# Obsolete MFA Profile FAQ

**A new FAQ is now available.**

This version of Refeds MFA Profile FAQ has been replaced with a new version. Visit the new REFEDS MFA Profile FAQ.

The following FAQ support the use of the REFEDS Multifactor Authentication Profile. This documentation is intended to be non-normative supporting information. If you have any questions about the use of the REFEDS MFA Profile, please direct them to the REFEDS mailing list (refeds@lists.refeds.org).



## What are the requirements for multifactor authentication in the profile?

In order to assert the REFEDS MFA profile, the Identity Provider must be using an authentication method that meets the following requirements:

- The authentication of the user’s current session used a combination of at least two of the four distinct types of factors defined in ITU-T X.1254: Entity authentication assurance framework, section 3.1.3, authentication factor (something you know, something you have, something you are, something you do).
- The factors used are independent, in that access to one factor does not by itself grant access to other factors.
- The combination of the factors mitigates single-factor only risks related to non-real-time attacks such as phishing, offline cracking, online guessing and theft of a (single) factor.

## What does the Profile Guarantee?

The Profile signals that the specific user in question is using multifactor authentication in a way that meetings the requirements shown above. Multifactor provides additional safeguards for both IdPs and SPs but is not completely resistant to all possible threats - this should be kept in mind when selecting appropriate approaches and technologies.

## What constitutes an acceptable "second" factor?

The REFEDS MFA Profile makes no statement about the types of technologies that could be used as the second or multi factor. The InCommon MFA Interoperability Profile Working Group has prepared some useful advice on approaches that might be useful.

## Why two of the four types?

Each factor in the MFA profile must be of a different type. This means that validating two separate passwords is not sufficient - the second (and further) factors must use a different factor approach. This is to address any generic point of failure with one factor type.

## What do you mean by "independent" factors?

- Implementors must work to ensure that the different factors used in the authentication process are independent, meaning that gaining access to one factor must not trivially grant access to the other factor.
- Any factor that is directly accessible using the first factor is no more secure than the single factor by itself, and so is NOT considered a second factor.
- Institutions are expected to provide safeguards to maintain the independence of their supported authentication factor.
    - a software/virtual phone that is authenticated using the enterprise password is not an appropriate second factor.
- The MFA profile does not enumerate specific requirements the institution must meet to protect against these forms of authentication dependence, but technical restrictions (where feasible) and user education are highly recommended to mitigate the risks of users deploying factors in a manner that decreases their independence.
- Processes that allow a user to immediately register a new second factor (re--registration) using only their “first factor” enterprise password are no more secure than use of the enterprise password itself.
- Implementors are expected to require greater scrutiny before allowing registration of replacement or additional second factors to prevent attackers with password access from simply registering and immediately using a new second factor. Additional second factors can use a existing second factor when registered or the same method as the first second factor.

## How Does the MFA Profile relate to (level) of assurance profiles?

Many assurance profiles will include approaches to MFA within them. REFEDS Assurance Framework does not reference the REFEDS MFA Profile but is intended to work in parallel with this profile.

## Is This Profile SAML-Specific?

The Profile is not intended to be SAML-specific and can be extended for use with other technologies, such as OIDC. This will be reviewed subject to demand in those areas and examples provided when appropriate and mature.

## How Do I Use This in SAML?

The recommended means of representing these profiles in a SAML assertion are via the `<AuthContextClassRef>` element (SAML 2.0). These are expressed in SAML authentication statements used to represent acts of authentication by the subject of an assertion.

In the case of SAML 2.0, the use of the Authentication Context mechanism has the benefit of enabling signaling of requirements by a relying party in its requests to an identity provider via the `<RequestedAuthnContext>` request element. The standard defines the rules for this in https://www.oasis-open.org/committees/download.php/56777/sstc-saml-core-errata-2.0-wd-07-diff.pdf in Section 3.3.2.2.1, on page 46. (Note that this revision of the standards document includes material adjusted by Approved Errata, though the clarificatons in this area are minimal.)

