LdapLogin_urls

Configure the LDAP endpoint. The URL must start with `ldap://` or `ldaps://`

In case of `ldaps://` you may have to import the certificate of the CA which has issued the certificate of the LDAP server into the `Backend Trust Store` on the `nevisAuth Instance`.

GenericDeployment_path

Absolute path of a directory on the target host(s) where the files will be deployed to. The command will run from the same path.

nevisAppliance targets only: if the files must be persisted across reboots, use a file name or path listed in the `/etc/rwdisk.conf` file on the nevisAppliance target host.

The path must not point into a directory (potentially) managed by a nevisAdmin 4 Instance Pattern. Thus, it is not possible to directly overwrite files generated by other patterns. See `Command` and `Command: Execution File Triggers` for an alternative solution to overcome this limitation.

Allowed Paths:

- `/tmp/generic-deployment`
- `/var/opt/<directory>`

Example:

- `/tmp/generic-deployment/patch01/`

GenericNevisFIDOSettings_nevisFidoYml

This setting provides a low-level way to add or overwrite configuration in `nevisfido.yml`.

Enter the configuration as it would appear in the `nevisfido.yml` using correct indentation.

Example:

```
fido-uaf:
  dispatchers:
  - type: png-qr-code
    registration-redeem-url: http://localhost:9080/nevisfido/token/redeem/registration
    authentication-redeem-url: http://localhost:9080/nevisfido/token/redeem/authentication
    deregistration-redeem-url: http://localhost:9080/nevisfido/token/redeem/deregistration
```

OutOfBandMobileDeviceRegistration_nevisfido

Assign a nevisFIDO instance.

This instance will be responsible for providing the device registration services.

PropertiesTestPattern_textProperty

Enter a text block.

OAuth2AuthorizationServer_consentScreen

Select `enabled` if you want to ask the user to grant consents for scopes.

Which scopes require consent can be configured in nevisMeta.

Select `disabled` if you don't have any scopes that require consents, or if you have to do custom consent handling.

AppleLogin_clientExtId

The ExtId of the client in nevisIDM that will be used to store the user

NevisLogrendConnector_url

Enter `hostname:port` of the nevisLogrend instance.

NevisFIDOLogSettings_serverSyslogFormat

Log4j 2 log format for the SERVER SYS logs.

Note: not relevant when Log Targets is set to `default`.

NevisAdaptDeployable_proxyHost

Enter the host for the forward proxy if available.

NevisAuthRealmBase_authParams

Add custom `init-param` elements to **each** `IdentityCreationFilter` generated by this pattern.

Most realms generate only 1 `IdentityCreationFilter` named `Authentication_<name>`, which is used to protect the application.

Multi-line values, as required for conditional configuration, can be entered by replacing the line-breaks with `\n`.

Examples:

Key	Value	
BodyReadSize	64000	
InterceptionRedirect	Condition:ENV:HTTP_USER_AGENT:mozilla\	Mozilla\ninitial\nnever
ClientCert	want	

NevisProxyDatabase_parameters

Enter parameters for the DB connection string.

Enter 1 parameter per line.

Lines will be joined with `&`.

The default for MariaDB:

```
ping_timeout=2
connect_timeout=10
```

and for PostgreSQl:

```
connect_timeout=10
```

The default value will be used **only** when no parameters are entered.

If you want to keep the default parameters, add them as well.

SwissPhoneChannel_serverAddress

The address of the server hosting the SwissPhone SMS Gateway.

NevisDetectAdminDeployable_port

Enter the port on which nevisDetect Admin service will listen.

SendgridChannel_key

API key to connect to Sendgrid.

MultipleFieldUserInput_title

Enter a text or *litdict key* for the form title (`<h1>`).

FIDO2Onboarding_onSuccess

Assign an authentication step to continue with after successful FIDO2 onboarding.

GenericSocialLogin_responseMode

The mode used for the responses of the server. It can be either `Query` or `Form POST` . The default value is `Query` .

NevisAdaptDeployable_ipReputationCron

Pick the update frequency of the IP reputation database.

Valid values:

- `disabled` - no update mechanism will be triggered. Not recommended for productive environment.
- `hourly`
- `daily`
- `weekly`
- `monthly`

When selecting 'disabled', it's highly recommended having a custom mechanism in place for keeping the database file up-to-date. We recommend setting up periodic update of IP geolocation and reputation mappings.

SamlIdp_issuer

Configure the `Issuer` used by this IDP.

The issuer can be an arbitrary String but it is a common practise to use the URL of the IDP.

Example: `https://idp.example.org/SAML2/`

OutOfBandMobileStepBase_trustStore

The trust store used to establish a connection with the nevisFIDO component.

The trust store must contain the certificate of the CA that has issued the certificated contained in the `Key Store` of the `nevisFIDO UAF Instance`.

In case both sides use automatic key management, trust can be established automatically and there is nothing to configure.

NevisAuthDeployable_frontendTrustStore

Assign the Trust Store provider for the HTTPs endpoint. If no pattern is assigned the Trust Store will be provided by the nevisAdmin 4 PKI.

GenericHostContextSettings_servlets

Configure `servlet` and/or `servlet-mapping` elements using the XML constructs described in the nevisProxy Technical Documentation.

You can also customize elements which have been generated by other patterns. Elements can be referenced as follows:

- `servlet : servlet-name`
- `servlet-mapping : url-pattern`

In Kubernetes side-by-side deployment a postfix is added to service names. Use the expression `${service.postfix}` connecting to a service deployed against the same inventory.

Example 1: Add or overwrite an `init-param` for an existing `servlet`:

```
<servlet>
  <servlet-name>Hosting_Default</servlet-name>
  <init-param>
    <param-name>NoMatchFile</param-name>
    <param-value>/index.html</param-value>
  </init-param>
</servlet>
```

Example 2: Remove a `servlet-mapping` :

```
<servlet-mapping>
    <url-pattern>/app/*</url-pattern>
</servlet-mapping>
```

Here we left out the `servlet-name` to tell the pattern to remove the `servlet-mapping` for the given `url-pattern` .

Note that the mapping of the hosted resources is an exception and cannot be removed this way (see the property `Hosted resources` of the Virtual Host pattern for more information).

Removing a `servlet` element is not supported.

CustomAuthLogFile_hide

Enter variables that should be hidden in logs.

The special option `auto` hides the following variables:

- variables used for GUI elements of type `pw-text`
- `dyncert.key`
- `connection.HttpHeader.Authorization`
- `client_secret`

The wildcard `*` may be appended (but not prepended). This way, every variable that starts with a certain string will be hidden from the log. For instance, `passw*` will hide `password` and `passwd` .

OAuth2UserInfo_signer

Configure the key material which is used to validate tokens. This signer must be the same signer that use to sign the tokens.

SamlSpConnector_subjectFormat

Set the `format` of the `NameID` element.

Examples:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
```

AppleLogin_clientId

ClientID is `Identifier` provided by Apple when you register Apple as IdP service.

NevisAdaptDatabase_oracleOwnerRoleName

Name of the owner role for the oracle database used for the Kubernetes migration. It's recommended to keep the default value unless the pattern is used with an existing database that has a different one.

AuthStatePatch_patchFile

Upload an XML file containing `AuthState` patch elements.

Example to illustrate the syntax:

```
<AuthState name="Check_FIDO2_Credential" class="ch.nevis.idc.auth.EventLoggingAuthState">
    <property name="eventLogger.wrappedState" value="ch.nevis.admin.v4.plugin.fido2.patterns.FIDO2Authentication"/>
    <property name="eventLogger.eventName" value="user.password.verified"/>
    <property name="eventLogger.failed.results" value="failed,noCredential,tmpLocked,locked"/>
    <property name="eventLogger.userDto" value="${sess:ch.adnovum.nevisidm.userDto}"/>
</AuthState>
```

NevisProxyObservabilitySettings_metricsTimeout

Configures a timeout for the metrics observable callback.

AuthCloudBase_authenticationType

Choose between:

- `QR code / deep link` : renders a QR code which should be scanned or shows a deep link
- `push / deep link` : sends a push notification to the user which tells them to check the access app or shows a deep link.

The first option is used for non-mobile browsers. The `deep link` is shown when using a browser on a mobile.

NevisIDMWebApplicationAccess_apiAccess

Enables REST API access for the NevisIDM web application. As of 2022 May it is only needed by the Terms & Conditions functionality. If Terms & Conditions is not used, then this can be disabled safely.

- `enabled` - the REST API will be exposed on the path `/nevisidm/api/*` .
- `disabled` - access to the path `/nevisidm/api/*` will be blocked.

If the REST API is enabled here, then the use of the `nevisIDM REST Service` pattern is not needed.

WARNING: if the `nevisIDM REST Service` pattern is also used, and has different realms or SecToken patterns assigned, then the configuration may lead to a requirement clash or a similar issue

NevisAuthRealmBase_authHostCheck

Enable to verify that the hostname on the certificate presented by nevisAuth matches the configured hostname in the `nevisAuth Instance` or `nevisAuth Connector` pattern.

GenericThirdPartyRealm_initialSessionTimeout

Define the idle timeout of the initial session. The user must complete the authentication within this time.

NevisIDMUserLookup_onUserNotFound

Assign a step to execute in the following error cases:

- User not found (`1`)
- User archived or disabled (`98`)

The variable `lasterror` is **not** cleared from the `notes` and thus an error message may be displayed in the next GUI which is rendered by nevisAuth.

This setting does **not** apply to technical errors. In case the call to nevisIDM fails the GUI will be shown (again) and the the message `error_99` will be displayed.

4.16.0

Full changelog:

Patterns 4.16.0 Release Notes - 2022-08-17

GUI Naming

The name of several `Gui` elements has been adapted. If you have a `Login Template` that expects certain names, you may have to adapt your `*.vm` and `*.js` files:

- `ConsentDialog -> oauth_consent`
- `cloud_mobile_auth -> authcloud`
- `oobloginform -> mauth`

Out-of-band Mobile Authentication

How this authentication step determines the userid has been changed. Check the release notes for details.

SAML IDP Dispatching

In previous versions the `SAML IDP` did not use the `Fallback Session Upgrade Flow` of `Authentication Realm`.

This has been changed so that a flow can be executed when there already is an authenticated session. Further, the `Fallback Session Upgrade Flow` has been renamed to `Default Session Upgrade Flow`.

The `SAML IDP` will now always dispatch into the `Default Session Upgrade Flow` when an authenticated session is found and none of the `Session Upgrade Flows` are applicable.

Note that `Session Upgrade Flows` will only be applied when their `Authentication Level` is required. This can be achieved by either:

- assigning an `Authorization Policy` requesting an `Authentication Level` to applications protected by `SAML SP Realm`,

- declaring a `Minimum Required Authentication Level` in `SAML SP Connector` patterns.

SAML Issuer and Audience Restriction

Commas and whitespaces are not allowed for SAML `Issuer` and `Audience Restriction`. If you need multiple `Audience` elements, enter multiple lines.

OAuth2Client_oidc

If enabled the scope openid is allowed for this client.

NevisDetectAdminWebApplicationAccess_admin

Reference for the pattern with the details of the web application.

Supported patterns:

- nevisDetect Admin Instance

FrontendKerberosLogin_onFailure

Assign authentication step that is processed if Kerberos authentication fails.

If no step is assigned an AuthState `Authentication_Failed` will be created automatically.

GenericSocialLogin_buttonLabel

The text that should be displayed for the end-user on the social login button, and provide translations for this label on the Authentication Realms.

SamlToken_keystore

Assign a pattern which sets the key material used for signing the token.

If no pattern is assigned automatic key management is used and the signer key will be created automatically.

NevisDetectRiskPluginBase_proxy

Outbound proxy, optional

Dispatcher_steps

Assign the steps to be used for `Transition(s)`.

SamlSpRealm_preProcess

Assign a step to apply custom pre-processing logic before executing SP-initiated SAML authentication. This pre-processing logic is executed for methods: `authenticate`, `stepup`, `unlock`, and `logout`.

You may assign a chain of steps to build a flow. The dispatching will continue when leaving this flow on the happy path.

For `On Success` exits this works automatically.

However, generic exits (i.e. `Additional Follow-up Steps` in `Generic Authentication Step`) must be marked a *success exits* by assigning the `Pre-Processing Done` pattern.

MobileTAN_onFailure

Assign the step to execute in case no mTAN can be sent or all attempts had been exhausted.

The step will be executed in the following cases:

- there is no session variable (`user.mobile` or `sess:ch.nevis.idm.User.mobile`) which contains the mobile number of the user
- the mobile number cannot be converted into a format supported by the `Connection Provider`
- all attempts had been exhausted and the user has failed to authenticate

If no step is assigned then the authentication flow will be terminated and an error GUI with label `error_99` (`System Problems`) will be shown.

TCPSettings_keepAlive

Pool TCP connections to backends for later reuse.

- `default` : does not generate any configuration so the default nevisProxy behaviour will apply.
- `disabled` : the TCP connection is closed after use, and a new connection will be established for the next request.
- `enabled` : the TCP connection is put in a pool so that it can be reused by future requests. Limiting factors are `Connection Pool Size` , `By Client` , `Inactive Interval` , and `Lifetime` .

AuthenticationFailed_code

Enter a status code for error page produced by nevisAuth. If not set the status code will be `200` .

Note that the error page from nevisAuth will not be shown, when error handling is applied by nevisProxy.

nevisProxy replaces the body of the HTTP response, when there is a page for this status code, uploaded to `Hosted Resources` of the `Virtual Host` , or to a `HTTP Error Handling` pattern.

SocialLoginBase_onSuccess

The step executed after a successful authentication. If no step is configured here the process ends with `AUTH_DONE`.

In case you change this to your custom step(s), you can assign pattern `Social Login Final Step` as the last step of the Authentication process to redirect back to original URL.

HostContext_cache

Add a Static Content Cache pattern to the Virtual Host.

Use it to cache the early hint resources as static content in nevisProxy to further increase the performance. Map the Static Content Cache pattern to the same paths as the Early Hints parameter.

RequestValidationSettings_rules

Use to **add**, **modify**, or **remove** ModSecurity rules.

Use the *Rule recommender* to white-list requests. Click the link to open the dialog, then paste log snippets from the nevisProxy `navajo.log` for requests which have been blocked by ModSecurity.

The log statement must contain the trace group `IW4ModsecF` and at least the `id` of the ModSecurity rule which has blocked the request. Example:

