



stonebranch

Universal Agent 7.2.x

Licensing

© 2022 by Stonebranch, Inc. All Rights Reserved.

1. Licensing	3
1.1 Auto-Start Components	6
1.2 Start-up Sequencing between Primary and Secondary z/OS UAGs	8
1.3 Using Universal Query to Display License Information	9

Licensing

- [Overview](#)
- [Acquiring a License](#)
- [Agent Start-up](#)
 - [When a Remote License Is Expected](#)
 - [When No Remote License Is Expected](#)
- [Managing License Expiration](#)
 - [Managing Expiration Dates for Universal Controller Licenses](#)
 - [Managing Expiration Dates for Local Licenses](#)

Overview


Prior to Universal Agent 6.9.0.0, licenses for individual Agent components were stored in their respective configuration files. Aside from verifying that the license information was valid (that is, the individual license parameters were capable of generating a key at runtime that matched the configured license key), no enforcement of license counts was done.

Universal Agent 6.9.0.0 provides a new method for managing license information which simplifies license administration and improves license enforcement.

The updated license management provided by Universal Agent 6.9.0.0 integrates seamlessly with existing Universal Controller-based deployments.

Universal Agent customers that do not use Universal Controller to schedule and execute their automated workloads can deploy [Universal Controller for Agent Management \(UCAM\)](#) to take advantage of centralized license management.

Note

 For Agents not connected to a Controller or UCAM, or for Agents that connect to a pre-6.9.x Controller, the Licensing method remains as it was for version 6.8.x.


Acquiring a License

The license acquisition process is the means by which a Universal Agent resolves the source of its license information.

A license's source will be either:

- a *remote* license, obtained from the Universal Controller (or UCAM).
- a *local* license, obtained from the Universal Agent's configuration files.

Note


 A Universal Agent eventually will be required to obtain its license information from a Universal Controller or UCAM installation. Support for locally-configured license information will be removed at that time.

Agent Start-up

When Universal Agent starts, the Universal Broker begins by caching the information that exists in all local configuration files. Locally-configured license information is part of this cached information. This functionality has existed for some time, and enables the Broker to serve as the Agent's configuration manager.

Starting with Universal Agent 6.9.0.0, the Broker not only caches the locally-configured license information, it also maps the information to values that are equivalent to those the Agent might receive from Universal Controller.

Note

 At this point, license acquisition is not complete. The Universal Agent does not have all the information it needs to determine whether it should proceed with this local license information or wait for a remote license (i.e., the license source is unresolved).

Later in Universal Broker start-up processing, it may attempt to start the UAG Server. At this point, Universal Broker makes a determination as to whether it should *expect* to receive a Controller license. The Universal Agent expects to receive a remote license when the following conditions are met:

- The `auto_start` value in the UAG Server's Component Definition is set to `yes`.

- The oms_servers value in the UAG Server's configuration contains a value. (The Broker cannot know whether this value will result in a successful OMS connection. It only knows that the UAG Server will attempt the connection.)

When a Remote License Is Expected

When the Agent expects to receive a Controller license, the license remains in an un-acquired state until the UAG Server receives a HELLO message response. If the UAG Server never gets a HELLO response – perhaps the oms_servers value is bad, OMS is not running, or licensed agent counts are exceeded for a pre-6.9.0.0 Controller – the license remains in an un-acquired state.

When the HELLO response is received, the license is acquired.

Note



The license that UAG Server receives does not have to be a 6.9.0.0 license, nor does it actually have to grant the Agent authorization to run licensed work. It only matters that a HELLO response is received and the Agent now can confirm that the license's source is resolved and license acquisition is complete.

When No Remote License Is Expected

When the Broker determines that it **should not** expect to receive a Controller license at start-up, the Broker changes the license's state to acquired and the locally configured license information is used.

Note



Even if the Broker does not expect to receive a Controller license at start-up, it still may eventually receive a Controller license at some point.

If the Broker receives a Controller 6.9.0.0 or later license, the license values being enforced will change and will be applied against any new work that starts.

It is possible for a Broker to replace locally-configured license information with a Controller license, but the reverse can never happen. That is, once a Controller license is acquired, the Broker will never use locally-configured license information.

Managing License Expiration

Managing Expiration Dates for Universal Controller Licenses

When a Universal Agent obtains a license from a Universal Controller (or UCAM), the single expiration date is applied to all licensed Agent components.

