Technical Report

FPolicy solution guide for ONTAP: IntraFind

Full-text and enterprise searches with iFinder5 elastic edition for NetApp

Brahmanna Chowdary Kodavali, NetApp
Alexander Kieslich, Manuel Brunner, Joerg Issel, IntraFind Software AG

June 2022 | TR-4670
Version 1.1

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

NetApp® FPolicy™ is a file access notification framework that allows users to monitor file access over NFS and CIFS protocols. This feature was introduced in NetApp clustered Data ONTAP® 8.2, a scale-out architecture that enables a rich set of use cases working with partners. The FPolicy framework requires that all the nodes in the cluster are running Data ONTAP 8.2 or later. FPolicy supports all SMB versions such as SMB 1.0 (also known as CIFS), SMB 2.0, SMB 2.1, and SMB 3.0. It also supports major NFS versions such as NFS v3 and NFS v4.0.

The FPolicy framework natively supports a simple file-blocking use case, which enables administrators to restrict end users from storing unwanted files. For example, an administrator can block audio and video files from being stored in data centers, which saves precious storage resources. This feature blocks files based on only extension. For more advanced features, partner solutions must be considered.

The FPolicy framework enables partners to develop applications catering to a diverse set of use cases.

The use cases include, but are not limited to, the following:

- File screening
- File access reporting
- User and directory quotas
- HSM and archiving solutions
- File replication
- Data governance

Audience

The target audience for this document is customers implementing IntraFind's iFinder5 elastic, a full-text search product, with the FPolicy server to index files stored on ONTAP® software. By using the FPolicy service, administrators can define the rules, which subsequently leads to very fast indexing of the respective documents. These documents are then searchable in iFinder5 elastic, enabling customers to tap into the wealth of information stored in the ONTAP shares. iFinder5 elastic is an enterprise-ready insight engine/enterprise search engine that delivers a single point of information access for all file service data and other data repositories such as intranets, wikis, collaboration platforms, or emails. Full-text content, metadata, and access rights are indexed. iFinder5 elastic includes ready-to-use UIs, a feature-rich knowledge worker UI, lightweight integration components such as search bar and hitlist, accessible UI, and even an iOS and Android app.

Purpose and scope

The purpose of this document is to provide an understanding of FPolicy framework and define steps to deploy the iFinder5 elastic. The scope of the document includes the required deployment steps and best practices for the solution.

FPolicy overview

The ONTAP FPolicy framework creates and maintains the FPolicy configuration, monitors file events resulting from client access, and sends notifications to external FPolicy servers. The communication between the storage node and the external FPolicy servers is either asynchronous or synchronous.

Asynchronous or synchronous communication depends on whether the FPolicy framework expects a notification response from the FPolicy server:

- Asynchronous notification is suitable for use cases for which ONTAP does not act based on the notification response from the FPolicy server. Therefore, it won’t wait for the response from the FPolicy server.
- Monitoring and auditing file access activities are examples of asynchronous notification use cases

- In contrast to asynchronous notification, synchronous notification is suitable for the use cases for which ONTAP must allow or deny the client access based on the notification response from the FPolicy server. Quotas, file screening, file archiving recall, replication, and so on are examples of synchronous notification use cases.

Role of ONTAP components in FPolicy configuration

The administrator storage virtual machine (SVM) (cluster), data SVMs, and data LIFs associated with SVM play a role in an FPolicy configuration.

Administrator SVM (cluster)

A cluster contains the FPolicy management framework and maintains and manages the information about all FPolicy configurations in the cluster.

Data SVMs

FPolicy configuration can be defined at the cluster or the SVM. The scope option in the FPolicy configuration defines the resources that are monitored in the SVM context; the scope option operates on only the SVM resources. One SVM configuration cannot monitor and send notifications for the data (shares) that belong to another SVM.

FPolicy configurations defined on the administrator SVM can be leveraged in all data SVMs.

Data LIFs

Connections to the FPolicy servers are made through data LIFs belonging to the data SVM with the FPolicy configuration. The data LIFs used for these connections can fail over in the same manner as data LIFs used for normal client access.

How FPolicy works with external FPolicy servers

The FPolicy framework runs on every node in the cluster. This framework is responsible for stablishing and maintaining connections with the external FPolicy servers. As part of the connection management, the FPolicy framework manages the following tasks:

- Make sure that file notification flows through the correct LIF to the FPolicy server.
- Load balances the notifications to the FPolicy server when multiple FPolicy servers are associated with a policy.
- Attempts to reestablish the connection when a connection to an FPolicy server is broken.
- Sends the notifications to FPolicy servers over an authenticated session.
- Establishes the connection with the data LIFs on all the nodes participating in the SVM.

For synchronous use cases, the FPolicy server accesses data on the SVM through a privileged data access path. To make this privileged data access path secure, ONTAP uses a combination of user credentials and the IP address of the FPolicy server configured as part of the FPolicy configuration on ONTAP. After the FPolicy server is enabled, the user credentials used in the FPolicy configuration are granted the following special privileges on the file system:

- The ability to bypass permissions checks while accessing the data. The user avoids checks on files and directory access.
- Special locking privileges. ONTAP allows the FPolicy server to read, write, or modify access to any file regardless of existing locks.

  Note: If the FPolicy server takes byte range locks on the file, it results in immediate removal of existing locks on the file.

- The ability to bypass any FPolicy checks. File access over a privileged data path does not generate FPolicy notification.

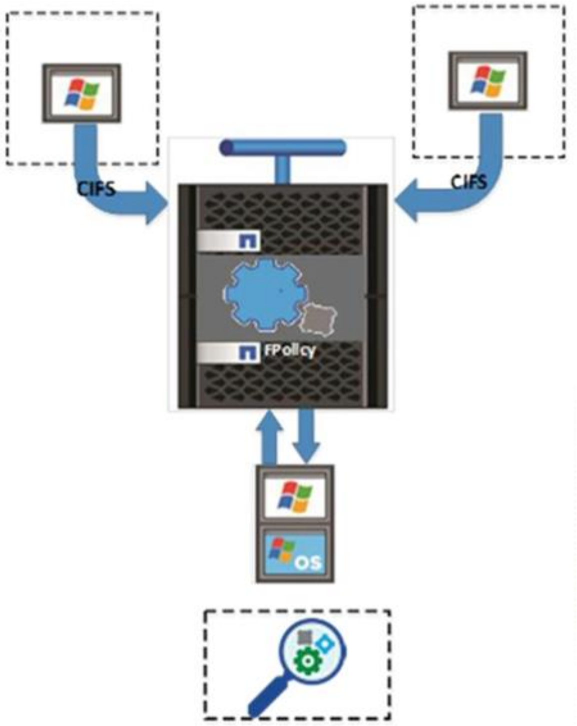
For more information about FPolicy functionality, see “File Access Management Guide for CIFS” on the NetApp Support site.

FPolicy solution architecture

The FPolicy solution consists of the following components, as shown in Figure 1:

- ONTAP FPolicy framework
- FPolicy application: iFinder5 elastic

Figure 1) FPolicy solution architecture.



FPolicy application software runs on an external server. The FPolicy framework is part of ONTAP software. The FPolicy framework connects to external FPolicy servers and sends notifications for certain file system events to the FPolicy servers when these events occur because of client access.

The external FPolicy servers process the notifications and send responses back to the node.

Components of FPolicy framework on ONTAP

The FPolicy framework on ONTAP includes the following components:

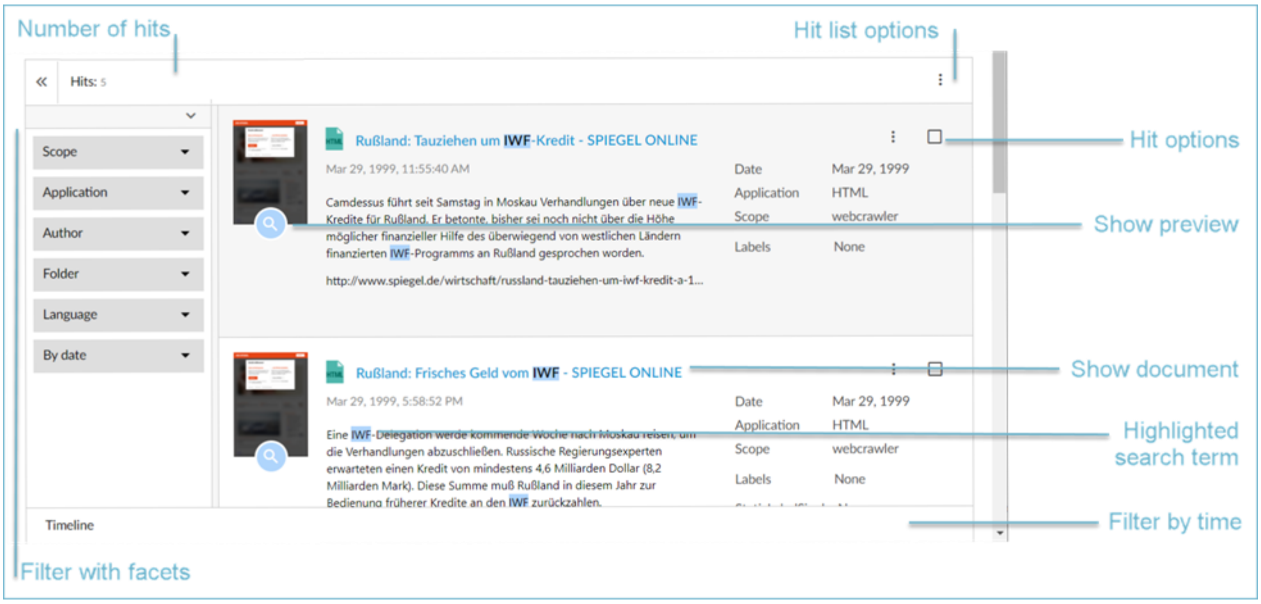
- **External engine.** This container manages external communications with the FPolicy server application, the iFinder5 elastic.
- **Events.** This container captures information about protocols and file operations monitored for the policy.
- **Policy.** This main container associates different constituents of the policy and provides the platform for policy management, such as policy enabling and disabling.
- **Scope.** This container defines the storage objects on which the policy acts: for example, volumes, sharers, exports, and file extensions.

FPolicy application software: iFinder5 Elastic edition for NetApp

The iFinder5 elastic is an enterprise search insight engine application. Through its connector framework, data from various sources is ingested into the search index of iFinder5 elastic, providing superfast and easy access to data and documents. These types of applications are also known as unified search, application-based search, insight engine, 360° view, and other names.

These applications all serve the same purpose: to allow users to find documents based on metadata and content, while still maintaining security on the documents: in other words, only documents for which the individual user is authorized to be part of the hitlist result from an individual search.

Figure 2: Example of an iFinder result list.



Rußland: Tauziehen um IWF-Kredit - SPIEGEL ONLINE

Mar 29, 1999, 11:55:40 AM

Camdessus führt seit Samstag in Moskau Verhandlungen über neue IWF-Kredite für Rußland. Er betonte, bisher sei noch nicht über die Höhe möglicher finanzieller Hilfe des überwiegend von westlichen Ländern finanzierten IWF-Programms an Rußland gesprochen worden.

http://www.spiegel.de/wirtschaft/russland-tauziehen-um-iwf-kredit-a-1...

Date	Mar 29, 1999
Application	HTML
Scope	webcrawler
Labels	None

Rußland: Frisches Geld vom IWF - SPIEGEL ONLINE

Mar 29, 1999, 5:58:52 PM

Eine IWF-Delegation werde kommende Woche nach Moskau reisen, um die Verhandlungen abzuschließen. Russische Regierungsexperten erwarteten einen Kredit von mindestens 4,6 Milliarden Dollar (8,2 Milliarden Mark). Diese Summe muß Rußland in diesem Jahr zur Bedienung früherer Kredite an den IWF zurückzahlen.

Date	Mar 29, 1999
Application	HTML
Scope	webcrawler
Labels	None

Companies currently encounter many challenges with regard to information retrieval. Data sources and data diversity are increasing, but at the same time, employees want a flexible and independent process to gain access to relevant corporate information at any time. For many years, IntraFind has been dealing with these challenges. It now offers an intelligent solution with the product iFinder5 elastic.

Innovation is the driving force for successful companies to open new markets, satisfy customers, and generate revenue. Innovation takes place when historical findings are matched with current and future requirements. IntraFind supports its customers by always providing knowledge in the form of on-hand information, regardless of the application in which it was created or stored.

With the search solution, iFinder5 elastic enables companies to gain full insight into their enterprise information. Through extensive and powerful connectors, IntraFind taps important data sources (NetApp ONTAP, standard file systems, portals, websites, and so on), which allows companywide documents and information to be quickly searchable and checked for access rights. Content is processed, evaluated, and connected based on the latest text and content analytics methods.

iFinder5 elastic goes far beyond classic full-text search. At any given time, employees can gain insight into important and relevant corporate data, making it possible to combine existing knowledge with new findings and requirements and to create innovation.

The iFinder5 elastic edition for NetApp includes the FPolicy server that connects to ONTAP cluster-mode FPolicy service. Without the benefit of FPolicy, content from file shares is typically crawled. Crawling features periodically analyze the file share and identify new, updated, or deleted documents. However, the more documents that are stored on the file share, the more cumbersome this task becomes. Instead, with the FPolicy server now part of the iFinder5 elastic, events are processed almost in real time. These events include creating, deleting, and updating files as well as updates to security information. With the FPolicy server, these events are reflected almost instantly in the search index. This enhancement provides users and administrators comfort while searching for data and documents, knowing that the search index is always up to date.

iFinder5 elastic offers several key features. One of the most popular features is linguistic processing. The IntraFind linguistic processes content with a very powerful function set. As a result, users do not need to know how the original content was written. For example, if the word "policies" is included in a document, the IntraFind linguistic allows the user to also find the word “policy.” While going beyond using stemmer, the IntraFind lemmatizer especially shows its strength while processing European languages, where stemming is not sufficient. For example, for the German word “laufen” (which means "to run"), all grammatical variations of this word would also be found, including the past tense (“lief”). This simple example demonstrates how stemming is not sufficient. With its state-of-the-art technology, the IntraFind linguistics solves these issues for almost 30 languages.

Installing and configuring iFinder5 elastic edition for NetApp

Server for iFinder

Table 1) Check and prepare Windows server.

Done	Task
	Microsoft Windows Server 2012 to 2022
	User who is local administrator on the server.
	Installation instructions are available: Installation under Windows

Table 2) Check and prepare Linux server.

Done	Task
	Linux (for example tested on Ubuntu 20.04 LTS and CentOS 7.6) with Python 3.5, f necessary update to newer version
	Users with sudo permissions
	Relevant tools are installed: • unzip • gcc • make • python (at least version 3.5) • fontconfig
	Installation instructions are available: Installation under Windows

Table 3) For all servers (Linux and Windows)

Done	Task
	The server must be registered in the domain.
	The host name (FQHN) and IP of the domain controller must be known.
	The URL to the LDAP server must be known (e.g. ldap://ad-domain.firma.de:3268/DC=rechte,DC=firma,DC=de)
	Open firewall ports on the server where Tomcat and the web app are installed so that they can be accessed from outside the server: iFinder: 80, 443, 8080 iFinder Administration: 9680
	Configure antivirus software: • Do not check location of index • Do not check installation directory of iFinder More information: Recommendations for the configuration of antivirus software.
	Enable access from the server to the respective data sources, see requirements of the respective connector: Prerequisites for connectors
	When using SSL: Relevant certificates have been generated and signed and are available. More information: • Configuring iFinder for SSL • Configuring iFinder service for SSL

Table 4) Prepare the iFinder configuration.

Done	Task
	Data for the user directory from which the user data for accessing iFinder comes, e.g. for LDAP the domain, the server URL, and the user. More information: • Administrating credentials • Configuring user directories
	Concept for the authorization of users via roles, see Roles and permissions
	Concept for the use of search profiles, see Configuring search profiles

Table 5) Clients for iFinder.