```
2020-07-21 13:00... IW4ModsecF ... Matched "Operator `Rx' with parameter ... against variable `REQUEST_BODY' ... [id "930100"]
```

The recommender will propose *ModSecurity modifications* to prevent these requests from being blocked in the future. The modifications will be as specific as possible, including the path, as well as parameters from the request. Please review the recommended modifications and adapt as

required.

You may also enter your own rules or modifications directly, skipping the recommender dialog. Check the ModSecurity documentation for further information on how to modify rules.

Both *exception modifications* or *whitelist modifications* are allowed in this box. The pattern ensures that the statements are included into the correct place in the generated ModSecurity configuration.

New ModSecurity rules require a *rule ID* which has to be unique within this pattern and must not used in the rule set. According to ModSecurity documentation the range `1–99999` is reserved for local (internal) use. The rule recommender will use the range `10001–10999`.

MultipleFieldUserInput_fields

List to contain `Custom Input Field` s and `Email Input Field` s, to retrieve information from the user.

CustomAuthLogFile_eventsLogFormat

Log4j 2 log format for the EVENTS logs.

Note: not relevant when Log Targets is set to `syslog`.

NevisIDMPasswordLogin_level

Set an authentication level if authentication of this step is successful. The level is relevant only if there are is an `Authorization Policy` assigned to applications.

NevisAdaptRestServiceAccess_backendHostnameCheck

Enable to verify that the hostname on the certificate presented by the backend matches the hostname of `nevisAdapt Instance`

GenericThirdPartyRealm_addons

Assign add-on patterns to customize the behaviour of applications protected by this realm.

A common case for redirect-based authentication is to assign a `Cookie Customization` here and to `Authentication Application Settings` to share cookies between applications and the authentication application.

AzureServiceBus_truststore

Assign a trust store which provides the `Microsoft Azure TLS Issuing CA 01` certificate.

You can access the `Host name` with your browser by adding `https://` in front, download the CA certificate, and then use a `PEM Trust Store` to provide it.

NevisAdaptServiceAccessBase_csrf

Cross-Site Request Forgery (*CSRF*) is an attack to force an authenticated user to send unwanted requests.

- `off (default)` - no CSRF protection. Recommended for applications which may be called from other sites.
- `header-based` - `GET` and `HEAD` requests are allowed (assumption: these methods must not manipulate server-side state). For other requests the `Referer` and `Origin` headers must match the `Host` header.

DatabaseBase_type

Choose between `MariaDB` and `Oracle` and `PostgresSQL`.

We recommend to use `MariaDB` as it is supported by all Nevis components that have a database.

Note: `PostgresSQL` database is only experimental configuration.

LdapLogin_loginidField

Specifies the attribute in the LDAP directory that should match the users login-ID input.

Examples:

- `uid`
- `cn`

HostContext_http2

Enables the support of HTTP/2 for incoming connections on this nevisProxy virtual host.

Note that mod_qos has limited support for HTTP/2, therefore only request level directives are supported if enabled.

LuaPattern_libraries

Upload addtional Lua libraries to be used within the `Lua Script` .

Uploaded files will be deployed to the following directory:

```
/var/opt/nevisproxy/${instance}/${host}/WEB-INF/lib/${name}/"
```

The Lua script **must** patch `package.path` so that the Lua libraries can be used.

For instance, add the following line at the beginning of the script:

```
package.path = package.path .. ";/var/opt/nevisproxy/${instance}/${host}/WEB-INF/lib/${name}/?.lua"
```

SamlSpIntegration_acsPath

Enter a sub-path of the application to sent the POST request to.

The POST request is sent by a DelegationFilter mapped in phase `AFTER_AUTHORIZATION`.

ErrorHandler_path

By default, the error pages are deployed to `/errorpages/<name>` but you can set a different location here.

GenericNevisProxySettings_instanceSettings

Customize the Navajo servlet container configuration (`navajo.xml`) using XML constructs described in the nevisProxy Technical Documentation.

The root element `<Service>` must be provided.

Examples:

Increase number of parallel requests (worker threads):

```
<Service>
    <Server MaxClients="1000"/>
```

```
</Service>
```

Increase the maximum allowed request body size:

```
<Service>
    <Server LimitRequestBody="10485760"/>
</Service>
```

Set a `Context` attribute for `some.domain.com`:

```
<Service>
    <Engine>
        <Host name="some.domain.com">
            <Context additionalStatusCodes="207,210,242,422,423,424,449,456,540,541,543,544,545,456,549,552,560" />
        </Host>
    </Engine>
</Service>
```

Overrule the allowed HTTP methods for `some.domain.com`:

```
<Service>
    <Engine>
        <Host name="some.domain.com">
            <Context allowedMethods="ALL-HTTP" />
        </Host>
    </Engine>
</Service>
```

Overrule the server aliases for `some.domain.com`:

```
<Service>
    <Connector name="some.domain.com" port="*" serverAlias="*.domain.com">
    </Connector>
</Service>
```

It is possible to use the following placeholders:

- `${instance.id}` : unique ID of the `nevisProxy Instance` pattern
- `${instance.name}` : name of the nevisProxy instance. For instance, use `/var/opt/nevisproxy/${instance.name}` to refer to the instance directory.

Limitations:

- customizing `Navajo` elements is not supported
- customizing `Host` (or its child elements) requires `name`

SamlSpConnector_sls

Enter the *Single Logout Service URL* of the SP.

If omitted the Assertion Consumer Service URL is used.

PermissionFilter_onSuccess

Assign the next authentication step (optional).

NevisFIDO2Database_encryption

Enables SSL/TLS in a specific mode. The following values are supported:

- `disabled` : Do not use SSL/TLS (default)
- `trust` : Only use SSL/TLS for encryption. Do not perform certificate or hostname verification. This mode is not safe for production applications but still safer than `disabled` .
- `verify-ca` : Use SSL/TLS for encryption and perform certificates verification, but do not perform hostname verification.
- `verify-full` : Use SSL/TLS for encryption, certificate verification, and hostname verification.

OAuth2PAREndpoint_requestTimeout

Configure how the PAR request shall be valid.

For security reasons, we suggest to keep this duration as low as possible.

If not set, the default in the nevisAuth component (90s) applies.

NevisFIDODeployable_displayNameSource

Defines the attribute of the user that will be populated into the `user.name` property in the PublicKeyCredentialCreationOptions object that nevisFIDO sends to the FIDO2 client during the Registration ceremony. Some browsers choose this `user.name` property to display to the user when they prompt for user interaction (as opposed to `user.displayName`). Supported values are `loginId` , `displayName` , `email` and `username` - this latter does not correspond strictly to a nevisIDM user property, but instead is the same `username` what nevisFIDO received in the `ServerPublicKeyCredentialCreationOptionsRequest` object.

The default is `loginId` .

HostContext_rules

Upload a `.zip` file containing configuration for ModSecurity. The `.zip` must contain a configuration file called `modsecurity.conf`.

The `modsecurity.conf` file will be included for all `Web Application` patterns which have `Request Validation` set to `standard`, `custom`, or `log only`.

Click `Download Default Configuration` to download the default configuration which is applied when no `.zip` is uploaded. There is one link per provided `OWASP ModSecurity CRS Version`.

NevisAdaptAuthenticationConnectorStep_onHighRisk

Will be considered only if `Profile` is set to either `balanced`, `strict` or `custom`.

Set the step to continue with if the calculated risk score exceeds the High threshold.

In case it remains unset:

1. `On Medium Risk` becomes mandatory
2. Applies the same next step as `On Medium Risk`

NevisIDMDeployable_addons

Assign add-on patterns to customize the configuration of nevisIDM.

NevisAuthRealmBase_logrend

Assign a pattern which defines the login renderer.

In case no pattern is assigned a nevisLogrend instance named `default` will be created and deployed on the same host as nevisProxy.

GroovyScriptStep_onFailure

Assign an authentication step which shall be executed when the Groovy script sets the result `error`.

