

# Recent Errors



## Planned feature

This page describes a feature planned for future midPoint versions.

This feature is roughly designed and it was evaluated as feasible. However, there is currently no specific plan when it will be implemented because there is no funding for this development yet. In case that you are interested in [supporting](#) development of this feature, please consider activating [midPoint Platform subscription](#).

## Motivation

MidPoint records errors in many different ways. All the errors are recorded in system logs. But that is not the most convenient way how to get information about the errors. The errors are also recorded in the tasks. But that approach does not scale, e.g. when a task run ends up producing large number of errors. And then there are tasks such as live sync with quick successive executions. In such case the errors of one execution is quickly overwritten with a success by a successive execution. Therefore the most recent improvement is to store the errors in the objects (users, shadows). This is an elegant and scalable solution as the information is neatly sorted and distributed. However, it is not very convenient to get information about recent errors: information in the tasks is too brief; information in the object is detailed, but it is scattered across the system and the logfiles are not readily accessible and difficult to work with.

## User Interface Improvement

We propose to create a new page to display recent errors:

- The page will display recent errors from all the objects in the whole system. E.g. it will display all errors that happened in that last 24 hours.
- There will be some basic filtering, e.g. show only errors concerning user objects, errors from a particular task and so on.
- This may be used to create a new dashboard items, to let administrator know about recent errors.

This page may be used to get information about errors in a user-friendly way - even in difficult cases such as livesync errors.

## Further Development

Later midPoint versions may extend the idea:

- Status flags for all the errors, indicating whether this particular error should be ignored, whether it was already taken care of, whether to keep the information (avoid automatic expiration) and so on.
- Support for complex (e.g. multi-node) tasks

## See Also

- [Task Manager](#)