Recast

Agent Configuration

Last Modified on 04.28.26

Agent.json

The agent.json file allows for configuration of the Application Workspace Agent.

Agent.json file location

The `Agent.json` file is by default placed in the Agent folder after an installation:

Platform	Path
Windows	%ProgramData%\Liquit\Agent\Agent.json
macOS	/Library/Application Support/com.liquit.Agent/Agent.json

The following are valid for both Windows and macOS operating systems:

- You need to restart both the UserHost and the Agent to apply changes made to the `Agent.json` file.
- In case you decide to place the JSON file in another location, we recommend you restart the Application Workspace service after each change of the file so that they are applied.

Example of configuration

Below you will find a sample of an agent.json configuration.

```
{
 "zone": "https://workspace.liquit.com",

 "registration": {
  "type": "Certificate"
 },

 "login": {
  "sso": true
 },

 "log": {
  "level": "Debug"
 }
}
```

Warning

Backslashes '\' should be escaped within a JSON file. So that a single '\' will be a '\\'.

JSON keys are case sensitive. For example, "autostart" will not work, but "autoStart" will work.

Zone

This should be the URL with the FQDN where the targeted Application Workspace Zone is reachable.

```
{
   "zone": "https://company.liquit.com/"
}
```

Prompt Zone

This key controls the display of the Zone Prompt dialog box where the user can configure the zone URL. The value entered in the dialog box is saved in the `zone` key.
This feature works only if there is no URL configured in the `zone` key.

Available options:

- Disabled: User is NOT prompted to enter the zone URL, even if it is not configured
- Show: User is prompted to enter the zone URL, if none is configured. This is the default value of the key.

```
{
   "promptZone": "Show"
}
```

Registration

Registration is the method to create a device in the Application Workspace zone. Registration is in general an action that only needs to be performed once per device. After successful registration, the device will get a "Liquit Agent Authentication certificate" that is tight to the device object in the Application Workspace. The "Liquit Agent Authentication certificate" is used for the device to log in to the Application Workspace.

For a detailed description of Device Registration options within Application Workspace UI, see Device Registration.

Below are described the 4 ways in which you can register a device within the Application Workspace.
These methods are applicable for devices running the Application Workspace Agent.

User Account registration type

For this method, the user account can be LOCAL or Microsoft Entra ID (Azure AD).

1. Install the Application Workspace Agent.

2. Modify the `Agent.json` file as in the following example:

```
{
   "registration": {
      "type": "User"
   }
}
```

3. Log into the Application Workspace and navigate to **Manage** > **System** > **Access policies**.

4. Assign the user an access policy that has the *Register device* privilege. By default, the predefined *User* access policy is granted this permission.

When the user logs in for the first time into the Application Workspace, his device is automatically registered.

Recast

Agent Credentials registration type

1. Install the Application Workspace Agent.
2. Modify the `Agent.json` file as in the following example:

```
{
  "registration": {
    "type": "Credentials",
    "username": "local\\wksimport",
    "password": "P@ssw0rd"
  }
}
```

3. Log into the Application Workspace and navigate to **Manage** > **Identities** > **Users**.
4. Create a user with the LOCAL identity source. For example, "username": "local\wksimport" and "password": "P@ssw0rd"
5. Assign the user an access policy that has the *Register device* privilege. By default, the predefined *User* access policy has this permission granted.

The specified credentials will be used to register the device.

Agent Certificate registration type

1. First choose one of the following three ways in which you want to proceed with the registration:

a. Place the certificate in the certificate repository of the Windows/Mac operating system on the device/user profile, install the Application Workspace Agent and then modify the `Agent.json` file as in the following example:

```
{
  "registration": {
    "type": "Certificate"
  }
}
```

b. Place the certificate in the device certificate store, install the Application Workspace Agent and then specify the `certificateThumbprint` in the `Agent.json` file as in the following example:

```
{
  "registration": {
    "type": "Certificate",
    "certificateThumbprint": "69f4db57b74e13415cd103323331d95022d840c1"
  }
}
```

c. Place the certificate in the device certificate store, install the Application Workspace Agent, and then specify the `certificateIssuer` in the `Agent.json` file.

