Applications**
3. Click **Add Application**
4. Since SERAM is not in the Gallery, click **Create you own application**
5. Give the new application a name, and choose **Integrate any other application you don't find in the gallery (Non-gallery)** > **Create**
6. In the application configuration, go to **Users and groups** and assign users and/or groups which shall have access to SERAM

   Note: In order to successfully test the sign-on later, you may need to assign yourself to the application.
7. In the application configuration, go to **Single sign-on** > **SAML**
8. Click **Edit** in the *Basic SAML Configuration* section
9. Enter the *Identifier (Entity ID)* and *Reply URL (Assertion Consumer Service URL)* as described in the *SAML2 IdP Configuration*
10. Click **Save** to save the configuration, and close the *Basic SAML Configuration*.

    Important: Don't test the sign-on yet if asked upon closing, it would fail since the SAML endpoint has not yet been configured in SERAM.
11. In the *SAML Certificates* section, download the *Certificate (Base 64)*
12. Add a row to the *SeramTenantManager* database for the issuer, using the *Microsoft Entry ID Identifier* from the *Set up SERAM Test* section, and the BASE64 contents of the certificate file downloaded in the previous step (open the file in a text editor):

```
INSERT [Shared].[tblClaimIssuer] ([sClaimIssuer], [iPriority],
 [xMetadata]) VALUES (
    N'https://sts.windows.net/12345678-1234-1234-1234-12345678ABCD/',
    75,
    N'<xml><samlCertificate>MII...</samlCertificate></xml>');
```

13. You can now test the sign-on. After sign-on, you should see the SERAM application with the *Welcome New User* page, and the pre-filled *E-mail address* as received from the *Entra ID* authentication provider.


SERAM
Welcome New User
This login is not yet registered as a SERAM user.
Full Name:
E-mail address:
[redacted]@[redacted].onmicrosoft.com
Register with SERAM
© 2023 Sirius Technologies AG - Version 2.2828.8578.2061 - Report an issue or give product feedback
swiss made software

Chapter 3

Users and Groups

Users and Groups are used to manage permissions in SERAM. Users, Groups and related entities are called *Subjects* in SERAM.

Users

The Users of SERAM are global (not per tenant). The permissions assigned to them directly or indirectly (via Groups) determines what they are allowed to see and perform.

Users may be active or inactive. Inactive users cannot log on to SERAM anymore, but their user informartion remains available for audit and history reasons.

Groups

Groups in SERAM are also global. By using groups, the management of permissions can be made more efficient since permission entries do not change when users are added or removed from groups.

Groups may contain both users and other groups as members. The groups a user is member of is thus determined by direct and indirect membership in groups.

Edit User Name

User names are not used as identifiers (each user has an internal GUID as identifier), therefore names can be changed freely.

- Log on and open the **Global Users** module.

1. Select the user which you want to edit.

Note: You may use the filter of the data grid to quickly narrow the list of users.

Note: You may only edit users on which you have the `ManageUser` permission.

2. Click the **Selected Subject** > **Edit ...** action.
3. Edit the user name (and/or toggle the Active flag).
4. Click the **Save** button.

Changes are immediately applied.

Manage Claims of User

In order to authenticate, users must be assigned claims which are used for identifying the user.

- Log on and open the **Global Users** module.

1. Select the user for which you want to modify claims.

Note: You may use the filter of the data grid to quickly narrow the list of users.

2. Click the **Selected Subject** > **Claims Of...** action.
3. Add, edit or remove claims with the actions in the **Claims** and **Selected Claim** categories.

Note: You may only manage the claims of users on which you have the `ManageUser` permission.

Changes are immediately applied.

Activate and De-Activate Users

Users may not be deleted, but they may be made inactive in order to prevent authentication.

- Log on and open the **Global Users** module.

1. Select the user which you want to activate or de-activate.

Note: You may use the filter of the data grid to quickly narrow the list of users.

Note: You may only activate and de-activate users on which you have the `ManageUser` permission.

2. Click the **Selected Subject** > **Activate ...** or **Selected Subject** > **De-Activate ...** action.

Changes are immediately applied.

Merge Users

Users cannot be deleted, but if users entries are duplicates, these can be merged.

- Log on and open the **Global Users** module.

1. Select the primary user entry. This user will remain in the system.

Note: You may use the filter of the data grid to quickly narrow the list of users.

Note: You may only merge users on which you have the `ManageUser` permission.

