Release Notes

/ Identity Gateway 7

Latest update: 7.0.2

Mark Craig
Joanne Henry

ForgeRock AS.
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Notes covering ForgeRock® Identity Gateway features, fixes, and known issues.

Table of Contents

Overview

IG integrates web applications, APIs, and microservices with the ForgeRock Identity Platform, without modifying the application or the container where they run. Based on reverse proxy architecture, IG enforces security and access control in conjunction with Access Management modules.

Quick Start

 What's New Discover new features and improvements in this version.	 Prepare for Deployment Learn about the requirements for running DS software in production.	 Check Compatibility Review key implementation changes and compatibility with previous deployments.
 Review Fixes Review bug fixes, limitations, and open issues.	 Check Doc Updates Track important changes to the documentation.	 Get Support Find out where to get professional support and training.

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

The ForgeRock Common REST API works across the platform to provide common ways to access web resources and collections of resources.

Chapter 1
What's New

What's New in IG 7.0.2

Stability	+ *Filter to rebase the scheme, host name, and port of requests* The ForwardedRequestFilter has been added to rebase a request URI with a computed scheme, host name, and port. Use this filter to configure redirects when the request is forwarded by an upstream application such as a TLS offloader. For more information, see "ForwardedRequestFilter" in the *Configuration Reference*.

What's New in IG 7.0.1

Stability	+ *AmService automatically obtains SSO token header name from AM* To reduce configuration errors, and simplify configuration, AmService no longer uses the default value, `iPlanetDirectoryPro`, for `ssoTokenHeader`. If `ssoTokenHeader` is not provided, IG queries the AM `/serverinfo/*` endpoint for the header name or cookie name of the SSO token.

What's New in IG 7.0.0

IG as a standalone Java executable	+ *Container-less execution environment* IG is now delivered as a .zip file, for installation in standalone mode. In standalone mode, IG provides a simple unzip installation path, a `classes` directory for support, a startup script, and support for custom extensions. A Vert.x-specific configuration block is available in the `connector` property of `admin.json`, and in the `websocket` property of ClientHandler and ReverseProxyHandler.

	For information about migrating from IG in web container mode to IG in standalone mode, see "*Migration*". For information about installing in standalone mode, see "Downloading and Starting IG in Standalone Mode" in the *Getting Started Guide*. + *Support for HTTP/2 when IG is client-side* In standalone mode, IG can use HTTP/2 or HTTP/1.1 to send requests to a proxied application, or request services from a third-party application. No additional configuration is required to use HTTP/2 over non-TLS. The Application Layer Protocol Negotiation ALPN extension is used for HTTP/2 over TLS. The protocol is negotiated according to the configuration of IG's `admin.json`, the `alpn` property in ClientTlsOptions, and by the new properties `protocolVersion` and `http2PriorKnowledge` in the ClientHandler and ReverseProxyHandler. + *Support for HTTP/2 when IG is server-side* IG in standalone mode provides a new object, ServerTlsOptions, to configure server-side properties of the the TLS-protected connector. Use ServerTlsOptions in `admin.json`. + *Support for patching* For IG installed in standalone mode, **classes** is a new directory in the classpath for patches from ForgeRock support.
Docker	+ *Evaluation Docker image* ForgeRock provides an unsupported base Docker image for IG, available in ForgeRock's public Docker registry. For information about using the Docker image, see the Deployment Guide. + *IG .zip provides Dockerfile* The IG .zip file now provides a Dockerfile that you can use to build a Docker image. For information, see the Deployment Guide.
API Security - separating API security concerns from business concerns	+ *Policies with new options for more flexibility*

- **resourceUriProvider** is a new property of the PolicyEnforcementFilter to ease the transition from an agent-based system. Use the property to request AM policy decisions with the original request URL as the resource URL, or with a script to generate the resource URL. In previous releases, IG could request policy decisions only by using the route `baseURI` as the resource URL.

  For more information, see the `"resourceUriProvider"` property of "PolicyEnforcementFilter" in the *Configuration Reference*.

- If an AM policy decision denies a request with supported advices, the PolicyEnforcementFilter can now redirect the request to a URL specified in a SingleSignOnFilter, such as the URL of the custom login page. Previously, the filter always redirected the request back to AM.

  The URL is passed in a new property, `loginEndpoint`, in the `ssoToken` context. To use the redirect, configure `loginEndpoint` in the SingleSignOnFilter.

  For information, see "SingleSignOnFilter" in the *Configuration Reference*.

- **sessionIdleRefresh** is a new property of AmService, to periodically refresh AM sessions.

  When the SingleSignOnFilter is used for authentication with AM, AM can view the session as idle even though the user is interacting with IG. The user session eventually times out and the user must re-authenticate.

  For information, see "AmService" in the *Configuration Reference*.

+ *Authenticate through AM authentication trees and chains*

A new property, `authenticationService`, in SingleSignOnFilter and CrossDomainSingleSignOnFilter lets users authenticate to AM by using AM's authentication trees and chains.

For more information, see "SingleSignOnFilter" in the *Configuration Reference* and "CrossDomainSingleSignOnFilter" in the *Configuration Reference*.

+ *Adding basic authentication to outgoing requests*

HttpBasicAuthenticationClientFilter is a new filter for service-to-service contexts, where IG needs to access remote resources that are protected by HTTP Basic Authentication. For more information, see "HttpBasicAuthenticationClientFilter" in the *Configuration Reference*.

+ *CORS to enable APIs, and control cross-origin access*

CorsFilter is a new filter to configure policies to allow user agents to make requests across domains. For more information, see "CorsFilter" in the *Configuration Reference*.

+ *Protection against cross-site request forgery*

CsrfFilter is a new filter to harden protection against CSRF attacks. For more information, see "*Protecting Against CSRF Attacks*" in the *Gateway Guide* and "CsrfFilter" in the *Configuration Reference*.

+ *Declarative authorization, for local evaluation of authorization rules*

AllowOnlyFilter is a new filter to authorize only requests that satisfy a set of rules based on the provenance, destination, and additional conditions of the request. When the rules are not satisfied, the request is rejected. For more information, see "AllowOnlyFilter" in the *Configuration Reference*.

+ *Financial API grade security*

- **FapiInteractionIdFilter** is a new filter to track the interaction ID of requests, according to the Financial-grade API (FAPI) WG. For more information, see "FapiInteractionIdFilter" in the *Configuration Reference*.
- **DateHeaderFilter** is a new filter to insert the server date in an HTTP date header on the response. For more information, see "DateHeaderFilter" in the *Configuration Reference*.

+ *SetCookieUpdateFilter to update cookie attributes*

SetCookieUpdateFilter is a new filter to update cookie attributes. Use SetCookieUpdateFilter for legacy applications, where cookies do not conform to requirements for newer browsers. For more information, see "SetCookieUpdateFilter" in the *Configuration Reference*.

+ *Temporary storage files in custom directory*

By default, IG writes temporary files to `$HOME/.openig/tmp`. You can now change the directory by setting the `temporaryDirectory` property in `admin.json`.

For information, see "AdminHttpApplication (`admin.json`)" in the *Configuration Reference*.