```
response.setResult('error')
```

If no step is assigned a default state will be added.

NevisAdaptObservationCleanupConfig_cleanupPeriodDays

This value indicates the buffer time beyond the base observation timeframe for removing trusted observations.

The default value is `1d`.

JWTToken_audience

The audience (`aud`) is an optional claim which may be checked by applications receiving this token.

OutOfBandMobileStepBase_policy

Enter the name of a policy provided by the assigned `nevisFIDO` instance.

Read the help of the `Policies` settings in the `nevisFIDO UAF Instance` pattern for details.

By default, no policy name is set here and thus the policy `default` will be used.

You can also enter a nevisAuth or EL expression to determine the policy based on the request or the user session.

Button_buttonValue

Enter a `value` to use for the `GuiElem`.

Configure only when you need a different value.

DummyLogin_onSuccess

Set the step to continue with on successful authentication. If no step is assigned, the process ends and the user will be authenticated.

JSONResponse_parameters

Define *Parameters* to be used in the `JSON Response`.

Examples:

```
backend-host: backend.siven.ch
```

The expression formats are:

`${param.<name>}`:

- `name` found: parameter value is used.
- `name` missing: expression is **not** replaced.

`${param.<name>:<default value>}`:

- `name` found: parameter value is used.

- `name` missing: default value will be used.

In `<default value>` the character `}` must be escaped as `\}`.

CustomNevisIDMLogFile_maxBackupIndex

Maximum number of backup files to keep in addition to the current log file.

This setting applies to `application.log` and `batch.log` only. The `audit.log` is rotated on a daily basis.

NevisAuthRadiusResponse_type

The Radius message type.

For instance, use `Access-Challenge` to prompt the user for input.

GenericNevisFIDOSettings_javaOpts

Add additional entries to the `JAVA_OPTS` environment variable.

Use the expression `${instance}` for the instance name.

For instance, you may configure nevisFIDO to create a heap dump on out of memory as follows:

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/var/opt/nevisfido/${instance}/log/
```

Be aware that this example will not work for Kubernetes as the pod will be automatically restarted on out of memory and the created heap dump files will be lost.

NevisIDMDatabase_oracleIndexTablespaceName

Name of the index tablespace for the oracle database. It's recommended to keep the default value unless the pattern is used with an existing database that has a different one.

SamlSpConnector_logoutMode

Configure the logout mode when a logout is initiated by or for this SP. Choose between:

- **ConcurrentLogout-Redirect**: IdP will send logout to all SP(s) at once.
- **SingleLogout**: IdP will send logout to 1 SP at a time.
- **SingleLogout-SOAP**: IdP will send SOAP logout to SP(s) one by one using SOAP method.

Maintenance_page

The page must contain two meta-tags which define the maintenance interval and will be patched during generation.

Example:

```
<head>
  <meta name="maintenance-start" content="${maintenance-start-value}">
  <meta name="maintenance-end" content="${maintenance-end-value}">
</head>
```

If the date and time on the target host are within this interval, the maintenance page will be shown. See also the introduction help text above.

NevisIDMDeployable_managementPort

This port is used in Kubernetes deployment to check if the instance is up after deployment.

NevisIDMUserLookup_rememberInput

Select `enabled` to add a `Remember Input` checkbox.

By ticking the checkbox the whatever has been entered by the user will be stored in a long-living cookie (named like this pattern).

Using this cookie, the login ID will be prefilled on subsequent authentications.

If no GUI is shown (e.g. to look up the user based on `Login ID Source`) you **must** select `disabled`.

NevisAuthRealmBase_maxSessionLifetime

Define the maximum lifetime of an authenticated session. The session will be removed after that time even if active.

NevisAdaptAnalyzerConfig_sharedDeviceAnalyzer

Used to disable the shared device analyzer. This means that the shared device analyzer will not be used to calculate risk scores.

SocialLoginBase_onFailure

The step that will be executed if the authentication fails. If no step is configured here the process ends with `AUTH_ERROR`.

In case you change this to your custom step(s), you can assign pattern `Social Login Final Failure Step` as the last step of the Authentication process to redirect back to original URL.

FIDO2Onboarding_welcomeScreenButton

Configure to add a dispatcher button to the welcome screen.

The button may have a special `Button Name` to render in a nice way by a customized `Login Template`.

For instance, Identity Cloud uses this mechanism to add a button which looks like a back arrow. This button takes the user to a previous step.

This is an advanced setting. Use only when you understand the concept.

NevisAuthDeployable_addons

Assign an add-on pattern to customize the configuration of nevisAuth.

NevisIDMAdvancedSettings_properties

Add properties for `nevisidm-prod.properties`. See nevisIDM Reference Guide (chapter Configuration files) for details.

AuthCloudOnboard_onSuccess

Assign a step to execute after successful onboarding.

If no step is configured, the flow ends and an authenticated session will be established.

This requires that the session contains an authenticated user.

A simple way to ensure that is to include `nevisIDM User Lookup` or `nevisIDM Password Login` steps in your flow.

NevisIDMChangePassword_currentPassword

Mandatory input value to use for old password if `Show GUI` is `disabled` and `Re-enter old Password` is `enabled`.

SamlSpIntegration_relayState

Enter a static value, or a nevisProxy expression, which defines the value of the POST parameter `RelayState` that shall be sent to the SP together with the SAML `Response`.

Whether a `RelayState` is required depends on the SP. Many SPs expect a URL and will redirect to this URL once the SAML `Response` has been successfully validated.

RealmBase_sessionTracking

Choose between:

- `COOKIE` : issue a session cookie.
- `AUTHORIZATION_HEADER` : track the session based on the value of the Authorization header.
- `CUSTOM` : track the session based on custom configuration. It generates an empty session filter which has to be replaced (see below).
- `disabled` : disable session tracking.

CUSTOM session tracking

Given a pattern name of SSO, the following empty filter will be generated:

```
    <filter>
        <filter-name>SessionHandler_SSO</filter-name>
        <filter-class>__REPLACE_USING_GENERIC__</filter-class>
    </filter>
```

For the filter-class, a placeholder (**REPLACE_USING_GENERIC**) will be used and that placeholder has to be overwritten.

Another pattern must complete the session filter. For example, use `Generic Virtual Host Context` pattern with the following Filters and Mappings configuration:

```
<filter>
    <filter-name>SessionHandler_SSO_RealmName</filter-name>
    <filter-class>ch::nevis::nevisproxy::filter::session::SessionManagementFilter</filter-class>
    <init-param>
        <param-name>Identification</param-name>
        <param-value>CUSTOM</param-value>
    </init-param>
    <init-param>
        <param-name>Custom.RequiredIdentifiers</param-name>
        <param-value>HEADER:Authorization</param-value>
    </init-param>
    <init-param>
        <param-name>Servlet</param-name>
        <param-value>LocalSessionStoreServlet</param-value>
    </init-param>
</filter>
```

OATHAuthentication_nevisIDM

Reference the nevisIDM Instance which has been used for first factor authentication.

ServiceBase_host

Assign a `Virtual Host` which shall serve as entry point.

CustomProxyLogFile_rotationCompression

Define rotated files will be compress or not

SamlIdpConnector_selector

The expression configured here will be used by nevisAuth to determine the IDP for SP-initiated SAML flows.

Configuration is required there are multiple `SAML IDP Connector` patterns assigned to the same `SAML SP Realm` .

For IDP-initiated flows the expression is not relevant as the IDP can usually be determined based on the `Issuer` contained in received SAML messages.

You may enter nevisAuth or EL expressions.

You must ensure that there is always exactly 1 expression which evaluates to `true`

If there is no match or multiple IDPs are applicable then `403 Forbidden` is returned.

Examples:

- IP of the user starts with `10.0.106` : `${request:clientAddress:^10.0.106}`
- Request path starts with `/myapp` : `${request:currentResource:(http.?.//[^/]+)/myapp.*}`

SamlSpRealm_samlSigner

Use a pattern to configure the signer certificate used by this Service Provider. If no pattern is assigned a key store will be provided automatically.

MicrosoftLogin_clientId

ClientID is `Application (client) ID` provided by Microsoft when you create an Application Microsoft.

Maintenance_enabled

Allows to easily enable / disable the maintenance with being forced to set a time window.

SharedStorageSettings_storageMountPath

The path where the volume will be mounted and used by the service.

For example: `/var/opt/shared`

For more information regarding persistent volumes in Kubernetes please visit this page

FrontendKerberosLogin_onSuccess

Configure the step to execute after successful authentication. If no step is configured here the process ends and the user will be authenticated.

TestingService_onGeneration

Use for testing only.

NevisAdaptEvent_followUpStep

Select which authentication step to continue with in case at least `Minimum Match Count` out of the selection provided in `Risk Events` are present in the report coming from the nevisAdapt service.

RESTServiceAccess_csrf

Cross-Site Request Forgery (*CSRF*) is an attack that forces an authenticated user to send unwanted requests.

- `off (default)` - no CSRF protection. Recommended for APIs which may be called from other sites.
- `header-based` - `GET` and `HEAD` requests are allowed. For other requests `Referer` and `Origin` headers must match the `Host` header.

SocialLoginCreateUser_unitExtId

The ExtId of the unit in nevisIDM that will be used to store the user

NevisIDMDeployable_smtpTruststore

Assign a Trust Store provider pattern to use for setting up trust between nevisIDM and the SMTP server.

OAuth2AuthorizationServer_restEndpoints

Add extension services for OAuth 2.0 Authorization Server / OpenID Provider

TCPSettings_dnsCache

Cache DNS lookup results.

- `default` : does not generate any configuration so the default nevisProxy behaviour will apply.
- `disabled` : the configured backend host names are resolved for each request. Use when IP addresses may change.
- `enabled` : host names are resolved only once. Use when the IP addresses are stable.

SamlSpConnector_assertionLifetime

On successful authentication this IDP will issue a SAML assertion.

The SAML assertion is re-created on each session upgrade to avoid replay attacks.

The lifetime of the assertion should be low but high enough so that the authentication works on slow network connections.

The SAML assertion will be consumed by the service provider. The service provider should then use a different mechanism to track the user session (e.g. a session cookie).

NevisAdaptServiceAccessBase_token

Propagate a token to the backend application. The token informs the application about the authenticated user.

Please assign a `NEVIS SecToken` . This is mandatory to have access to the Administration UI.

SamlSpConnector_authRequestLifetime

SAML authentication requests have a maximum lifetime which may be validated by this identity provider.

Enter `unlimited` to disable the maximum lifetime check for received SAML `AuthnRequests`. This sets `in.max_age` to `-1` in the generated `IdentityProviderState`.

CustomNevisIDMLogFile_batchSyslogFormat

Log4j 2 log format for the BATCH SYS logs.

Note: not relevant when Log Targets is set to `default`.

GenericSMTPChannel_protocol

Select the protocol of the SMTP server.

SMTPS is usually mentioned as the TLS secured SMTP protocol.

JSONResponse_json

Enter the JSON response.

NevisIDMServiceAccessBase_realm

Mandatory setting to enforce authentication.

NevisIDMCheckUserCredentials_allCredentialFound

Configure the step to execute if the user has at least one credential from all type selected in `Credential Types`. If no step is configured here the process ends with `AUTH_DONE`.

GenericSocialLogin_clientSecret

The secret of the client ID that has been set in the OAuth/OpenID Connect configuration of the social account.

SecurosysKeyStoreProvider_configFiles

The two necessary configuration files for accessing the HSM.