2. Click the **Selected Subject** > **Merge User into ...** action.
3. Select the duplicate user entry. This user will be removed from the system, but his permissions and claims will be transferred to the primary user.
4. Click **Merge User** to perform the merge.

Warning: The operation cannot be undone, make sure that the accounts are indeed duplicates.

Warning: The old user name will no longer be resolved correctly, therefore you may see the GUID of the user in some places of the application.

The user is merged. Changes are immediately applied.

Impersonate a User

Adminstrators may act as a different user by the means of impersonation. This is useful for debugging purposes or to test user account permissions.

- This action is only available when using a session-based user provider (see *Authentication* topic), and for users which you may manage. Managing users implies being able to edit their claims and thus impersonate their account, therefore this functionality has no negative security impact. Impersonation of users is recorded in the audit log.
- Log on and open the **Global Users** module.

1. Select the user which shall be impersonated.

Note: You may use the filter of the data grid to quickly narrow the list of users.

Note: You may only impersonate users on which you have the `ManageUser` permission.

2. Click the **Selected Subject** > **Impersonate this User** action.
3. Your session is now impersonating the selected user, and you may see and use the system as if you were that user.

   Note: Only one level of impersonation is allowed. Therefore, while impersonating a user, it is not possible to impersonate another user even if the impersonated user has sufficient permissions to use impersonation.
4. To end impersonation, click the **Revert** link on the top right of the screen.

   Warning: Since the session is switched to impersonate the user, using the back button of the browser or closing the browser will not end impersonartion. Therefore make sure to always end the impersonation with **Revert**.

Manage Group Members

- Log on and open the **Global Users** module.

1. Select the user which you want to edit.

   Note: You may use the filter of the data grid to quickly narrow the list of users.
2. Click the **Selected Subject** > **Members of ...** action.
3. Add or remove members using the actions in the **Members** category.

   Note: You may only manage members of groups on which you have the `ManageMembers` or `ManageUser` permission.
4. Click the **Save** button.

Changes are immediately applied after saving the members.

Chapter 4

Multitenancy

SERAM is built for supporting multiple *Tenants* on a single instance by default.

Global Scope

In the *Global Scope* of the application, *Users*, *Groups*, *Tenants* and *Settings* are managed.

Each *User* has its own set of *Claims* which are used to identify the user. This allows using different (including multiple concurrent) means of external authentication systems for accerssing SERAM. See the *Authentication* chapter for details.

Groups may be used to group multiple users and other groups together for simplified permissions management. See the *Users and Groups* chapter for details.

Tenants

Each tenant has its own configuration and database. It is therefore independent of the global instance except that is uses the user IDs from the global scope for permission entries and may inherit some settings and permissions defined globally.

Tenant Access

Visibility and access to tenants is controlled via permissions granted to subjects (users and groups). Therefore a user may have access to none, one or several tenants when logging in.

User who do not have the right to create a new tenant (e.g. no `ManageTenant` permission globally) will see the following behavior:

No Tenant

The user will be presented with a simple message that he has no access to any tenant.

One Tenant

If the user has access to one tenant only, he will be redirected to that one tenant automatically. For the user the application behaves like a single tenant system.

Multiple Tenants

The user will be presented with a list of tenants to which he has access. Each entry of the list is a link which will redirect the user to the specific tenant.

Chapter 5

Configuration

SERAM is very configurable at different levels.

Configuration settings may be configured in the `web.config` `appSettings`, globally or by tenant.

Settings configured in `web.config` `appSettings` take precedence over all other settings.

Global settings are inherited to all tenants, but any setting re-defined on the tenant level will override the global setting.

Settings

All SERAM settings.

Global Settings (`web.config`)

Key	Type	Default	Description
ModuleStore:autoUpdate	Boolean	true	Automatically perform DB updates by ModuleStore.  **Note:** If not specified in the config file, this setting can be set at the Tenant Manager level, in which case it applies to the tenant databases (but not the tenant manager database).
ModuleStore:snapshotIsolation	Boolean	false	Controls whether snapshot isolation transactions are used or not.  **Note:** This setting has no effect if the database is set up to use "*Read-Commited Snapshot Isolation*".
ModuleStore:ignoreInventory	Boolean	false	Skip the inventory checks of ModuleStore - this may lead to outdated DBs being used by the application
ModuleStore:forceUpdateCheck	Boolean	false; true if debugger is attached	Controls whether ModuleStore is to perform an update check even if the hash in the module list matches the inventory hash.
RequireJS:minimize	Booloean	true; false if debugger is attached	Controls whether scripts loaded by the means of the RequireJS configuration are to be automatically minimized.
RequireJS:bundle	Boolean	Opposite of no-cache	Controls if related JS files are bundled on the server to reduce the number of required requests and improve application performance.

