

Important Notice!

Breaking changes in LS Central release 21.0

1 Obsolete code marked with tag 19.0 and below removed

Our solution gets bigger with every release, but we intend to clean up by removing obsolete code in major releases as soon as the code has been marked obsolete for at least a year.

In release 21.0 we removed constructs like procedures and events, but there are no breaking schema changes yet, like Tables and Table fields.

Recommended action: Check if you are using a feature that has been marked obsolete for more than a year. The warning should state the reason and the alternative that is intended to be used.

2 OnPrem-only procedures are now obsolete

All procedures and objects that are marked `#if _LS_SAAS_`, previously marked `[Scope('OnPrem')]`, will become internal.

From 21.0 these procedures and objects will be marked *Obsolete* for OnPrem extensions, until they will finally become internal in 22.0.

If your extension is using any of these, they will now get a warning.

Recommended action: If you have an OnPrem extension, check if there are any warnings. Usually, there are alternative procedures available, but partners can also request that a list of these is made public through the same flow as event requests.

3 Autotest object names now have an LSC prefix

In version 17.4 we added an *LSC* prefix to LS Central, but this was not done for our Autotests extension, which we include in our releases.

In version 21.0, autotests will have a prefix as well.

Recommended action: If you are using or extending the LS Central Autotests extension, be aware of this change.

4 Integration event triggers are now internal/local

In rare cases, it has been technically possible to trigger some of our events from an extension.

These cases are usually an oversight, since the *internal/local* keyword was missing for the *IntegrationEvent* so they become public, by default.

This has prevented us from adding new arguments to existing events, and therefore unnecessarily forcing new versions for events where a breaking change would not have been needed.

Recommended action: This should not require any work. There is a chance that somewhere an upgrade might fail, if it is calling an event function, for example in POS Transaction Event, which usually has no effect. Instead, the events are intended to be subscribed to, with an *EventSubscriber*.