'primus.cfg' must contain the configuration settings for connecting to the HSM, and '.secrets.cfg' must contain the credentials to access the materials on HSM.

Keep in mind that the files are not validated, first set up a working configuration, and use the already validated files here.

KeyObject_trustStore

Reference a trust store provider pattern or leave empty to let nevisAdmin establish a trust store. This reference property is considered when type `trust store` is selected.

HeaderCustomization_subPaths

Set to apply the header customization on some sub-paths only.

Sub-paths must be relative (e.g. not starting with `/`) and will be appended to the frontend path(s) of the virtual host (`/`) or applications this pattern is assigned to.

Sub-paths ending with `/` are treated as a prefix, otherwise an exact filter-mapping will be created.

The following table illustrates the behaviour:

Frontend Path	Sub-Path	Effective Filter Mapping
`/`	`secure/`	`/secure/*`
`/`	`accounts`	`/accounts`
`/`	`api/secure/`	`/api/secure/*`
`/`	`api/accounts`	`/api/accounts`
`/app/`	`secure/`	`/app/secure/*`
`/app/`	`accounts`	`/app/accounts`
`/app/`	`api/secure/`	`/app/api/secure/*`
`/app/`	`api/accounts`	`/app/api/accounts`

SamlIdpConnector_artifactResolutionService

Configure to enable HTTP Artifact Binding.

Enter the `Location` of the `ArtifactResolutionService`. This information can usually be found in the SAML metadata provided by the IDP.

The location must be a valid URL. In case of `https://` import the CA certificate of the endpoint into the backend truststore of nevisAuth.

When a SAML artifact is returned by the IDP the service provider will send a request to the artifact resolution service to retrieve the SAML assertion.

JWTAccessRestriction_header

By default, the JWT will be extracted from the `Bearer` type `Authorization` request header:

```
Authorization: Bearer <token>
```

Optionally, this behavior can be overwritten by this property by specifying a request header from where the token should be extracted, for example if the token is sent like:

```
CustomAuthHeader: <token>
```

Then configure `CustomAuthHeader` for this property.

NevisIDMChangePassword_fail

Assign an authentication step to execute when the status of the URL ticket or credential is **failed**.

OAuth2AuthorizationServer_meta

Assign a `nevisMeta Instance` or `nevisMeta Connector` .

nevisMeta is used to lookup metadata for the given OAuth2 / OpenID Connect Setup (see `Setup ID`).

NevisFIDODeployable_relyingPartyOrigins

Enter all URLs from where FIDO 2 registration and authentication is invoked.

Example: `https://www.example.com`

nevisFIDO will use this information to check the `Origin` header of incoming REST calls.

URLs must be entered without a path.

URLs must have a common base domain which will be used as the ID for the relying party.

For Android Applications using the non-WebauthN standard compliant Origins enter the origin in the format `android:apk-key-hash:<your-apk-key-hash>` .

CustomAuthLogFile_serverLogFormat

Log4j 2 log format for the default SERVER logs.

Note: not relevant when Log Targets is set to `syslog` .

FacebookLogin_clientId

ClientID is `App ID` provided by Facebook when you register Facebook as IdP service.

NevisIDMUserLookup_userNotFoundError

When no user is found error code `1` is set.

If you flow shows another GUI after taking the `On User Not Found` exit, an error text may be displayed.

The default translation for English is: `Please check your input.`

In some flows (e.g. self-registration) this is not desired. Thus, you can select `disabled` here to remove the error code.

NevisProxyDeployable_apacheSSLCache

Configures the Apache storage type of the global/inter-process SSL Session Cache.

Uses the default high-performance cyclic buffer inside a shared memory segment in RAM.

This is the recommended and default SSL Cache for nevisProxy, which is required to enable SSL session resumption.

For more information, see the official Apache documentation about the SLLSessionCache directive.

NevisAdaptDeployable_ipReputationHostnameVerifier

Enabling this option will set an Apache hostname verifier (which also handles certificate checks) instead of the default one.

Default: `disabled` (backwards compatibility)

HeaderCustomization_responseHeadersRemove

Removes HTTP headers from responses.

The syntax is: `<header name>`

Examples:

```
X-Content-Type-Options
```

Headers set by Apache cannot be removed:

- `Server`

Note: change the `Filter Phase` to remove headers early / late.

NevisIDMAccountRecovery_onSuccess

Configure the step to execute after the user was successfully authenticated.

NevisAuthRealmBase_logrendHostCheck

Enable to verify that the hostname on the certificate presented by nevisLogRend matches the configured hostname in the `nevisLogrend Instance` or `nevisLogrend Connector` pattern.

This setting only applies if nevisLogrend is used in the `Login Renderer` setting and the connection to nevisLogrend uses HTTPs.

AutomaticKeyStoreProvider_owner

Select an instance pattern which defines the target hosts of this `Automatic Key Store` . This setting is required only when this pattern is assigned to an `Automatic Trust Store` .

NevisConnectorPattern_kubernetesNamespace

Enter the Kubernetes namespace.

Configuration is required when `Kubernetes` is set to `other_namespace` .

NevisAuthDeployable_languages

Configure the language codes that shall be supported.

Each language code must be entered on a new line. By default, translations are provided for the following codes:

- `en` : English
- `de` : German
- `fr` : French
- `it` : Italian

nevisAuth uses the `Accept-Language` header sent by the browser to determine the user language. In case this header is not available the first configured language code will be used as a default.

NevisLogrendDeployable_logging

Add logging configuration for nevisLogrend.

NevisFIDODeployable_nevisidm

For user and credential management, nevisFIDO needs nevisIDM.

Assign a `nevisIDM Instance` or `nevisIDM Connector` here.

This connection uses *Client TLS* and the trust is **not** built up automatically.

NevisLogrendLogSettings_maxFileSize

Maximum allowed file size (in bytes) before rolling over.

Suffixes "KB", "MB" and "GB" are allowed. 10KB = 10240 bytes, etc.

Note: not relevant when rotation type is `time`.

NevisIDMJmsQueues_dlq

NevisIDM JMS Queue to which Dead Letter messages should be sent.

Only accepts URIs starting with `amqp`, `amqps` or `Endpoint=sb`. Validates only URIs with `amqp` or `amqps` schemes.

Dead letter messages are those messages which are not in the expiryQueue and their delivery was unsuccessful. For further reference check `NevisIdm Technical documentation > Configuration > Components > Provisioning module > Provisioning providers`.

NevisAdaptDeployableBase_secTokenTrustStore

Assign the Trust Store provider for verifying the NEVIS SecToken. If no pattern is assigned the signer key will be provided by the nevisAdmin 4 PKI.

GenericDeployment_group

Owner of the directory at path. All files and subdirectories will have the same owner.

NevisIDMPasswordLogin_customEmailSentRedirect

Enter a URL, path, or nevisAuth expression which defines where to redirect to after the ticket has been created (and sent to the user via email).

NevisIDMCheckUserLoginInfo_userPreviouslyLoggedIn

Configure the step to execute if the user has previously logged in. If no step is configured here the process ends with `AUTH_DONE`.

CookieCustomization_clientCookies

Cookies listed here will be allowed to pass through.

Use for cookies which should be returned to the caller (e.g. browser).

Regular expressions are supported.

Example:

- `LANG.*`

NevisIDMProperty_uniquenessScope

If set then values stored in the property must be unique within the configured scope.

- `ABSOLUTE` : The property's values have to be unique overall. Two property values with the same content must not exist.

NevisFIDODeployable_deepLinkAppFiles

Upload resources required for deep links.

Installation Page

You can upload the HTML page that your `Deep Link` points to.

This page is shown only when the mobile app is not installed and should provide installation instructions.

You can also upload static resources (e.g. CSS and images) used by this page.

Uploaded files will be hosted at the root location (`/`) of the `Deep Link Host` .

If you want to host them on a sub-path, use the `Hosting Service` pattern instead.

App Link Files

App link files are JSON files which provide information about the mobile app. Apple and Google use different terms, and expect different filenames and contents.

Visit our official documentation for more information.

iOS

The file must be named `apple-app-site-assocation` **without** any extension and will be hosted at `/.well-known/apple-app-site-association` .

The file must be created manually and match the following structure:

```
{
  "applinks": {
    "details": [
      {
        "appIDs": [
          "<team id>.<bundle id>"
        ],
        "components": [
          {
            "/": "open",
            "?": {
              "dispatchTokenResponse": "*"
            },
            "caseSensitive": false
          }
```

```
            ]
          }
        ]
      }
    }
```

appID: refers to the app. It consists of two components as follows: `<TeamID>.<bundleID>` , for details or about how to obtain it, see Apple documentation about app links.

deep-link-base-path: The path configured here is the one supported in the deep links. Make sure the path used in the `Deep Link` corresponds to this value.

When the mobile app is installed or updated, iOS fetches this file from the server and stores it for later, to verify the paths in deep links the user clicks on.

For more information, visit app links.

Android

The file must be named `assetlinks.json` **with** the extension and will be hosted at `/.well-known/assetlinks.json` .

The file can be generated with the Statement List Generator using the following information:

- *Hosting site domain*: enter the domain used in the `Deep Link` . It should point to the assigned `Deep Link Host` .
- *App package name*: enter the package name of your app. If you are using a NEVIS branded Access App, that would be `ch.nevis.security.accessapp` .
- *App package fingerprint (SHA256)*: enter the fingerprint of the certificate your app has been signed with.

For more information, visit Android App Links.

Note that certain Chinese browsers do not support *Android App Links*: 360, QQ, UC.

The file must be created manually and match the following structure:

```
[
  {
    "relation": [
      "delegate_permission/common.handle_all_urls"
    ],
    "target": {
      "namespace": "android_app",
      "package_name": "<bundle id>",
      "sha256_cert_fingerprints": [
        "<certificate fingerprint>"
      ]
    }
  }
]
```

GenericAuthService_authStatesFile

Enter `AuthState` elements as XML.

The `Domain` element is optional.

- If missing the element will be created. The `Entry` methods `authenticate` and `stepup` will be set to the first provided `AuthState`. The method `logout` is not set and thus the nevisAuth default behaviour applies.
- If provided the `Domain` must come before all `AuthState` elements. The attributes `name` and `default` are not supported and should be omitted. Attributes are sorted by name. The `Entry` elements are sorted by `method`.

The `AuthState` linked to `stepup` should be able to dispatch the request. For instance, you may have assigned an `Authorization Policy` to your application(s) and thus you need a state which decides based on the request variable `requiredRoles`.

The following example dispatches level `2` into an `AuthState` named `TAN` which provides authentication via mTAN:

```
<AuthState name="EntryDispatcher" class="ch.nevis.esauth.auth.states.standard.ConditionalDispatcherState" final="false">
    <ResultCond name="nomatch" next="Authentication_Done"/>
    <ResultCond name="level2" next="TAN"/> <!-- TAN state is expected to set authLevel="2" -->
    <Response value="AUTH_ERROR">
        <Arg name="ch.nevis.isiweb4.response.status" value="403"/>
    </Response>
    <property name="condition:level2" value="${request:requiredRoles:^2.*$:true}"/>
</AuthState>
```

The following expressions are supported:

- `${instance}` : name of the nevisAuth instance
- `${request_url}` : generates a nevisAuth expression which returns the URL of the current request
- `${realm}` : name of the Realm (see below)
- `${service_url}` : generates a nevisAuth expression which evaluates to true for requests received on the configured `Frontend Path`
- `${service.postfix}` : in Kubernetes side-by-side deployment a postfix is added to service names. Use this expression when connecting to a service deployed against the same inventory.
- `${keystore}` : name of the `KeyStore` element provided by this pattern. Assign a pattern to `Key Objects` to add a `KeyObject` into this `KeyStore` .