+ *Multiple OIDC providers use same clientID*

	In OpenID Connect with multiple client registrations, the same `clientId` can now be used for multiple client registrations if if the `issuerName` for each registration is different. The `clientId` must be unique in the context of a single issuer. In the OAuth2ClientFilter login service URI, specify both the `clientId` and the `issuerName`, For more information, see "OAuth2ClientFilter" in the *Configuration Reference* and "ClientRegistration" in the *Configuration Reference*, "Issuer" in the *Configuration Reference*. + *sameSite flag maintained* Cookies that arrive at IG with the `sameSite` flag set are correctly maintained. + *Skew allowance in StatelessAccessTokenResolver* The property `skewAllowance` has been added to the StatelessAccessTokenResolver to manage the validity period of access_tokens. For information, see "StatelessAccessTokenResolver" in the *Configuration Reference*.
OAuth 2.0, to separate API security concerns from business concerns	+ *Caching OAuth 2.0 access tokens* **CacheAccessTokenResolver** is a new object to enable and configure caching of OAuth 2.0 access_tokens, based on *Caffeine*. For more information, see "CacheAccessTokenResolver" in the *Configuration Reference*. + *mTLS through HTTP headers for certificate-bound access tokens* IG 7 adds support for draft 12 of the OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens specification, a key component of ForgeRock's Open Banking and Revised Payment Services Directive (PSD2) support. **CertificateThumbprintFilter** is a new filter to verify of certificate-bound access_tokens. **ConfirmationKeyVerifierAccessTokenResolver** is a new access token resolver to verify that certificate-bound OAuth 2.0 bearer tokens presented by clients use the same mTLS-authenticated HTTP connection. Use these objects when IG is running behind a TLS termination point, such as a load balancer or other ingress point.

For more information, see "CertificateThumbprintFilter" in the *Configuration Reference*, "ConfirmationKeyVerifierAccessTokenResolver" in the *Configuration Reference*, and "Validating Certificate-Bound Access Tokens" in the *Gateway Guide*.

+ *OAuth 2.0 client credentials filter*

ClientCredentialsOAuth2ClientFilter is a new filter to authenticate a client, using the client's OAuth 2.0 credentials. Use this filter in a service-to-service context, where a service needs to access resources protected by OAuth 2.0.

The filter obtains an access_token from an authorization server, and injects the access_token into the inbound request as a Bearer Authorization header, and refreshes the access_token as required. For information, see "ClientCredentialsOAuth2ClientFilter" in the *Configuration Reference*.

+ *Discovery and dynamic registration using private_key_jwt*

The `private_key_jwt` authentication method can now be used for authentication during discovery and dynamic registration with an OpenID Connect provider. In previous releases, only `client_secret_basic` and `client_secret_post` authentication methods could be used.

For more information, see "OAuth2ClientFilter" in the *Configuration Reference*, and the example in "Discovering and Dynamically Registering With OpenID Connect Providers" in the *Gateway Guide*.

Secrets

+ *SecretsProvider provides improved control over where to search for secrets*

SecretsProvider is an updated heap object to specify a secrets service to resolve secrets for IG configuration objects, using the property `secretsProvider`.

For backward compatibility, if SecretsProvider is not configured, objects use the global secrets service, which searches for keys across the whole configuration. If multiple keys have the same label, there is a bigger risk that the wrong key is used.

For information, see "SecretsProvider" in the *Configuration Reference*.

+ *New secret stores*

- **JwkSetSecretStore** is a new secret store for JSON Web Keys (JWK) in a JWK Set. For information, see "JwkSetSecretStore" in the *Configuration Reference*.
- **Base64EncodedSecretStore** is a new secrets store for generic secrets, such as passwords or simple shared secrets, whose base64-encoded

values are hard-coded in the route. Use this store for testing or evaluation only. In production, use an alternative secret store. For information, see "Base64EncodedSecretStore" in the *Configuration Reference*.

+ *Secret formats*

SecretsKeyPropertyFormat is to define the format and algorithm used for the secrets.

Use this object with FileSystemSecretStore or SystemAndEnvSecretStore, when symmetric keys are provided in files, environment variables, or system properties, by external secret management systems, such as Kubernetes Secrets or Docker Secrets. In previous releases, symmetric keys had to be declared in a KeystoreSecretStore.

+ *KeyManager and TrustManager provided as secrets*

- **SecretsKeyManager** is available for IG in standalone mode to provide secrets for KeyManager. Use with ClientTlsOptions and ServerTlsOptions to prove the identity of the local peer during TLS handshake. For more information, see "SecretsKeyManager" in the *Configuration Reference* and
- **SecretsTrustManager** is available for IG in standalone mode to provide secrets for TrustManager. Use with ClientTlsOptions and ServerTlsOptions to manage trust material for peer credentials.

  For more information, see "SecretsTrustManager" in the *Configuration Reference*.

+ *Secrets in FileSystemSecretStore or SystemAndEnvSecretStore for signing and encryption*

Secrets stored in a FileSystemSecretStore or SystemAndEnvSecretStore can now be used for symmetric signing keys and symmetric encryption keys. In previous releases, keys had to be declared in a KeystoreSecretStore.

For more information, see the `mappings` property of "FileSystemSecretStore" in the *Configuration Reference* and "Packing Data Into a JWT Signed With a Symmetric Key" in the *Configuration Reference*.

+ *Secrets in CDSSO*

IG can now verify the signature of signed CDSSO tokens in cross-domain single sign-on.

	For information, see the properties `verificationSecretId` and `secretsProvider` in "CrossDomainSingleSignOnFilter" in the *Configuration Reference*. + *Deterministic ECDSA for JWT signatures* When elliptic curve keys are used for signing, and Bouncy Castle is installed, and, by default, JWTs are signed with a deterministic ECDSA. In previous releases, JWTs were signed with a non-deterministic ECDSA, which is less secure. The new system property `org.forgerock.secrets.preferDeterministicEcdsa` is by default `true`. To use the less secure algorithm, set the property to `false`. For more information, see "Algorithms for Elliptic Curve Digital Signatures" in the *Gateway Guide*.
HTTP sessions	+ *Authenticated encryption for stateless sessions* JWT tokens can now be secured by authenticated encryption with symmetric keys. There is now no need to sign these JWTs as a separate step, leaving more space for session data. Before this release, JWT tokens could only be encrypted *and then* signed. For information, see "JwtSession" in the *Configuration Reference*. + *Support for SameSite Cookies* **sameSite** is a new property in CrossDomainSingleSignOnFilter and JwtSession to manage the circumstances in which a cookie is sent to the server. Use this property to manage the risk of cross-site request forgery (CSRF) attacks. For information, see the `authCookie` property of "CrossDomainSingleSignOnFilter" in the *Configuration Reference*, or the `cookie` property of "JwtSession" in the *Configuration Reference*. + *SetCookieUpdateFilter* **SetCookieUpdateFilter** is a new filter to change the attributes of generated cookies.
Stability	+ *Session attributes retrieved without AM session properties whitelist* IG can now retrieve specified session properties or all session properties from AM, without relying on AM's Session Properties Whitelist. Properties with a value are returned; properties with a null value are not returned.