License Expiration for Active Agent

The Universal Broker executes a daily check on the license's expiration date. Starting 30 days prior to a license's expiration, Universal Broker will issue the following message when it performs its periodic check on the license:

UNV0130W [] Product license expires in nn day(s). Contact Stonebranch, Inc. for license renewal.

Individual licensed components also will display this message at start-up.

When the expiration date elapses, Universal Broker must ensure that no *new* work can start on that Universal Agent instance. Universal Broker and any auto-started components will continue to run, but any requests to start new work will fail upon a check of the license's expiration date.

When a license expires, you can expect to see the following message in the Universal Broker log and from licensed components:

UNV0129E [] License is expired. Contact Stonebranch, Inc. for license renewal.

Universal Agents do not change state when a Universal Controller's license expires. Any "online" Agents will continue to show online after the license expires.

License Expiration at Agent Start-up

Universal Agents that connect to a Universal Controller whose license is expired still will receive a HELLO response. The Agent will be granted the license provided that the number of licensed Agents is not exceeded.

However, when a Controller's license expires, it will enter a paused state, which will prevent any Universal Agent task execution. After a new license is applied, you must select the Controller's [Server Operations](#) option and execute the Resume Cluster Node task.

Managing Expiration Dates for Local Licenses

If a Universal Agent does **not** receive a remote license from Universal Controller (or UCAM), individual expiration dates specified in local configuration files are still applied to their respective components just as they were prior to 6.9.0.0.

When locally-configured licenses are used, the Broker does not perform a daily check of license expiration dates. Expired licenses are only detected and enforced when the respective component is executed.

Auto-Start Components

- [Agent Component Start-Up](#)
- [Starting an OMS Server](#)
- [Starting a UAG Server](#)
- [Starting All Other Auto-Start Components](#)

Agent Component Start-Up

To ensure proper start-up behavior, the Agent components responsible for obtaining license information from Universal Controller must be started in an orderly fashion. (Prior to 6.9.0.0 releases, where the order in which auto-start components were started did not matter.)

Note



An auto-start component is one in which the AUTOMATICALLY START (auto_start) option value is set to *yes* in that component's [Component Definition](#), thus specifying that the component starts automatically when the Universal Broker is started.

An OMS Server must exist to enable message exchange between the Controller and Agent. An OMS Server does not have to run on every Agent installation, but the OMS Server must start first for those Agents that do have auto_start set to *yes*.

Next, if UAG Server is installed and has an auto_start option set to *yes*, it must start after the OMS Server in order to register with the Controller and receive its license information.

After OMS Server and/or UAG Server (if necessary) start – and valid license information is available to the Universal Broker – the order in which Universal Broker starts any remaining auto-start components is not important.

The Universal Broker will start OMS Server first, provided:

1. OMS Server is defined for the Agent's platform (OMS is not available for z/OS).
2. The auto_start option in the OMS Server Component Definition is set to *yes*.

If OMS Server starts as expected, the Universal Broker will start the UAG Server, provided:

1. UAG Server is installed (the Windows system install allows it not to be installed.)
2. The auto_start option in the UAG Server Component Definition is set to *yes*.

Finally, the startup of all other auto-start components will occur in an indeterminate order.

Starting an OMS Server

When the Universal Broker starts, it attempts to start a component whose Component Definition component_name value is *oms*. This is the OMS Server component.

The OMS Server may not be available on the current Agent platform. For example, OMS Server is not implemented for z/OS and it may not have been selected to be installed on Windows. For such situations, Universal Broker will not find the OMS Server Component Definition.

If an OMS Server Component Definition exists, the value of its auto_start option determines whether the Universal Broker will attempt to start the OMS Server.

- If auto_start is set to *yes*, the Universal Broker will try to start the OMS Server. If OMS Server fails to start, the Universal Broker will issue error message UNV0257E and will stop.
- If auto_start is set to *no*, the Universal Broker will proceed with its start-up processing.

Starting a UAG Server

When the Universal Broker starts, it attempts to start a component whose Component Definition component_name value is *uag*. This is the UAG Server component.

The UAG Server may not be available on the current Agent platform. For example, UAG Server may not have been selected to be installed on Windows. For such situation, Universal Broker will not find the UAG Server Component Definition.

If a UAG Server Component Definition exists, the value of its auto_start option determines whether the Universal Broker will attempt to start the UAG Server.

- If auto-start is set to *yes*, Universal Broker will try to start the UAG Server. If the UAG Server fails to start, the Universal Broker will issue error message UNV0257E and will stop.

Starting All Other Auto-Start Components

After starting the OMS Server and, if necessary, the UAG Server, Universal Broker will start all other auto-start components.

The order in which Universal Broker starts these components does no matter.

If any of these components fail to start, the Universal Broker will issue error message UNV0257E. However, Universal Broker does not stop if any of these components fail to start.

Start-up Sequencing between Primary and Secondary z/OS UAGs

- [Introduction](#)
- [Overview](#)
- [Primary/Secondary Agent Communication](#)

Introduction

This page describes the start-up message sequencing between a Primary and a Secondary UAG in a z/OS Sysplex; non-Sysplex UAGs are not affected.

The primary use of the start-up messaging is to control the number of Secondary agent licenses that are in use.

Overview

New licensing functionality provided by the 6.9.0.0 release simplifies and improves management of Universal Agents deployed within a z/OS Sysplex. The Universal Controller license contains a number of z/OS agents that are licensed to execute. A z/OS Universal Agent is granted permission to run as long the number of licensed z/OS agents is not exceeded.

The Universal Agent identified as the primary agent (via the Universal Broker `sysplex_role` configuration option) is responsible for obtaining the remote license from Universal Controller and distributing it to the secondary agents running in the sysplex. Only a primary agent may communicate with the Controller. Secondary agents can only communicate to the primary agent.

Primary/Secondary Agent Communication

When a Primary agent starts, it will initialize normally (Like a non-Sysplex UAG). Additionally, it will broadcast an (internal) XCF message to all members connected to the UAG Sysplex Group. (A group that encompasses all Primary and Secondary agents in the Sysplex that share the same `system_id` value.) When a Secondary agent receives this message, it will reply to it. Once the Primary agent receives a reply, it will send another XCF message to the replying Secondary agent containing its license information. The Secondary agent will use this information to decide to start processing, or shut down.

When a Secondary agent starts, it will initialize to the point where it can start to process. It will remain in a state where no tracking data of any kind will be processed. It will wait to be contacted by a Primary agent.

Note that tracking data will be collected by the various UAG exits and cached in the high common storage area while there is space. (The size of this area is defined by the `high_common_storage` configuration parameter.) Jobs will continue to run and UAGRERUN will function normally. However, job tracking, File Monitor and Syslog message information will not be processed until the Secondary receives a valid license.

If a Primary shuts down after granting licenses to Secondary agents, all Secondary agents will return to their initial state, waiting to be contacted by the Primary. Secondaries will have to re-acquire a license when the Primary restarts. If the number of licenses has been reduced for any reason (or the number of Secondaries increased), some Secondaries will not receive a new license and will have to stop processing.

Licenses are handed out by the Primary on a first come, first serve basis.

In case a Secondary agent starts after the Primary agent, the Primary agent will detect the startup and broadcast the same XCF message it broadcast on startup. Processing will then proceed as described above. (Agents can detect the startup, and shutdown, of any agent in their group through XCF events that are sent to all agents connected to the group.)

Note



If any of the Secondary Agents start before the Primary Agent, after 15 seconds, the following console message is issued:

```
UAG1100W Waiting for synchronization with Primary UAG.
```

The message is repeated at intervals that double with each iteration. That is the second message is issued 30 seconds after the first, the third after an additional minute, the fourth after two additional minutes, and so on, until the wait ends or the thread is terminated.

When the Primary Agent is available, the following message is issued:

```
UAG1101I Synchronization with Primary UAG complete.
```

Using Universal Query to Display License Information

- [Overview](#)
- [UQUERY Output for a Universal Controller-Obtained License](#)
- [UQUERY Output for Locally-Configured Licenses](#)
- [UQUERY Output for an Un-Acquired License](#)

Overview

Starting with 6.9.0.0, Universal Query can display license information for Universal Agent 6.9.0.0 and later. The information can be used to quickly confirm the source and status of a Universal Agent's license.

UQUERY Output for a Universal Controller-Obtained License

The following information is returned when license information is obtained from a Universal Controller.

```

Universal Query Report
for
Tue 13 Oct 2020 03:48:30 PM EDT

host: 192.168.1.183 port: 30690 ping: YES report: NORMAL

Ubroker Host Name.....:
Ubroker IP Address.....: *
Ubroker Host Port.....: 30690
Ubroker Description.....: Universal Broker Service (x64)
Ubroker Version.....: 6.9.0 Level 0 Release Build 66
Ubroker Service.....: UNKNOWN
Ubroker Status.....: Active
Ubroker Managed.....: NO
Ubroker Start Time.....: 12:12:29 PM
Ubroker Start Date.....: 10/09/2020
UAG Netname(s).....: WIN2016-6900


License Information

License Source.....: Universal Controller
License Customer.....: UA Development
License Status (Agent).....: Granted
License Expiration Date.....: Mon Dec 31 23:59:59 2029
Licensed Distributed Agents.....: Unlimited
Licensed z/OS Agents.....: Unlimited
Licensed UPPS Connections.....: Unlimited
Licensed USAP Connections.....: Unlimited
Licensed SOA Protocols.....: HTTP, JMS, SOAP
    
```

This information shows the following:

- The Agent was granted a license by Universal Controller.
- An unlimited number of distributed agents (that is, Linux and Windows) and z/OS agents are permitted, along with an unlimited number of concurrent USAP and UPPS connections.
- The Universal Agent for SOA is licensed and permits HTTP outbound connections and inbound and outbound JMS and SOAP connections.
- The license expires on December 31, 2029 at 23:59:59 local time.

Note

 If the licenses were expired, the expiration date would be followed by an **(EXPIRED)** tag.

UQUERY Output for Locally-Configured Licenses

The following information is returned when license information is obtained from local configuration files.

```

Universal Query Report
for
Tue 13 Oct 2020 03:54:51 PM EDT

host: 192.168.1.183 port: 30887 ping: YES report: NORMAL

Ubroker Host Name.....:
Ubroker IP Address.....: *
Ubroker Host Port.....: 30887
Ubroker Description.....: Universal Broker Service (x64)
Ubroker Version.....: 6.9.0 Level 0 Release Build 66
Ubroker Service.....: UNKNOWN
Ubroker Status.....: Active
Ubroker Managed.....: NO
Ubroker Start Time.....: 12:12:30 PM
Ubroker Start Date.....: 10/09/2020
UAG Netname(s).....: WIN2016

License Information

License Source.....: Local
License Status (UCMD).....: Granted
License Expiration Date (UCMD).....: Mon Dec 31 00:00:00 2029
License Status (UDM).....: Granted
License Expiration Date (UDM).....: Mon Dec 31 00:00:00 2029
Licensed UPPS Connections.....: Unlimited
License Expiration Date (UPPS).....: Mon Dec 31 00:00:00 2029
Licensed USAP Connections.....: Not authorized
Licensed SOA Protocols.....: HTTP, JMS, SOAP, XD SOAP, IBM WebSphere MQ
License Expiration Date (SOA).....: Mon Dec 31 00:00:00 2029

```

This information shows the following:

- The Agent is still using locally-configured licenses. In this case, the Agent is connected to a pre-6.9.0.0 Controller, which cannot provide an Agent with a remote license.
- The UCMD Manager, UDM Manager, and UPPS have valid licenses that will expire at midnight on December 31, 2029.
- USAP is not licensed on this Agent.
- Universal Agent for SOA is licensed for the Agent and will allow connections for all supported protocols.

Note



If any of the licenses were expired, the expiration date would be followed by an **(EXPIRED)** tag.

UQUERY Output for an Un-Acquired License

The following information is returned when a Universal Agent has not yet acquired its license.

```
Universal Query Report
for
Tue 13 Oct 2020 04:12:35 PM EDT

host: 192.168.1.183 port: 30887 ping: YES report: NORMAL

Ubroker Host Name.....:
Ubroker IP Address.....: *
Ubroker Host Port.....: 30887
Ubroker Description.....: Universal Broker Service (x64)
Ubroker Version.....: 6.9.0 Level 0 Release Build 66
Ubroker Service.....: UNKNOWN
Ubroker Status.....: Active
Ubroker Managed.....: NO
Ubroker Start Time.....: 04:12:20 PM
Ubroker Start Date.....: 10/13/2020
UAG Netname(s).....: WIN2016

License Information

License Status (Agent).....: Not available
```

This information shows the following:

- The Agent expects to receive a license from Universal Controller, but has not received a response to its HELLO message. In this case, the **auto_start** value in the uag component definition was set to yes, but the **oms_servers** value in the UAG Server configuration file did not contain a valid port number for the OMS Server.