The `name` of `AuthState` elements is prefixed with the sanitized name of the Realm (referred to as `${realm}`).

The realm prefix must be added when using `propertyRef` to reference AuthStates generated by other patterns (e.g. `<propertyRef name="${realm}_SomeState"/>`).

An exception is the AuthState which defines the nevisIDM connection (as generated by `nevisIdm Password Login` or `nevisIDM Connector for Generic Authentication`). Here the `propertyRef` must be defined as follows:

```
<propertyRef name="nevisIDM_Connector"/>
```

This pattern does not validate that labels are translated. Translations can be provided on the `Authentication Realm` pattern.

FacebookLogin_claimsRequest

The claims request parameter. This value is expected to be formatted in JSON and does not accept trailing spaces nor tabs.

NevisIDMDeployable_encryptionCipher

Encryption cipher.

NevisAdaptFeedbackConfig_feedbackKey

Enter a 256-bit encryption key represented in Base64.

To generate a new random key, you may run the following console command:

```
openssl rand -base64 32
```

Regular expression for valid values: `[a-zA-Z0-9+/]{43}=`

Example: `fq7J7E1xVFNHcEJ2MSQojLibKOQOMIlp2qXVqvv5y9w=`

NevisAdaptLogSettings_maxBackupIndex

Maximum number of backup files to keep in addition to the current log file. When `Rotation Type` is `time`, this property is used as Logback's maxHistory property. This means that logs will be archived for this number of time units where time unit is as defined in `Rotation Interval`.

NevisIDMClient_remarks

Any other additional information about the client.

EmailInputField_variable

Enter `<scope>:<name>` of the variable which shall be set.

The following scopes are supported:

- `inargs`
- `notes`
- `sess` or `session`

For instance, enter `notes:loginid` to prefill the login form which is produced by the `nevisIDM Password Login` pattern.

NevisAdaptDeployable_logging

Assign `nevisAdapt Log Settings` to change the log configuration.

NevisIDMPasswordLogin_useDefaultProfile

Should in the Authentication flow assume default profile is selected if the user has multiple profiles, or should it display a selection dialog for the user.

4.17.0

Full changelog:

Patterns 4.17.0 Release Notes - 2022-11-16

Improved key-value settings

Improved handling and display of key-value settings. Keys and values are now displayed in separate boxes.

So far, a multi-line text box was used with the following separators: `->`, `:`, `=`.

The new widget uses a structured format to store its configuration. The widget is able to import legacy configuration, but requires you to confirm the migration.

Check your project for issues and follow the instructions given in the patterns.

Some separators (usually `->`) were used the "wrong way round" in previous releases. Therefore, you may have to switch the content of the left and right boxes after you have clicked the "Migrate" button. Check the help of the setting for what is expected there.

Refactored Social Login patterns

Social login patterns had to be refactored to address a security vulnerability.

It was possible to take over another nevisIDM user by changing the email at the social login provider.

To address this issue, the social login patterns don't automatically link the user anymore.

Some exits in the patterns have to be re-configured.

The exit `On User Found` will be taken when a user was found in nevisIDM but the user is not linked yet. We recommend to assign a step to validate that possession of the nevisIDM user, e.g. by asking for the password, and then end the flow with the `Social Login Link User` pattern.

The exit `On User Not Found` will be taken when the email provided by the social login provider was not found in nevisIDM. In this case you should validate that the user has access to the email, e.g. by sending a TAN code, and then complete the flow with the `Social Login Create User` pattern.

NevisIDMCheckUserCredentials_noCredentialFound

Configure the step to execute if the user has no credential from credential types defined in `Credential Types`. If no step is configured here the process ends with `AUTH_DONE`.

OAuth2AuthorizationServer_idTokenJWKSetProxy

Forward proxy for the connection to the JWK Set endpoint for ID token encryption. Enter the hostname:port here

Example: `proxy.your-internal-domain:3128`

InBandMobileDeviceRegistration_authenticationServicePath

Configure the path of the authentication service.

AuthCloudBase_userNamePrefix

Optional prefix which will be added to the Authentication Cloud username.

WARNING: Changing this option means that all existing users will have to register their Access Apps again.

The Authentication Cloud *username* consists of the *user ID* and the optional `Username Prefix`.

The *user ID* is looked up from the following sources:

- session variable `ch.adnovum.nevisidm.user.extId`
- request field `userId`

NevisIDMDeployable_mailSMTPPort

Port of the SMTP server.

GroovyScriptStep_scriptTraceGroup

Use a different category for logging in your Groovy script.

PemKeyStoreProvider_dirName

Enter a name for the key store directory which is used instead of the pattern name.

This configuration may be used to prevent key stores overwriting each other and is only required in complex setups with multiple projects or inventories.

NevisLogrendDeployable_addons

Assign an add-on pattern to customize the configuration of nevisLogrend.

SamlResponseConsumer_path

Enter a path where SAML `Response` messages sent by an external IDP shall be consumed.

The external IDP may send messages using POST or redirect binding.

NevisIDMAccountRecovery_onFailure

Configure the step to execute after the authentication failed.

If no step is configured here the process ends.

FIDO2Authentication_onSuccess

Assign an authentication step to continue with after successful authentication.

FacebookLogin_scope

Select the request scopes for getting user information from Facebook.

The default is `email` and thus minimal information will be returned.

Select `public_profile` to return additional user information.

Scope `offline_access` is not supported as Facebook has removed this scope.

BackendServiceAccessBase_sendCertificateChain

Choose which certificates are sent to the backend during mutual authentication:

- `disabled` : Send the client certificate from the **Key Store**;
- `enabled` : Send the certificate chain from a **PEM Key Store** or a **nevisKeybox Store**. The certificate chain file must contain the client certificate and the intermediate CA certificates.

FacebookLogin_buttonLabel

Enter the text that should be displayed for the end-user on the social login button, and provide translations for this label on the Authentication Realms.

SamlIdpConnector_issuer

Enter the `Issuer` of the IDP.

Example: `https://idp.example.org/SAML2`

The `Issuer` is used to look up the trust store containing the signer certificate of the IDP.

For this purpose a `KeyObject` element will be configured in the nevisAuth `esauth4.xml` using the `Issuer` for the attribute `id`.

AzureServiceBusRemoteQueue_policy

Enter the `Policy` that shall be used to connect.

Also known as: `SAS Policy`, `Shared access policy`

GenericSocialLogin_subjectClaim

The claim that contains the subject of the logged-in user in the social account. The default value is `sub`.

NevisDetectAuthenticationConnectorStep_jmsClientTrustStore

Reference a trust store provider pattern or leave empty to manage the trust store with nevisAdmin.

PropertiesTestPattern_attachmentProperty

Upload 1 or multiple files. No support for subdirectories but some patterns unpack uploaded zip files. Should have support for in-place file edit for known file extensions.

ErrorHandler_redirectStatusCodes

Redirect to a given location **instead** of rewriting the response body.

Locations can be entered as:

- URLs (starting with `http://` or `https://`)
- paths (starting with `/`)

Internal and external locations are supported.

Examples:

```
404,500-599 -> /some/super/redirect/
403 -> https://www.google.com
```

ObservabilityBase_configuration

Configuration file of the selected agent.

Use `${...}` expressions to refer parameter values. Default parameters:

- `${name}` : component name
- `${instance}` : instance name
- `${version}` : version
- `${service.name}` : service name (kubernetes deployment)

Sample configuration for OpenTelemetry:

```
otel.service.name = ${service.name}
otel.resource.attributes = service.version=${version}
otel.exporter.otlp.protocol = http/protobuf
otel.exporter.otlp.traces.protocol = http/protobuf
otel.exporter.otlp.traces.endpoint = ${tracesEndpoint}
otel.exporter.otlp.metrics.protocol = http/protobuf
otel.exporter.otlp.metrics.endpoint = ${metricsEndpoint}
otel.exporter.otlp.metrics.temporality.preference = cumulative
otel.exporter.otlp.logs.protocol = http/protobuf
otel.exporter.otlp.logs.endpoint = ${logsEndpoint}
```

Sample configuration for Application Insights:

```
{
  "connectionString": "${connectionString}",
  "role": {
    "name": "${service.name}"
  },
  "customDimensions": {
    "service.version": "${version}"
  },
  "sampling": {
    "percentage": 100
  },
  "instrumentation": {
```

```
      "logging": {
        "level": "OFF"
      }
    }
  }
```

KerberosLogin_proxyHostNames

Enter the `Frontend Addresses` of the nevisProxy `Virtual Host` patterns for which this pattern provides authentication.

Example:

- `www.siven.ch`

In case multiple values are configured you can define which `Keytab File` or `Keytab File Path` to use by referencing its file name.

Example:

- `www.siven.ch -> kerberos_ch.keytab`
- `www.siven.de -> kerberos_de.keytab`

AuthServiceBase_addons

Assign add-on patterns to customize the behaviour of this authentication service.

Example use cases:

- `URL Handling` with phase `AFTER_AUTHENTICATION` to redirect after the authentication flow completes.
- `Access Restriction` to restrict access based on source IPs.
- `HTTP Header Customization` to add, replace, or remove HTTP headers in requests or responses.

FrontendKerberosLogin_kerberosRealms

Enter the allowed `Kerberos realms ( AD domains )`.

Example:

- `SIVEN.CH`

In case multiple values have to be configured you can define which `Keytab File` or `Keytab File Path` to use by referencing its file name.

Example:

- `SIVEN.CH -> kerberos_ch.keytab`
- `SIVEN.DE -> kerberos_de.keytab`

GenericNevisMetaSettings_javaOpts

Add additional entries to the JAVA_OPTS environment variable.

Use the expression `${instance}` for the instance name.

For instance, you may configure nevisMeta to create a heap dump on out of memory as follows:

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/var/opt/nevismeta/${instance}/log/
```

Be aware that this example will not work for Kubernetes as the pod will be automatically restarted on out of memory and the created heap dump files will be lost.

OAuth2UserInfo_idm

Assign a `nevisIDM Instance` or `nevisIDM Connector` to get user information.

GenericNevisAdaptSettings_javaOpts

Add additional entries to the JAVA_OPTS environment variable.

For instance, you may configure nevisAdapt to create a heap dump on out of memory as follows:

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/var/opt/nevisadapt/log/
```

Be aware that this example will not work for Kubernetes as the pod will be automatically restarted on out of memory and the created heap dump files will be lost.

CustomNevisMetaLogFile_maxFileSize

Maximum allowed file size (in bytes) before rolling over.

Suffixes "KB", "MB" and "GB" are allowed. 10KB = 10240 bytes, etc.

Note: not relevant when rotation type is `time` .

OutOfBandMobileDeviceRegistration_host

Assign the `Virtual Host` which serves the domain where the nevisFIDO services shall be exposed so that this pattern can generate the required configuration.

