

# KEYTYPE

*The information on this page refers to v5.0 and later, which introduced the upgrade license type. For documentation applicable to earlier versions, see [documentation for previous versions](#).*

The KEYTYPE directive specifies whether a network license is exclusive, additive, [upgrade](#) or [token-based](#).

- By default, all network licenses are exclusive. Exclusive licenses are taken into account before other license types. If two exclusive licenses of the same feature name are present, the first is used and the second is discarded.
- Additive licenses coexist with exclusive licenses and other additive network licenses of the same feature name, and are seen as a separate license pool of the same feature.
- [Upgrade licenses](#) are similar to additive licenses, but rather than co-existing with an exclusive license of the same feature name, they are applied to the corresponding exclusive license during license server startup, transparent to the client. Upgrade licenses increase the license count so that only one feature exists.
- [Token-based licensing](#) is always additive, and lets you specify that a particular feature should use one or more other licenses to fulfill checkout requests.

## Usage and recommendations

The KEYTYPE directive can be used only with network licenses; accordingly, the [COUNT](#) directive or [TOKEN\\_DEPENDENCY](#) directive must be set if KEYTYPE is set.

- To prepare an [upgrade license](#), create a feature with the same name as the original license, set the number of licenses using the [COUNT](#) directive, and set KEYTYPE="UPGRADE" in the XML license template file. (See [Upgrade licensing](#) for more information about how to define an upgrade license.)
- To prepare an additive license, set KEYTYPE="ADDITIVE" in the XML license template file.
- To prepare a token-based license, the "KEYTYPE=TOKEN" directive must be used together with the [TOKEN\\_DEPENDENCY](#) directive. Token-based licenses are always additive. (See [Token-based licensing](#) for more information about how to define a token-based license.)

## Additive and upgrade licenses

To add licenses for software features that a customer buys after the initial purchase of a license without changing the original license, you can use either additive or upgrade licenses.

While upgrade licenses increase the license count for an existing feature, additive licenses are treated as a separate license pool of the same feature; therefore, you must checkout licenses in the proper order when using additive licenses.

It is important to understand that multiple additive licenses (even when combined with an exclusive license) do not behave exactly the same as one license with a combined license count (as opposed to [upgrade licenses](#)).

If a client tries to acquire multiple additive licenses, the maximum number of licenses that can be acquired per [LMX\\_Checkout](#) call will be equal to what one license holds. The license server will not perform checkouts across multiple additive licenses.

For example: A license server has one exclusive and two additive licenses of feature "ABC" available, with the license counts 1, 2 and 3. When a client requests one "ABC" license, the exclusive license with count 1 will be used. If this same client later attempts to acquire three more "ABC" licenses, only the additive license with the license count of 3 will be used rather than using one license from the first additive license and two from the third additive license.

To avoid problems, you should checkout all needed additive licenses at one time, or preferably, use [upgrade licenses](#) instead, which eliminates such potential problems completely.