Key	Type	Default	Description
RequireJS:no-cache	Boolean	true for DEBUG build, false otherwise	Controls whether scripts are cacheable, which implies using filenames based on their content hash. For development, it has to be disabled or the names are cryptic.
Session:timeoutMinutes	Integer	0	If larger then 0, the sessions will time out after the specified amount of inactivity minutes.
Session:purgeDays	Integer	0	If larger then 0, expired sessions will be purged from the database with the given frequency.
DebugMode	Boolean	depends on build type: only true for debug builds	Changes some debugging settings such as caching behavior.
ClientCacheDuration	*Timespan*	10 hours	Changes the time static files may be cached by the client (only applies when DebugMode is false).  **Note:** This setting does not affect JavaScript file caching, which is controlled through the `RequireJS:-` Settings.

Global Settings (Tenant Manager)

Key	Type	Default	Description
CustomerHostPattern	Regex Pattern	^(?!(www|login|admin|seram)$)[a-z0-9][a-z\-0-9]{,29}[a-z0-9]$	The pattern is used to verify if a name has the correct format. Reserved names (www etc.) are excluded, and the length is restricted to 31 characters.
DatabaseNamePattern	Format String	SeramTenant{0}	The placeholder will receive the host name specified by the customer.
DomainRoot	String		If specified, controls what the domain for the application is. This disables the heuristic host name detection when matching tenant names.
ManagementHost	String	login	Only applicable when a DomainRoot has been set. Used when the user is redirected to a management URL, such as for logging on.
SetupMode	Boolean	true	As long as the SetupMode is enabled, all new users will receive full administrative permissions over the full system. This ensures that the first user on a new installation will be able to administer the system.
ActiveTenantUpdateAction	string	none	none: no actionshutdown: shutdown the hosting environmentrecycle: recycle the IIS app pool

Key	Type	Default	Description
Licensing:enabled	Boolean	true	Controls whether licensing and usage tracking are enabled for the whole installation.
Security:claimTemplates	JSON	(undefined)	Specifies the E-Mail issuer when creating an E-Mail claim. SERAM OpenID is: `[{ "name": "E-Mail", "issuer": "https://openid.seram.ch/openid/provider", "type": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress" }]`
Security:unmanagedDomainUserCreationNotifyEmail	String	(null)	Can be set to an e-mail address to be notified when a user self-registers without being assigned to a managed domain.

Tenant Settings

Key	Type	Default	Description
Company Name	String	(undefined)	The name of the company. Note the space in the key name.
DefaultLanguage	String	en	The language to be used by default (e.g. the code of the neutral language)
CommonSnippets	JSON	["CO_2", "²", "³", "∑"]	
Languages	JSON	(undefined)	A JSON array of objects with id and name, like this: `[{"id":"de","name":"German"},{"id":"fr","name":"French"},{"id":"it","name":"Italian"}]`
SpecificCulture	String	(undefined)	When specified, used as LCID for specifying the client-side culture to use
HideGlobalUsers	Boolean	false	When true, the Tenant Users module does not show global users.
ExcelVersion	Spire.XLS ExcelVersion	Version2007	Controls the Excel file format used when doing Excel exports. Can be Version97to2003, Version2007 or Version2010
EnableActiveDirectory	Boolean	true	Enable the Active Directory functionality in the UI (query and importing of users). The server AD sync is only enabled when defined globally.
ApplicationSetValue:errorHttpStatus	String	422 Unacceptable	
ApplicationSetValue:includeIgnored	Boolean	true	When true, ignored items will be returned along in the result set.
Caching:enabled	Boolean	true	When true, values will be cached for the tenant.

Key	Type	Default	Description
			Note: Caching must be active globally for this setting to have any effect on the tenant level.
Caching:maintenanceEnabled	Boolean	true	When true, performs maintenance on the cache. **Note:** Disabling the maintenance will cause the cache to grow unlimited. This should only be used temporarily, or on tenants where no changes occur.
Caching:directory	String	%TEMP%\FasterLogs	When specified, controls the root directory for storing the cache files in the file system.
Caching:indexEntries	Integer	7'340'032	The size of the cache index. Each index entry uses approximately 9 bytes. The memory footprint of the index will be rounded up to the next power of 2, so the default index uses 64MB. .
Caching:memorySizeMB	Integer	64	Size of the memory part of the cache. The given value is rounded up to the next power of 2.
Caching:maxSizeMB	Integer	1024	The maximum size of the cache including on-disk cache. Default is 1GB. **Note:** The cache performs garbage collection and consolidation every few minutes, so the size used on disk may temporarily exceed the given setting in rare situations.
Caching:consolidateExistingEntries	Boolean	false	When performing cache maintenance, move (upsert) all valid entries to the head of the LRU cache in order to prevent discarded entries from being kept in the cache. Enabling this option will generate much more IO and cache log activity when performing maintenance.
DataEntry:allowReadOnlyAccess	Boolean	false	When true, users which have read-only access may use the Data Entry module as well.
DataEntry:showTagCode	Boolean	true	When false, only the name of the tag is shown, no code
DataEntry:maxDirectAccessStructureCount	Integer	5	How many structures a user can be assigned for a Data Entry role before the system refuses to compute a list of pending indicators.
DataEntry:pendingItemTemplate	Metadata array (or single string)	{3}: {1} indicators on {2} in {4} ({0} values)	Format args:0: Open value count1: Per tag: Tag-Name / All tags: Open status name2: Structure Name3: "Past" or "Current"4: Time-Index-Name
DataEntry:showPendingIndicatorsPerTag	Boolean	true	Controls how the list of pending indicators is build for the Pending Data Entry view - one entry per site or one entry per site&tag
DataEntry:skipTagSelectionStep	Boolean	false	Allow skipping the tag selection in the Data Entry "wizard"

Key	Type	Default	Description
DataEntry:skipUniqueSelectionSteps	Boolean	true	When doing Data Entry, the user may encounter some steps where the selection is unique (e.g. exactly one item). When this setting is true, the user will not be asked and the selection process just skips to the next step.
DataAnalysis:allowSetToZero	Boolean	false	If true, a batch operation for setting empty values is offered in Data Analysis
DataAnalysis:alwaysAllowDrilldown	Boolean	false	If true, always allows offerts the Drill-Down action, even when a user does not have the Value Analytics permission.
DataAnalysis:autoRefresh	Boolean	true	When set to "false", editing values does not cause a forced reload of the Data Grid
DataAnalysis:boldNonPersistentValueCells	Boolean	false	When true, non-persistent (e.g. calculated) values are displayed in a bold font
DataAnalysis:maxRestatementCount	Integer	2000	Allows to change the max number of values returned when performing an restatement analysis
DataAnalysis:maxValueCount	Integer	100000	Allows to change the max number of values (as in cells) in Data Grid which are computed by the system
DataAnalysis:onlyYearlyTimes	Boolean	false	Only allow to pick from yearly values in the times selection of Data Grid
ExcelDataGrid:compactUnit	Boolean	true	When true, the unit is formatted into the number cell in the Excel Data Grid instead of using a separate column.
ExcelDataGrid:includeEmptyComments	Boolean	false	List all values in the comments
ExcelDataGrid:useNumberText	Boolean	true	If true, exports the textual representation of numbers (e.g. for Choice indicators)
ExcelInputValues:inputFrequencyComments	Boolean	false	If true, the import/export handles comments on the input frequency instead of the reporting frequency (concatenation upon import)
ExcelIndicatorAssignments:transpose	Boolean	false	If true, lists indicators as rows (instead of structures)
ExcelIndicatorAssignments:codeAndName	Boolean	true	If true, shows both name and code in the indicator headings (when transposed)
FileStorage:directory	String		Base directory where files are to be stored (instead of a database table).  **Note:** Multiple directories are also allowed, separated using a | (pipe) character.
FileStorage:inDatabase	Boolean	true	When true, files are stored in a database table even if a file storage directory has been specified.
Layout:displayActionsOnTop	Boolean	false	Display the uncategorized actions (those shown as buttons on the bottom of the screen) in the top portion of the screen as well, which may be more accessible when dealing with large lists.

Key	Type	Default	Description
Layout:displaySwissMadeLogo	Boolean	true	Display the Swiss Made Software logo.
Layout:displayVersion	Boolean	true	Display version information in footer for logged-on users
Indicators:computeValueFractions	Boolean	true	Controls whether fractions are computed for sum indicators on higher frequencies (e.g. a quarter value is divided by 3 to create month values).
Indicators:enableDataQuality	Boolean	false	Controls whether indicators use the Data Quality
Indicators:enableForecasted	Boolean	true	Controls whether the indicators provide a forecasted flag
Indicators:statusOfLockedMethodFactor	Guid	(empty)	Controls the Data Status to attribute to locked method factor values.
Indicators:statusOfUnlockedMethodFactor	Guid	(empty)	Controls the Data Status to attribute to unlocked method factor values.
RapidAPI:concurrency	Number	1	The number of concurrent requests to be sent to the API. The amount of allowed requests/second is dependent on the API subscription.
RapidAPI:host	String	meteostat.p.rapidapi.com	API host, default should be correct in most cases.
RapidAPI:key	String	(empty)	The API application key to use. Please create a dedicated application in the API console to obtain a key.  **Important:** The `meteoData()` formula function will not return any result if no API key has been configured.
RapidAPI:timeoutSeconds	Number	30	The time amount of time the system waits for starting the call if the calls are congested (see also concurrency setting).
Security:administatorsSeeAllSharedDataEntryForms	Boolean	true	When false, tenant-wide users with the Administrator role also only see their own Shared Data Entry Forms.
Security:requireAccessOnLinkedStructures	Boolean	false	Controls behavior related to "site filtering". When disabled (default), users do not need access on all linked structures; when enabled, the users need access to linked structures to see one structure.
Style:applicationName	String	SERAM	The name shown in the upper left corner
Style:chartColors	JSON array	(rainbow)	Used as colors for the slices, lines and bars in charts.
Style:copyrightHtml	HTML		Additional copyright notice, can contain a link, image etc.
Style:barChartCustomization	JSON object	(empty)	Properties to apply to the *amCharts serial chart configuration object*
Style:lineChartCustomization	JSON object	(empty)	Properties to apply to the *amCharts serial chart configuration object*

Key	Type	Default	Description
Style:pieChartCustomization	JSON object	(empty)	Properties to apply to the *amCharts pie chart configuration object*
Style:customCSS	CSS	(empty)	The CSS to be injected into the rendered page. `.data-analysis .grid .cell { width: 22em !important; } .data-analysis .grid .cell .partialwidth { width: 65% !important; }`
Style:customerLogo	*CSS URL*	url(seram.png)	Sets a variable used in the stylesheet for the centered background logo
Style:lineBreak	*CSS white-space*	normal	Can be set to nowrap to avoid line breaks in the DataGrid components (tables)
Style:userCustomerLogo	Boolean	false	If true, the customerLogo as above can be stored as part of the customer metadata.
Style:fieldsetColor	*CSS Color*	(gradient)	When specified, the light cyan gradient is replaced by the given CSS color for all fieldsets.
Style:selectedColor	*CSS Color*	#9fd987	Sets the color of active selections
Style:seramDark	*CSS Color*	#005020	Sets the color to be used for the dark elements (heading, footer, some borders)
Value:commentPlaceholder	Metadata array (or single string)	(empty)	The placeholder attribute text to use for comment fields. Example: `[{ "@xml:lang": "fr", "#text": "Commentaire / Source des données" }, { "@xml:lang": "en", "#text": "Comment / Data origin" }, "Kommentar / Ursprung Daten"]`
Value:commentSchema	Schema array	(empty)	The schema to be used for storing comments as JSON objects. The placeholder can be a metadata array (like commentPlaceholder). Example: `[{ "key": "details", "label": "Details of Calculation", "placeholder": "Details of calculation" }, { "key": "source", "label": "Data Source & Responsibility", "placeholder": "Data source and responsible person" }, { "key": "period", "label": "Time Period",`

Key	Type	Default	Description
			<pre> "singleline": true, "placeholder": "Time period (e.g. 1.1.-31.12.)" }]</pre>

Value:dataQualities	JSON object with numeric keys and metadata values		The data qualities selections. Example:
			<pre>{
			 "20":"Estimated",
			 "60":[
			  {
			  "@xml:lang":"fr",
			  "#text":"Calculé"
			  },
			  {
			  "@xml:lang":"de",
			  "#text":"Berechnet"
			  },
			  "Calculated"
			],
			 "100":"Measured"
			}</pre>

Value:publishedColor	*CSS Color*	#0099ff	The color used in Data Grid for showing published values
Licensing:dashboard	Boolean	true	Controls whether roles with the tenant-wide "Tenant:ViewLicenseUsage" permission are presented with a license summary on the dashboard.
Licensing:freeDataPoints	Integer	0	Number of free data point allowance per month.