The domain is coded into the mobile app and has to be communicated when ordering the app.

The `Virtual Host` assigned here will also be considered when calculating the `Frontend Address` in the `nevisFIDO UAF Instance`.

NevisIDMDeployable_jobStore

Select `db` to track job execution in the database. This ensures that a given batch job can only run once at the same time. Use this configuration when you have multiple lines / replicas.

Select `ram` to store track job execution in memory. You may use this value when you have only 1 line / replica.

NevisAuthDeployable_sessionIndexing

Enables session indexing.

WARNING: Other patterns, such as `nevisAdapt Instance` may overrule this configuration.

This is required by ThrottleSessionsState.

Set `Session Index Attribute` if you need to index a non-default attribute.

AuthCloudBase_hashUserName

Enable to use a hash (MD5) for the Authentication Cloud username.

WARNING: Changing this option means that all existing users will have to register their Access Apps again.

There are 2 motivations for enabling this feature:

- the Authentication Cloud username is limited to 50 characters. Hashing makes it shorter.
- you avoid storing sensitive user information in the Authentication Cloud instance.

DatabaseBase_password

Password for the database connection user.

This setting is used in the following cases:

- Classic deployments (VM)
- In Kubernetes when 'Database Management' (Advanced Settings) is set to 'disabled'.

LdapLogin_baseDN

Specifies the directory subtree where all users are located.

Example:

- ou=people,o=company,c=ch

NevisAdaptDeployable_ipToLocationCron

Pick the update frequency of the IP-to-location database.

Valid values:

- `disabled` - no update mechanism will be triggered. Not recommended for productive environment.

- `hourly`
- `daily`
- `weekly`
- `monthly`

When selecting `disabled`, it's highly recommended having a mechanism in place for keeping the database file up-to-date. We recommend setting up periodic update of IP geolocation and reputation mappings.

NevisIDMURLTicketConsume_onExpired

Assign an authentication step to execute when the URL ticket is **expired**.

If not set a screen with `title.url_ticket` and `error.url_ticket.expired` will be shown in that case.

LuaPattern_subPaths

Set to apply this pattern on some sub-paths only.

Sub-paths must be relative (e.g. not starting with `/`) and will be appended to the frontend path(s) of the virtual host (`/`) or applications this pattern is assigned to.

Sub-paths ending with `/` are treated as a prefix, otherwise an exact filter-mapping will be created.

The following table provides examples to illustrate the behaviour:

Frontend Path	Sub-Path	Effective Filter Mapping
`/`	`secure/`	`/secure/*`
`/`	`accounts`	`/accounts`

Frontend Path	Sub-Path	Effective Filter Mapping
/	api/secure/	/api/secure/*
/	api/accounts	/api/accounts
/app/	secure/	/app/secure/*
/app/	accounts	/app/accounts
/app/	api/secure/	/app/api/secure/*
/app/	api/accounts	/app/api/accounts

NevisFIDODeployable_facets

Facets are required configuration for mobile authentication scenarios. The FIDO AppID and Facet Specification defines facets as *identities of a single logical application across various platforms*.

The following wildcard facet IDs are included for ease of use, to speed up integration of the Nevis Access App or Nevis Mobile Authentication SDK:

```
android:apk-key-hash:*,
ios:bundle-id:*
```

The wildcard facet ID entries are compatible with *integration* flavor Access Apps or *debug* flavor mobile SDKs. *Production* Access Apps or *release* mobile SDKs do not accept wildcard facet ID entries. If you want to use one of the Nevis Mobile Authentication SDK example applications with the *release* SDK, you will need to add one or multiple of the following facetID entries:

- `android:apk-key-hash:ch.nevis.mobile.authentication.sdk.android.example`
- `android:apk-key-hash:ch.nevis.mobile.authentication.sdk.flutter.example`

- `android:apk-key-hash:ch.nevis.mobile.authentication.sdk.react.example`
- `ios:bundle-id:ch.nevis.mobile.authentication.sdk.ios.example`
- `ios:bundle-id:ch.nevis.mobile.authentication.sdk.flutter.example`
- `ios:bundle-id:ch.nevis.mobile.authentication.sdk.objc.proxy.example`
- `ios:bundle-id:ch.nevis.mobile.authentication.sdk.react.example`

For **production deployment** you have to replace the default and add your own facets for your iOS or Android applications. The following documentation provides additional information:

- For **Access Apps**, refer to the FacetID Calculation documentation.
- For the **mobile SDK**, refer to the FacetID chapter in the configuration section.

OAuth2AuthorizationServer_idTokenClaims

Define claims for the OpenID Connect ID token.

For the value you can use a constant, a nevisAuth expression, an EL expression, or refer to an inventory variable by using the `${var.<name>}` syntax.

Note that you also have to do this for standard OpenID Connect claims. The only exception are `sub`, `iss` which will always be added.

Here are some examples:

Claim	Value
`given_name`	`${sess:ch.nevis.idm.User.firstName}`
`family_name`	`${sess:ch.nevis.idm.User.name}`
`email`	`${sess:ch.nevis.idm.User.email}`

Claim	Value
`mobile`	`${sess:ch.nevis.idm.User.mobile}`
`customer`	`${var.customer-number}`

Which claims will be added to the ID token depends on the incoming request. Non-standard claims have to be requested using the claims request parameter. Standard claims are added when a certain OpenID Connect `scope` is requested:

Requested Scope	Added Claims
`profile`	`name` , `family_name` , `given_name` , `middle_name` , `nickname` , `preferred_username` , `profile` , `picture` , `website` , `gender` , `birthdate` , `zoneinfo` , `locale` , `updated_at`
`email`	`email` , `email_verified`
`address`	`address`
`phone`	`phone_number` , `phone_number_verified`

OAuth2AuthorizationServer_idm

Assign a `nevisIDM Instance` or `nevisIDM Connector` . Required if nevisMeta is not used to store user consent.

GenericNevisProxySettings_bcProperties

Customize the low-level configuration (`bc.properties`) using properties described in the nevisProxy Technical Documentation.

For instance, when request validation is enabled this requires a buffer and this buffer has to be big enough to store the entire request.

The following example increases the maximum size of the request buffer to 10 MB:

```
ch.nevis.navajo.request.BufferSize=10485760
```

You also may have to increase the maximum allowed request size. See `Configuration: navajo.xml` for an example.

Note that increased buffer sizes may lead to increased demand of RAM and disk space.

When the required buffer exceeds `ch.nevis.navajo.request.MemBufferSize` then nevisProxy will buffer to disk instead.

The demand caused by request buffers can be estimated as follows:

- RAM: `MaxClients * ch.nevis.navajo.request.MemBufferSize`
- disk: `MaxClients * ch.nevis.navajo.request.BufferSize`

See `Configuration: navajo.xml` for a description of `MaxClients`.

NevisFIDODeployable_allowedAuthenticators

Here you can configure which authenticators are allowed.

This configuration is used and required **only** when `Allowed Authenticators` is `enabled`.

Proceed as follows:

1. Download the official FIDO Alliance Metadata file (JWT) from the FIDO Alliance Metadata Service.
2. Decode the downloaded JWT.
3. Copy out the complete metadata statements of the desired authenticators into a new JSON file.
4. (optional) Remove the optional entries to “slim down” the metadata entry, only required entries are `aaguid` and `attestationRootCertificates`.

5. Safe the JSON file and upload it here.

You can find more information in our FIDO2 Concept and Integration Guide.

NevisLogrendDeployable_https

Choose between plain HTTP, normal HTTPs and mutual (2-way) HTTPs. If `enabled a Key Store` is required. If set to `mutual`, a `Trust Store` is required as well.

OAuth2Client_refreshTokenLifetime

Enter a custom lifetime for the refresh token.

If not set the value of the `OAuth 2.0 Authorization Server / OpenID Provider` is used.

HostContext_defaultEntry

Set to redirect requests for the root path (/) to an absolute path or a full URL.

FIDO2StepBase_username

The `username` is used by nevisFIDO to look up the user in nevisIDM.

Depending on how the `nevisFIDO FIDO2 Instance` is configured, either the `extId` or the `loginId` have to be used.

NevisMetaServiceAccessBase_backendHostnameCheck

Enable to verify that the hostname on the certificate presented by the backend matches the hostname of `nevisMeta`

NevisDetectPersistencyDeployable_database

Add a database connection reference pattern.

Required properties to be set in the connector pattern are as follows:

- JDBC Driver (Oracle or MariaDB)
- JDBC URL
- DB user/password

BackendServiceAccessBase_hostnameCheck

Enable to verify that the hostname on the certificate presented by the backend matches the hostname configured in `Backend Addresses`

NevisAuthDatabase_schemaPassword

The password of the user on behalf of the schema will be created in the database.

NevisDetectAuthenticationConnectorStep_core

Pattern reference for the nevisDetect Core Instance to connect to.

CustomProxyLogFile_maxFileSize

Maximum allowed file size (in bytes) before rolling over.

Suffixes "KB", "MB" and "GB" are allowed. 10KB = 10240 bytes, etc.

If not set the following defaults will be used:

- `apache.log` : 1MB
- other logs: 10MB

SocialLoginBase_arbitraryAuthRequestParam

Arbitrary additional request parameters used in the authentication request. The property supports variable substitution.

Example:

```
[paramName]=[paramValue]
```

InBandMobileDeviceRegistration_realm

Assign an `In-band Mobile Authentication Realm` or `Authentication Realm` here.

Assignment is required.

The assigned realm will be used to protect the path `/nevisfido/uaf/1.1/request/registration/` .

If `Authentication Service` is `enabled` , a simple authentication flow will be added to this realm.

GroovyScriptStep_classPath

Set the `classPath` attribute of the `AuthState` element.

Lines will be joined with `:` . Enter 1 path per line.

When set, the `classLoadStrategy` attribute will be set to `PARENT_LAST` .

NevisIDMDeployable_smtp

Host:port of the SMTP server used for sending emails.

Configure if you prefer to provide the SMTP server with a single configuration, instead of configuring both `SMTP Host` and `SMTP Port` .

NevisFIDODeployable_relyingPartyName

Enter a name for the relying party.

This name is displayed by the user agent when performing a FIDO 2 registration or authentication.

GenericThirdPartyRealm_authenticationFilter

Define the filter that shall be application to applications to enforce authentication.

The following variables may be used:

- `${realm.id}` - unique ID of this realm pattern
- `${realm.name}` - name of this realm pattern
- `${auth.servlet}` - name of the servlet of the `Authentication Application` . May be used to perform a side-call.

NevisDetectPersistencyWebApplicationAccess_backendHostnameCheck

Enable to verify that the hostname on the certificate presented by the backend matches the hostname of `nevisDetect Persistency`

ServiceAccessBase_responseRewrite

Use this feature to replace backend hostnames in responses or set to `custom` to configure complex rewriting use cases.

- `off` disables automatic response rewriting
- `header` enables auto rewrite for response headers (including Set-Cookie header)
- `complete` enables auto rewrite for the entire response (including body)
- `custom` configure `Response Rewriting Settings` via `Additional Settings`

NevisAdaptDeployable_ipToLocationToken

Provide a secret download token for authentication.

DeployableBase_initialMemory

Use the given percentage of `Memory Limit` for the initial memory usage (`-Xms`).