	In previous releases, only whitelisted session properties were returned, irrespective of whether they had a value. For information, see `sessionProperties` in "AmService" in the *Configuration Reference*.
	+ *Infinite loop prevented in SSO* The SingleSignOnFilter has been adapted to prevent an infinite loop when a final redirect is returned without an AM session cookie name. For information, see "SingleSignOnFilter" in the *Configuration Reference*, and "CrossDomainSingleSignOnFilter" in the *Configuration Reference*.
	+ *Skew allowance in JwtSession* A new property `skewAllowance` has been added to JwtSession to manage small differences in system clocks. For information, see "JwtSession" in the *Configuration Reference*.
	+ *Improved protection from MIME sniffing* To help prevent MIME sniffing of responses from the StaticResponseHandler, the `X-Content-Type-Options` response header is now set by default to `nosniff`. In previous releases, the header was not set, allowing the user agent to interpret the response entity as a different content type. For information about how to protect against cross-site scripting, see "General Security Considerations" in the *Maintenance Guide*.
	+ *Ping endpoint* A ping endpoint is available after IG startup to check whether IG is available. When IG is installed and running as described in the Getting Started Guide, the endpoint is at `http://openig.example.com:8080/openig/ping`.
Cloud Readiness	+ *Global log level configurable through a variable* To make it easier to deploy IG without modifying the default configuration, the global log level is now defined as a variable in the default `logback.xml`. To change the global log level, set an environment variable or system property.

	For information, see "Changing the Global Log Level" in the *Maintenance Guide*.
Studio	+ *Freeform Designer* Freeform Designer has moved from `Technology Preview` to `Stable`, as defined in "*Release Levels and Interface Stability*". The Studio Welcome page has been replaced by the Routes page. + *Global decoration of routes in Studio* The `globalDecorators` property can now be configured for a route in Studio.
Logs and Audits	+ *Whitelisting for audit event fields in logs* To prevent logging of sensitive data for an event, the Common Audit Framework uses a whitelist to specify which event fields appear in logs. By default, only event fields that are whitelisted are included in the logs. For more information, see "Safelisting Audit Event Fields for the Logs" in the *Maintenance Guide*. + *Audit of custom events* IG can now record custom audit events as well as `access` audit events. For an example of how to configure custom audit events, see "Recording Custom Audit Events" in the *Gateway Guide*. For information about auditing, see "*Audit Framework*" in the *Configuration Reference*. + *AuditClientHandler for Splunk* You can now configure a client handler named `AuditClientHandler` in the heap, to relay audit events to Splunk. If a client handler named `SplunkClientHandler` is configured in the heap, it is used by priority. For information, see "SplunkAuditEventHandler" in the *Configuration Reference*. + *NoOpAuditService provided by default*

	NoOpAuditService is a new audit service to add an empty audit service to the top-level heap and its child routes. When an AuditService is not defined, auditing is delegated to the parent audit service. For more information, "NoOpAuditService" in the *Configuration Reference*.
Core	+ *URI path rewriting* **UriPathRewriteFilter** is a new filter to rewrite the path of a request URL. Use this filter to expose applications that are on a different path. Continue to use `baseURI` to override the scheme, host, and port of a request URL. UriPathRewriteFilter does not re-write the content of a message. For more information, see "UriPathRewriteFilter" in the *Configuration Reference*. + *ResourceHandler serves static content* **ResourceHandler** is a new handler to serve static content from a directory. In previous releases, IG could not serve static content so easily. For information, see "ResourceHandler" in the *Configuration Reference*. + *IG agent in AM* AM now provides a simplified process to create an agent profile for IG. When the IG agent is authenticated, the token can be used for tasks such as getting the user's profile, making policy evaluations, and connecting to the AM notification endpoint. Procedures in the Gateway Guide that previously used a Java agent in AM now use a the new profile for an IG agent in AM. + *toJson function* The `toJson` function has been added for expressions. For information, see "toJson" in the *Configuration Reference*.
IG-X Foundations	+ *JDBC data source configured outside container* JDBC data sources can now be set up independently of the web container configuration. In previous releases, JDBC data sources were configured at the web container level. For more information, see "JdbcDataSource" in the *Configuration Reference*.
Others	+ *SingleSignOnFilter logout can be triggered by any aspect of a request*

The new SingleSignOnFilter property `logoutExpression` can trigger logout based on any aspect of a request. Before this improvement, logout could be triggered only when a request matched the URI path.

For information, see "SingleSignOnFilter" in the *Configuration Reference*.

+ *CrossDomainSingleSignOnFilter logout triggered by any aspect of request*

New CrossDomainSingleSignOnFilter properties, `logoutExpression` and `defaultLogoutLandingPage`, are available to trigger logout of the associated AM session token based on any aspect of a request.

For information, see "CrossDomainSingleSignOnFilter" in the *Configuration Reference*.

+ *CaptureDecorator for mask header and attribute values*

The CaptureDecorator can now be configured to mask the value of headers and attributes in the logs. Use this feature to prevent disclosure of sensitive information in the logs.

For more information, see "CaptureDecorator" in the *Configuration Reference*.

+ *SAML support for all name ID formats*

In SAML SP-initiated SSO, IG can now act as an SP with an IDP that does not support the `transient` NameID Format. For SP-initiated SSO as well as for IDP-initiated SSO, the NameID Format can be any format supported by the IDP.

In previous releases, for SP-initiated SSO, the NameID Format could be only `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`.

For more information, see "Using a Non-Transient NameID Format" in the *Gateway Guide*.

+ *Eviction of revoked OAuth 2.0 access tokens*

(From AM 6.5.3.) The CacheAccessTokenResolver and OAuth2ResourceServerFilter can now receive a notification when AM revokes an OAuth 2.0 access_token, and can evict the token from the cache.

For information, see "CacheAccessTokenResolver" in the *Configuration Reference*, and the `cache` property of "OAuth2ResourceServerFilter" in the *Configuration Reference*.

+ *ipMatch function*

The function `ipMatch()` is added to check whether an IP address matches an IP range.

For information, see "ipMatch" in the *Configuration Reference*.

+ *Spaces maintained in cookie values*

As a cookie passes through IG, if the cookie value is not enclosed in quotes, spaces in the cookie value are not removed. In previous releases, spaces were removed.

+ *Class imported automatically for Groovy scripts*

The `org.forgerock.http.header` class is now imported automatically for Groovy scripts.

+ *Enhanced command-line for sample application*

When you launch the sample application, new command-line options are available to configure the ports, session timeout, AM URL base for the OpenID provider configuration, and help display. For more information, see "Configuration Options for the Sample Application" in the *Getting Started Guide*.

+ *ResourceException Not Logged at Error Level when AM Returns 401*

Previously, if the user's SSO session had expired or become otherwise invalid and was used in a request to IG, calling the AM session info endpoint to get session status would return a 401 response. This 401 response was valid but ended up being logged by IG at Error level, which was misleading, and would generate a large amount of additional logging data.

IG now logs an error message only when the response from an AM session info endpoint is not a 401. IG still logs it as a debug message to show that it was a 401 response.

+ *Warning if decoded secret starts or ends in a non-ascii character*

IG logs a warning when the decoded value of a BASE64-encoded secret starts or ends with a non-ASCII character.

If a text editor adds a carriage return to the end of a plain string value before it is encoded, non-ASCII characters can be added to the BASE64-encoded value. When the decoded value is used as part of a username/password exchange, it can then cause an authentication error.

+ *New functions for URL-safe and filename-safe encoding and decoding*

The functions `encodeBase64url` and `decodeBase64url` are added to facilitate URL-safe and filename-safe encoding and decoding.

For information, see "encodeBase64url" in the *Configuration Reference* and "decodeBase64url" in the *Configuration Reference*.

+ *ConnectionFactory heartbeat can be disabled*

The heartbeat of the ConnectionFactory used in the `org.forgerock.openig.ldap.LdapClient`, enabled by default, can now be disabled. In previous releases, it could not be disabled.

For an example of how to disable the ConnectionFactory heartbeat, see "Authenticate to an LDAP Server" in the *Gateway Guide*.

+ *URL query strings preserved*

A new property in `admin.json` lets you preserve query strings as they are presented in URLs. Select this option when query strings must not change during processing; for example, in signature verification.

By default, IG tolerates characters that are disallowed in query string URL components, by applying a decode/encode process to the whole query string.

For information, see `preserveOriginalQueryString` in "AdminHttpApplication (`admin.json`)" in the *Configuration Reference*.

+ *Option to reuse connections after a request*

Not supported for IG in standalone mode.

stateTrackingEnabled is a new property of ClientHandler and ReverseProxyHandler to specify whether a connection can be kept open and reused after a request.

For information, see "ClientHandler" in the *Configuration Reference* or "ReverseProxyHandler" in the *Configuration Reference*.

+ *Join function can process an iterable string*

	The `join` function can now return a string joined with the given separator, from an Iterable value. In previous releases, it used only an array of string values. For more information, see "join" in the *Configuration Reference*.

Chapter 2

Requirements

Important

ForgeRock supports customers using the versions specified here. Other versions and alternative environments might work as well. When opening a support ticket for an issue, however, make sure you can also reproduce the problem on a combination covered here.

Downloads

Download the following product software from the ForgeRock BackStage download site:

- `IG-7.0.2.zip`: For deployment in standalone mode
- `IG-7.0.2.war`: For deployment in web container mode
- `IG-sample-application-7.0.2.jar`: Web application for testing IG configurations

For information about using the Docker image provided with the product software, see the Deployment Guide.

Operating Systems

IG is tested on Windows and Linux operating systems.

Web Application Containers

In web container mode, IG runs in the following containers:

- Apache Tomcat 9
- Jetty 9
- JBoss EAP 7.3

Deploy IG to the root context of a container. Deployment in other contexts causes unexpected results, and is not supported.

Java

IG supports the following Java environments:

Supported Java Versions

Vendor	Versions
OpenJDK, including OpenJDK-based distributions: • AdoptOpenJDK/Eclipse Adoptium • Amazon Corretto • Azul Zulu • Red Hat OpenJDK ForgeRock tests most extensively with AdoptOpenJDK/Eclipse Adoptium.	11
Oracle Java	11

ForgeRock recommends that you keep your Java installation up-to-date with the latest security fixes.

FQDNs

IG replication requires use of fully qualified domain names (FQDNs), such as `openig.example.com`.

Hostnames like `example.com` are acceptable for evaluation. In production, and when using replication across systems, you must either ensure DNS is set up correctly to provide FQDNs, or update the hosts file (`/etc/hosts` or `C:\Windows\System32\drivers\etc\hosts`) to supply unique, FQDNs.

Certificates

For secure network communications with client applications that you do not control, install a properly signed digital certificate that your client applications recognize, such as one that works with your organization's PKI, or one signed by a recognized CA.

To use the certificate during installation, the certificate must be located in a file-based keystore supported by the JVM (JKS, JCEKS, PKCS#12), or on a PKCS#11 token. To import a signed certificate into the server keystore, use the Java **keytool** command.

Third-Party Software for Encryption

Bouncy Castle is required for signature encryption with RSASSA-PSS or Deterministic ECDSA. For information, see *The Legion of the Bouncy Castle*.

Third-Party Software

ForgeRock provides support for using the following third-party software when logging ForgeRock Common Audit events:

Software	Version
Java Message Service (JMS)	2.0 API
MySQL JDBC Driver Connector/J	8 (at least 8.0.19)
Splunk	8.0 (at least 8.0.2)

Tip

Elasticsearch and Splunk have native or third-party tools to collect, transform, and route logs. Examples include Logstash and Fluentd.

ForgeRock recommends that you consider these alternatives. These tools have advanced, specialized features focused on getting log data into the target system. They decouple the solution from the ForgeRock Identity Platform systems and version, and provide inherent persistence and reliability. You can configure the tools to avoid losing audit messages if a ForgeRock Identity Platform service goes offline, or delivery issues occur.

These tools can work with ForgeRock Common Audit logging:

- Configure the server to log messages to standard output, and route from there.
- Configure the server to log to files, and use log collection and routing for the log files.

ForgeRock provides support for using the following third-party software when monitoring ForgeRock servers:

Software	Version
Grafana	5 (at least 5.0.2)
Graphite	1
Prometheus	2.0

For hardware security module (HSM) support, ForgeRock software requires a client library that conforms to the PKCS#11 standard v2.20 or later.

Studio Browser

ForgeRock has tested many browsers with Studio, including:

- Chrome, latest stable version
- Firefox, latest stable version

Features Using ForgeRock Access Management

Feature	Supported in AM Version
Support for refresh of idle sessions when the SingleSignOnFilter is used for authentication with AM. For more information, see the `sessionIdleRefresh` property of "AmService" in the *Configuration Reference*.	AM 6.5.3 and later versions.
Eviction of revoked OAuth 2.0 access_tokens from the cache. For more information, see "CacheAccessTokenResolver" in the *Configuration Reference*, and the `cache` property of "OAuth2ResourceServerFilter" in the *Configuration Reference*.	AM 6.5.3 and later versions.
Support for OAuth 2.0 Mutual TLS (mTLS). For more information, see "ConfirmationKeyVerifierAccessTokenResolver" in the *Configuration Reference*, and "Validating Certificate-Bound Access Tokens" in the *Gateway Guide*.	AM 6.5.1 and later versions.
Eviction of entries from the AmService `sessionCache`, using WebSocket notifications from AM. For more information, see "AmService" in the *Configuration Reference*.	AM 5.5 when the user manually whitelists the `AMCtxId` session property, and with AM 6 and later versions.
AM password capture and replay, as described in "*Getting Login Credentials From AM*" in the *Gateway Guide*.	AM 5 and later versions, and AM 6 and later versions when the `AES` keyType is used to decrypt the password.
AM policy enforcement, as described in "*Enforcing Policy Decisions From AM*" in the *Gateway Guide*.	AM 5 and later versions
OpenID Connect dynamic registration and discovery, as described in "Discovering and Dynamically Registering With OpenID Connect Providers" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions
Token transformation, as described in "*Transforming OpenID Connect ID Tokens Into SAML Assertions*" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions
User Managed Access 2.x, for IG 5.5, as described in "*Supporting UMA Resource Servers*" in the *Gateway Guide*.	AM 5.5 and later versions
User Managed Access 1.x, for IG 5 and earlier versions.	AM 5.1 and earlier versions

Feature	Supported in AM Version
Single sign-on, as described in "*Single Sign-On and Cross-Domain Single Sign-On*" in the *Gateway Guide*.	AM 5 and later versions
Cross-domain single sign-on, as described in "Authenticating With CDSSO" in the *Gateway Guide*.	AM 5.5 and later versions
Capture and storage of AM session information, as described in "SessionInfoFilter" in the *Configuration Reference*.	AM 6 and later versions
Capture and storage of AM user profile attributes, as described in "UserProfileFilter" in the *Configuration Reference*.	AM 5 and later
Support for transactional authorization, as described in "*Hardening Authorization With Advice From AM*" in the *Gateway Guide*.	AM 5.5 and later versions
Validation of stateless access_tokens, as described in "Validating Stateless Access_Tokens With the StatelessAccessTokenResolver" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions
Retrieval of specified session properties or all session properties from AM, without relying on AM's Session Properties Whitelist. Described in "AmService" in the *Configuration Reference*.	AM 5.1.2 and later versions

Chapter 3
Migration

IG is delivered in this release as a standalone Java executable in a .zip file, as well as in a .war file. Consider these points to migrate from IG in web container mode to IG in standalone mode.

Session replication between IG instances

Use only stateless sessions, as described in "Sessions" in the *Gateway Guide*. Stateful sessions are not supported.

Streaming of asynchronous responses and events

In ClientHandler and ReverseProxyHandler, use only the default mode of `asyncBehavior`:`non_streaming`; responses are processed when the entity content is entirely available. If the property is set to `streaming`, the setting is ignored. For information, see "ClientHandler" in the *Configuration Reference* and "ReverseProxyHandler" in the *Configuration Reference*.

Connection reuse when a client certificate is used for authentication

In ClientHandler and ReverseProxyHandler, use only the default mode of `stateTrackingEnabled`:`true`; when a client certificate is used for authentication, connections cannot be reused. If the property is set to `false`, the setting is ignored. For information, see "ClientHandler" in the *Configuration Reference* and "ReverseProxyHandler" in the *Configuration Reference*.

Tomcat configuration alternative

Port number

- **Tomcat**: Configure in the `Connector` element of `/path/to/tomcat/conf/server.xml`:

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

- **Standalone**: Configure the `connectors` property `$HOME/.openig/config/admin.json`, where `$HOME/.openig` is the instance directory. For information, see "AdminHttpApplication (`admin.json`)" in the *Configuration Reference*.

Configuring IG for HTTPS server-side

- **Tomcat**: Create a keystore, and set up the SSL port in the `Connector` element of `/path/to/tomcat/conf/server.xml`. For information, see "Configuring IG for HTTPS (Server-Side) in Tomcat" in the *Gateway Guide*.

- **Standalone**: Create a keystore, set up secrets, and configure secrets stores, ports, and ServerTlsOptions in `$HOME/.openig/config/admin.json`, where `$HOME/.openig` is the instance directory. For information, see "Configuring IG For HTTPS (Server-Side)" in the *Gateway Guide*.

Session cookie name

- **Tomcat**: Configure `WEB-INF/web.xml` when you unpack the IG .war file.
- **Standalone**: Configure the `session` property of `$HOME/.openig/config/admin.json`, where `$HOME/.openig` is the instance directory. For information, see "AdminHttpApplication (`admin.json`)" in the *Configuration Reference*.

Access logs

- **Tomcat**: Configure with AccessLogValve.
- **Standalone**: Configure in the Audit framework. For information, see "*Auditing Your Deployment*" in the *Maintenance Guide* and "*Audit Framework*" in the *Configuration Reference*.

JDBC datasource

- **Tomcat**: Configure in the `GlobalNamingResources` element of `/path/to/tomcat/conf/server.xml`.
- **Standalone**: Configure with the JdbcDataSource object. For information, see "JdbcDataSource" in the *Configuration Reference*. For an example, see "Logging In With Credentials From a Database" in the *Gateway Guide*.

Environment variables

- **Tomcat**: Configure in `/path/to/tomcat/bin/setenv.sh`, where `$HOME/.openig` is the instance directory.
- **Standalone**: Configure in `$HOME/.openig/bin/env.sh`, where `$HOME/.openig` is the instance directory.

Jar files

- **Tomcat**: Add to to web container classpath; for example `/path/to/tomcat/webapps/ROOT/WEB-INF/lib`.
- **Standalone**: Add to `$HOME/.openig/extra`, where `$HOME/.openig` is the instance directory.

Chapter 4

Incompatible Changes

Incompatible Changes in IG 7.0.2

+ *Validation of `goto` Parameter in OAuth2ClientFilter*

To prevent redirects to malicious web sites, IG now validates the `goto` query parameter in requests to OAuth2ClientFilter `/login` and `/logout` endpoints.

The goto URL must use the same scheme, host, and port as the original URI, or be a relative URI (just the path). Otherwise, the request fails with an error. To redirect a request to a site that does not meet the goto URL criteria, change the original URI by using a ForwardedRequestFilter.

For more information, see "OAuth2ClientFilter" in the *Configuration Reference* and "ForwardedRequestFilter" in the *Configuration Reference*.

Incompatible Changes in IG 7.0.1

No incompatible changes have been introduced in this release.

Incompatible Changes in IG 7.0.0

The following changes introduced in this release can impact your migration from IG 6.5:

+ *SAML 2.0 Deployments Require Additional Configuration*

When IG uses AM federation libraries generated from AM 6.5.2 or earlier, add the following lines to the `FederationConfig.properties` file:

```
# Specifies implementation for
    # org.forgerock.openam.federation.plugin.rooturl.RootUrlProvider interface.
    # This property defines the default base url provider.

 com.sun.identity.plugin.root.url.class.default=org.forgerock.openam.federation.plugin.rooturl.impl.FedletRootU
```

+ *Content-Type is a required header when entity used in StaticResponseHandler*

When `entity` is used in the StaticResponseHandler, `Content-Type` is a required header. In previous releases, `Content-Type` was optional.

For an example configuration, see the `headers` property of "StaticResponseHandler" in the *Configuration Reference*.

+ *Java 11 required*

IG 7.0 requires Java 11. Java 8 is not supported.

+ *Connections to DS secure by default*

ForgeRock Directory Services (DS) is now secure by default. Connections between IG and DS must therefore be configured for TLS.

+ *Groovy 3.0*

IG now supports Groovy 3.0. For information about the Groovy version, see the *Groovy Documentation*.

+ *JwtSessionFactory not an alternative type for JwtSession*

JwtSessionFactory is no longer an alternative type for JwtSession.

+ *Default value of skew allowance in JwtSession*

The default skew allowance in JwtSession has been reduced from 2 minutes to zero, and a property to configure the skew allowance has been added in JwtSession. For information, see "JwtSession" in the *Configuration Reference*.

+ *KeyStore and KeyStoreSecretStore default type based on keystore extension*

Oracle recommends the use of PKCS12 keystores. From Java 9, Oracle has provided more support for PKCS12. From Java 11, Oracle has changed the default keystore to PKCS12.

Following this lead, the default type for KeyStore and KeyStoreSecretStore is now based on the keystore extension. If the keystore extension is not recognized, the default type is PKCS12. In previous releases, the default type was the one used by the platform.

To ensure backward-compatibility, where keys are generated using a non-PKCS12 type (for example, JKS), specify `type` in KeyStore or `storeType` in KeyStoreSecretStore.

For information, see "KeyStore" in the *Configuration Reference* and "KeyStoreSecretStore" in the *Configuration Reference*.

+ *OAuth2ResourceServerFilter doesn't check access_token expiry*

In previous releases, after an access_token resolver validated an access_token, the OAuth2ResourceServerFilter checked that the access_token was not expired. From this release, the OAuth2ResourceServerFilter considers any token returned by an AccessTokenResolver as valid, and checks only that the required scopes are present.

For information, see "OAuth2ResourceServerFilter" in the *Configuration Reference*.

+ *gracefulStop In ScheduledExecutorService*

When `gracefulStop` is true, the ScheduledExecutorService now removes submitted jobs and attempts to end running jobs, after respecting the `gracePeriod`. In previous releases, when `gracefulStop` was true, it did not remove or end jobs.

For information, see "ScheduledExecutorService" in the *Configuration Reference*.

+ *Whitelisting of audit event fields in logs*

To prevent logging of sensitive data for an event, the Common Audit Framework now uses a whitelist to specify which event fields appear in logs. Compared to previous releases, different event fields are included by default in the logs.

The AuditService `includeIf` property has been implemented to include non-whitelisted event fields in the logs. For information about how to include or exclude event fields, see "Safelisting Audit Event Fields for the Logs" in the *Maintenance Guide*.

+ *Identification of OAuth2ClientFilter registrations*

In OAuth2ClientFilter, registrations are now identified by the ClientRegistration property `clientId` instead of `name`. In this release, IG automatically rewrites OAuth2Session tokens that use `name` to use `clientId`. Registration by `name` will be removed in a later release.

When a user initiates a login with the OAuth2ClientFilter, the login endpoint uses the ClientRegistration property `clientId`:

```
{clientEndpoint}/login?registration={clientId}[&goto={url}]
```

In previous releases, the login endpoint used the ClientRegistration property `name`:

```
{clientEndpoint}/login?registration={name}[&goto={url}]
```

Similarly, the login endpoint in Nascar pages uses `client_id` instead of `name`.

For information, see "ClientRegistration" in the *Configuration Reference*, and the example route in "Using Multiple OpenID Connect Providers" in the *Gateway Guide*.

+ *SplunkClientHandler*

A client handler named `SplunkClientHandler` can now be declared in the heap of a route that uses a SplunkAuditEventHandler. The client handler relays audit events to Splunk.

In previous releases, it was necessary to configure a client handler named `ElasticsearchClientHandler`, or use the route's default client handler. For more information, see "SplunkAuditEventHandler" in the *Configuration Reference*.

Chapter 5
Deprecation

Deprecated is defined in "ForgeRock Product Stability Labels".

Deprecated Functionality in IG 7.0.2

No additional functionality has been deprecated in this release.

Deprecated Functionality in IG 7.0.1

No additional functionality has been deprecated in this release.

Deprecated Functionality in IG 7.0.0

The following features and properties are deprecated, and likely to be removed in a future release:

+ *Delivery of IG war file*

  The delivery of a .war file is deprecated in this release and may be removed in the next release.

+ *IG route monitoring endpoint*

  The IG Route Monitoring Endpoint is deprecated and will be removed in a later release. As a replacement, IG provides Prometheus Scrape Endpoint and Common REST Monitoring Endpoint.

  For more information, see "Monitoring at the Prometheus Scrape Endpoint" in the *Maintenance Guide*, and "Monitoring the Common REST Monitoring Endpoint" in the *Maintenance Guide*,

Configuration Object	Deprecated Settings	Replacement Settings
AmService	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.

Configuration Object	Deprecated Settings	Replacement Settings
AuditService	`event-handlers`	Replaced by `eventHandlers`.
CapturedUserPasswordFilter	`key`	Replaced by `keySecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
ClientHandler	`proxy` subproperty `password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	• `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see "ClientTlsOptions" in the *Configuration Reference*.
	`websocket` subproperties: • `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see "ClientTlsOptions" in the *Configuration Reference*.
ClientRegistration	• `keystore` • `privateKeyJwtAlias` • `privateKeyJwtPassword`	Replaced by `privateKeyJwtSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`name`, when used to identify a `registration`	Replaced by `clientId`. For information, see "ClientRegistration" in the *Configuration Reference*, and the example route in "Using Multiple OpenID Connect Providers" in the *Gateway Guide*.
	`clientSecret`	Replaced by `clientSecretId`.

Configuration Object	Deprecated Settings	Replacement Settings
		If the deprecated and replacement properties are both provided, the replacement property takes precedence.
CryptoHeaderFilter	Whole object	Not replaced. For information, see "CryptoHeaderFilter" in the *Configuration Reference*.
DesKeyGenHandler	Whole object	Not replaced. For information, see "DesKeyGenHandler" in the *Configuration Reference*.
JwtBuilderFilter	`signature` subproperties: • `keystore` • `alias` • `password`	Replaced by `signature` property `secretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
JwtSession	`encryptionSecretId` and `signatureSecretId`	Replaced by `authenticatedEncryptionSecretId` and `encryptionMethod`.
	`cookieName` and `cookieDomain`	Replaced by `cookie`, and its subproperties `name`, `domain`, `path`, `secure`, `httpOnly`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`password`	Replaced by `passwordSecretId` If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	Combination of `password`, `alias`, and `keystore` Combination of `passwordSecretId`, `alias`, and `keystore`	Replaced by `encryptionSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`sharedSecret`	Replaced by `signatureSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
KeyManager	`password`	Replaced by `passwordSecretId`.

Configuration Object	Deprecated Settings	Replacement Settings
		If the deprecated and replacement properties are both provided, the replacement property takes precedence.
KeyStore	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
OpenAmAccessTokenResolver	Whole object	Not replaced. For information, see "OpenAmAccessTokenResolver" in the *Configuration Reference*.
ReverseProxyHandler	• `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see "ClientTlsOptions" in the *Configuration Reference*.
	`websocket` subproperties: • `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see "ClientTlsOptions" in the *Configuration Reference*.
Route	`monitor`	Replaced by the Prometheus Scrape Endpoint and Common REST Monitoring Endpoint. For information, see "Monitoring Endpoints" in the *Configuration Reference*.
SingleSignOnFilter	`logoutEndpoint`	Replaced by `logoutExpression`.
SqlAttributesFilter	`dataSource` as a JNDI lookup name	Replaced by `dataSource` as a `JdbcDataSource` configuration object.
StatelessAccessTokenResolver	`signatureSecretId`	Replaced by `verificationSecretId`.
	`encryptionSecretId`	Replaced by `decryptionSecretId`.
UserProfileFilter	`ssoToken`	Replaced by `username` in UserProfileFilter.

Configuration Object	Deprecated Settings	Replacement Settings
	`amService` and `profileAttributes`	Replaced `amService` and `profileAttributes`, as sub-properties of `userProfileService`
The environment variable and system property that define the file system directory for configuration files.	`OPENIG_BASE` and `openig.base`	Replaced by `IG_INSTANCE_DIR` and `ig.instance.dir`. If neither the deprecated setting nor the replacement setting are provided, configuration files are in the default directory `$HOME/.openig` (on Windows, `%appdata%\OpenIG`). If the deprecated setting and the replacement setting are both provided, the replacement setting is used.

Chapter 6

Removed

Removed is defined in "ForgeRock Product Stability Labels".

Removed Functionality in IG 7.0.2

No functionality was removed in this release.

Removed Functionality in IG 7.0.1

No functionality was removed in this release.

Removed Functionality in IG 7.0.0

The following features and properties were removed in this release.

+ *Support for AM policy agents*

  Support for the use of AM policy agents in password capture and replay is removed in this release.

  By using CapturedUserPasswordFilter, you can get login credentials from AM without setting up an AM policy agent. For more information, see "*Getting Login Credentials From AM*" in the *Gateway Guide*, and "CapturedUserPasswordFilter" in the *Configuration Reference*.

Configuration Object	Removed Settings	Newer Evolving Settings
ClientRegistration	Deprecated previously, removed in this release: `keyStore`.	Previously replaced by `keystore`.
OAuth2ResourceServerFilter	Deprecated previously, removed in this release: `cacheExpiration`.	Previously replaced by `cache` and its sub-properties `enabled`, `defaultTimeout`, and `maxTimeout`.
PolicyEnforcementFilter	Deprecated previously, removed in this release: `cache` subproperty `maxTimeout`.	Previously replaced by `cache` property `maximumTimeToCache`.

Chapter 7

Fixes

Fixes in IG 7.0.2

The following important issues were fixed in this release:

- OPENIG-5084: WebSocket connections are not being proxied when baseURI scheme is wss
- OPENIG-5219: Vert.x HTTP Client does not replicate current CHF behaviour when request fails and headers have been received
- OPENIG-5258: IG Standalone must populate the originalUri.port from Host header

Fixes in IG 7.0.1

The following important issues were fixed in this release:

- OPENIG-4034: AuditService does not delete old files when maxDiskSpaceToUse is reached
- OPENIG-4900: AMService cannot connect to AM via TLS with Standalone

Fixes in IG 7.0.0

The following important issues were fixed in this release:

- OPENIG-3221: OpenIG is decoding special character ' while sending to the backend which is causing issues
- OPENIG-3275: SamlFederationHandler Doesn't Support Filtering
- OPENIG-3296: UserProfileFilter and usernames with colons
- OPENIG-3403: ContentTypeHeader quoted directives should be maintained
- OPENIG-3488: IG fails to stop when started with a config.json with invalid json syntax.
- OPENIG-3492: Request and response logged in different files when capture:all and global captureDecorator are in config.json
- OPENIG-3659: SSOFilter logoutEndpoint does not take query parameters into consideration

- OPENIG-3755: IG's decodeBase64 function returns null on JWTs generated by IG or AM
- OPENIG-3783: ClassCastException in scriptable access token resolver occurs when invalid token is returned by delegated access token resolver
- OPENIG-3819: WebSocket requests should be built using the raw query parameters
- OPENIG-3837: WebSocketAdapter#writeBuffersIfStreamIsReady should check if stream is ready before calling flush
- OPENIG-4037: Global decorators declared in a route cannot refer to decorators declared in the same route
- OPENIG-4168: CacheAccessTokenResolver : missing requests to amService (not available in capture)
- OPENIG-4190: A WebSocket Origin header is missing the scheme from the URL

Security Advisories

ForgeRock issues security advisories in collaboration with our customers and the open source community to address any security vulnerabilities transparently and rapidly. ForgeRock's security advisory policy governs the process on how security issues are submitted, received, and evaluated as well as the timeline for the issuance of security advisories and patches.

For details of all the security advisories across ForgeRock products, see Security Advisories in the *Knowledge Base library*.

Chapter 8

Limitations

Limitations in IG 7.0.2

No additional limitations have been introduced in this release.

Limitations in IG 7.0.1

No additional limitations have been introduced in this release.

Limitations in IG 7.0.0

The following limitations are inherent to the design, not bugs to be fixed:

+ *Host information not forwarded for HTTP/2 requests*

  OPENIG-4817

  When IG is acting as a reverse proxy, and receives HTTP/2 requests, it does not forward the host information provided in the HTTP/2 pseudo-header `:autority:` to the protected application.

  If the protected application is using the HTTP/1 `Host` header or HTTP/2 `:authority:` pseudo-header to route requests, an error occurs.

+ *Multiple spaces in unquoted cookie values are changed to a single space in JBoss*

  OPENIG-4395

  In JBoss, multiple spaces in unquoted cookie values are reduced to one space. For example:

  ```
  testCookieName=cookie    value
  ```

  is changed to

  ```
  testCookieName=cookie value
  ```

+ *No access to common time related functions in expressions*

OPENIG-4201

The value of `System.currentTimeMillis()` cannot currently be used in filters, such as JwtBuilderFilter, for claims such as `exp` and `iat`.

+ *IG scripts can access anything in their environment*

OPENIG-3274

IG scripts are not sandboxed, but instead have access to anything in their environment. You must make sure that the scripts that IG loads are safe.

+ *Persist UMA shares*

OPENIG-3273

Shared resources cannot be persisted when IG restarts. They must be shared each time that IG restarts. For more information, see "*Supporting UMA Resource Servers*" in the *Gateway Guide*.

+ *Proxy WebSocket traffic*

OPENIG-3248

When IG is running in the Jetty application container, it cannot proxy WebSocket traffic.

For more information, see "*Proxying WebSocket Traffic*" in the *Gateway Guide*, and the `websocket` property of "ClientHandler" in the *Configuration Reference* or "ReverseProxyHandler" in the *Configuration Reference*.

+ *Blocked ClientHandler with asynchronous HTTP clients*

OPENIG-2417

IG processes responses from asynchronous HTTP clients by using two thread pools of the same size:

- the first thread pool receive the response headers,
- the second thread pool completes the promise by to executing the callback and writing the response content to the stream. Reading and writing to the stream are synchronous, blocking operations

synchronous operation can cause routes to declare a blocked ClientHandler.

To recover from blocking, restart the route, or, if the route is `config.json`, restart the server. To prevent blocking, increase the number of worker threads.

+ *Cannot use custom config.json in Studio*

OPENIG-1557

When a customized `config.json` is configured in Studio, Studio cannot deploy routes.

+ *Log file of audit events can be overwritten*

OPENIG-813

The log file of audit events can be overwritten when the log file is rotated.

When `CsvAuditEventHandler` is used to log audit events, the log file is overwritten if it is rotated before the file suffix, `rotationFileSuffix`, changes. By default, `rotationFileSuffix` is defined as a date in the format `_yyyy-MM-dd`.

Log files are rotated when one of the following limits is reached: `maxFileSize`, `rotationInterval`, or `rotationTimes`.

Set the log rotation parameters so that the log is not likely to rotate before `rotationFileSuffix` changes.

+ *CookieFilter Is Not JwtSession compatible*

OPENIG-458

The CookieFilter heap object stores a java.net.CookieManager reference in the `session`, so that cookies are linked to the HTTP session. This behavior is not compatible with the use of a JwtSession.

+ *Cannot use SAML with AM policy agent*

OPENIG-291

When SAML is used with an AM policy agent, class cast exceptions occur.

+ *SAML fails with incorrect user-defined mapping*

OPENIG-234

When the user defined mapping is incorrectly set, missing SAML assertions produce an infinite loop during authentication attempts.

+ *For mutual authentication in HTTPS cannot specify which certificate to present*

OPENIG-221

IG can check server certificates for HTTPS. However, for mutual authentication, the client certificate must be the first certificate in the KeyStore.

Chapter 9

Known Issues

IG issues are tracked at https://bugster.forgerock.org/jira/browse/OPENIG.

Known Issues in IG 7.0.2

The following important issue remained open at the time of release:

- OPENIG-5401: Retries on a ReverseProxyHandler not being triggered

Known Issues in IG 7.0.1

No additional issues were opened in this release.

Known Issues in IG 7.0.0

The following important issues remained open at the time of release:

- OPENIG-659: CryptoHeaderFilter - error on handling header value with incorrect length

Chapter 10
Documentation

Date	Description
April 2021	Information has about the `session` property has been added to "AdminHttpApplication (`admin.json`)" in the *Configuration Reference*. Clarification of the role of the property `version` in "AmService" in the *Configuration Reference*.
February 2021	Release of Identity Gateway 7.0.2 software.
October 2020	Release of Identity Gateway 7.0.1 software.
September 2020	The Identity Cloud Guide for IG has been added to provide examples of how to integrate your business application and APIs with ForgeRock Identity Cloud for Single Sign-On and API Security.
August 2020	Initial release of Identity Gateway 7 software. In addition to the changes described elsewhere in these notes, the following important changes were made to the documentation: **Reorganization** • The Maintenance Guide has been added to describe tasks and configurations you might repeat throughout the life cycle of a deployment in your organization. The guide is for anyone who sets up and maintains IG services for their organization. The following chapters have been moved from the Gateway Guide: • "*Auditing Your Deployment*" in the *Maintenance Guide* • "*Monitoring Services*" in the *Maintenance Guide* • "*Troubleshooting*" in the *Maintenance Guide* • The Studio User Guide has been added to describe how to use the IG studio to design and develop routes to protect applications. Examples that used Studio have been moved from the Gateway Guide. • The Deployment Guide now refers to the Docker image provided with the product. Previously, it referred to a Dockerfile provided on Github. • Information about how decorators are implemented in IG has been moved from the Reference Guide to "Decorators" in the *Gateway Guide*. Information about the decorators provided by IG, remains in "*Decorators*" in the *Configuration Reference*.

Date	Description
	Best Practices • Information about how to install IG in standalone mode, installed from a .zip file, has been added to "Downloading and Starting IG in Standalone Mode" in the *Getting Started Guide*. Pointers for migrating from web container mode to standalone mode, have been added to "*Migration*". • Information about how create a Docker image from the Dockerfile provided in the IG .zip file, has been added to Deployment Guide. • Information about how to do a basic installation of IG in Tomcat and JBoss has been moved from the *Gateway Guide* to the Getting Started Guide. • Information about how to configure IG for HTTPs (server-side) has been added to "*Installation in Detail*" in the *Gateway Guide*. **More Overviews** • "Sessions" in the *Gateway Guide* has been added to provide an overview of how IG manages sessions. • Information about the Delegate object has been added to the `target` property of "OAuth2ClientFilter" in the *Configuration Reference*. • Information about the properties of `${attributes.openid}` has been added to "AttributesContext" in the *Configuration Reference*. **More Examples** • Information about routes that do not enforce authentication for specific request URLs or URL patterns has been added in "*Implementing Not-Enforced URIs for Authentication*" in the *Gateway Guide*. • An example that sets up a deployment with three instances of IG, that share a JwtSession has been added to "Sharing JWT Session Between Multiple Instances of IG" in the *Gateway Guide*. • ForgeRock Directory Services (DS) is now secure by default, and connections between IG and DS must therefore be configured for TLS. The example of scripting authentication to non-secure, LDAP-enabled servers has been removed from "Scripting Authentication to LDAP-Enabled Servers" in the *Gateway Guide*. **Standards** "*Supported Standards*" in the *Configuration Reference* has been added to list the standards supported by IG.

Appendix A. Release Levels and Interface Stability

This appendix includes ForgeRock definitions for product release levels and interface stability.

ForgeRock Product Release Levels

ForgeRock defines Major, Minor, Maintenance, and Patch product release levels. The release level is reflected in the version number. The release level tells you what sort of compatibility changes to expect.

Release Level Definitions

Release Label	Version Numbers	Characteristics
Major	Version: x[.0.0] (trailing 0s are optional)	• Bring major new features, minor features, and bug fixes • Can include changes even to Stable interfaces • Can remove previously Deprecated functionality, and in rare cases remove Evolving functionality that has not been explicitly Deprecated • Include changes present in previous Minor and Maintenance releases
Minor	Version: x.y[.0] (trailing 0s are optional)	• Bring minor features, and bug fixes

Release Label	Version Numbers	Characteristics
		• Can include backwards-compatible changes to Stable interfaces in the same Major release, and incompatible changes to Evolving interfaces • Can remove previously Deprecated functionality • Include changes present in previous Minor and Maintenance releases
Maintenance, Patch	Version: x.y.z[.p] The optional `.p` reflects a Patch version.	• Bring bug fixes • Are intended to be fully compatible with previous versions from the same Minor release

ForgeRock Product Stability Labels

ForgeRock products support many features, protocols, APIs, GUIs, and command-line interfaces. Some of these are standard and very stable. Others offer new functionality that is continuing to evolve.

ForgeRock acknowledges that you invest in these features and interfaces, and therefore must know when and how ForgeRock expects them to change. For that reason, ForgeRock defines stability labels and uses these definitions in ForgeRock products.

ForgeRock Stability Label Definitions

Stability Label	Definition
Stable	This documented feature or interface is expected to undergo backwards-compatible changes only for major releases. Changes may be announced at least one minor release before they take effect.
Evolving	This documented feature or interface is continuing to evolve and so is expected to change, potentially in backwards-incompatible ways even in a minor release. Changes are documented at the time of product release. While new protocols and APIs are still in the process of standardization, they are Evolving. This applies for example to recent Internet-Draft implementations, and also to newly developed functionality.
Legacy	This feature or interface has been replaced with an improved version, and is no longer receiving development effort from ForgeRock. You should migrate to the newer version, however the existing functionality will remain. Legacy features or interfaces will be marked as *Deprecated* if they are scheduled to be removed from the product.

Stability Label	Definition
Deprecated	This feature or interface is deprecated and likely to be removed in a future release. For previously stable features or interfaces, the change was likely announced in a previous release. Deprecated features or interfaces will be removed from ForgeRock products.
Removed	This feature or interface was deprecated in a previous release and has now been removed from the product.
Technology Preview	Technology previews provide access to new features that are considered as new technology that is not yet supported. Technology preview features may be functionally incomplete and the function as implemented is subject to change without notice. DO NOT DEPLOY A TECHNOLOGY PREVIEW INTO A PRODUCTION ENVIRONMENT. Customers are encouraged to test drive the technology preview features in a non-production environment and are welcome to make comments and suggestions about the features in the associated forums. ForgeRock does not guarantee that a technology preview feature will be present in future releases, the final complete version of the feature is liable to change between preview and the final version. Once a technology preview moves into the completed version, said feature will become part of the ForgeRock platform. Technology previews are provided on an “AS-IS” basis for evaluation purposes only and ForgeRock accepts no liability or obligations for the use thereof.
Internal/Undocumented	Internal and undocumented features or interfaces can change without notice. If you depend on one of these features or interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Appendix B. Getting Support

ForgeRock provides support services, professional services, training through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. For details on ForgeRock's support offering, including support plans and service level agreements (SLAs), visit https://www.forgerock.com/support.

ForgeRock publishes comprehensive documentation online:

- The ForgeRock Knowledge Base offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

  While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock product documentation, such as this document, aims to be technically accurate and complete with respect to the software documented. It is visible to everyone and covers all product features and examples of how to use them.