Done	Task
	A supported browser is installed on the clients. All modern and up-to-date browsers are supported, including, but not limited to: • Google Chrome • Apple Safari • Microsoft Edge • Mozilla Firefox (For problems with warnings, see the Problems and solutions information). It is recommended to keep your browser up to date. Internet Explorer 11 will no longer be supported after August 2021.
	Currently, Java 8 and Java 11 are supported. Support for Java 8 will end on March 31, 2022.
	If File Launcher is to be used for opening files in the original location, it is installed on the clients. More information: File Launcher

Hardware prerequisites IOPS (input/output operations per second)

The performance of indexing and searching depends, among other things, significantly on the speed of the hard disks used.

We strongly recommend the use of SSDs, but in any case hard disks that perform at least 20,000 IOPS (input/output operations per second). The value of 20,000 IOPS must be available for both reading and writing. The random IOPS are particularly important for searching. The rule is: the higher the IOPS value, the faster the data medium. New storage media, such as an SSD, achieve up to 50,000-100,000 IOPS. A standard notebook has a performance of around 8,000 IOPS. For an installation, we assume a minimum of 20,000 IOPS for writing and reading in the IntraFind directories. Tools such as “AS SSD Benchmark” for Windows and “fio” for Linux can support you to measure the IOPS of your server. If you need assistance with this, feel free to contact your IntraFind contacts.

Recommendations for the configuration of antivirus software

We recommend not to check the following with the antivirus software:

- The directory where the index is located. The antivirus scanner slows down the access and thus slows down the search.
- The directory where the software is installed. Here are JAR files that are important for the operation of the software. It may happen that the antivirus software deletes JAR files. This would make the software unusable. Furthermore, temporary files that are important for the extraction of textual information are stored here. When the antivirus software accesses the files, they are blocked for a certain time and the process is delayed. If these files are deleted by the antivirus software, the indexing of the files may be incomplete.

Table 6) General prerequisites.

Done	Task
	User for the installation with administration rights
	User for access to the data source with appropriate rights

Done	Task
	Address for access of the connector
	Concept, which data should be indexed
	Documentation for the connector is available: iFinder Connectors
	if-sv-search: Access to the data source for indexing and searching. More information: Configure permission checks.
	if-sv-converter: Access the data source to generate thumbnails. More information: Configuring authentication for if-sv-converter.

Installing with the ZIP installer

For test installations and non-complex installation scenarios, use the ZIP installer. The ZIP installer sets the environment variables and runs all installation programs with the default settings. The following is installed:

- All services for the search
- JDK
- Apache Tomcat
- Apps for searching
- Administration application

For detailed information on the installation and the necessary configuration steps, see Manual installation.

Note: Observe the requirements under iFinder Installation.

Preparing installation

1. Log on with a user who is in the local administrators group.
2. Create a directory for the installation, for example, `C:\IntraFind` with the subdirectory license. In the following, this directory is referred to as <Installation directory>.
3. Copy the license file `intrafind.lic` to `<Installation directory>\license`.
4. Install Microsoft Visual C++ Redistributable using `vcredist_x64.exe`. The file is available in the installation directory and here: https://www.microsoft.com/de-de/download/details.aspx?id=40784.

Installing

1. Execute the `file 1_Environment.bat` as administrator: Context menu > Run as administrator.
2. If you want to use LDAP users and not local users:
   a. Execute the file `2_Services.bat` as administrator: Context menu > Run as administrator.
   b. Optional for impersonation: After installation, extract the archive `waffle-for-if-sv-access.zip` and copy the files from the lib directory to `if-sv-access\lib`.
   c. Extract the archive waffle.zip and copy the file `web.xml` to `<Installation directory>\services\tomcat\webapps\iFinder5\WEB-INF\`.
3. If you want to use local users and not LDAP users:
   a. Execute the file `Local_Users.bat` as administrator: Context menu > Run as administrator.
   b. The following users are created:
      - Users `local.ifinder` for searching, password `iFinder5`
      - User `local.admin` for the administration tool, password `iFinder5`
4. After successful execution, the operating system automatically starts the iFinder interface in the browser defined as default.

5. Open the user interfaces:
   – iFinder5: http://localhost:8080/iFinder5
   – Administration: http://localhost:9680/resource/login.html

Testing search

1. Start the search interface iFinder5: http://localhost:8080/iFinder5.
2. Search for terms that occur in your own documents.

Verifying installation

After the installation is complete, verify that the services work properly:

- Enter the following URL on the server being tested: http://localhost:<PORT>/json/state/get?param0=[health]
- Sample answer: {"health.status":"green"}

Table 7) List of preconfigured services including ports.

Service	Port/request	Expected result
if-sv-thesaurus	9601	Contains "status":"green"
if-sv-converter	9602	Contains "status":"green"
if-sv-search	9605	Contains "status":"green"
If-app-adminui	9680	Contains "status":"green"
If-sv-elasticsearch	http://localhost:9200/_cluster/health	Contains "status":"green"
Tomcat	http://localhost:8080/	HTML is returned.

Manual installation

This installation guide describes a single-node installation and the basic post-installation configuration.

Packages for a default installation

The installers listed in Table 8 are relevant. For information about the relevant version numbers, see the Release Notes on the iFinder Extranet.

Table 8) List of additional services including ports.

Installer	Description	Required
if-meta-openjdk-installer	IntraFind Installer for Java Developer Kit.	Required
if-elasticsearch7-base-installer	Installs if-elasticsearch7. The service contains two interfaces and provides the basic indexing and search functionality.	Required
if-sv-search7-installer	Installs the search service that is based on Elastic Search 7	Required
if-sv-configstore-installer	Installs the config store.	Required
if-app-admin-ui-installer	Installs IntraFind Administration (Admin UI).	Required
if-sv-access-installer	Installs if-sv-access that provides access to sources and files.	Required
if-sv-converter-installer	Installs if-sv-converter. The service extracts text and metadata from documents.	Required
if-sv-thesaurus-installer	Installs the thesaurus service. The service supports two thesaurus input formats: skos and csv.	Required

Installer	Description	Required
apache-tomcat	Installs Apache Tomcat server. Only relevant for search front-end.	Required
waffle	Provides Windows authentication if the Access Service is not used. Only relevant if you want to use LDAP users and not local users.	Only Windows
waffle-for-if-sv-access	Additionally required for Windows authentication when using the Access Service. Only relevant if you want to use LDAP users and not local users as well as impersonation.	Only Windows
if-app-ifinder5	Contains the iFinder5 frontend to be deployed as Tomcat app.	Required
searchbar-standard	Contains the search bar integration component to be deployed as Tomcat app.	Required

Installation under Windows

Preparing for installation

The user for the installation must have administrator rights.

1. Log on with a user who is in the local administrators group.
2. Create a separate directory for the installation, for example `C:\IntraFind` with the following subdirectories: license, install.

   In the following, this directory will be referred to as <installation directory>.
3. Copy the installation files to `<installation directory>\install`.
4. Copy the license file to `<Installation directory>\license`.
5. Install Microsoft Visual C++ Redistributable using `vcredist_x64.exe`. The file is available in the installation package in the Microsoft Download Center.

   Note: You can also run the Windows installers all as silent installations with the `/S` option. The software will then be installed with the default settings.

   Note: You can get a list of available parameters for silent installation with `/S /?`.

Installing components

Installing Java Developer Kit (JDK)

1. Start `if-meta-openjdk-installer-<Version>.exe`.
2. Follow the instructions of the installation assistant.

Setting the environment variables

1. Set the following environment variables in the Control Panel.

   Control Panel > System > Advanced System Settings > Environment Variables > System Variables.

```
JAVA_HOME=<installation directory>\jdk
INTRAFIND_LICENSE=<installation directory>\license\intrafind.lic
```

2. Set the variable `JAVA_HOME` if you have only this Java installation on your system. If you have multiple Java installations, you might need to define the path when you reference your Java installation.
3. Add the following paths to the path variable:

```
<installation directory>\jdk
<installation directory>\services\tomcat
```

Installing Elasticsearch and Linguistics

The installation program `if-elasticsearch7-installer-<version>.exe` installs the elasticsearch7 service and the linguistic features. You can use the linguistic features only if your license includes them.

1. Install `if-elasticsearch7-installer-<VERSION>.exe`.
2. Follow the instructions of the installation assistant.
3. The default settings are:
   - **Installation directory**: `C:\IntraFind\services\if-elasticsearch7-1`. If you want to install into a directory with spaces in the name, see the information under Probleme und Lösungen.
   - **Service port**: 9200. You can check if the port is available. If not, change the port number.
   - **Select license**: Use the license from INTRAFIND_LICENSE. You can see the value under the option.
   - **Value for maximum heap size**: 3GB
   - **Clustername**: ifinder-prod

   Note: NetApp does not recommended installing multiple instances of Elasticsearch for purposes other than testing.

Installing services and apps

1. Install the following services and apps one by one:

Table 9) Services and apps.

Installation program	Default values
`if-sv-search7-installer-<VERSION>.exe`	• Destination: C:\IntraFind\services\if-sv-search-1 • Service port: 9605
`if-sv-configstore-installer-<VERSION>.exe`	• Destination: C:\IntraFind\services\if-sv-configstore-1 • Service port: 9600
`if-app-admin-ui-installer-<VERSION>.exe`	• Destination: C:\IntraFind\apps\if-app-admin-ui-1 • Port: 9680
`if-sv-access-installer-<VERSION>.exe`	• Destination: C:\IntraFind\services\if-sv-access-1 Port: 9611 **Note:** If you want to use impersonation, Waffle is required for this installation . After installation, unzip the archive waffle-for-if-sv-access.zip and copy its contents to the *if-sv-access\lib* folder.
`if-sv-converter-installer-<VERSION>.exe`	• Destination: C:\IntraFind\services\if-sv-converter-1 • Options to use: Converter, Preview, Thumbnail Port: 9602 **Note:** Microsoft Visual C++ Redistributable is required for this installation. Use vcredist_x64.exe for installation.
`if-sv-thesaurus-installer-<VERSION>.exe`	• Destination: C:\IntraFind\services\if-sv-thesaurus-1 Port: 9601

2. Follow the instructions of the installation assistants for each service.
3. If you want to install additional services, install them one by one as well.

Install frontend for search

The search UI requires an Apache Tomcat installation.

1. Create a subdirectory services\tomcat in the installation `directory <Installation directory>\services\tomcat`.

2. Unzip apache-tomcat-<VERSION>.zip and copy the contents to `<Installation directory>\services\tomcat`.
3. Open a command prompt as an administrator and type the following:

```
cd C:\IntraFind\services\tomcat\bin service.bat install Tomcat-iFinder
```

4. Start Services and set the Apache Tomcat 9.0 Tomcat iFinder service to Automatic (Delayed Start).
5. Place files as `if-app-ifinder5-VERSION.war` and `if-app-searchbar-standard-VERSION.war` in the `<installation directory>\services\tomcat\webapps` and rename them:
   - `if-app-ifinder5-VERSION.war` to `iFinder5.war`
   - `if-app-sesrchbar-standard-VERSION.war` to `searchbar.war`
6. Start Apache Tomcat 9.0 Tomcat-iFinder, so that the files are unpacked automatically.
7. If you want to use LDAP users and not local users:
   - Optional for impersonation only: After installation, extract the archive `waffle-for-if-sv-access.zip` and copy the files from the `lib directory to if-sv-access\lib`.
   - Unpack the file waffle.zip and copy the file web.xml to `<Installation directory>\services\tomcat\webapps\iFinder5\WEB-INF\`.
8. Start the service Apache Tomcat 9.0 Tomcat-iFinder and all installed iFinder services. (The names of the services start with IntraFind.)
9. Check if you can access the user interfaces:
   - iFinder5: http://localhost:8080/iFinder5
   - Administration: http://localhost:9680/resource/login.html

   Note: If you have not installed with LDAP users but with local users, the iFinder interface cannot yet be called following the initial configuration. In this case, enable the following configuration in iFinder Administration: iFinder5 > Connection > User credentials > Use login filter for local users?

Installation under Linux

- Preparing installation
- Installing components
  - Setting the environment
  - Installing the Java Developer Kit
  - Increasing the virtual memory
  - Installing Elasticsearch and Linguistics
  - Installing services and apps
  - Start Services
  - Installing the frontend
- Installing connectors
- Little helpers in a Linux environment
- Problems and solutions

Preparing installation

The user for the installation must have sudo privileges.

1. Make sure you have access to the relevant tools in your Linux system:
   - unzip
   - gcc
   - make
   - python (at least version 3.5)

  - fontconfig

2. Create a user IntraFind. The user should have sudo privileges. Alternatively, you must have another user with sudo permissions.

   User for the installation:

   - The software is installed in directories whose owner is the user IntraFind.
   - The installation is done either by the user IntraFind or by another user with sudo privileges.
   - If another user with sudo privileges is used for the installation, the user IntraFind must be passed during the installation.
   - If the user IntraFind is used for the installation and does not have sudo privileges, you must use the -- no-daemon suffix. Next, make sure that autostart of services is set up with a sudo user.

3. Log in with the user `intrafind`.
4. Create a dedicated directory for the installation, for example `/opt/intrafind`. The IntraFind user must be the owner of the directory.

```
sudo mkdir /opt/intrafind
sudo chown -R intrafind:intrafind /opt/intrafind
```

5. Copy the license file `intrafind.lic` to `/opt/intrafind/license/`.
6. Copy the installation files to `/opt/intrafind/install/`.
7. Make sure that all `*.bin` files are executable.(chmod +x *.bin).

Installing components

Setting the environment

Define the variables before installation.

1. With a user with sudo privileges, add the following text to the PATH setting in the `/etc/environment` file: `:/opt/intrafind/jdk/bin:/opt/intrafind/tomcat/bin`. Your default path may be different.

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
:/opt/intrafind/jdk/bin:/opt/intrafind/tomcat/bin"
```

2. Add three lines to the end of the file and adjust the paths if necessary.

```
JAVA_HOME=/opt/intrafind/jdk
INTRAFIND_LICENSE=/opt/intrafind/license/intrafind.lic
CATALINA_HOME=/opt/intrafind/tomcat
```

   Note: You cannot use variables in `/etc/environment`.

3. As user IntraFind add the following lines at the end of the `~/.profile` file:

```
export JAVA_HOME=/opt/intrafind/jdk
export INTRAFIND_LICENSE=/opt/intrafind/license/intrafind.lic
export CATALINA_HOME=/opt/intrafind/tomcat
export PATH="$JAVA_HOME/bin:$CATALINA_HOME/bin:$PATH"
```

4. Log out and log back in to use the new settings.

   Note: `su - intrafind` is not enough. To activate the settings, you must also log in again with the current user.

Installing the Java Developer Kit

1. As user IntraFind, install the file `if-meta-openjdk-installer-xx.bin`.

```
intrafind@localhost:/opt/intrafind/install$ ./if-meta-openjdk-installer-xx.bin
```

2. Log out and log back in to use the new settings.

Increasing the virtual memory

You need sufficient virtual memory to install Elasticsearch, see also https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html.

As a user with sudo privileges, increase the available memory:

1. Edit the `/etc/sysctl.conf` file.
2. Add the following to the end of the file:

```
vm.max_map_count = 262144
```

Restart the machine or enter the following command to change temporarily without a restart:

```
sudo sysctl -w vm.max_map_count=262144
```

The permanent increase in size as described above is nevertheless necessary.

Installing Elasticsearch and Linguistics

The installation program `if-elasticsearch7-installer-<version>.bin` installs the elasticsearch7 service and the linguistic functions.

Note: You can use the linguistic features only if your license includes them.

1. Install `if-elasticsearch7-installer-<version>.bin` as a user with sudo privileges.

```
intrafind@localhost:/opt/intrafind/install$  sudo ./if-elasticsearch-installer-<version>.bin -
u intrafind
```

2. You get the following output:

```
IntraFind Installer <Version>
Installing "elasticsearch"

Summary of selected options:
INSTALLATION DIR : /opt/intrafind/services/if-elasticsearch
JRE              : /opt/intrafind/jdk
PORT             : 9200
USERNAME         : intrafind

Continue? [Y/n]
```

3. Press Enter to install Elasticsearch.

   Note: It is not recommended to install multiple instances of Elasticsearch for other than testing reasons.

Installing services and apps

1. Install the following services and apps one by one:

   Note: If your license includes additional services and applications, install them as described below as well.

Installation program
• if-sv-search7-installer-<version>.bin
• if-sv-configstore-installer-<version>.bin
• if-app-admin-ui-installer-<version>.bin
• if-sv-access-installer-<version>.bin
• if-sv-converter-installer-<version>.bin
• if-sv-thesaurus-installer-<version>.bin

2. Run the installation script for each package:

```
intrafind@localhost:/opt/intrafind/install$  sudo ./<name>-<version>.bin -u intrafind
```

Alternatively, you can run the installation directly under the IntraFind user without sudo privileges with the --no-daemon option. Then you need to set autostart for the service with a sudo user afterwards.

3. After the execution, the installation process begins:

```
IntraFind Installer 1.0
Installing "search"

Summary of selected options:
INSTALLATION DIR : /opt/intrafind/services/if-sv-search
JRE              : /opt/intrafind/jdk
LICENSE          : /opt/intrafind/license/intrafind.lic
PORT             : 9605
USERNAME         : intrafind

Continue? [Y/n]
```

4. If you have changes, cancel the installation process by typing *n* and pressing Enter or by pressing CTRL + C.

   Otherwise, press the Enter key. The service or application is now installed.

5. The service and admin UI are available at `/etc/system/system/<name>.service`. You can now enable autostart at system startup:

```
sudo systemctl enable <name>.service
```

Start services

How to start the `if-elasticsearch` service:

```
sudo systemctl start if-elasticsearch.service
```

To start the other services:

```
sudo systemctl start if-sv-<servicename>
```

Installing the frontend

1. Download Apache Tomcat in version 9.0.50, see http://tomcat.apache.org/. `apache-tomcat-[version].tar.gz` is the platform-independent version without precompiled components.

2. Log in with the user IntraFind.

   Because the IntraFind user is also the owner of the directory, you can run most commands without sudo. Only some commands require sudo privileges.

3. Extract Apache Tomcat:

```
cd ~/install
unzip apache-tomcat.zip -d /opt/intrafind/
cd ..
mv apache-tomcat tomcat
```

4. Configure Apache Tomcat for your environment:

```
cd /opt/intrafind/tomcat/bin
tar xzf commons-daemon-native.tar.gz
cd commons-daemon-x.x.x.-native-src/unix
./configure --with-java=$JAVA_HOME
make
```

5. Usually you can ignore the warnings.

6. Copy the created file and edit the `daemon_sh` file.

```
cp jsvc /opt/intrafind/tomcat/bin/jsvc
nano /opt/intrafind/tomcat/bin/daemon.sh
```

7. Edit the following settings:

Setting	Notes

`--tomcat-user )`
`TOMCAT_USER="$2"`
`shift; shift;`
`continue`
`;;`
	• If you have only local users on your installation, for example on a DEV installation, replace $2 with IntraFind.
	• If you have an AD environment, replace $2 with a domain user.

Setting	Notes
`test ".$TOMCAT_USER" = . && TOMCAT_USER=tomcat`	• If you have only local users on your installation, for example on a DEV installation, replace $2 with IntraFind. • If you have an AD environment, replace tomcat with a domain user.

8. Enter the following commands:

```
cd /etc/init.d
sudo ln -s /opt/intrafind/tomcat/bin/daemon.sh tomcat
cd /opt/intrafind/tomcat/bin/
chmod +x *.sh
```

9. Copy the `if-app-ifinder5<VERSION>.war` file to `<installation directory>/tomcat/webapps/iFinder5.war`.
10. Copy the `if-app-searchbar-<VERSION>.war` file to `<installation directory>/tomcat/webapps/searchbar.war`.
11. Start Tomcat.

```
sudo /etc/init.d/tomcat start
```

Little helpers in a Linux environment

To check which `system.ctl` services are running, run the following command:

```
systemctl | grep if-
```

To test a service and view the log file directly, run `./runConsole.sh` in the bin directory of a service. This is the interactive mode.

What is the process on the specified port?

```
lsof -i :4680
# returns the process id
```

Which instance is running in this process ID?

```
ps -p 19817 -o command
```

Problems and solutions

Problems might occur during the installation of the Search Service. Check the following:

Owner and group of service points:

```
intrafind intrafind 4096 Aug  7 10:42 if-app-admin-ui
intrafind intrafind 4096 Aug  7 07:11 if-elasticsearch7
intrafind intrafind 4096 Aug  7 08:47 if-sv-access
intrafind intrafind 4096 Aug  7 10:41 if-sv-configstore
intrafind intrafind 4096 Aug  7 08:47 if-sv-converter
intrafind intrafind 4096 Aug  7 13:39 if-sv-search
intrafind intrafind 4096 Aug  7 08:48 if-sv-thesaurus
```

Change owner recursively:

```
sudo chown -R intrafind:intrafind ~/services/*
```

User name in the wrapper configuration:

```
nano ~/services/if-sv-search/conf/wrapper.conf
```

Check for the following line:

```
wrapper.app.account=intrafind
```

Output to wrapper.log:

```
less ~/services/if-sv-search/logs/wrapper.log
```

Possible errors in the `wrapper.log` file and how to fix them

Error: “Language: de not licensed for Analyzers!”

```
WrappedException: java.util.concurrent.ExecutionException: RemoteTransportException[[node-
1][127.0.0.1:9300][indices:admin/create]]; nested:
NotSerializableExceptionWrapper[license_exception: Language: de not licensed for Analyzers!
HostIds: error: *java.io.IOException: Cannot run program "hostname": error=13, Permission
denied
```

Solution: In this case, move the IntraFind license `~/license/inrafind.lic` to user/home and restart the Elasticsearch service and then the Search service.

```
intrafind@localhost:~$ mv ~/license/intrafind.lic ~/.
intrafind@localhost:~$ sudo systemctl stop if-elasticsearch.service
intrafind@localhost:~$ sudo systemctl start if-elasticsearch.service
intrafind@localhost:~$ sudo systemctl start if-sv-search
```

Error: Sudden interruption/restart of the service.

```
INFO|7143/0|if-sv-search|18-08-07 11:49:36|2018-08-07 11:49:36,639 {07143} [INFO ] <BeanConfig
> |server pipeline subnet-restriction| subnet-restriction allows 'LOCALHOST'
INFO|7143/0|if-sv-search|18-08-07 11:49:37|2018-08-07 11:49:37,320 {07143} [INFO ] <GuardJvm
> |server pipeline jvm| throttling if heap < '210'mb
INFO|7143/0|if-sv-search|18-08-07 11:49:37|2018-08-07 11:49:37,373 {07143} [INFO ] <Server
> || serving 'state' --> 'com.intrafind.common.state.StateDispatcher@3fb6cf60'
INFO|7143/0|if-sv-search|18-08-07 11:49:37|2018-08-07 11:49:37,375 {07143} [INFO ] <Server
> || serving 'search' --> 'com.intrafind.services.search.SearchStats@12192604'
exit code bsd process 1
INFO|wrapper|if-sv-search|18-08-07 11:49:38|restart process due to default exit code rule
INFO|wrapper|if-sv-search|18-08-07 11:49:38|restart internal RUNNING
INFO|wrapper|if-sv-search|18-08-07 11:49:38|stopping process with pid/timeout 7143 45000
INFO|wrapper|if-sv-search|18-08-07 11:49:38|killing 7143
INFO|wrapper|if-sv-search|18-08-07 11:49:38|process exit code: 1
```

Solution: Check the `hs_err_pid*` log in the service directory. You might run out of memory, causing the service to incessantly try to restart.

1. Open the `wrapper.conf` file.

```
nano ~/services/if-sv-search/conf/wrapper.conf
```

2. Search for the following line:

```
wrapper.java.additional.2 = -Xmx2048M
```

   Change it to:

```
wrapper.java.additional.2 = -Xmx4G
```

3. Check the total memory with Top.

   If your machine does not have enough memory, ask your administrator to allocate more memory for your machine.

Installation with Docker

To install iFinder with all relevant components, you can use Docker containers and combine them via Docker Compose. You use a YAML file to configure the services. Then you can start all components with a single command.

Note: Contact your technical contact to obtain the access data.

- Prerequisites:
  - Software
  - IntraFind Docker Registry
  - System configuration
- Docker Compose configuration:

  - docker-compose.yml (download)
  - docker-compose.override.yml (download)
- Other necessary configuration files
- Target structure
- First steps
- Configuration
- Adding connectors

Prerequisites

Software

You need a Linux system with:

- Docker, version 19.03.0 or later (installation instruction)
- Docker Compose, version 1.25.5 or newer (installation instruction)

  Note: In principle, you can also work with Docker Desktop on Windows or Mac, but these products are explicitly not intended for production use. This manual therefore only describes the installation under Linux.

IntraFind Docker Registry

You will need access credentials to IntraFinds's Docker Registry. Log in once upon receiving them. You are asked for the password:

```
docker login -u <user name> docker-registry.intrafind.net
```

System configuration

For the installation of Elasticsearch you need sufficient virtual memory, see also https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html.

Increase the available memory:

1. Edit the `/etc/sysctl.conf` file.
2. Add the following: vm.max_map_count = 262144

   Note: To temporarily increase the memory without restarting, enter the following: sysctl -w vm.max_map_count=262144. The permanent enlargement as described above is nevertheless necessary.

Docker Compose configuration

The recommended Docker Compose configuration consists of two files:

- Basic configuration in `docker-compose.yml`
- Customizable configuration in `docker-compose.override.yml`

  Note: If you keep this separation, you can replace only the `docker-compose.yml` file when upgrading to a newer version. Your customizations will then be preserved.

3. Create a new directory (in the following `intrafind-docker` as an example)
4. Place the files described below there.

docker-compose.yml (download)

```
version: "3.8"

x-license: &license
  type: bind
  source: ./license
  target: /opt/intrafind/license
  read_only: true
```

```
x-logging: &logging
  logging:
    driver: json-file

x-configstore-env: &configstore-env
  intrafind.common.config-store: new
com.intrafind.common.config.configstore.ConfigstoreRest('http://if-sv-
configstore:9600/configstore-api/v1', null, null)

services:
  if-elasticsearch:
    image: docker-registry.intrafind.net/intrafind/if-elasticsearch:7.10.2.2.0
    command:
      - elasticsearch
      - -Enetwork.host=_site_,_local_
    networks:
      - if-elasticsearch
    volumes:
      - *license
    <<: *logging

  if-sv-configstore:
    image: docker-registry.intrafind.net/intrafind/if-sv-configstore:7.10.2.2.0
    depends_on:
      - if-elasticsearch
    environment:
      ES_CLIENT_HOSTS: if-elasticsearch
      ES_CLIENT_PORTS: '9200'
    networks:
      - if-elasticsearch
      - if-backend
      - if-frontend
    volumes:
      - *license
    <<: *logging

  if-app-admin-ui:
    image: docker-registry.intrafind.net/intrafind/if-app-admin-ui:5.5.1.0
    depends_on:
      - if-sv-configstore
    environment:
      intrafind.common.config-store:
com.intrafind.common.config.configstore.ConfigStores.versioned(new
com.intrafind.common.config.configstore.ConfigstoreRest('http://if-sv-
configstore:9600/configstore-api/v1', 'admin-ui', 'intrafind'))
    networks:
      - if-backend
    volumes:
      - *license
    <<: *logging

  if-sv-search:
    image: docker-registry.intrafind.net/intrafind/if-sv-search:5.5.1.0
    depends_on:
      - if-sv-configstore
    environment:
      <<: *configstore-env
      intrafind.es.client.hosts: if-elasticsearch
      intrafind.net.server.pipeline.subnet-restriction.subnet:
10.0.1.0/24,10.0.2.0/24,localhost
    networks:
      - if-elasticsearch
      - if-backend
      - if-frontend
    volumes:
      - *license
    <<: *logging

  if-sv-thesaurus:
    image: docker-registry.intrafind.net/intrafind/if-sv-thesaurus:5.5.1.0
    depends_on:
      - if-sv-configstore
    environment:
      <<: *configstore-env
    networks:
      - if-backend
      - if-frontend
    volumes:
      - *license
    <<: *logging

  if-sv-access:
    image: docker-registry.intrafind.net/intrafind/if-sv-access:5.5.1.0
    depends_on:
      - if-sv-configstore
    environment:
      <<: *configstore-env
    networks:
      - if-backend
      - if-frontend
    volumes:
      - *license
    <<: *logging

  if-sv-converter:
    image: docker-registry.intrafind.net/intrafind/if-sv-converter:3.7.13.0
    depends_on:
      - if-sv-access
      - if-sv-configstore
    environment:
      <<: *configstore-env
    networks:
      - if-backend
      - if-frontend
    volumes:
      - *license
    <<: *logging

  if-app-ifinder5:
    image: docker-registry.intrafind.net/intrafind/if-app-ifinder5:5.5.1.0
    depends_on:
      - if-app-admin-ui
      - if-sv-search
      - if-sv-thesaurus
    environment:
      CATALINA_OPTS: -Dintrafind.common.config-store="new
com.intrafind.common.config.configstore.ConfigstoreRest('http://if-sv-
configstore:9600/configstore-api/v1', null, null)"
    networks:
      - if-frontend
    volumes:
      - *license
    <<: *logging

networks:
  if-elasticsearch:
    ipam:
      config:
        - subnet: 10.0.0.0/24
  if-backend:
    ipam:
      config:
        - subnet: 10.0.1.0/24
  if-frontend:
    ipam:
      config:
        - subnet: 10.0.2.0/24
```

Code Block 1 docker-compose.yml

docker-compose.override.yml (download)

```
version: "3.8"

services:
  if-elasticsearch:
    volumes:
      - type: bind
        source: ./intrafind.yml
        target: /opt/intrafind/app/config/intrafind.yml
        read_only: true
      - type: volume
```

```
        source: if-elasticsearch-index
        target: /opt/intrafind/app/data
      - type: volume
        source: elasticsearch-logs
        target: /opt/intrafind/app/logs

  if-sv-configstore:
    volumes:
      - type: bind
        source: ./additionalConfig.json
        target: /opt/intrafind/app/additionalConfig.json
      - type: volume
        source: configstore-logs
        target: /opt/intrafind/app/logs

  if-app-admin-ui:
    ports:
      - "127.0.0.1:9680:9680"
    volumes:
      - type: volume
        source: admin-ui-auditlog
        target: /opt/intrafind/app/auditlog
      - type: volume
        source: admin-ui-logs
        target: /opt/intrafind/app/logs

  if-sv-search:
    volumes:
      - type: volume
        source: search-logs
        target: /opt/intrafind/app/logs

  if-sv-thesaurus:
    volumes:
      - type: volume
        source: thesaurus-logs
        target: /opt/intrafind/app/logs

  if-sv-access:
    volumes:
      - type: volume
        source: access-logs
        target: /opt/intrafind/app/logs

  if-sv-converter:
    environment:
      intrafind.kerberos.activate: "false"
    volumes:
      - type: volume
        source: converter-logs
        target: /opt/intrafind/app/logs
      - type: volume
        source: converter-export
        target: /opt/intrafind/app/export
      - type: volume
        source: converter-store
        target: /opt/intrafind/app/store

  if-app-ifinder5:
    ports:
      - "127.0.0.1:8080:8080"

volumes:
  if-elasticsearch-index:
    external: true
  elasticsearch-logs:
  configstore-logs:
  admin-ui-auditlog:
  admin-ui-logs:
  search-logs:
  thesaurus-logs:
  access-logs:
  converter-logs:
  converter-export:
  converter-store:
```

Code Block 2 docker-compose.override.yml

Other necessary configuration files

- The `additionalConfig.json` file contains configuration options that are loaded into the configstore at first startup (`additionalConfig.json`):

```
{
  "admin.ui.extconfigs": "{\"converter.security.mapping\":{\"value\":\"{\\n
\\\"urlConfigs\\\": [\\n    {\\n      \\\"regex\\\": \\\".*if-sv-access:9611.*\\\",\\n
\\\"loginUsername\\\": \\\"user\\\",\\n      \\\"loginPassword\\\":
\\\"DEJlrDMYbJxKo5zcxmxudWZstti9wGTP9NF0Yx5b36di\\\",\\n      \\\"authMethod\\\":
\\\"BASIC\\\"\\n    }\\n
]\\n}\",\"cfggroup\":\"converter\",\"isJson\":\"true\",\"id\":\"converter.security.mapping\"},
\"chrome.custom-arguments\":{\"value\":\"--no-
sandbox\",\"cfggroup\":\"converter\",\"isJson\":\"false\",\"id\":\"chrome.custom-
arguments\"}}",
  "chrome.custom-arguments": "--no-sandbox",
  "converter.security.mapping": "{\n  \"urlConfigs\": [\n    {\n      \"regex\": \".*if-sv-
access:9611.*\",\n      \"loginUsername\": \"user\",\n      \"loginPassword\":
\"DEJlrDMYbJxKo5zcxmxudWZstti9wGTP9NF0Yx5b36di\",\n      \"authMethod\": \"BASIC\"\n    }\n
]\n}",
  "ifinder.access.url": "http://if-sv-access:9611/hessian/access",
  "ifinder.autocomplete.url": "http://if-sv-search:9605/hessian/autocomplete",
  "ifinder.click2rank.url": "http://if-sv-search:9605/hessian/click2rank",
  "ifinder.connectors.exchange.access.url": "http://if-sv-access:9611/bytes/access",
  "ifinder.converter.url": "http://if-sv-converter:9602/hessian/converter",
  "ifinder.didyoumean.url": "http://if-sv-search:9605/hessian/didyoumean",
  "ifinder.index-jobstore.url": "http://if-sv-search:9605/hessian/index-jobstore",
  "ifinder.index.url": "http://if-sv-search:9605/hessian/index",
  "ifinder.knowledgegraph.url": "http://if-sv-search:9605/hessian/knowledgegraph",
  "ifinder.metadata.url": "http://if-sv-search:9605/hessian/metadata",
  "ifinder.preview.url": "http://if-sv-converter:9602/hessian/preview",
  "ifinder.search-jobstore.url": "http://if-sv-search:9605/hessian/search-jobstore",
  "ifinder.search.url": "http://if-sv-search:9605/hessian/search",
  "ifinder.searchexposed.url": "http://if-sv-search:9605/hessian/search-exposed",
  "ifinder.simdocs.url": "http://if-sv-search:9605/hessian/similardocs",
  "ifinder.stats.index.url": "http://if-sv-search:9605/hessian/index-stats",
  "ifinder.stats.search.url": "http://if-sv-search:9605/hessian/search-stats",
  "ifinder.store.url": "http://if-sv-search:9605/hessian/store",
  "ifinder.thesaurus.url": "http://if-sv-thesaurus:9601/hessian/thesaurus",
  "ifinder.thumbnail.url": "http://if-sv-converter:9602/hessian/thumbnail",
  "ifinder.userinfo.url": "http://if-sv-search:9605/hessian/userinfo"
}
```

Code Block 3 additionalConfig.json

- The `intrafind.yml` file is referenced in `docker-compose.yml` and included in the Elasticsearch container (`intrafind.yml`):

```
security.subnet: 10.0.0.0/24,localhost
```

Code Block 4 intrafind.yml

- Additionally, you need a license file from IntraFind. Place it under `license/intrafind.lic` so that it can be found by Docker Compose.

Target structure

Your directory should now have the following structure:

- intrafind-docker/
  - additionalConfig.json
  - docker-compose.override.yml
  - docker-compose.yml
  - intrafind.yml
  - license/
    - intrafind.lic

First steps

1. Create a Docker volume for the Elasticsearch data.

```
docker volume create --name=if-elasticsearch-index
```

2. Start the Configstore.

```
docker-compose up -d if-sv-configstore
```

The first start takes some time, because Elasticsearch is started in the background. You can watch the log with the following command (cancel with Ctrl + C):

```
docker-compose logs -f if-sv-configstore
```

3. The startup is successful when the messages "Initialization successful" and "Started Configstore [...]" are displayed in the log.

   Note: It might happen that the process in the Docker container restarts several times while Elasticsearch is not yet accessible. In most cases this is not a problem and the Configstore will start successfully after a few attempts. If this is not the case, try restarting the container or investigate whether the Elasticsearch container might have failed to start.

4. Start all other services, iFinder Administration (Admin UI) and the iFinder web application.

```
docker-compose up -d
```

5. The iFinder Administration can be reached at http://localhost:9680/resource/index.html after startup. If you are accessing from another machine, you need to set the port shares from 127.0.0.1 to the desired IP address in the `docker-compose.override.yml` file, for example, `10.1.40.174:9680:9680`.
6. Configure your system. For more information, read the next section Configuration.
7. The iFinder can be accessed after startup at http://localhost:8080/iFinder5/. If you are accessing from another machine, you need to set the port shares from `127.0.0.1` to the desired IP address in the `docker-compose.override.yml` file, for example, `10.1.40.174:9680:9680`.

Configuration

You can configure the system using the technical documentation. You can find a first overview in the “Configuring the system” section.

In conjunction with Docker, you should use one of the following strategies:

- Wherever possible, make configuration settings using iFinder Administration. If necessary, use the Extended configuration.
- If configuration is not possible this way, you can set environment variables for the corresponding service in the `docker-compose.override.yml` file. Prefix the configuration key with IntraFind. For example:

```
services:
[…]
  if-sv-converter:
    environment:
      intrafind.converter.activate.image.store: "false"
[…]
```

  Note: Make sure that the indentation is correct. Changes to environment variables are applied when you recreate the container with docker-compose up -d.

- The documentation often refers to a configuration file called `config.cfg`. If you want to edit this file, the following procedure is recommended:

  a. Copy the file from the corresponding container to your directory. You can find them each under `/opt/intrafind/app/config.cfg`.

     For example:

```
mkdir if-sv-converter
```

```
docker cp intrafind-docker_if-sv-converter_1:/opt/intrafind/app/config.cfg if-sv-
converter/config.cfg
```

   b. Edit the file.

   c. Include the file in the container using statements in `docker-compose.override.yml`.

      For example:

```
services:
[…]
  if-sv-converter:
    volumes:
      - type: bind
        source: ./if-sv-converter/config.cfg
        target: /opt/intrafind/app/config.cfg
        read_only: true
[…]
```

Note: Make sure that the indentation is correct. Changes to this file will be applied when you restart the respective container.

- If a service requires additional files for configuration, include them in the container analogously to the above procedure.
- Basic rule:
  - – Settings through environment variables cannot be overridden through the `config.cfg` file or iFinder Administration.
  - – Settings through the `config.cfg` file cannot be overwritten through iFinder Administration.

Configuring the system

This guide only describes some basic configuration steps in a standard one-node deployment.

For detailed information, see the following guides:

- Documentation for the services (for example, for if-sv-search7 and if-sv-converter)
- Documentation for connectors (for example, for if-app-indexer-share)
- iFinder Customizing for information about customizing the Search bar and iFinder
- iFinder Administration or integrated help file for information about setting up the system using iFinder Administration (Admin Tool).

Enabling iFinder5

Perform a basic user configuration of iFinder in iFinder Administration: iFinder Administration.

1. Start iFinder Administration: http://localhost:9680/resource/index.html
2. Define the credentials:
   a. Select product > Credential Administration
   b. Add a credential.
   c. Enter the data. Enter the username in the following format: <user>@<NetBiOSDomainName>, for example ifinder@muc
3. Define an LDAP connection (configuring user directories):
   a. Select product > User Administration > LDAP Connection > LDAP Credentials.
   b. Add an LDAP connection.
   c. Enter the data:

      The NETBios domain name is the short name of LDAP connection, f. ex. muc.

      The credential ID refers to the credential that you created in the last step.
4. Add an LDAP group:
   a. Select product > User Administration > LDAP Connection > LDAP Group Administration.
   b. Add an LDAP group.

   c. Enter the data:

      Select the LDAP connection that you created before.

      Enter the group names exactly as they are defined in your Active Directory. If you add more than one, separate the entries with commas or semicolons.

   d. To identify the existing Active Directory groups, type in the following in a command prompt: `whoami /groups`.

5. Map iFinder roles to the LDAP groups.

   a. Select product > User administration > Groups > iFinder.

   b. Edit the group and add the relevant roles.

6. Map search profile roles to the LDAP groups.

   a. Select product > User administration > Groups > Search profiles.

   b. Edit the group and add the relevant roles.

Defining security settings (Windows and Linux)

If you want to enable access to the search service from outside:

1. Change the firewall rules, at least enable access to port 8080 for the user front end.
2. Edit the following file:

   Windows

```
<Elasticsearch node>\config\ intrafind.yml
```

   Linux

```
<Elasticsearch node>/config/intrafind.yml
```

3. Comment out the following line:

```
security.subnet: localhost
```

4. If required, edit the file:

   Windows

```
<installation directory>\if-sv-search\config.cfg
```

   Linux

```
<installation directory>/if-sv-search/config.cfg
```

5. Extend the security settings for the services (ports 9600-9620) with the host IPs or range that are allowed to access the services.

```
## network security
net.server.pipeline.subnet-restriction.subnet: localhost
es.security.subnet:                            localhost
```

Starting and configuring conversion (Windows and Linux)

If you are installing in an AD domain, you need to start the `if-sv-converter` service with a domain user with read permissions to all files. This is required so that the converter service can open and read the documents for the thumbnail and preview creation. The converter service in turn will create images of the documents to provide the thumbnails and previews.

Depending on the connector, further settings are necessary, see the documentation for the connectors.

If you are using the e-indexer and want to index files on the local server, you need to change the configuration of the converter service. If the e-indexer runs on another machine than the converter service, do not change the setting.

To configure the converter for processing local files:

1. Edit the following file:

Windows

```
<Installation directory>\services\if-sv-converter\config.cfg
```

Linux

```
/opt/intrafind/services/if-sv-converter/config.cfg
```

2. Change the line to true.

```
converter.allow.local.files: false
```

Note: By changing False to True you will be able to convert local files.

Avoiding broken thumbnails and previews for web pages (Linux)

During conversion, you might get the following error in the log file:

```
INFO|1812/0|Service if-sv-converter|17-05-02 15:19:21|2017-05-02 15:19:21,587 {convert 01812}
[ERROR] <PhantomRunner  > |@2| error executing external command: ./phantomjs/phantomjs: error
while loading shared libraries: libfontconfig.so.1: cannot open shared object file: No such
file or directory
```

Install the following font configs:

```
sudo apt-get install libfontconfig
```

Configuring the search service

For detailed information about configuring the search service, see the search service documentation.

Configuring access for the search service

Define the access settings in the Admin Tool. For detailed information, see iFinder Administration Guide.

If you have no AD (for testing only):

1. Edit the `config.cfg` file.

   Windows

```
<Installation directory>\services\if-sv-search\
```

   Linux

```
<installation directory>/services/if-sv-search/
```

2. Look for `permissionsearch.permission-check` and change it.
3. Comment `aclRetriever` and the first `permCheck` and uncomment the second `permCheck`:

```
permissionsearch.permission-check:
var secure = com.intrafind.ifinder.secure;

#  var aclRetriever = new secure.ACLRetrieverUserInfo(Beans.of("permissionsearch.userinfo"),
"_store.groupsids");
#  var permCheckACL = new secure.ACLBasedPermissionCheck(aclRetriever)

  # allows all documents which have a field "_raw.aclallow" containing a value "S-1-1-0"
  var permCheckPublic = new secure.AllowAllPermissionCheck()
```

Configuring LDAP access (Linux only)

1. If you are trying to access the iFinder from a remote machine, grant access.
2. Comment following lines out `in ~/services/if-sv-search/config.cfg` (prepend #):

```
## network security
net.server.pipeline.subnet-restriction.subnet: localhost
es.security.subnet:                            localhost
```

3. In the Kerberos configuration file, replace the placeholders with your LDAP information.

```
Nano ~/tomcat/webapps/iFinder5/WEB-INF/classes/krb5.conf
```

4. In the conf file, for the principal parameter replace the placeholders with the path to your machine.

```
nano ~/tomcat/webapps/iFinder5/WEB-INF/classes/login.conf
```

5. In iFinder Administration:
   a. Select product > iFinder5 > Connection > User credentials.
   b. Activate the Kerberos Security Filter option and add the relevant information.
   c. Path to KRB5 configuration:

```
/home/<USER>/tomcat/webapps/iFinder5/WEB-INF/classes/krb5.conf
```

   d. Path to KRB5 login configuration:

```
/home/<USER>/tomcat/webapps/iFinder5/WEB-INF/classes/login.conf
```

6. Restart tomcat after configuration.

```
sudo /etc/init.d/tomcat stop
sudo /etc/init.d/tomcat start
```

Installing multiple instances in a test scenario

Note: It is not recommended to install multiple instances of Elasticsearch for reasons other than testing.

If you want to install an additional instance on the same server for test purposes, complete the following steps:

1. Install a second instance of the `elasticsearch`.
2. Install a second instance of the search service `if-sv-search7`.
3. Install a second instance of the iFinder app.
4. During installation, the installation assistant detects that there already is an instance.
5. During installation, as cluster name, enter a different name, for example `ifinder-test`.
6. In addition, change the cluster name manually in the `if-sv-search/config.cfg` file of the second search service:

```
## elasticsearch
es.client.settings: cluster.name, ifinder-test
```

Finalizing installation

Starting the system

1. Restart the system.
2. After restarting, have a look at the following logs:
   – elasticsearch log at

     Windows

```
<Installation Directory>\services\if-elasticsearch7\logs\<clustername>.log
```

     Linux

```
<Installation Directory>/services/if-elasticsearch7/logs/<clustername>.log
```

   – Search the service log at:

     Windows

```
<Installation Directory>\services\if-sv-search\logs\wrapper.log
```

     Linux

```
<Installation Directory>/services/if-sv-search/logs/wrapper.log
```

Verification steps

Verify that the services are up and running:

- Basic command that is called from the server for testing:

```
curl 'http://localhost:<PORT>/json/state/get?param0=[health]'
```

- Example of a response:

```
{"health.status":"green"}
```

Table 10) List of preconfigured services including ports.

Service	Port	Request
If-sv-thesaurus	9601	Contains "status":"green"
If-sv-converter	9602	Contains "status":"green"
If-sv-search	9605	Contains "status":"green"
If-app-adminui	9680	Contains "status":"green"
If-sv-elasticsearch	curl 'http://localhost:9200/_cluster/health'	Contains "status":"green"
Tomcat	curl ‘http://localhost:8080’	HTML wird zurückgegeben.

For convenience, you can run the following script on Linux:

```
#!/usr/bin/env bash

HOSTNAME=$( hostname )
exec 1> >(tee integration-testing.log) 2>&1
echo "Testing open ports on $HOSTNAME"
for a in 9601 9602 9605 9680 8080;
do
  curl "http://localhost:$a/json/state/get?param0=[health]" 2>/dev/null > /dev/null && echo
"$a available" || echo "$a not available";
done
curl 'http://localhost:9200/_cluster/health' 2>/dev/null | grep '"status":"green"' > /dev/null
&& echo "9200 green" || echo "9200 red"
echo ""
```

Note: The script writes a file named `integration-testing.log`.

Install and configuration of Share-Indexer push

The Share connector (also Share-Indexer or Share-Crawler) with the technical name `if-app-indexer-share` indexes file share into the InfraFind index used by iFinder5 and searchbar.

Typically, Windows file systems are indexed together with access right information. The search then shows only those documents to which the user also has access on the file system.

Table 11) Feature list Share-Indexer.

Supported content types	Files, embedded files, directories
Supported functions of the connected system	Push-Event-Unterstützung in Kombination mit if-sv-indexer-share-push
Authorization check	According to the source system with the LDAP retriever
Authentication of the Converter Service	Through a domain user with access rights to the file system
Synchronization	Pull, in combination with if-sv-indexer-share-push also push
Secure connection between iFinder and the connected system	Optionally with SSL

Connector configuration	In iFinder Administration or with configuration file config.cfg
Highlighting in the hit list	✓
Thumbnail in the hit list	✓
Highlighting in the preview	✓
Templates in iFinder for the display of hits	✓

Prerequisites for if-app-indexer-share

Make sure the requirements listed in Table 12 have been met.

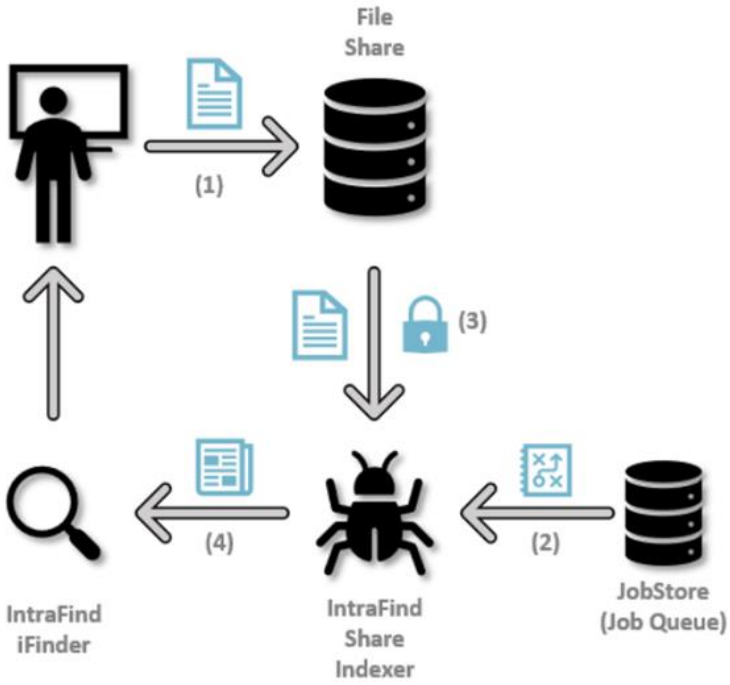
Table 12) Prerequisites for if-app-indexer-share.

Done	Prerequisite
	Users with full read permissions on the files and directories: Since the connector must access all files, the existence of a user with full access to the file systems to be crawled is mandatory. Contact your system administrator to set up such a user.
	Converter Service with this user: To convert binary data to a textual format, the indexer users the Converter Service. This must also have full read permissions on the files and directories.
	Target index: The indexing target must be available as an index and search service. This is typically done when installing if-sv-search.
	Databse JobStore: The indexer needs a second index and search service as data store (JobStore) When you install more than one indexer, the indexers can use the same data store together.

How if-app-indexer-share works

- The user changes a file on the file share.
- After a certain time, the folder is reindexed.
- An index worker takes over the reindexing, detects that the file has changed, and retrieves it including the access rights from the file share.
- The Index Worker updates the index of iFinder.

Figure 3) iFinder index process.



Access rights

The file system is accessed with the user set in the connector. The user must therefore be assigned the rights by the system administrator to access all documents. Since the Share indexer uses the Converter service to convert files in PDF, Word, Excel formats into a format suitable for indexing, the user running the Converter service must also have full access to the file system to be indexed.

Update behavior

The share connector works in pull mode, such as all directories are checked for changes at regular intervals (standard configuration: every 24 hours). A file is re-converted and indexed if the modification date of the file has changed since the last indexing. If only the access rights (= ACL = Access Control List) were modified, the document is re-indexed, but it does not have to be re-converted.

On average, 12 hours (a maximum of 24 hours) elapse between changing a file and transferring the change to the index.

Indexer-Contexts

Every connector is defined with a context. A context defines a configuration and a separate area in the index. This means that the same source document may exist several times in the index if it is indexed in different contexts - but only once for every context.

Background information

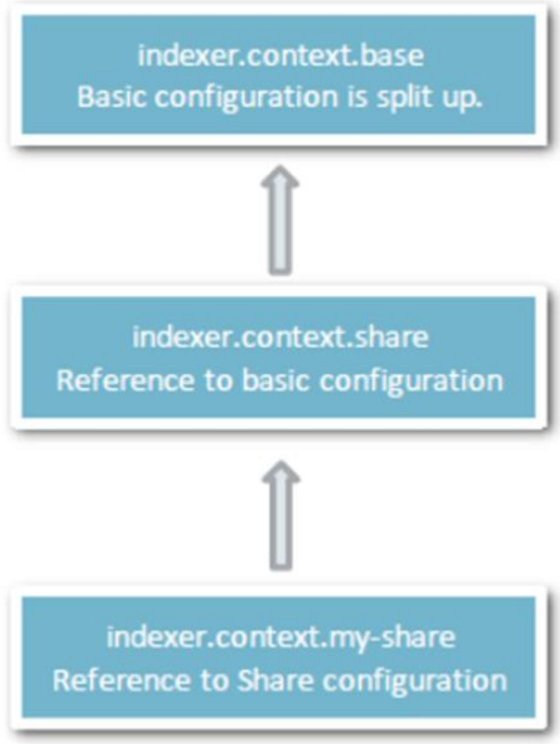
Contexts are type-specific, that is, they are used for a specific source system. Therefore, there must be a corresponding context for each document source: for the indexing of network file systems, a context of type Share exists; for the indexing of an Exchange server, a context of type EWS (Exchange Web Services) must be used.

The configuration options in a context can be of a general nature (index name, client, target index), but there are also always specific parameters that depend on the type of context. For example, you must specify the JDBC connection string for an SQL connector and the Exchange Server address for an EWS indexer.

Inheritance of configuration settings

Configuration in environments with many source systems can be simplified by allowing contexts to refer to higher-level contexts (_parent key). If a key is not set for a context, this key is looked up in the parent context. An example is the share context, which refers to the base context.

Figure 4) Context settings in the iFinder.


indexer.context.base
Basic configuration is split up.
indexer.context.share
Reference to basic configuration
indexer.context.my-share
Reference to Share configuration

When configuring a context, stick to the reference chain `indexer.context.base <= indexer.context.<typ> <= indexer.context.< name>!` Basic configuration and type configuration can make undocumented settings.

After changing the configuration, restart the respective services.

Installation if-app-indexer-share

Installation prerequisites

Installing Share indexer requires the following:

- **User with full access**. Since the connector must access all files, the existence of a user with full access to the file systems to be crawled is mandatory. Contact your system administrator to set up such a user.
- **Converter user under this user**. To convert binary data to a textual format, the indexer users the Converter Service. Also this service must have full access.
- **Target index**. The indexing target must be available as an index and search service. This is typically done when installing if-sv-search.
- **Database JobStore**. The indexer needs a second index and search service as data store (JobStore) When you install more than one indexer, the indexers can use the same data store together.

Dependencies on other IntraFind components

The following IntraFind-Services must be installed and configured:

- IntraFind Converter Service `if-sv-converter`

- IntraFind Index und Search Service `if-sv-search7`
- IntraFind Access Service `if-sv-access`

Installation process

1. Start the installation with `if-app-indexer-share-installer-<Version>.exe`.
2. Click on Next to start installing.

   As a rule, you can proceed with the default settings in the following steps.

   a. Select an installation directory and click on Next.

   b. Default: C`:\IntraFind\apps\if-app-indexer-share-1`.

   c. Enter a name for the service that is in accordance with the directory name and click on Next.

   Default: IntraFind indexer-share service.

   d. As a rule, an infrastructure of IntraFind services and an associated, reusable license already exists prior to this installation, so that the configuration of the license is only necessary here in exceptional cases.

   Default: Use the license from the INTRAFIND_LICENSE environment variable.

   e. Click on Finish. In the following, configure the application.

Licensing

The connector requires a license file provided by IntraFind Software AG. Contact support or your project manager.

Logging

The log files are by default stored in the subdirectory: logs of the installation directory.

Configuring if-app-indexer-share

Before you start if-app-indexer-share, perform at least a minimal configuration:

1. Starting directories
2. Root

You can perform the configuration either in the Admin Tool in the Connectors section, or directly in `conf/config.cfg` in the installation directory of the service. The parameters are the same.

Configuration in iFinder Administration

Setting up the iFinder Administration for the configuration

If you want to use iFinder Administration for configuration, the settings are stored in the config store and no longer in the config.cfg file. Before you start the configuration in the Admin Tool, you have to define once where the Config-Store is located.

Open the `<installation directory>\config.cfg` file with an editor and modify the following key.

Key	Description
common.config-store	Activates the Config Store for storing the configuration. If you want to configure the connector using the Admin Tool, the key and enter the Config Store URL. The configuration is saved there by the Admin Tool. Default: http://localhost:9600/hessian/configstore If you configure a Config Store, the other configuration parameters are read from there. More settings are not necessary here.

Do not define any further keys in the `config.cfg` file, but comment out all other parameters. Define all other keys in the Admin Tool.

Configuring with iFinder Administration

1. Open iFinder Administration under http://my_ifinder_server:9680/resource/index.html.
2. Open Connectors > Share.
3. The basis for further configuration is the context, which represents a separate index area. Create a new context based on the share context. For detailed information on contexts and their use, see Indexer contexts.
4. To create a new context, click on Add. Enter a meaningful name for the context and select the share context as `_parent`.
5. Save with Save settings.
6. Click on Edit for the new context.
7. Configure at least the keys root and seeds, see Root and seeds.
8. Adjust other keys if necessary. For available keys, see Parameters for configuration.
9. Save with Save settings.
10. After each saving, indexing is scheduled and the service is restarted.
11. The first indexing may take some time. Check the log files to see the status. After indexing has been performed, a corresponding status is displayed there.

Connector configuration in the config.cfg (expert configuration)

If you do not want to configure via tiFinder Administration, for example because you need special settings, you can configure the settings directly in the `config.cfg` file. It is available in the installation directory of the indexer.

Note: If a central configuration service is installed, you must make sure not to mix local and central settings as this may lead to inconsistencies. In this case, only use the central configuration with the Admin Tool (Admin Tool > Connectors).

The file looks as follows. There might be changes depending on the respective version.

```
indexer.context.share.seeds:      //share/root1, //share/root2

indexer.context.share.root:       //share

indexer.context.share.skip-stubs: false
```

To configure the starting directory and root complete the following steps:

1. In the `C:\IntraFind\apps\if-app-indexer-share-1` installation directory, open the `config.cfg` file.
2. If the keys are commented out with #, remove it for the respective line to activate the setting.
3. Enter the values for seeds and root. See Root and Seeds.
4. After changing the configuration, restart the respective services.

Parameters for configuration

The tables in this section describe the basic configuration that is supported in every context.

Background information for experts

The settings of a context consist of a set of key-to-value pairs that are grouped in a namespace/key space. The prefix/namespace plus period plus the respective key listed in the left column result in the key to be entered.

Example: `indexer.context.share.index-name`.

In the `indexer.context` namespace, the collection of all context is configured. The individual namespaces for a context are hierarchically subordinates to this. In this instance, the context share with the namespace is `indexer.context.share`. In this namespace, only the `index-name` key exists.

Note: When you configure with iFinder Administration, you only see the key with a field name and not the additional context information, such as the key index-name with the field name (and not `indexer.context.share.index-name`). Some keys are not available in the Admin Tool because they are expert settings

Table 13) Context type and higher-level context.

Key	Field in iFinder administration	Description
_parent	Parent context	Contexts can use this key to refer to a higher-level context, that is, settings that are not explicitly set for this context are copied from the higher-level context. Here it makes sense to reference the general higher-level context: `indexer.context.base`.

Table 14) Root and seeds.

Key	Field in iFinder administration	Description
root	Root directory	Configuring the root key is important for generating the folder facet and defines the root folder of the indexed directory. A useful configuration for the Share Indexer, for example, is the share itself //share, whereas the start directories are located at least one level below this root: `//share/start1`, `//share/sub/start2`.
seeds	Comma separated list of starting paths	The seeds are a comma-separated list of context-dependent starting points for the connector. For the Share connector, this list is obligatory and configures the start directories for traversing the file system. The seeds are also relevant for checking if a share is still available. If a previously existing directory is not accessible, the system first checks whether the corresponding seed still exists as a path. If this is not the case, the operation is aborted (message: Seed '<path>' not available). This prevents documents from being deleted from the index in the event of a temporary failure of a share. **Note:** To remove documents of a no longer existing share from the index, the corresponding seed must be deleted. Otherwise the documents remain in the index.

Table 15) Scheduler.

Key	Field in iFinder Administration	Description
Scheduler	Not available	Expert setting: A scheduler is a (replaceable) time schedule for the times at which jobs are started. It could for example be possible, to update some directories for often than other directories. This is an expert setting. In the default, a global strategy is referenced that checks the system for changes every 24 h (per default). To define an update every 12 hours, the setting is: `com.intrafind.indexer.scheduler.Scheduler Simple(12).` **Note:** The more often you update, the higher the system load is. Default: `indexer.scheduler` (in `indexer.context.base`).

Table 16) Index and search.

Key	Field in iFinder administration	Description
Index	Location of index service	Define the target index for the connector- All indexing operations are performed in this index, that can be configured as URL or reference. Default: `indexer.index` (in `indexer.context.base`)
Search	Location of search service	Some operations require a search configuration that search exactly in the index that is configuration with the index value. Default: `indexer.search` (in `indexer.context.base`)

Table 17) Name/tenant and connector.

Key	Field in iFinder administration	Description
Index-name	Name of the search scope, value for the fields `_str.indexname` and `_facet.indexname`	This value is used in the `_str.indexname` field. Default: index (in `indexer.context.base`)
Tenant	Tenant	iFinder supports multi-tenancy and thus offers the possibility to manage several clients in one index. Each document is assigned to exactly ONE tenant. Only the respective tenant later has access to these documents. If the document is not linked to a specific tenant, it will automatically be assigned to the public tenant (default configuration and always available). This means that all documents are associated with a specific tenant or are public and therefore viewable to all authorized users. In other words, a document is either public or is assigned to a tenant. Default: Public (in `indexer.context.base`)
connector	Value for field _str.connector	This value is used in the _str.connector field. Default: Filesystem

Table 18) Configuration of the Converter service.

Key	Field in iFinder administration	Description
Converter	Location of converter service	When indexing, typically also binary data such as files in doc or pdf format are relevant. The Converter Service converts these formats to a text format. It is configured with this key and can be a URL or a reference It is also possible (expert setting) to distribute the load among several target services. Default: `indexer.converter` (in `indexer.context.base`)
Converter.params	Not available	Expert setting: Conversion can be configured with a list of komma-separated key-value pairs. Here, it is important to configure the time after which a converting attempt is aborted. It is also recommended to define a maximum field size in order to avoid main memory bottlenecks. For

Key	Field in iFinder administration	Description
		information about possible parameters, see Converter Service documentation. Default: Maxfieldlength, 5242880, maxconvertertime, 30 (configured in `indexer.context.base`)
Converter.tries	Maximum number of conversion attempts of the Converter service	The conversion of binary data by the converter service can fail, for example, due to the time limitation - however, there are also situations in which the conversion basically fails (for example, because the input data is corrupt). This key configures the maximum number of conversion attempts on the same file. If the file is changed, the attempts are restarted. Default: 3 (in `indexer.context.base`)
Converter.error-doc	Index an error document when conversion fails	If conversion fails after the defined number of attempts, optionally an error document can be indexed that contains the error message and has the same access settings (ACL) as the original document. If this value is not set to true, no indexing is done. Default: True

Table 19) Processor, salt, acceptor.

Key	Field in iFinder Administration	Description
Processor	Not available	Expert setting: The processor defines a Java object of the type Processor<List<Document>> and is applied to the documents generated by the converter service after the conversion. Here, the document content can be changed; for example, by enriching it with the Tagging Service.
Salt	Version of documents, change for complete reindexing	The change date of an object is stored in an internal database. Documents are only re-indexed if the date changes. You might want to reindex all documents after changing the processor or updating the converter service. This invalidates the stored data and leads to a complete re-conversion and re-indexing. **Note:** This can create a very high load on the system.
Meta-salt	Version of meta data, change for complete reindexing	Changes to meta data (for example, index-name, tenant or connector) are not necessarily adopted for unmodified documents. If this value is changed, the meta data for all documents is rewritten. **Note:** This can create a very high load on the system.

Table 20) Share-Indexer-specific.

Key	Field in iFinder Administration	Description
Skip-stubs	Skip stub files	Expert setting: Stub files are not normal files, but references to data, which are completely or partially stored in an archive system. They are typically used to move less used files from a fast and expensive storage system (e.g. SSD) to a cheap secondary storage (NAS). The user usually does not notice more than an increased delay during access. The indexing of these files can lead to an uncontrolled transfer of all files into the primary system. For this reason, you can prevent stub files from being indexed. For this, you have to activate this key, i.e. set it to True. Default: False (in `indexer.context.share`)

Key	Field in iFinder Administration	Description
Skip-links	Skip links	This key controls whether links are skipped during indexing. Default: True (in `indexer.context.share`).
Case-sensitive	Use case-sensitive names	This key controls whether the names found by the connector are case-sensitive. A use case here is the indexing of a Windows share: here you do not want to distinguish between an uppercase and a lowercase directory name (because they address the same directory). For file names under Linux, however, "file" and "FILE" denote different objects, so case-sensitive must be set to true. Default: False (in `indexer.context.share`).
Acceptor.extensions	Comma separated list of accepted file extensions	This is a list of file extensions accepted by the system, i.e. only files whose extensions are contained in this list are indexed. There is a special key for files without extensions: `NO_EXT`. **Standard:** csv, doc, docm, docx, dot, dotx, htm, html, mht, mhtml, mpp, msg, ods, odt, pdf, pdfa, php, pot, potm, potx, pps, ppsm, ppsx, ppam, ppt, pptm, pptx, rtf, shtm, shtml, txt, vdx, vsd, vss, vst, vsx, vtx, wps, xla, xlam, xls, xlsb, xlsm, xlsx, xlt, xltm, xltx, xml, xps
Meta-only.extensions	Comma separated list of meta-only file extensions	List of file extensions. Only fields with this ending are potentially indexed as meta document. Default: Not defined.

Table 21 contains a selection of possible settings. If you have any questions, contact your technical contact. These settings are not available in the Admin Tool.

Table 21) Special settings.

Key	Field in iFinder administration	Description
Acceptor	Script for accepted objects	The Share Indexer combines the Basic Acceptor with a size limit (see below). Even more complex configurations are possible here, but these are expert settings that should be made by IntraFind as required. In the default at this point various forms of temporary files are excluded (check for `AcceptorsPath.VALID_NAME`) directories are accepted (`AcceptorsPath.IS_DIR`). Files with one of the file extensions configured under `acceptor.extensions` are accepted and all files accepted by the meta-only acceptor. (`META_ONLY_ACCEPTOR`) For more information, see Excluding- and including files.
Meta-only	Only index meta-data	For performance and robustness reasons, it may be useful not to index a subset of the documents to be indexed with full text, but to make only the file path, file name, and some other metadata searchable. By default, this is done for all files that exceed the file size limit configured in meta-only.minsize.mb and for files with file extensions defined in meta-`only.extensions`.
Meta-only.minsize.mb	Minimal size of meta-only files in MB	Minimum size of a file that is only indexed as meta document. Default: 50 MB
Check-acl		With shares, there are often rights violations (such as the visibility rights of a folder or a file are different from those of the parent folder). The share indexer can find and mark

Key	Field in iFinder administration	Description
		such rights violations during indexing (meta documents are indexed for folders in this case). The ACL lists are provided with additional values that reflect the kind of rights violation. Default: False. **Note:** This setting increases the load on the system!
Acl	Not available	The default value uses the permissions assigned to the document itself. Depending on the configuration in your system, however, there may be documents that a user cannot find via the file explorer because he does not have permission for the parent directories. However, from a purely technical point of view, the user could open these documents if he knows the exact path. These documents may be found via search. This can be prevented by the following value: `indexer.context.share.acl-sids-inherit`. In addition, the Search Service must be configured accordingly, see Configuring permission checks. This setting may lead to a higher load on the system. Standard: `indexer.context.share.acl-sids`
Dir-doc	Create a document for each folder: 0 = never, 1 = always, 2 = on ACL conflict	Only together with check-acl: True. Creation of documents for folders: 0 = never 1 = always 2 = when there are ACL conflicts
Semantic-fields-filter	Fields used as semantic terms	You can insert a script that defines fields as semantically significant terms. For example, you can copy individual fields completely or edit or filter their values. By default, the field `_str.filename` is defined as semantically significant term.

Scaling

Any number of instances of a crawler can be operated in parallel. They share the tasks involved. The following points are important:

- All indexer instances share the same database (JobStore).
- All indexer instances have full access to the shares.

Note: The limiting factor for indexing is usually the converter service, so that a stronger distribution of the indexing processes should be accompanied by an increase in the conversion capacity.

Excluding and including files (expert setting)

Files are included or excluded through the Acceptor setting. The default setting is:

```
//groovy
  import com.intrafind.common.functional.predicate.Predicates
  import com.intrafind.indexer.acceptor.Acceptors
  import com.intrafind.indexer.share.AcceptorsPath

  Predicates.and(
      AcceptorsPath.VALID_NAME,
      Predicates.or(
          AcceptorsPath.IS_DIR,
          AcceptorsPath.onName(Acceptors.extensions(config.getStrings("extensions"))),
          META_ONLY_ACCEPTOR
      )13.       )
```

At first glance, this setting appears complicated, but this system allows you to define very complex sets of rules by combining individual predicates.

Note: For questions about the acceptor configuration, contact your IntraFind contact.

The meaning of the above rule is (pay attention to the brackets):

Accept the paths (which can be directories or files) if they have valid names, this excludes typical special names such as `~file`, `thumbs.db`, `$recycle.bin`, `.file`, and are either a directory, have one of the allowed extensions, or have one of the allowed file types.

Example: If, in addition to the above rules, you exclude all files that have an "e" or "u" in their name, you can configure this:

```
//groovy
  import com.intrafind.common.functional.predicate.Predicates
  import com.intrafind.indexer.acceptor.Acceptors
  import com.intrafind.indexer.share.AcceptorsPath

  Predicates.and(
      AcceptorsPath.VALID_NAME,
      Predicates.or(
          AcceptorsPath.IS_DIR,
          AcceptorsPath.onName(
              Predicates.and(
                   Acceptors.extensions(config.getStrings("extensions"))
                   ),
                   Predicates.not(
                            Predicates.contains("e", "u")
                        )
                   )
              ),
          META_ONLY_ACCEPTOR
      )
   )
```

The following constructs are available:

- **Predicates.and(<predicate1>, <predicate2>, …).** This allows various individual decisions to be combined via an AND link. A value is accepted if all predicates accept it.
- **Predicates.or(<predicate1>, <predicate2>, …).** This allows various individual decisions to be combined via an OR link. A value is accepted if one predicate accepts it.
- **Predicates.not(<predicate>).** This inverts the decision of an predicate. An example can be seen above: here all files are to be excluded which correspond to a certain criterion.
- **Predicates.is(<value1>, <value2>, ...).** This predicate is used to check for exact correspondence with a set of values.
- **Predicates.contains(<infix1>, <infix2>, ...).** This can be used to restrict names or paths that must contain a certain character string. You can specify several strings here (see example above), the value is accepted if it contains one of these strings.
- **Predicates.startsWith(<prefix1>, <prefix2>, ...).** This can be used to restrict names or paths that must start with a certain character string. Also, here you can specify several strings here (see example above), the value is accepted if it contains one of these strings.
- **Predicates.endsWith(<suffix1>, <suffix2>, ...).** Analogous to `startsWith,` only with suffixes.
- **Predicates.find(<regex>)**With **find**, you can apply regular expressions to paths or names, and only paths or names corresponding to the regular expression will be accepted. By nesting in a not construct, this can be converted into a predicate that discards all values that are not covered by the regular expression.
- **AcceptorsPath.onName(<predicate>).** The string based predicates are usually applied to the entire path. This can result in many temporary strings. If you only want to apply a rule to the filename, it is best to nest this predicate in an `onName(...)` construct.
- **META_ONLY_ACCEPTOR**: This accepts the files defined under `meta-only`.

Editing the relevant execution files and executing them

Before running the Share Indexer, you need to adjust the `seed.bat` file according to your context. By default, the context is `share`.

Note: This adjustment is only necessary if you do not configure through the Admin Tool.

Adjusting seed.bat

```
java -Xmx4G -cp .;*;lib/* com.intrafind.indexer.IndexerTool -c share -a seed
```

Alternative to the java call `java -Xmx4G -cp.;*;lib/*` you can reference an existing `j.bat` file.

In this instance, the `seed.bat` contains the following entry:

```
j com.intrafind.indexer.IndexerTool -c share -a seed
```

Executing if-app-indexer-share

- Start `if-app-indexer-share` as a service.
- To start the initial seeding, start the file `bat\seed.bat` (Windows) or `bin/seed.sh` (Linux). All start folders to be crawled are then set as jobs in the job database (JobStore).

  For detailed information about how to control indexing, see Indexer Tool.

if-sv-indexer-share-push

With network drives, the entire share must not only be run through completely during the initial indexing, but also later to be able to track changes. This is known as pull indexing: the connector runs through all directories and takes changes into the index. This also results in a considerable load during the incremental run: although you can check a very large number of files within one second to see whether content or access rights have been changed, the often very large number of files means that a complete check of a network drive can take days under certain circumstances. The time span (change latency) between changing a file and the inclusion of this change in the search index is therefore often very long.

To reduce this time, some systems that provide network drives therefore offer the ability to send notifications when a file is modified. These so-called push events can be used to reduce change latency from the order of days to a few seconds.

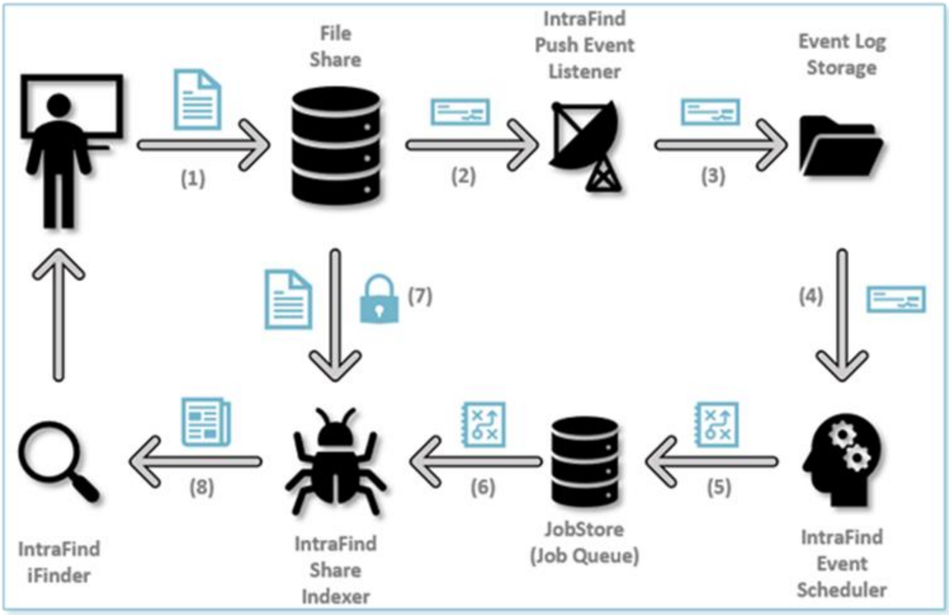
Push event support

For network drives, the entire share must be run through not only during initial indexing, but also later to track changes. This is called pull indexing: the connector traverses all directories and incorporates changes into the index. This also creates a considerable load during the incremental run: although it is possible to check a very large number of files within a second to see whether content or access rights have been changed, the often very large number of files means that a complete scan of a network drive can sometimes take days. The time span between the modification of a file and the transfer of this modification into the search index (change latency) is therefore often very high.

Therefore, to reduce this time, some systems that provide network drives offer the possibility to send notifications as soon as a file is modified. These so-called push events can be used to reduce the change latency from the order of days to a few seconds.

The installation and configuration of the share indexer is described below.

Figure 5) Processing push events.



Processing push events

- A user modifies a file (or folder) on the network share.
- The storage system informs the IntraFind push event listener about this.
- This persists the information locally in an event log file.
- A second service continuously evaluates the event logs.
- This service enters tasks into the connector task list according to the update strategy.
- A worker of the share indexer processes the instruction.
- The worker fetches the new version of the file and its access rights from the network share.
- The worker updates the index for the search in iFinder.

This form of decoupling enables processing that is characterized by high robustness and yet ensures low latencies.

Update behavior

The event listener receives notifications about changes in the file system and converts them into the corresponding instructions to the share indexer. This can significantly reduce the latency between change and index update (often from an offset of several hours to a range of a few seconds).

In a system consisting of many distributed components, even with careful planning and the use of redundant services, it can never be completely ruled out that (for example, due to temporary connection problems) messages (such as push events) are lost. Therefore, a push event crawl is always accompanied by a pull crawl that matches the index with the share. As a result, changes in the share are included in the index even if notifications are missing or lost, albeit with a significantly greater latency.

This mechanism also ensures that systems automatically resynchronize after service failures and no manual operations are required to correct errors.

Note: If file changes have not been applied to the system even after a crawl cycle (usually configured to 24 hours), contact your IntraFind representative.

Important

- **Push events do not update the index in real time.** 30 seconds usually pass between the time a file is changed and the time it is added to the index. This time exists due to technically induced latencies, however artificial delays are also deliberately built in to implement intelligent crawl strategies: For example, when moving a large number of files between directories, it makes sense not to perform individual file processing, but to crawl the affected directories in their entirety incrementally (incrementally means that only changed files are indexed). Often files are changed every second (for example, log files) - also in such a case it makes sense to index this file less often (such as with higher latency). Furthermore, the time depends on the base load of the cluster and the availability of other participating systems and is therefore subject to fluctuations.

- **Pull crawls cannot be dispensed with despite push events.** Push events can be lost for many reasons - for example, due to network problems, event listeners may be unreachable. This can potentially create inconsistencies between share and index, which are compensated for by matching (incremental) pull indexing that continues to be performed (albeit less frequently).
- **Push events increase the load on the system.** Contrary to what you might suspect, using push events increases the load on the system. This is due to the following reasons:
  - Although a push event crawl can reduce the frequency of regular pull crawls, it is not possible to do without them completely (see above). Therefore, the load saving achieved is limited.
  - A file push event always performs a file conversion, but this is orders of magnitude more expensive than a change check. In a pure pull crawl, each modified file is converted at most once in the crawl cycle. Push crawls convert after each file change - possibly dozens or even hundreds of times as often as pull crawls.
- **Temporary inconsistencies are possible**. Push events are processed in a distributed system that only guarantees eventual consistency. This means that the file system and index are not consistent. Under unfavorable circumstances, it may take up to 24 hours to ensure consistency.

Dependence on other IntraFind components

For a push event evaluation, at minimum, the IntraFind services listed in Table 22 must be installed and accessible to the service.

Table 22) IntraFind services.

Service/component	Technical name
IntraFind Share Indexer	if-app-indexer-share
IntraFind Converter Service	if-sv-converter
Index and Search Service	if-sv-search
Access Service (for access from iFinder)	If-sv-access

If necessary, additional instances of a converter service are required, running under a user that occupies a global visibility on all documents of the share. (this is usually a user with backup administrator rights).

Installing technology-specific components

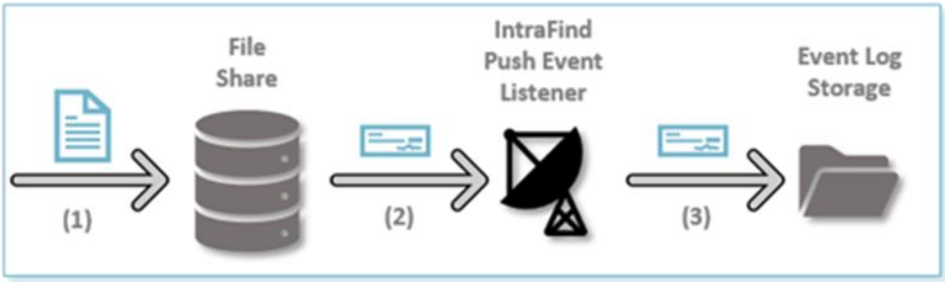
For each share technology supported by IntraFind there are suitable installation packages for the push event listener. Run the appropriate installer and refer to the associated documentation.

The installed IntraFind infrastructure consists of the following components:

- Push event listener
- Event scheduler

To close the processing chain to iFinder5, the installation of the share indexer `if-sv-indexer-share` is mandatory.

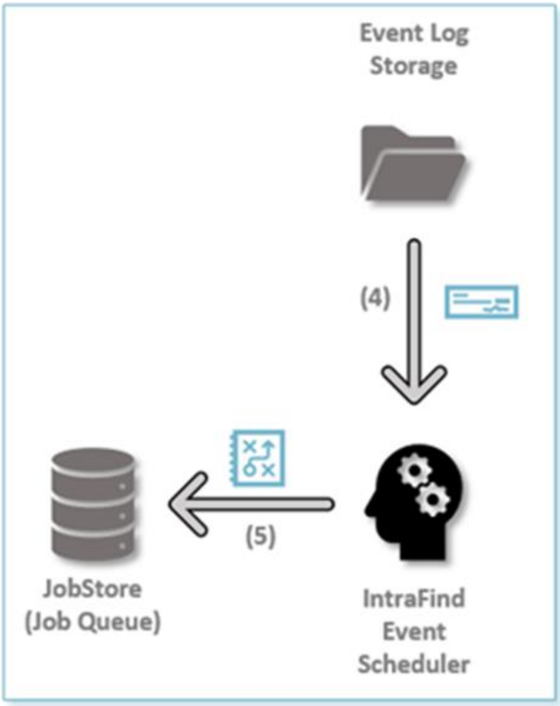
Figure 6) Push listener flow.



Configuring event processing

Event processing continuously reads the event logs persisted on the local system and converts the notifications contained therein into jobs for the share indexer, which are set in its database (JobStore).

Figure 7) Event process flow.



The processing of push events is only possible if the following three components work together correctly:

- **Push event server.** Receives the push messages and persists them in a local file (event log).
- **Event processing (event scheduler).** Reads the event log, groups instructions together if necessary and stores them in the JobStore.
- **Share indexer.** Processes the tasks, performing conversion and indexing.

The event logs are realized through text files, which are created in the `events-<typ>` directory and are recreated every hour. For example, the file `2017-02-17_13.xx.txt` contains all entries of February 17, 2017, in the period 13:00-13:59.

The event logs contain one notification per line and consist of date, event type, event type (FILE or DIR), operation, other type-dependent information, and a JSON-encoded relative path:

```
2016-07-27 11:17:34,089  <TYPE>  <FILE|DIR>  <OP>  <TYP-INFO>  <PATH>
```

Note: The format might change from version to version.

Configuring multiple event logs

In rare exceptional cases it is necessary to read in additional event logs. This is the case if there are several shares based on different technology or if the server configurations should not be the same for all shares.

Is this case, an additional source needs to be added to the configuration `wrapper.conf` from the event scheduler.

Example configuration:

```
#service_base parameter
wrapper.app.parameter.1 = -p
wrapper.app.parameter.2 = indexer.share-push.events2jobstore
wrapper.app.parameter.3 = -s
wrapper.app.parameter.4 = -7200
wrapper.app.parameter.5 = -S
wrapper.app.parameter.6 = 2099-01-01
wrapper.app.parameter.7 = <dir-1>/%.14sxx.txt##era=hour&maxdays=1
wrapper.app.parameter.8 = <dir-2>/other-%.14sxx.txt##era=hour&maxdays=1
```

Configuration in the config.cfg file

The configuration file `config.cfg` contains the settings for the IntraFind services. The configuration key `indexer.share-push.shares` is of central importance here. This key makes it possible to assign the events to a share and to map the information from the events to an absolute path.

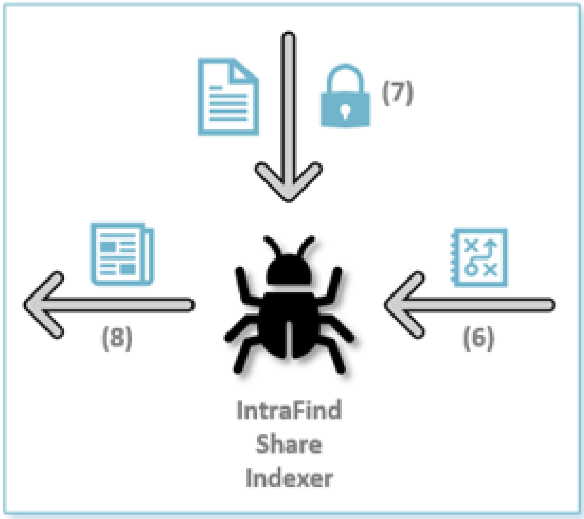
A minimal configuration has the following structure:

```
indexer.share-push.shares:
 <TYP>, <INFO1>, <INFO2>, <CONTEXT>, <MAPPING>, <PREDICATE>
```

This meaning of the fields `info1/info2/mapping` depends on the type of share and is described in the documentation provided for it.

Configuring share indexer for push events

Figure 8) ACL fetcher.



The share indexer is responsible for the actual task of fetching files and their ACL (Access Control List = file permissions) as well as indexing for searching.

In the default configuration, the share indexer is not configured to process push events, but only performs the operations for a regular pull crawl.

To configure push event processing, complete the following steps:

1. Open the file `if-app-indexer-share-<version>/conf/wrapper.conf`. The configuration looks like this example:

```
wrapper.app.parameter.1 = --context
wrapper.app.parameter.2 = share
wrapper.app.parameter.3 = --profile
wrapper.app.parameter.4 = share
```

2. Change the configuration as follows:

```
wrapper.app.parameter.1 = --context
wrapper.app.parameter.2 = share
wrapper.app.parameter.3 = --profile
wrapper.app.parameter.4 = share-push
```

Note: Due to the changed profile, the crawler now also processes push events in addition to the normal file crawl operations.

3. Stop and restart the service.

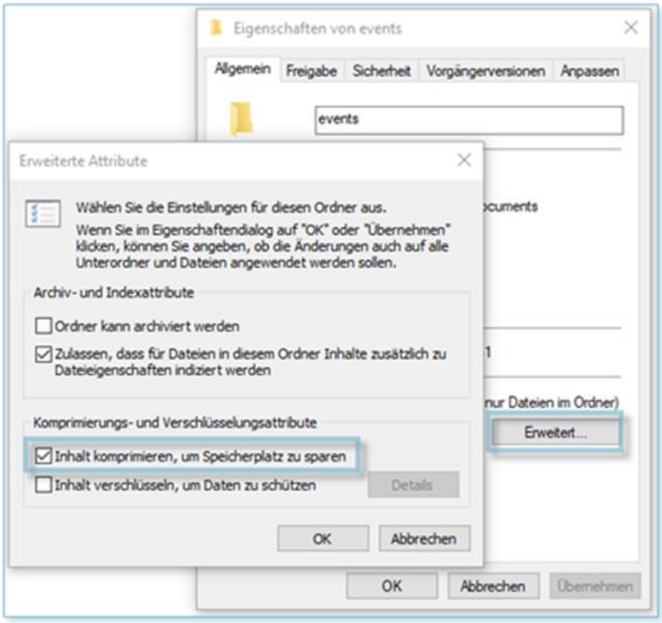
Logging

The log files are by default stored in the subdirectory: `logs` of the installation directory.

Compression of event logs

On Windows, it is recommended to compress the event directory by completing the following steps:

1. In Windows Explorer, open the context menu to the folder where the events are stored.
2. Click Properties > General tab > Advanced and enable Compress content to save disk space.



events

Wählen Sie die Einstellungen für diesen Ordner aus.

Wenn Sie im Eigenschaftendialog auf "OK" oder "Übernehmen" klicken, können Sie angeben, ob die Änderungen auch auf alle Unterordner und Dateien angewendet werden sollen.

Archiv- und Indexattribute

☐ Ordner kann archiviert werden

☑ Zulassen, dass für Dateien in diesem Ordner Inhalte zusätzlich zu Dateieigenschaften indiziert werden

Komprimierungs- und Verschlüsselungsattribute

☑ Inhalt komprimieren, um Speicherplatz zu sparen

☐ Inhalt verschlüsseln, um Daten zu schützen

Initial indexing

Even when push-crawling a share, all documents must initially be indexed through a pull crawl. During this phase, the load on the system is high, and it lasts longer than the later incremental phase, when only changed documents are indexed.

Note: Therefore, it is recommended to turn off the processing of push events during the initial crawl.

Push event support FPolicy

- Dependence on other IntraFind components
- Requirements for the server components

- Installation of the FPolicy server for ONTAP
  - FPolicy mode
  - FPolicy port
- Handling missing connections
- Scaling/Failover
- Memory settings
- Licensing
- Checklist
- Troubleshooting
- Technical questions related to execution

Dependence on other IntraFind components

The components listed in Table 23 must be installed:

Table 23) Service components.

Service/component	Technical name
IntraFind Share Indexer	if-app-indexer-share
IntraFind Converter Service	if-sv-converter
Index and Search Service	if-sv-search

Requirements for the server components

It must be possible to establish socket connections between the ONTAP Vservers and the FPolicy servers (default: port 9000). If necessary, firewall rules must be created.

See also “Configuring FPolicy in ONTAP.”

Note: The software has been tested and certified with Data ONTAP 8.3.

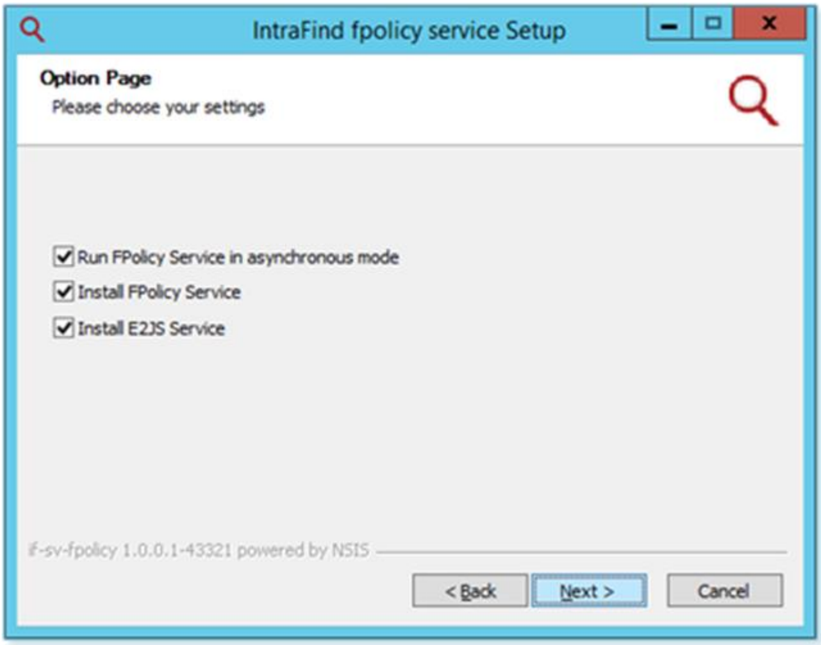
Note: The software has been tested with ONTAP 9.5.

Installation of the FPolicy server for clustered mode

During the installation, there are two important configurations where the chosen settings should be applied. However, these can also be changed later.

FPolicy mode

An important decision is the selection of the FPolicy mode, which can be synchronous or asynchronous.



IntraFind fpolicy service Setup

Option Page

Please choose your settings

☑ Run FPolicy Service in asynchronous mode

☑ Install FPolicy Service

☑ Install E2JS Service

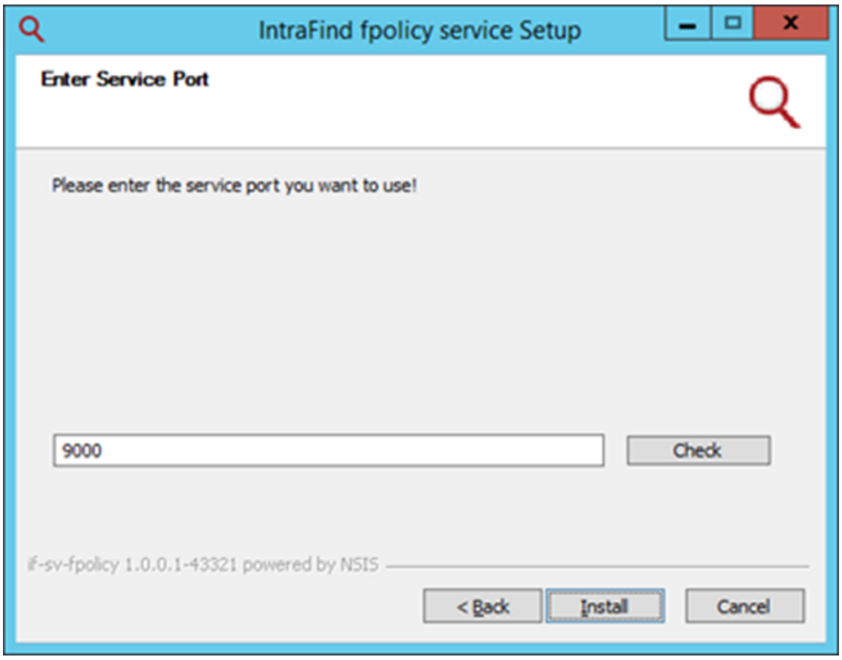
- In synchronous mode ONTAP waits for confirmation of events, this slightly increases the latency of write operations, but ensures that they have been processed.
- The recommended setting is asynchronous mode. Here there is no need to wait for the confirmation, thus the work with the share is done without delay, however under unfavorable circumstances (network problems) events can be lost.

When configuring FPolicy in ONTAP (see “Setting Up the FPolicy Engine”), the appropriate setting must be selected.

Note: If the settings between the ONTAP Vserver and the IntraFind FPolicy server differ on this point, an initial connection can be established, but there will be a communication breakdown after the first processing of an event.

FPolicy port

The setting of the ports must also be synchronous with the later configuration in ONTAP (see “Setting Up FPolicy Engine”).


IntraFind fpolicy service Setup
Enter Service Port
Please enter the service port you want to use!
9000
Check
if-sv-fpolicy 1.0.0.1-43321 powered by NSIS
< Back
Install
Cancel

Note: If the settings differ, ONTAP and FPolicy server cannot communicate with each other.

After installation, the following steps are required:

1. Configure FPolicy in ONTAP
2. Configure FPolicy server

Handling missing connections

If there is a prolonged loss of communication between the FPolicy server and a Vserver, the Vserver does not reestablish the connection automatically. In this instance, you have to switch off the FPolicy and switch it on again. Use the following command to turn it off:

```
fpolicy disable -policy-name intrafind -vserver <vserver>
```

The FPolicy server for ONTAP can be configured to constantly check that all Vservers configured in the indexer.share-push.shares table have established a connection.

If missing connections are detected, a Java processor is called (this can also be generated by scripting). To do this, specify the name of this Java object in `wrapper.conf`:

```
#service_base parameter

wrapper.app.parameter.1 = --port
wrapper.app.parameter.2 = 9000

wrapper.app.parameter.4 = --processor
wrapper.app.parameter.5 = indexer.share-push.processor.reconnect-bat

indexer.share-push.processor.reconnect-bat
```

The installation provides a prebuilt object for reconnecting named `indexer.share-push.processor.reconnect-bat`. This passes the missing UUIDs as parameters to a Windows batch file.

```
indexer.share-push.processor.reconnect-bat:
    Beans.of("FPolicy.processor.factory-process")(Lists.of("cmd.exe", "/K", "reconnect.bat"))
```

For example, you can configure this batch file so that the administrator of the NetApp share receives an email and takes care of the reconnection.

The following example executes the necessary commands remotely in the ONTAP console using `plink.exe` (from a PuTTY).

```
plink -pw <pass> <user>@<netapp> " fpolicy disable -vserver * -policy-name intrafind ; fpolicy
enable -vserver * -policy-name intrafind -sequence-number 1 "
```

The following adjustments might be necessary:

- Specifically address a Vserver instead of ***** (any Vserver) when specifying Vservers.
- If multiple FPolicy servers are in use, you might need to specify a different value for sequence-number for IntraFind FPolicy.

  Note: In this instance, the NetApp administrator password is in plain text in a file. Take the appropriate security precautions to ensure that no unauthorized person gains access to these passwords.

Scaling/failover

Any number of Vservers can communicate with an IntraFind ONTAP FPolicy server, even several hundred events per second can be processed without any problems.

You can run multiple FPolicy servers in parallel to increase resilience. To do this, install additional instances with the same configuration settings and specify the IP addresses of the additional servers using the `-secondary-servers` parameter when configuring ONTAP.

Tip: If you have any questions regarding scaling, contact your IntraFind contact person.

Memory settings

To change the memory requirements of the services you can use the following parameters:

- -XMS for initial memory allocation
- -XMX for the maximum memory allocation.

The settings are made in `wrapper.conf` in the `conf` subdirectory.

For more information about memory settings, contact your IntraFind-representative.

Licensing

The connector requires a license file provided by IntraFind Software AG. Contact support or your project manager.

Table 24) Checklist.

Step	See
Installation if-app-indexer-share	Documentation if-app-indexer-share
Installation if-sv-indexer-share-push	Documentation if-sv-indexer-share-push
Modifying wrapper.conf in if-app-indexer-share	Documentation if-sv-indexer-share-push
Configuring a third party system	Configuring FPolicy in ONTAP
Configuring indexer.share-push.shares in config.cfg: In all workers (if-app-indexer-share) In the push infrastructure (if-sv-indexer-share-push) **Note:** This table must be available everywhere.	Configuring the IntraFind FPolicy server for ONTAP

Troubleshooting

- No share is configured for '<UUID>:<MSID>’ missing configuration in share-push.shares table.
- Cannot create context `share_d` (class or _parent missing?).

  Context `share_d` not (correctly) configured; possibly missing `class` or `_parent` key.

  Note: The context must also be available in the FPolicy server.

- Events are generated (see event-log), but no files are crawled.
- Is the key `seeds` configured (correctly) in context? ONTAP: Was the IP or the host name consistently used for the same share in seeds and share-push.shares?
- Do the share-indexer-instances run under a user with sufficient rights?
- Were all event directories (or patterns) also configured for the event scheduler?
- Has the share indexer profile been changed to share-push?
- The seed '<path>' is not available.

  This is a security mechanism. A path or file is not accessible and the corresponding seed is not accessible either. This indicates that there is a technical problem with the share. In order not to lose documents in the index in such a case, the jobs are canceled. If the seed/share has actually been removed, this must also be done in the configuration. The jobs are repeated automatically after a waiting period.
- [cDot] missing Vserver UUID(s) '[<UUID>]'

  The Vserver does not connect to the FPolicy server after network problem. Solution: Turn FPolicy off (`fpolicy disable -policy-name intrafind -vserver <vserver>`) and turn it on again (see above).

  You can configure the FPolicy server to automatically run a bat file when it detects that the necessary connections are missing. To do this, set the following parameters:

```
--minutes 1 --processor fpolicy.processor.reconnect-bat
```

Technical questions related to execution

Failure mechanisms

- Job (worker): Retry happens at increasing intervals (10 minutes, 30 minutes, 1 hour)
- Converter service: Dummy document with an exception is created (can be switched on and off)
- Dummy files present in the index: 3x attempts to reindex the dummy file. The new attempt happens only in the new job run.
- For load balancing, round-robin can be used, for example. All jobs must be reset.

How is it ensured that jobs never starve?

- New directories are preferred.
- Old jobs are handled before a new crawl.
- New jobs are always done before re-crawls to get new files quickly.

How do you determine if the initial indexing is completed?

When there are no or few jobs left that have long-schedule=1, the initial indexing is complete.

Configuring FPolicy in ONTAP

- Creating FPolicy Event
- Setting up FPolicy engine
- Configuring FPolicy
- Creating scope
- Switching on FPolicy

Figure 9) NetApp ONTAP to iFinder Fpolicy Server.

NetApp
ONTAP
IntraFind
FPolicy
Server
FPOL
(2)

In order for the ONTAP Vserver and the IntraFind FPolicy server to communicate, you must execute the commands described below in the ONTAP console.

After configuration, the Vserver sends information to the IntraFind FPolicy server for each change on the file system.

If multiple Vservers are configured, the configuration must be set on each Vserver.

Note: If several systems (for example, production and test) access the same database (share), you need one ONTAP rule per system.

Creating an FPolicy event

The following command creates an event for the CIFS file share. Replace <vserver> with the name of the appropriate Vservers.

```
fpolicy policy event create -vserver <vserver> -event-name intrafind -protocol cifs -file-
operations
delete_dir,rename_dir,create_dir,rename,delete,setattr,open,close -filters
close_with_modification,open_with_delete_intent
```

Setting up an FPolicy engine

This command configures the actual connection. Make sure that the firewall does not prohibit communications. Replace the placeholders accordingly. Typically, you configure an asynchronous connection where the Vserver does not have to wait for confirmation from the FPolicy server.

```
fpolicy policy external-engine create -vserver <vserver> -engine-name intrafind -primary-
servers <ip-fpolicy-server> -port <port-FPolicy-server> -extern-engine-type asynchronous -ssl-
option no-auth
```

Note: The FPolicy server must be configured according to the FPolicy synchronously or asynchronously. See Konfiguration des IntraFind FPolicy-Servers für ONTAP.

Configuring FPolicy

The following command configures the FPolicy.

Note: If `is-mandatory` is set, file operations are blocked if the FPolicy server cannot be reached, and working with the share is not possible. Therefore, set `is-mandatory` to `true` only if timely tracking of the index is absolutely necessary.

```
fpolicy policy create -vserver <vserver> -policy-name intrafind -events intrafind -engine
intrafind -is-mandatory false
```

Creating the scope

The following command configures the scope of the FPolicy. If the rules are to apply to all volumes and FPolicies, then the scope is configured as follows:

```
fpolicy policy scope create -vserver <vserver> -policy-name intrafind -volumes-to-include "*"
-export-policies-to-include "*"
```

If the Vserver manages multiple volumes and not all of them are to be indexed through push into the IntraFind index, NetApp recommends specifying these volumes as a comma-separated list in the volumes-to-include parameter. This prevents the generation of unnecessary events and offloads the system.

Switching on FPolicy

Make sure that the FPolicy server is running and activate the policy.

```
fpolicy enable -policy-name intrafind -sequence-number 1 -vserver <vserver>
```

ONTAP best practices

NetApp recommends the FPolicy best practices described in this section for server hardware, operating systems, patches, and so on.

Policy configuration

FPolicy external engine for SVM

Providing additional security comes with a performance cost. Enabling SSL communication has a performance effect on CIFS.

FPolicy events for SVM

Monitoring file operations influences the overall user experience. In fact, filtering unwanted file operations on the storage side improves the overall user experience. NetApp recommends monitoring the minimum number of file operations and enabling the maximum number of filters without breaking the use case. The CIFS home directory environment has a high percentage of getattr, read, write, open, and close operations. NetApp recommends using filters for these operations. For recommended filters, see the "Create an FPolicy Event" section.

FPolicy scope for SVM

Restrain the scope of the policies to relevant storage objects, such as shares, volumes, and exports, rather than enabling them throughout the SVM. NetApp recommends checking directory extensions. If is-file-extension-check-on-directories-enabled is set to true, directory objects are subjected to the same extension checks as regular files.

Network configuration

Network connectivity between the FPolicy server and the controller should be of low latency. NetApp recommends separating FPolicy traffic from client traffic by using a private network.

Note: In a scenario where the LIF for FPolicy traffic is configured on a different port than the LIF for client traffic, the FPolicy LIF might fail over to another node due to a port failure. This situation can make the FPolicy server not reachable from the node and can also make the FPolicy notifications for the file operations on the node fail. Make sure that the FPolicy server is reachable through at least one LIF on the node to process FPolicy requests for the file operations performed on that node.

Hardware configuration

The FPolicy server can be on either a physical server or a virtual server. If the FPolicy server is in a virtual environment, make sure to allocate dedicated resources (CPU, network, and memory) to the virtual server.

Multiple policy configuration

The FPolicy policy for native blocking has the highest priority, respective of the sequence number. Decision-altering policies have a higher priority than others. Policy priority depends on use cases. To determine the appropriate priority, NetApp recommends working with partners.

Managing FPolicy workflow and dependency on other technologies

NetApp recommends disabling an FPolicy policy before making any configuration changes. For example, if you want to add or modify an IP address in the external engine configured for the enabled policy, then first disable the policy.

If you configure FPolicy to monitor NetApp FlexCache® volumes, NetApp recommends that you do not configure FPolicy to monitor read and getattr file operations. Monitoring these operations in ONTAP requires the retrieval of inode-to-path (I2P) data. Because I2P data cannot be retrieved from FlexCache volumes, it must be retrieved from the origin volume. Therefore, monitoring these operations eliminates the performance benefits that FlexCache can provide.

When both FPolicy and an off-box antivirus (AV) solution are deployed, the AV solution receives notifications first. FPolicy processing starts only after AV scanning is complete. A slow AV scanner could affect overall performance, so AV solutions must be sized properly.

Sizing considerations

FPolicy performs inline monitoring of CIFS operations, sends notifications to the external server, and waits for a response, depending on the mode of external engine communication (synchronous or asynchronous). This process affects the performance of CIFS access and CPU resources. To mitigate any issues, NetApp recommends assessing and sizing the environment before enabling FPolicy. Performance is affected by the number of users, workload characteristics such as operations per user, data size, and network latency.

iFinder5 elastic edition for NetApp best practices

In every enterprise search project there is low-hanging fruit. All projects start in the sales cycle, determining what the customer wants or needs to achieve. Key questions to ask at this starting point include:

- Secure search: Does the search software need to reflect the authorization information?
- Which sources need to be connected? Which metadata or entities are supposed to be indexed?

  Note: With iFinder5 elastic edition for NetApp, the file share source is a given. Indexing a file share provides quick success in a project.

Troubleshooting

Problem 1: The FPolicy server is disconnected

Potential solution: If the server is not connected, try to connect it by running the engine-connect command. Look for the reason for FPolicy server disconnection by running the show-engine –instance command and take appropriate action.

Example command:

```
1.      fpolicy show-engine delete_dir,rename_dir,create_dir,rename,delete,setattr,open,close
-filters close_with_modification,open_with_delete_intent
2.      fpolicy engine-connect –node <node name> -vserver <vserver name> -policy intrafind -
server <ip address of FPolicy server>
fpolicy show-engine -instance
```

Problem 2: The FPolicy server does not connect

Precheck: Verify that the SVM has a data LIF through which the FPolicy server is reachable.

Example command:

```
network interface show
network ping -lif <vserver_data_lif> -destination <fpolicy server IP address> -lif- owner
vserver_name>.
```

Potential cause number 1: There are issues with routing.

Potential solution: Check the routing table entries by running the routing-groups route show command to verify whether a route is available for the SVM. If not, add a route by running the routing-groups route create command.

Example command:

```
routing-groups route create -vserver <vserver name> -routing-group d10.X.0.0/18 -destination
0.0.0.0/0 -gateway 10.X.X.X
```

Potential cause number 2: The FPolicy server is not listening on the port specified.

Potential solution: Look for the log entry connect failed. errno = 61 Establish TCP connection returned error in the FPolicy user space log file (fpolicy.log). Then, check the port on which the FPolicy server is listening and modify the external engine configuration to use the same port.

Example command:

```
fpolicy policy external-engine modify -vserver <vserver name> -engine-name intrafind -port
<tcp port no>
```

Potential cause number 3: The security options for the external engine are not the same as for the FPolicy server.

Potential solution: Run the `fpolicy policy external-engine show –instance` command. If the FPolicy server is using SSL, then the field SSL Option for External Communication is either mutual-author server-auth.

Also, check the fields FQDN or Custom Common Name, Serial Number of Certificate, and Certificate

Authority to verify that the certificates are properly configured. To correct this problem, modify ssl-auth to no-auth if the FPolicy server is not using SSL. Otherwise, use mutual-auth/server-auth, depending upon the level of security needed.

Example command:

```
fpolicy policy external-engine modify -vserver <vserver name> -engine-name intrafind -primary-
servers <ip address> -port <tcp port no> -ssl-option no-auth
```

Potential cause number 4: The dedicated LIF for the FPolicy traffic failed over to a different node.

Potential solution: Make sure that the FPolicy server is reachable through at least one LIF for that SVM on the node to process FPolicy requests for the file operations performed on that node.

Example command:

```
network interface show
fpolicy show-engine
```

Problem 3: The external engine is not native for the policy

Potential solution: Run the `fpolicy policy show` command to verify whether the Engine field is set to Native. Then create an external engine for the FPolicy server and attach it to the policy.

Example command:

```
fpolicy policy external-engine create
fpolicy policy modify
```

Problem 4: Notifications are not being received for the file operations on volume, share, and export

Potential cause: The FPolicy policy scope is not set properly.

Potential solution: Run the `fpolicy policy scope show` command to verify whether the scope contains the vol/share on which the ops are performed. Then, create or modify the scope for the policy to add the necessary volume, share, or export.

Example command:

```
fpolicy policy scope create/modify
```

Performance monitoring

FPolicy is a notification-based system. Notifications are sent to an external server for processing, and a response is then sent back to the ONTAP software. This roundtrip process adds latency to client access.

Monitoring the performance counters on the FPolicy server and ONTAP helps to identify bottlenecks in the solution and allows you to tune the parameters necessary for an optimal solution. For example, an increase in FPolicy latency has a cascading effect on CIFS latency. Therefore, you should monitor both workload (CIFS) and FPolicy latency. Also, you can use quality-of-service policies in ONTAP to set up a workload for each volume or SVM that is enabled for FPolicy.

NetApp recommends displaying workload statistics by running the statistics show –object Workload command. NetApp also recommends that you monitor the average read and write latencies, the total number of operations, and the read and write counters. You can also use the ONTAP FPolicy counters described in this section to monitor the performance of FPolicy subsystems.

Note: To collect statistics related to FPolicy, you must be in diagnostic mode.

Collect and display FPolicy counters

To collect FPolicy counters, run the following commands:

```
statistics start -object fpolicy -instance <instace name> -sample-id <id>
statistics start -object fpolicy_policy -instance <instace name> -sample-id <id>
```

To display FPolicy counters, run the following commands:

```
statistics start -object fpolicy -instance <instace name> -sample-id <id>
statistics start -object fpolicy_policy -instance <instace name> -sample-id <id>
```

Counters to monitor

Table 25 lists the FPolicy counters that can be monitored.

Table 25) FPolicy counters.

Counters	Description
`max_request_latency`	Maximum screen request latency
`outstanding_request`	Total number of screen requests in process
`request_latency_hist`	Histogram of latency for screen requests
`request_dispatcher_rate`	Number of screen requests dispatched per second
`request_received_rate`	Number of screen requests received per second
`max_request_latency`	Maximum latency for a screen request
`outstanding_requests`	Total number of screen requests waiting for response
`request_latency`	Average latency for screen request
`request_latency_hist`	Histogram of latency for screen request
`request_sent_rate`	Number of screen requests sent to FPolicy server per second
`response_received_rate`	Number of screen responses received from FPolicy server per second

Performance monitoring for iFinder5 elastic edition for NetApp

Various log files are maintained. These log files also contain statistical data (number of items processed over time) that can be used to compare real-life performance against planned performance.

In future versions of the IntraFind administration, review of the worker queue of the underlying processing engine will be possible.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- ONTAP 9 Documentation
https://docs.netapp.com/us-en/ontap/index.html

Version history

Counters	Date	Document version history
Version 1.0	April 2018	Initial release.
Version 1.1	June 2022	Update to the latest version of iFinder.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright information

Trademark information

TR-4670-0622