This setting applies to classic VM deployments only.

PemKeyStoreProvider_keyPass

Enter the passphrase of the private key.

The passphrase will be used to decrypt the uploaded private key, if it is encrypted.

As the passphrase is considered sensitive information it should not be published with the project. It is therefore required to use a variable and define the value in the inventory (as a secret).

The default value of the variable is not relevant as the key is not loaded during background validation.

AuthCloudLookup_onUserNotExists

Assign an authentication step to continue with when the user does not exist or has no active authenticator.

AzureServiceBusRemoteQueue_key

Enter the `Primary Key` of the `Policy` as shown in the Azure portal.

PropertiesTestPattern_numberProperty

Enter a number.

GenericSocialLogin_secondNameClaim

The claim that contains the second name of the logged-in user in the social account. The default value is `family_name`.

HostContext_proxy

Assign the `nevisProxy Instance` this virtual host should be assigned to.

GoogleLogin_redirectURI

The callback URI to go to after a successful login with Google.

This will create an endpoint in your host config.

The URL will be a combination of the `Frontend Address` of the `Virtual Host` and the value configured here. For example, let's assume that you have configured:

- Return Path: `/oidc/google/`
- Frontend Address: `https://nevis.net`

Then the URL will be `https://nevis.net/oidc/google/` .

Use the `exact:` prefix to use the given path as-is. Without this prefix a normal mapping with `/*` will be generated and thus sub-paths will be accessible as well.

GenericSocialLogin_tokenEndpoint

The token endpoint of the OAuth2 server. It's required when `providerType` has the value `OAuth2` .

NevisAuthDeployable_frontendKeyStore

Assign the Key Store provider for the HTTPs endpoint. If no pattern is assigned a Key Store will be provided by the nevisAdmin 4 PKI.

FrontendKerberosLogin_level

Authentication level that is set on success.

TestingService_onDeployment

Use for testing only.

OAuth2RestEndpointBase_secure

Set Basic authentication for REST Service of OAuth 2.0 Authorization Server / OpenID Provider.

When this property is `enabled`, the request must include Authentication Header. The header is a combination of clientID and clientSecret with base64 encoded

NevisDetectDatabase_parameters

Enter parameters for the DB connection string.

Enter 1 parameter per line.

Lines will be joined with `&`.

The default is:

```
useMysqlMetadata=true
```

The default value will be used **only** when no parameters are entered.

If you want to keep the default parameters, add them as well.

NevisDetectServiceAccessBase_csrf

Cross-Site Request Forgery (*CSRF*) is an attack to force an authenticated user to send unwanted requests.

- `off (default)` - no CSRF protection. Recommended for applications which may be called from other sites.
- `header-based` - `GET` and `HEAD` requests are allowed (assumption: these methods must not manipulate server-side state). For other requests the `Referer` and `Origin` headers must match the `Host` header.

SamlSpRealm_template

By default, the Service Provider does not need any rendering template.

However, a GUI will be shown when `Logout Reminder` is `enabled` and may be shown when `Custom Pre-Processing` is used.

nevisLogrend: Simple Mode

Point your browser to a protected application to have a look at the login page. Download any resources (e.g. images, CSS) that you want to adapt. Then upload the changed files here.

To change the outer HTML upload a file named `template.html` . Here is a simple example:

```
<!DOCTYPE html>
<html lang="${lang.code}">
  <head>
    <title>${label.title}</title>
    <link href="${resources}/bootstrap.min.css" rel="stylesheet" type="text/css">
    <link href="${resources}/default.css" rel="stylesheet" type="text/css" media="all">
  </head>
  <body>
    <header id="header" class="container-fluid">
```

```
        <img class="logo center-block" src="${resources}/example.svg" alt="NEVIS Security Suite">
      </header>
      <main id="content" class="container">
        ${form}
      </main>
    </body>
  </html>
```

Please also upload file resources referenced by your template (e.g. images, CSS, Javascript). Use this when you reference additional files, or if you want to override the default files provided.

The template must contain `${form}` and may contain additional expressions.

Expression	Description
`${form}`	generated login form (required)
`${lang.switch}`	language switcher component
`${lang.code}`	current language code (i.e. EN, DE)
`${label.title}`	a human-readable title
`${label.myLabel}`	a custom text which must be defined via `Custom Translations`
`${resources}`	path to static resources (e.g. CSS, images, Javascript)

Some resources (i.e. bootstrap.min.css, default.css) are provided out of the box because they are required by the default template. Feel free to use them.

nevisLogrend: Expert Mode

Expert users may upload Velocity templates and resources to nevisLogrend.

Zip files will be extracted into the nevisLogrend *application*:

```
/var/opt/nevislogrend/<instance>/data/applications/<realm>
```

Flat files will be added to the following subdirectories:

- `webdata/template` : Velocity templates (`*.vm`)
- `webdata/resources` : additional resources (e.g. images, CSS)

nevisProxy: Simple Template

nevisProxy provides a simple login page renderer which can be used instead of nevisLogrend. See `Login Renderer` for details.

For each enabled language (e.g. `en`) upload a file named `<lang>_template.html` . The template must contain the placeholder `NEVIS_AUTH_FORM` .

If your templates require additional resources (e.g. CSS, images) upload them as `Hosted Resources` on the nevisProxy virtual host.

MobileDeviceDeregistration_token

Assign a `NEVIS SecToken` pattern.

This pattern must also be assigned to `Application Access Tokens` in the `Authentication Realm` .

NevisMetaDeployable_frontendKeyStore

Assign the Key Store for the HTTPs endpoint.

If no pattern is assigned a Key Store will be provided by nevisAdmin 4 automatic key management.

HostContext_sessionStore

Assign a `nevisProxy Remote / Hybrid Session Store` pattern here if you want to store sessions in a remote session store.

A remote session store must be used when the nevisProxy instance is deployed with redundancy and there is no sticky load balancer in front.

ProxyPluginPattern_serviceMapping

Mapping entries between RESTful addressees and services. One line per mapping, for example:

```
requestData=/processRequestData
terminateSession=/processSessionTermination
getVersion=/getVersion
```

GenericNevisAuthSettings_envVariables

Add additional environment variables to the nevisAuth `env.conf` .

The standard environment variables `RTENV_SECURITY_CHECK` and `JAVA_OPTS` will always be present in `env.conf` and can't be overwritten using this setting.

NevisFIDODeployable_metadata

The FIDO UAF specification describes metadata as follows:

```
It is assumed that FIDO Server has access to a list of all supported authenticators and their corresponding Metadata. Authenti
```

```
  * Supported Registration and Authentication Schemes
  * Authentication Factor, Installation type, supported content-types and other supplementary information, etc.

  To make a decision about which authenticators are appropriate for a specific transaction, FIDO Server looks up the list of aut
```

The nevisFIDO server ignores any authenticators and halts all operations in relation to them, which do not have metadata data entries accessible for the server.

Note that the default value of this field represents the metadata required for nevisFIDO to be able to work with the `NEVIS Access App`. If you're using a custom app based on the NEVIS Mobile Authentication SDK or a customized Whitelabel Access App, these values will need to be updated.

The *Android* metadata statements contain the Google root certificates to support *Android Key Attestation | FIDO UAF Basic Full Attestation*. These entries must be kept up-to-date.

NevisAdaptDatabase_user

Provide the DB user name here.

NevisMetaWebApplicationAccess_token

A `NEVIS SecToken` pattern must be assigned here.

The token will be issued after authentication and propagated to nevisMeta.

The user must have the role `nevisMeta.admin`.

OAuth2AuthorizationServer_jwkSetKeyId

When set to `enabled` a `kid` header value will be added to issued access and ID tokens.

The value allows the authorization server to explicitly signal a change of key material to recipients.

The meaning of the `kid` header is slightly different for signed and encrypted tokens.

ServiceAccessBase_backendTrustStore

Assign the trust store for outbound TLS connections.

If no pattern is assigned a trust store will be provided by nevisAdmin 4 automatic key management.

GenericSocialLogin_buttonCss

The css class that apply for the social login button. Ensure that the Login Template used in your realm pattern includes a CSS file which defines the CSS class.

NevisAdaptDatabase_jdbcDriver

Due to licensing, nevisAdapt cannot ship the JDBC driver to connect to Oracle databases, Therefore, those who want to use an Oracle database need to obtain and provide the Oracle JDBC driver on their own.

The `.jar` files can be downloaded from Oracle

Uploading any other `.jar` files containing JDBC drivers is possible as well.

SocialLoginBase_onUserFound

Configure the Authentication Flow in case no user with Subject/ID from social account was found but email does exist in nevisIDM. The Authentication Flow must contain:

- `Social Login Link User` pattern to link an existing user in IDM with Subject/ID of social account.
- `Social Login Done` to end the social login flow after some other action(s).

Note: Please select scope `email` and `profile` for getting user's information from social account.

LogSettingsBase_serverLog

Select the type of appender.

In Kubernetes the `default` appender writes to system out so that log messages appear in the docker logs.

Choose between:

- `default` - log to default target
- `default + syslog` - log to default target and forward to a Syslog server
- `syslog` - forward to a Syslog server only

CustomAuthLogFile_auditLog

Configure audit logging of nevisAuth.

Select `enabled` to use the default audit channel implementation provided by the NevisAuditChannel class.

If you want to use your own channel, you have to assign `Audit Channels` and select one of the following options here:

- `enabled` : use you own channel **in addition** to the `NevisAuditChannel` .
- `custom` : use only own channel.

UserInput_rememberInput

Enable this feature to show a `Remember Input` checkbox.

If selected the user input is stored in a cookie (named like this step) so that the value can be prefilled on subsequent authentications.

UserInformation_onSubmit

Define a follow-up step.

The `Button Type` should be set to `submit`, but the form can also be submitted by other means (e.g. refreshing the browser).

BackendServiceAccessBase_params

Add custom `init-param(s)` for the Http(s)ConnectorServlet. For example: ConnectionRetries=10

Please check the nevisProxy technical documentation for supported `init-params` of the servlet classes
`ch::nevis::isiweb4::servlet::connector::http::HttpConnectorServlet` and
`ch::nevis::isiweb4::servlet::connector::http::HttpsConnectorServlet`.

GenericNevisDetectSettings_javaOpts

Add additional entries to the JAVA_OPTS environment variable.

Use the expression `${instance}` for the instance name.