2. Log into the Application Workspace and navigate to **Manage** > **System** > **Device registration**.
3. Create a *Certificate* type registration.
4. Create or upload a certificate with a private key

Recast

Agent Certificate Enrollment registration type

1. First choose one of the following two ways in which you want to proceed with the registration:

a. Place the enrolled certificate which must have a private key on the device/user profile, install the Application Workspace Agent and then specify the `certificateThumbprint` in the `Agent.json` file.

b. Place the enrolled certificate which must have a private key on the device/user profile, install the Application Workspace Agent and then specify the `certificateIssuer` in the `Agent.json` file as in the following example:

```
{
   "registration": {
      "type": "CertificateEnrollment",
      "certificateIssuer": "Sectigo RSA Domain Validation Secure Server CA"
   }
}
```

2. Log into the Application Workspace and navigate to **Manage** > **System** > **Device registration**.
3. Create a *Certificate enrollment* type registration.
4. Select an intermediate certificate or the root CA in a certificate chain. It does not need a private key.
5. (Optional) You can specify an OID, for example, to limit the number of valid certificates.

Agent certificate location

If you choose to certificate the device, the certificate must be placed in:

- Windows: Local machine certificate store (under the HKEY_LOCAL_MACHINE root).
- macOS: /Library/Keychains/System.keychain

If you choose to certificate a user, the certificate must be placed in:

- Windows: Current user certificate store (under the HKEY_CURRENT_USER root)
- macOS: ~/Library/Keychains/login.keychain-db

Keys

Key	Description	Default
type	• *User* – The user that is currently logged on will be used to register the device with the server. • *Credentials* – The specified credentials will be used to register the device with the server. • *Certificate* – The system uses a certificate to register the device. • *Certificate Enrollment* – The system uses an erolled certificate to register the device. See Device Registration for more information.	User
username	When *type* key is set to *Credentials*, this username is used to register the device.	
password	When *type* key is set to *Credentials*, this password is used to register the device.	
certificateThumbprint	The hexadecimal string that uniquely identifies the certificate, for example: ea7f07211fddc0df73bac1437a3ff932ce*	

Key	Description	Default
certificateIssuer	The authority that issued the certificate.	

When *type* key is set to *Certificate* or *Certificate Enrollment*, the `certificateThumbprint` and `certificateIssuer` are optional settings that can be used to verify the certificate.

If the value of `certificateThumbprint` is not specified: the certificate must persist on the device in the following locations:

- **Windows**: %ProgramData%\Liquit\Agent\AgentRegistration.cer
- **macOS**: /Library/Application Support/com.liquit.Agent/AgentRegistration.cer

In this case, the certificate which matches the value and persists on the device, will be installed in the Windows Certificate Store or macOS Keychain and the certificate file will be removed from the device.

If the value of `certificateThumbprint` is specified: then the Windows Certificate Store or macOS Keychain will be checked for a certificate with the specified value.

Deployments

```
{
  "deployment": {
              "zoneTimeout": 60,
              "enabled": false,
              "start": true,
              "context": "User",
              "cancel": false,
              "triggers": true,
              "autoStart": {
                    "enabled": true,
                    "deployment": "RRO Retry deployment",
                    "timer": 10
     }
   }
}
```

Key	Description	Default
zoneTimeout	When the deployment is running and the connectivity to the zone is lost, the Agent will wait for the zone to become available for the set period of time. This unit of measure of this value is minutes.	10
enabled	If set to *true*, Application Workspace will enable the deployment process on this machine.	false
start	If set to *true*, the process of deployments is automatically started. • When context value is *Device*, the deployment will start after 5 seconds of the agent being started. • When context value is *User*, the deployment will start the first time the userhost has started. If set to *false*, this process can be manually started by starting: • Windows: *C:\Program Files\Liquit Universal Agent\ShellAPI.exe --deployment --run* • macOS: */Applications/Liquit.app/Contents/MacOS/ShellAPI --deployment --run*	true

Key	Description	Default
context	The context in which the deployment should run. • *User* – User login is required before the deployment wizard is shown. Before starting the deployment, the user will be prompted to select which of the available deployments to run. • *Device* – No user login is required and the deployment will begin automatically. Autostart should be enabled and the specified deployment under autostart should match a single deployment. The autostart timer will be ignored.	Device
cancel	If set to *true*, the deployment process can be cancelled.	true
triggers	If set to *true*, Application Workspace events (refresh/ login) can still be executed.	false

Autostart

Key	Description	Default
enabled	If set to true, Application Workspace will automatically start the deployment process when a corresponding deployment is found as configured in the deployment key or when only one deployment is available for this device.	false
deployment	The targeted deployment which can be either the name or the ID of the deployment.	
timer	This key accepts an integer and represents the number of seconds Application Workspace will wait before automatically starting the deployment.	0

Log

The Application Workspace Agent logs events it initiates.

```
{
   "log": {
      "level": "Debug",
      "agentPath": "Agent.log",
      "userHostPath":"UserHost.log",
      "rotateCount": 5,
      "rotateSize": 1048576
   }
}
```

Key	Description	Default
level	This element is used to define the level of logging. The following levels are available: • *None* – Nothing will be logged to the log file • *Critical* – Only critical errors will be logged to the log file. • *Error* – Only errors and critical errors will be logged to the log file. • *Warning* – Only warnings, errors and critical errors will be logged to the log file. • *Info* – Basic information, warnings, errors and critical errors will be logged to the log file. • *Debug* – Detailed information about all actions, including all web activity will be logged to the log file. You can use this information when troubleshooting.	Info
agentPath	You can define an alternate path of the Agent log files here. The default location is: • Windows: *%ProgramData%\Liquit\Agent\Logs\Agent.log* • macOS: */Library/Logs/com.liquit.Agent/Agent.log*	Agent.log

Key	Description	Default
userHostPath	You can define an alternate path of the userhost log files here. The default location is: • Windows: *%LOCALAPPDATA%\Liquit\UserHost\Logs\UserHost.log* • macOS: */Users/xxx/Library/Logs/com.liquit.Agent/UserHost.log*	UserHost.log
rotateCount	The number of logfiles that will be archived.	5
rotateSize	The limit of logfile size in bytes. When this limit is reached, a new logfile will be created and the old file will be archived.	1048576

Login

This element controls the login behaviour for the Application Workspace Agent.

```
{
   "login": {
      "enabled": true,
      "sso": true,
      "identitySource": "LIQUIT",
      "timeout": 4
      "showConnectionPendingScreen": false
   }
}
```

Key	Description	Default
enabled	If set to *true*, the user will be prompted for login.	true
sso	If set to *true*, the Universal Agent will use the value of the identity source key you provide, to facilitate SSO.	false
identitySource	The default identity source used to log in the user. Use the name of the identity source as you defined it within Application Workspace.	
timeout	The interval in seconds after which the Application Workspace login prompt will be displayed if SSO could not be performed within the interval.	4
showConnectionPendingScreen	If the agent is unable to reach the targeted zone during the SSO login, the connection pending window is shown.	false

Icon

Controls the behaviour of the system tray icon.

```
{
   "icon": {
      "enabled": true,
      "exit": true,
      "timeout": 30
   }
}
```

Key	Description	Default
enabled	If set to *false*, the tray icon is hidden from the user.	true

Key	Description	Default
exit	If set to *false*, the quit option from the icon's context menu in the system tray is hidden. The quit option will always be disabled if the launcher is enabled and is not allowed to close.	true
timeout	The number of seconds Application Workspace waits for the Windows shell to load in order to display the system tray icon. Note that there is no maximum time limit.	30

Cache

This element controls the settings of the cache.

```
{
   "cache": {
      "enabled": true,
      "offline": true,
      "path": "Cache",
      "tempPath": "Temp",
      "packageTempPath": "${TEMP}",
      "autoClean": true,
      "expireContent": 90
   }
}
```

Key	Description	Default
enabled	When an identity is entitled to a package, the package is automatically downloaded and cached on the end-user device. It will remain there even after the session ends, as long as the end-user is entitled to it. When the entitlement is removed, the cache is cleaned depending on how `autoClean` and `expireContent` are configured.	
offline	If set to *true*, the offline mode will be available for the local device. If set to*false*, the packages marked offline will not be automatically downloaded and the Application Workspace Launcher won't switch to offline mode.	true
path	The location on the local device where all the content used by the Agent is cached. Note that this path must be relative to the Application Workspace directory. By default, this path is configured to: • Windows: *%ProgramData%\Liquit\Agent\Cache* • macOS: */Library/Caches/Liquit/Agent/Cache*	Cache
tempPath	The temp directory on the local device, used for uploading folders to the Application Workspace. By default, this path is configured to: • Windows: *%ProgramData%\Liquit\Agent\Temp* • macOS: */Library/Application Support/com.liquit.Agent/Temp*	Temp
packageTempPath	When an Application Workspace app is entitled to a user, it is automatically deployed in the local cache directory: %ProgramData%\Liquit\Agent\Cache. The name of the app will actually be a GUID with the `.dat` extension. Once it gets installed on the local device, the app is copied in the directory configured for the `${PackageTempDir}` variable and renamed to its true name. By default, the path of `${PackageTempDir}` is configured to: • Windows: %LOCALAPPDATA%\Temp\ <package-id> For example: C:\Users\< Username >\AppData\Local\Temp\ <package-id> • macOS: $TEMP/<package-id> For example: /var/folders/rm/<device-id?>\T\	`${TEMP}`

Key	Description	Default
autoClean	If enabled, the stale content from the local cache of devices is automatically deleted if one of the following conditions are met: • the content is superseded by a new version. • the content is no longer entitled to a user and the period of time set in the `expireContent` has passed.	true
expireContent	The period in days after which the local cache of devices is deleted automatically if it meets one of the conditions defined above for the `autoClean`.	90

Native Icons

Modify the behaviour of the native icons configured for package entitlements that allow you to display Liquit icons on certain locations within the device's operating system.

```
{
   "nativeIcons": {
      "enabled": true,
      "primary": true,
      "startMenuPath": "${Programs}\\Liquit"
   }
}
```

Key	Description	Default
enabled	Allow Application Workspace to push native icons configured for package entitlements to the operating system of devices.	true
primary	If set to *true*, only icons from the zone defined in the zone key in the Agent.json file will be pushed.	true
startMenuPath	The location where the Start Menu items will be displayed; it allows you to specify a different directory than the default one. This option is available only for Windows.	`${Programs}\\Liquit`

Warning
For the Native Icons to work the Launcher needs to be enabled.

Triggers

Modify the behaviour of the events configured for package entitlements.

```
{
   "triggers": {
      "enabled": true,
      "primary": true
   }
}
```

Key	Description	Default
enabled	Allow Application Workspace to execute events like refresh or login for example.	true

Key	Description	Default
primary	If set to *true*, the Agent will trigger events only for the primary zone defined in the zone key in the Agent.json file.	false

Refresh

Modify the behaviour of the agent refresh process.

```
{
   "refresh": {
      "manual": true,
      "interval": 3600
   }
}
```

Key	Description	Default
manual	If set to *false*, the refresh option from the icon's context menu in the system tray is hidden.	true
interval	This value represents the time interval between Application Workspace refreshes.	3600

Launcher

```
{
   "launcher": {
      "enabled": true,
      "state": "Default",
      "start": "Auto",
      "tiles": false,
      "minimal": false,
      "contextMenu": true,
      "sideMenu": "Tags",
      "close": true
      "startupDelayMilliseconds": 5000
      }
}
```

Keys	Description	Default
enabled	Enables the background service of the Application Workspace Launcher.	true
state	Defines how the Application Workspace Launcher will be shown on start up: • *Default* – Default sized window • *Minimized* – Minimized window • *Maximized* – Maximized window	Default

Keys	Description	Default
start	Defines when the launcher is visible and accessible to the user: • *Disabled* – Application Workspace Launcher does not open during login. • *Auto* – Application Workspace Launcher opens when connected to the zone or offline mode is available. • *Connected* – Application Workspace Launcher opens when connected to the zone. • *Always* – Application Workspace Launcher always opens, even if connection to the zone fails or offline mode isn't available. For Windows OS, the configuration of this option overrides the Windows Startup configuration. This means that even if you disable the Application Workspace Launcher in the Windows Startup list, the launcher will still start automatically when you log in, as long as this setting has a value different than *Disabled*.	Auto
tiles	If set to *true*, Application Workspace will use the tile-themed skin.	false
minimal	If set to *true*, the Side Menu and tabs (**Workspace**, **Contacts**, **Catalog**, **Manage**) are hidden. Only the toolbar without the side menu toggler is displayed.	false
contextMenu	If set to *true*, the context menu is available across Application Workspace Launcher.	true
sideMenu	Choose which tab(s) should be opened by default in the Side Menu.: • *filters* • *tags* • *teams* • *categories*	
close	If set to *false*, the **X** close button of the Application Workspace Launcher window is disabled and the quit option from the icon's context menu in the system tray is hidden.	true
startupDelayMilliseconds	The startup delay is applied only when UserHost is launched at system startup; if the user closes the UI and starts it again manually, the UI opens immediately and the delay is not applied.	0

User Host

Available startup modes

You can choose one of the following modes depending on how the Application Workspace Launcher should be visible when it's started by the ShellAPI at device startup or user login.

Normal

In Normal mode, the Application Workspace Launcher starts at device startup or user login, and logs in the user immediately.
This mode is most suitable for traditional desktop environments where users expect Application Workspace Launcher to be visible after signing in.

Background (default)

In Background mode, the Agent and UserHost are fully initialised, but the user does not see the Application Workspace Launcher unless they open it explicitly later. The login of the user can be triggered. The tray icon is available and displays a

context menu.

This is the default behaviour and is recommended for most environments, including device-centric and automated deployments.

Hidden

In Hidden mode, the Agent and UserHost are fully initialised. The UserHost does not attempt to connect to a zone until UserHost.exe is run again by the user or by a script. The user does not see the Application Workspace Launcher and is not logged in automatically. The tray icon is available but does not display a context menu.

This mode is useful in scenarios where Application Workspace must be running, but user interaction should be deferred or tightly controlled, such as in zero-trust environments.

Example

```
{
  "launcher": {
    "enabled": true,
    "start": "Disabled",
    "close": true
  },
  "userHostStartupOperatingMode": "Hidden",
  "zone": "https://company.liquit.com/"
}
```

Restrict Zones

Define which Liquit Zones are always allowed to communicate with the local Agent.

```
{
    "restrictZones": false
}
```

If set to *true*, only the zones you list will have access to the local Agent. If set to*false*, a warning will be displayed asking the user if the zone where he wants to navigate is allowed to access the agent. The default value is *false*.

Trusted Zones

Here you can add additionally trusted zones.

```
{
    "trustedZones": [
        "zone1.liquit.com",
        "zone2.liquit.com",
        "*.dev.liquit.com"
    ]
}
```

Events

Configure how event data is uploaded to server when the Agent triggers the event.

```
{
   "events": {
      "enabled": true,
      "interval": 30
   }
}
```

Keys	Description	Default
enabled	If set to *true*, event data is uploaded from the agent to the server.	true
interval	The time interval in seconds when the event data is uploaded from the Agent to the server.	30

Web Socket

This protocol allows a more efficient way of handling data. This option is mandatory for the push event feature described in the Events section.

```
{
 "webSocket": {
  "enabled": true
 }
}
```

Further reading

For a detailed description of Device Registration options within Application Workspace UI, see Device Registration.