While this mechanism has a number of advanced features, the simple form and the one that interoperates with most open source SAML IdPs and SPs is the use of the default ("exact") `Comparison` operator and the inclusion of one or more `<AuthContextClassRef>` elements. An IdP that receives a request containing such an element is obligated to authenticate the user in a manner that allows it to return one of the requested `<AuthContextClassRef>` values in its assertion, or fail the request. Returning a SAML error is required in the event of a failure, though there are some cases where an IdP cannot always guarantee a response to the SP. Any other behavior by an IdP is non-compliant with the standard.

Typically, an IdP by default would not be expected to recognize the REFEDS profile context class, and so requesting it would be expected to result in an error. Configuration examples for several identity provider technologies are available to demonstrate how to incorporate this context class into an IdP.

## What Does a (SAML) SP need to do?

Most IdPs and campuses that support MFA services do not provide universal MFA coverage for their user communities. This means that even when a given IdP is capable of supporting this profile, there is a significant probability that any given user may not be able to authenticate using MFA. There is no defined mechanism at present to identify whether a given IdP is configured to assert `<AuthnContextClassRef>` values, and SAML itself does not rely on that knowledge; it assumes that IdPs will respond in accordance with the standard when handling a request containing requirements it cannot meet. If the SP does not have any information about an IdP's capabilities, it may not be able to distinguish between a case of specific users being unable to satisfy the profile, and an IdP as a whole not supporting it. Whether this distinction is relevant will depend on the SP.

The `<AuthnContextClassRef>` value is returned in the SAML assertion; it is not sufficient to configure an SP to request MFA and assume all responses will therefore contain the MFA context. This is because users can generally bypass an SP's SAML request configuration using unsolicited responses from an IdP, or by handcrafting a SAML request that does not include the MFA requirement. It is therefore essential that an SP requiring MFA MUST enforce the presence of the value in the incoming assertion before proceeding; merely requesting the value does not ensure that the result will contain the proper value *even if an assertion is returned*, for a variety of reasons that are outside the scope of this FAQ.

## What Goes in My SAML Requests?

Typically when somebody accesses a secure service, the SAML SP software of choice kicks in and relays a SAML `<AuthnRequest>` message through the browser to trigger the IdP to log them in. This is usually a very automatic and "under the covers" process and there's very little in the request that needs to be customized.

When producing a SAML authentication request in which use of the MFA profile is desired, the content is straightforward:

1. Explicitly list every `<AuthnContextClassRef>` value that your SP is willing to accept in the `<RequestedAuthnContext>` element in your SAML request. The actual values you list will depend on your use case (see "Use Cases" below for some general guidance).
2. No matter how carefully you specify context class values, some IdPs may be unable to respond due to software or process limitations. (This issue is not specific to the MFA profile but affects any requests that includes explicit `<RequestedAuthnContext>` elements). If you want to support IdPs that are not able to support the values you list, then on receiving a SAML error you can try reissuing your SAML request without the same content.

Note that this is a protocol-level description of what needs to happen. The actual mechanics of customizing the SAML requests issued by an SP will vary by software implementation and in some cases may not even be possible if the software is not fully SAML compliant.

## What Are Some Use Cases for Including the <RequestedAuthnContext> Element?

### SP Requires MFA

To require that all users must authenticate using MFA, a SAML authentication request must include:

```
<samlp:RequestedAuthnContext Comparison="exact">
        <saml:AuthnContextClassRef>https://refeds.org/profile/mfa</saml:AuthnContextClassRef>
</samlp:RequestedAuthnContext>
```

That is, MFA is the (only) requested value.

Even if an IdP supports the MFA Profile, it can only respond successfully to such a request if MFA is actually performed. If the user can authenticate to the IdP, but is not able to use MFA, the IdP MUST respond with an error, and the SP will not receive any information about the user who tried to authenticate.

If this distinction is important, and it's important to know the identity of the user even if MFA is not possible, consider one of the later uses case of preferring MFA, but accepting less. Application error messages when using this model should explicitly note that MFA is required to access the SP's services. Note that most SPs do not provide sufficient default error handling capability for this and so this will generally require some effort.

### SP Prefers, But Does Not Require, MFA

In some cases, an SP may prefer that users authenticate with MFA but is willing to accept non-MFA authentication. Some scenarios where this approach would make sense:

- Applications that can implement a local scheme to do "stronger authentication" of specific users but prefer to allow users to use familiar campus mechanisms when available.
- Applications that will allow access to some services to all users, but have other services that are limited to those that authenticate using MFA.
- Applications that wish to offer their own opt-in feature for users to elect to use MFA for that service.
- An application that only allows access to users who authenticate with MFA, but wants to personalize error messages to users who do not use MFA as part of the authentication process.

Unfortunately, SAML 2.0 does **not** support this use case directly. It does not have the concept of "prefer but not require" and while the list of context classes in a request is ordered (in fact this was clarified via errata), this is only useful when different types of authentication are not "composed" together, which is a very common situation with MFA deployments. MFA tends to combine password authentication with "other things" and so it is impossible to reliably ask for "MFA or Password" because one step happens first, satisfies the request, and may well short-circuit any further work in the interest of minimizing the user's burden.

Ordering can be useful when methods are discrete, such as Password vs. X.509 certificates vs. Kerberos/SPNEGO. In such a case, the IdP can be expected to try methods in order when it can, but even here IdPs may "prioritize" SSO so that a later method the user has already completed would be favored over an earlier method that would require a new challenge.

Because this is not directly supported, there are a couple of alternative strategies to consider for these use cases, though neither is simple to implement.

#### "Step Up" MFA

The step-up model is often the most elegant if it can be orchestrated by the application and SP and is particularly well-suited to cases where either a minority of access needs MFA or where the use of MFA can naturally be required later in the interaction with the user.

With this model, the first interaction with the SP should be handled with a "vanilla" authentication request that does not stipulate anything regarding MFA and simply attempts to get the user identified and into the application.

Assuming this works, the application can proceed for some set of activities (or not), but eventually may either prompt the user to indicate whether MFA is possible, or just unilaterally attempt to elevate the authentication of the user by triggering a second authentication request back to the IdP with the requirement for MFA as described previously. Upon receipt of the new assertion, either the user's privileges can be elevated or in the event of an error, left as is with an indication to the user that the request for MFA was not successful.

Again, this may be very non-trivial with some application and SP software or integration patterns.

Note that while many IdPs and many MFA deployments are architected such that the second request for MFA may skip the password step and make for a relatively seamless step-up experience, this is by no means a guarantee. Some IdPs may not have the ability to do this, and some MFA technologies may rely on factors that don't include the same password.

#### MFA with Retry

Another common approach is to initially attempt authentication via MFA, but then handle an error from the IdP by downgrading and reissuing a request for regular authentication. In this scenario, the first request is issued with the additional content noted above, but if the IdP responds with an error, the SP has to detect this and either immediately reissue a request without the additional content, or present the user with an explanation and perhaps provide the option to do so.

This can be a bit more interoperable and result in fewer challenges to the user in some cases, but also requires more advanced error handling by the SP and application, so may not always be possible.

## Why Reference ITU-T X.1254?

The REFEDS Multifactor Authentication Profile references ITU-T X.1254 as it references four different factors for authentication in a well described way.  Other frameworks define 3 factors, which is more limiting for implementors.  These four areas are:

- something an entity has (e.g., device signature, passport, hardware device containing a credential, private key);
- something an entity knows (e.g., password, PIN);
- something an entity is (e.g., biometric characteristic);
- something an entity typically does (e.g., behaviour pattern).

Please note that this reference is purely to the good definitions used by ITU-T X.1254 for authentication factors - there is no other relationship between the MFA Profile and this, or any other, assurance framework at this time.

## Do You Provide Profiles for the Authentication Approaches?

REFEDS is developing a Profile to flag approaches that do not provide multifactor.  The scope of this is still to be decided, but this FAQ will be updated as appropriate.