For instance, you may configure nevisDetect to create a heap dump on out of memory as follows:

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/var/opt/nevisdetect/${instance}/log/
```

Be aware that this example will not work for Kubernetes as the pod will be automatically restarted on out of memory and the created heap dump files will be lost.

NevisIDMDeployable_mailSMTPUser

Set if a user is required to connect to the SMTP server.

NevisDetectEntrypointDeployable_jms

Add reference for the pattern providing Java Messaging Service.

Two different options are allowed at this time:

- `nevisDetect Message Queue Instance` - deployment pattern for a dedicated MQ component
- `ActiveMQ Client Configuration` - connect to an external ActiveMQ service via SSL

WARNING: In case of Kubernetes deployment, only `ActiveMQ Client Configuration` is supported.

ICAPScanning_url

URL(s) of the ICAP server(s). Each URL must have the same path.

Example: `icap://my-clamav-server1/avscan`

TCPSettings_requestTimeout

Timeout waiting for the response.

NevisProxyObservabilitySettings_traceContextExtraction

Choose one of:

- **enabled**: if present, extract the trace context from the HTTP request header and set it as parent for the current span
- **disabled**: ignore the trace context from the HTTP request header

CustomAuthLogFile_auditLogFormat

Log4j 2 log format for the AUDIT logs.

Note: not relevant when Log Targets is set to `syslog`.

OAuth2Client_redirectUri

Enter allowed URIs to return the code / token to.

Single-page and classic Web applications should use URLs, mobile applications sometimes use custom scheme URIs.

Regular expressions are not supported.

CustomAuthLogFile_levels

Set log levels.

The default is:

Category	Level
`EsAuthStart`	`INFO`
`org.apache.catalina.loader.WebappClassLoader`	`FATAL`
`org.apache.catalina.startup.HostConfig`	`ERROR`

The default gives you log messages during startup but is rather silent during runtime.

A good setting for troubleshooting is:

Category	Level
`AuthEngine`	`INFO`
`Vars`	`INFO`

When using `nevisAuth Database` with MariaDB the category `org.mariadb.jdbc` can be set. The levels behave as follows:

- `ERROR` : log connection errors
- `WARNING` : log query errors
- `DEBUG` : log queries
- `TRACE` : log all exchanges with server

Check the documentation for other important trace groups.

In classic deployment nevisAdmin 4 does **not** restart nevisAuth if you only change log levels. The log configuration will be reloaded within 60 seconds after deployment.

HostContext_allowedMethods

Define the HTTP methods which are allowed on this virtual host.

The setting `default (complete)` is quite relaxed as it enables most methods. Only two are excluded:

- `CONNECT` : no use case of nevisProxy.
- `TRACE` : may be useful for debugging but can be a security vulnerability.

If you do not have any applications using WebDav select `basic` .

The allowed HTTP methods can be restricted further in application patterns.

For more fine-grained control you may use `Generic nevisProxy Instance Settings` to overwrite the `allowedMethods` (see pattern help for details).

CustomAuthLogFile_eventsSyslogFormat

Log4j 2 log format for the EVENTS SYS logs.

Note: not relevant when Log Targets is set to `default` .

NevisAuthDeployable_logging

Add logging configuration for nevisAuth.

NevisAdaptUserNotification_idm

Reference for the nevisIDM service. The `nevisAdapt Authentication Connector` uses nevisIDM's REST API to send notification emails to the user if the calculated weighted risk score exceeds the configured threshold.

NevisIDMPasswordCreate_onExists

If the user already has a password credential and error will occur.

You can assign a step here to handle this case.

TANBase_maxRetry

The maximum attempts for **each** code.

When this threshold is reached, the behaviour depends on `Max Regenerations` .

As long as `Max Regenerations` is not exhausted, a new code will be generated and sent to the user.

Once `Max Regenerations` is reached as well, the `On Failure` exit will be taken.

NevisAdaptAnalyzerConfig_deviceCookieAnalyzer

Used to disable Device Cookie creation.

AuthCloudLookup_onUserExists

Assign an authentication step to continue with when the user exists and has an active authenticator.

NevisDetectAuthenticationConnectorStep_adapt

Optional pattern reference for the nevisAdapt Instance to help configure the device cookie name.

SecretTestAddon_secretValues

Set a variable and insert secret value(s) in the inventory.

The file `/var/opt/nevisproxy/<instanceName>/run/secret_values.txt` should then contain:

- classic: resolved value(s)
- Kubernetes: `secret://` reference(s)

OutOfBandManagementApp_resources

Upload a ZIP to provide your own resources.

By default, the following resources are provided:

- `index.html`
- `logo.png`

SamlSpConnector_audienceRestrictionMode

Configure **if** an `<AudienceRestriction>` element shall be added to generated SAML assertions and **what** the element shall contain.

Choose between:

- `automatic` : use `Custom Audience` , if configured, and `SP Issuer` otherwise.
- `issuer` : use `SP Issuer` .
- `custom` : use `Custom Audience` .
- `none` : no `<AudienceRestriction>` element is added.

GenericAuthenticationStep_keyObjects

This pattern adds a XML element `KeyStore` to `esauth4.xml` .

Each pattern referenced here creates an additional `KeyObject` which will be added to this `KeyStore` as a child element.

AzureServiceBus_dlq

Remote Azure Service Bus Queue to which Dead Letter messages should be sent.

Dead letter messages are those messages which are not in the expiryQueue and their delivery was unsuccessful. For further reference check `NevisIdm Technical documentation > Configuration > Components > Provisioning module > Provisioning providers` .

NevisDPLogSettings_maxBackupIndex

Maximum number of backup files to keep in addition to the current log file.

This configuration applies to non-Kubernetes deployment only.

NevisIDMCheckUserLoginInfo_neverLoggedIn

Configure the step to execute if the user never logged in. If no step is configured here the process ends with `AUTH_DONE` .

NevisFIDODeployable_frontendAddress

Enter the address of the `Virtual Host` where the services of this instance are exposed.

Enter the address without any path component.

Example:

```
https://example.com
```

If no address is provided, the pattern tries to automatically determine a value based on the `Virtual Host` patterns, that are associated with this instance through patterns for out-of-band use-cases.

The entered value is used to calculate:

- AppID
- *Dispatch payload*

The *dispatch payload* informs the mobile device where to access nevisFIDO for the following use cases:

- Out-of-band Registration
- Out-of-band Authentication

NevisProxyDatabase_peerStrategy

Controls the used strategy of the Peer Servlet:

- `FAILOVER` : The loadbalancer sends all requests to the same instance (instance A). If instance A goes down, the loadbalancer will send now all requests to instance B. The loadbalancer should only switch back to instance A if it has been restarted.

- `DISTRIBUTED` : The loadbalancer assure at least 90% session stickiness to both instances, for example by using the client IP address. Once the request for a session goes to the other instance, this one will get the session information from the first instance and copy into its local session store.

MicrosoftLogin_buttonLabel

Enter a label for the social login button.

Translations for this label can be configured in the `Authentication Realm` pattern.

ObservabilityBase_agentLibrary

Path to the selected agent's library that is available locally to the deployed application.

NevisIDMDeployable_messagingPort

Port of the messaging service.

Enter a different port to deploy multiple nevisIDM instances on the same target host in classic VM deployment.

SecToken_keystore

Assign a pattern which sets the key material used for signing the token.

If no pattern is assigned automatic key management is used and the signer key will be created automatically.

StaticContentCache_responseHeaderMode

Response headers can indicate whether clients and intermediate servers should cache the response.

Choose one of:

- **comply** : Follow the `Cache-Control` directives sent by the backend.
- **ignore** : Store the response even if the backend sent a `Cache-Control` directive to prevent caching. Be aware that ignoring `Cache-Control` directives can lead to sharing sensitive data between clients.

Some clients or content providers try to switch off caching even for mostly static content like images or style sheets. You can limit the load on your content providers as follows:

- Add a **Static Content Cache** pattern and link it to your application via **Additional Settings**;
- Configure **Apply only to sub-paths** to store responses on paths that only emit static content, for instance images;
- Set **Response Header Mode** to **ignore**;
- Configure the **Max Lifetime** of stored responses.

KerberosLogin_level

Authentication level that is set on success.

SAPLogonTicket_userIdSource

Source of the user ID to set for the issued SAP ticket.

The default is `${request:userId}` .

NevisProxyDatabase_databaseSchemaCheck

Select one of:

- `enabled` - the database schema and integrity constraints are checked on startup to ensure they match the requirements of the `Remote Session Store`.
- `disabled` - the database schema and integrity constraints are not checked on startup.

Note: On certain MariaDB versions, the check produces fake errors due to a MariaDB bug. By setting this parameter to `disabled`, you can skip the check.

DummyTAN_level

Set an authentication level.

NevisDetectDatabase_hikariValues

Specify custom values for Hikari datasource configuration. Separate keys and values with `=`. The valid keys can be found at HikariCP - GitHub.

Example to set the same as if selecting `recommended`:

```
maxLifetime=300000
idleTimeout=100000
maximumPoolSize=50
```

GenericIngressSettings_ingressClassName

Defines the `ingressClassName` of the generated ingress. It can be used instead of the `kubernetes.io/ingress.class` annotation to select which ingress controller should handle the generated ingress. For more information see Multiple Ingress controllers.

OAuth2AuthorizationServer_invalidClient

Configure the step to execute after error when the client sending the request is not registered.

If no step is configured here the process ends and the error will display on UI.

NevisFIDODeployable_database

Configure a database to store nevisFIDO sessions.

If no pattern is assigned, sessions will be stored in memory. We recommend to use a database in production.

Webhook_key

Set a unique key for the property name.

AuthCloudBase_title

Enter a label to use for the title.

You can use a different standard label (e.g. `title.login`) or invent your own.

Translations for custom labels can be defined in the `Authentication Realm / GUI Rendering / Translations` .

The default label `title.authcloud` has the following translations:

- `en` : Authenticate with Access App
- `de` : Mit Access-App anmelden
- `fr` : S'authentifier avec l'application Access
- `it` : Autenticazione con l'app Access

ManagementDemo_path

Enter the path where this example shall be exposed on the nevisProxy `Virtual Host` .

NevisIDMChangePassword_addConfirmationField

If `enabled` , a confirmation field is also rendered on GUI.

NevisIDMPruneHistoryJob_skipList

Comma-separated list of versioned tables (which are used to provide history data) to be ignored by the prune history job and left with their original content.

Possible values (Any combination of the following):

- `tidma_application_v`
- `tidma_authorization_appl_v`
- `tidma_authorization_client_v`
- `tidma_authorization_erole_v`
- `tidma_authorization_unit_v`
- `tidma_authorization_v`

- tidma_cert_info_v
- tidma_client_application_v
- tidma_client_v
- tidma_consent_v
- tidma_cred_login_info_v
- tidma_credential_v
- tidma_dict_entry_v
- tidma_dict_entry_value_v
- tidma_enterprise_auth_v
- tidma_enterprise_role_v
- tidma_erole_member_v
- tidma_fido2_v
- tidma_fido_uaf_v
- tidma_mobile_signature_v
- tidma_oath_v
- tidma_personal_answer_v
- tidma_personal_question_v
- tidma_policy_configuration_v
- tidma_policy_parameter_v
- tidma_profile_v
- tidma_property_allowed_val_v
- tidma_property_v
- tidma_property_value_v
- tidma_role_v
- tidma_saml_federation_v

- `tidma_template_collection_v`
- `tidma_template_text_v`
- `tidma_template_v`
- `tidma_terms_application_v`
- `tidma_terms_url_v`
- `tidma_terms_v`
- `tidma_unit_cred_policy_v`
- `tidma_unit_v`
- `tidma_user_login_info_v`
- `tidma_user_v`

For further information about historical tables visit Versioned DB tables .

Dispatcher_transitions

Define how to dispatch based on *conditions*.

In the first column enter the *transition*. A *transition* may be:

- a condition `name`
- a comma-separated list of conditions

All conditions in the transition **must** match in order for the transition to be applicable. The most specific transition is chosen.

In the second column enter the *position*. Position refers to the list of `Conditional Step(s)` . The first step has position `1` .

Examples: