



## One Identity Manager 8.2

### Process Monitoring and Troubleshooting Guide

**Copyright 2021 One Identity LLC.**

**ALL RIGHTS RESERVED.**

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.  
Attn: LEGAL Dept  
4 Polaris Way  
Aliso Viejo, CA 92656

Refer to our Web site (<http://www.OneIdentity.com>) for regional and international office information.

**Patents**

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <http://www.OneIdentity.com/legal/patents.aspx>.

**Trademarks**

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at [www.OneIdentity.com/legal](http://www.OneIdentity.com/legal). All other trademarks are the property of their respective owners.

**Legend**

 **WARNING:** A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

 **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

# Contents

<b>About this guide</b> .....	<b>5</b>
<b>Monitoring handling of processes</b> .....	<b>6</b>
Working with the Job Queue Info program .....	6
Views in the Job Queue Info .....	7
Updating views in the Job Queue Info .....	8
Changing the column configuration in the Job Queue Info program .....	8
Changing program settings in the Job Queue Info .....	9
Creating and using Job queue filters .....	10
Helpful keyboard shortcuts for working with Job Queue Info .....	11
Monitoring process handling in the Job Queue Info .....	12
Details about process handling .....	13
Details about handling process steps .....	15
Details of a process step's parameters .....	16
OUT parameters .....	16
Show hidden parameters .....	17
Re-enabling process steps and processes .....	18
Enabling and disabling extended logging of process steps .....	19
Determining Job server and web server status .....	20
Configure process collection check for Job servers .....	21
Initializing One Identity Manager Service queues .....	22
Displaying DBQueue processing .....	22
Displaying Job queue progress .....	23
Emergency stop .....	23
<b>Support for error localization in One Identity Manager</b> .....	<b>25</b>
Overview of the system configuration and transport history .....	25
Displaying error messages in the One Identity Manager tools .....	26
The error message window in One Identity Manager tools .....	26
Displaying error log messages .....	28
Displaying system journal messages .....	31
Displaying the One Identity Manager Service log file .....	33
Querying One Identity Manager Service availability .....	34

Displaying an application server's status .....	35
Which authentication module is the current user using? .....	35
Which system user is the current user using? .....	36
Which permissions apply to the current user? .....	37
Which program functions are available to the current user? .....	37
Which permission level does the user use? .....	38
<b>Configuring logs in One Identity Manager .....</b>	<b>40</b>
Configuring retention times of messages in the system journal .....	40
Logging process handling errors in the system journal .....	41
Logging logins and logouts in the system journal .....	42
Logging information about OAuth 2.0/OpenID Connect authentication .....	42
Global configuration of logging with NLog .....	43
Logging the One Identity Manager components .....	45
Configuring One Identity Manager Service logging .....	46
Prerequisites for displaying the One Identity Manager Service log file .....	47
Configuring the One Identity Manager Service log file .....	48
Authentication method for displaying the One Identity Manager Service log file .....	49
Advanced logging in the One Identity Manager Service .....	50
Extended debugging in One Identity Manager Service .....	50
Outputting custom messages in the One Identity Manager Service log file .....	51
Logging One Identity Manager Service messages in the event view .....	52
Changing the event log for the One Identity Manager Service .....	53
HTTPLogPlugins log file .....	53
Output of extended return values from individual process components .....	54
Configuring notification behavior for DBQueue Processor initialization .....	55
Enabling the crash recorder .....	55
<b>Appendix: One Identity Manager configuration files .....</b>	<b>57</b>
Application-specific configuration files .....	57
Global configuration file for One Identity Manager tools .....	59
<b>About us .....</b>	<b>60</b>
Contacting us .....	60
Technical support resources .....	60
<b>Index .....</b>	<b>61</b>

## About this guide

One Identity Manager describes the various methods of monitoring processing and of localizing errors in *One Identity Manager Process Monitoring and Troubleshooting Guide*. It also explains advanced log configuration in One Identity Manager.

It is assumed that you understand the concept and the architecture of One Identity Manager. It is also assumed that you are thoroughly familiar with the One Identity Manager tools.

You can find additional notes about error localization and troubleshooting in the other One Identity Manager guides.

### Available documentation

You can access One Identity Manager documentation in the Manager and in the Designer by selecting the **Help > Search** menu item. The online version of One Identity Manager documentation is available in the Support portal under [Technical Documentation](#). You will find videos with additional information at [www.YouTube.com/OneIdentity](http://www.YouTube.com/OneIdentity).

### Detailed information about this topic

- [Monitoring handling of processes](#) on page 6
- [Support for error localization in One Identity Manager](#) on page 25
- [Configuring logs in One Identity Manager](#) on page 40

## Monitoring handling of processes

The Job Queue Info program helps you check the current status of the services running in the One Identity Manager network. It enables a detailed and comprehensive overview of the requests in the Job queue and various One Identity Manager Service requests on the servers. This program makes it easier to work with processes, supplies status information during run-time and allows errors to be quickly recognized and debugged.

### Detailed information about this topic

- [Monitoring process handling in the Job Queue Info](#) on page 12
- [Details about process handling](#) on page 13
- [Details about handling process steps](#) on page 15
- [Details of a process step's parameters](#) on page 16
- [Re-enabling process steps and processes](#) on page 18
- [Enabling and disabling extended logging of process steps](#) on page 19
- [Determining Job server and web server status](#) on page 20
- [Configure process collection check for Job servers](#) on page 21
- [Displaying DBQueue processing](#) on page 22
- [Displaying Job queue progress](#) on page 23
- [Emergency stop](#) on page 23

## Working with the Job Queue Info program

The Job Queue Info program has several views for the layout of processes and process steps in the Job queue. In the Job Queue Info program, you can:

- Monitor handling of Job queue processes.
- Monitor processing of the DBQueue.

- Monitor the status of the Job server and web server.
- Display the One Identity Manager Service log file.
- Display the system journal.

Some of the functions in Job Queue Info, for example certain menu items or key combinations, can only be used if the current user has the corresponding program functionality. For more information about controlling permissions with program functions, see the *One Identity Manager Authorization and Authentication Guide*.

## Views in the Job Queue Info

The Job Queue Info has several views for displaying and editing processes and process steps in the Job queue.

**Table 1: Job Queue Info views**

View	Description
Job queue	This view shows the contents of the Job queue grouped by processes. In the first level of the hierarchy, all the processes are shown with a process count. If a process node is opened, all the processes are shown with start times. The complete process with its hierarchy is displayed under a process node. Each process step contains its success and failure branches as sub elements.
Job server	This view shows the Job queue contents sorted by executing servers. At the first hierarchy level, all Job servers are displayed, with their counts of the different processes, that exist in the Job queue for the Job server. If a Job server node is opened, the process tasks are listed and the number of process step per process task is shown. The process steps are listed by start time under the process task node.
Process History	The view show processes that are already finished. The flow of process handling is displayed in the order of the processes. You can limit the list of processes in the process history to only processes with errors in the program settings. If you select a failed process step, the entire error message is shown in a tooltip.
Base objects	The process history entries and the current Job queue entries are summarized here in this view for the object being processed. If an error occurs during processing and the process handling is stopped ( <b>Frozen</b> or <b>Overlimit</b> status), you can analyze the previous processing sequence in this view. Once all processes have been successfully handled for this object the error messages are removed from the view.
Process	This view gives an overview of how process steps are linked within a process. In this way, the handling sequence of individual process steps for large processes can be monitored better. After selecting a process, all its

View	Description
	process steps are displayed.
Process step	In this view detailed information is displayed for each process step. The view shows the data structure for a process step at compilation time. After selecting a process step, specific information from the Job queue is mapped as well as each parameter of the selected process step with its values.
Parameters	After selecting a process step, the passing parameters of the process step are displayed with their names and their values. If the selected node does not represent a process step, the parameter view is cleared.
Affected objects	This view shows all objects that are affected by a process step.
Progress	This view displays the number of entries in the Job queue is queried. The current value is represented by a number and inserted, at the same time, into a bar graph. The process step progress state is shown in different colors.
Server state	This view gives you a faster overview of all the Job servers and Web servers available in the network.
DBQueue	Calculation tasks in the DBQueue used for DBQueue Processor processing are displayed in this view. The number, sort order and name of the queued requests are displayed.
System journal	Displays entries in the system journal.

## Updating views in the Job Queue Info

To update the views in the Job Queue Info, choose **F5**. If the view focus is on a base object then the whole display is updated and the hierarchy tree is closed. This update refreshes the contents of all views. This update also refreshes the contents of other views.

The views can only ever display a snap-shot of the queue because the contents of the Job queue is continually changing. Therefore, when a node is opened or the view is updated, the necessary information may have already been deleted from the Job queue. If this is the case, the corresponding entry in the hierarchical display is deleted or the corresponding element is not shown.

## Changing the column configuration in the Job Queue Info program

In some of the program views, you can specify which columns are to be displayed.



### ***To specify which columns to display***

- Select a node in the hierarchical display and select **Configure columns** from the context menu.

Select the columns you want to display by moving through the list and accepting with the arrow buttons, then change the order in which they are displayed.

### ***To change the width of the columns on display***

- Double-click a column boundary to optimize the column width.
- Use **Shift + double-click** for a column boundary to optimize the width of all columns.

## **Changing program settings in the Job Queue Info**

The Job Queue Info's general configuration settings are defined in the `Manager.exe.config` configuration file. Valid global configuration settings can also be defined through the `Global.cfg` global configuration file in One Identity Manager's own format. The configuration files are stored in the program directory. For more information, see [One Identity Manager configuration files](#) on page 57.

### ***To change the program settings***

- In the Job Queue Info, select the **Database > Settings** menu.

You can find the program settings for the user configuration in the One Identity Manager database.

### **Language settings**

- **Language:** Language used for formatting data, such as date formats, time formats, and number formats.
- **Other user interface language:** Language for the user interface. The initial program login uses the system language for the user interface. Changes to the language settings take effect after the program has been restarted. The language is set globally for all One Identity Manager programs, which means the language setting does not have to be configured for each program individually.

### **Database queries**

- **Result limit:** Number of entries to load and display for processes or process steps.
- **Polling interval:** Number of seconds between queries. The views are updated at the end of every interval. If the value is **0**, the views are not updated. In this case, use **F5** to update.

## Server state

- **Job servers HTTP port:** HTTP port at which the One Identity Manager Service operates for polling the server state of the Job server. The default value is port **1880**.
- **Status query timeout (s):** Maximum duration of a status query. Job servers that do not respond within this time limit are considered unavailable.

## Process history

- **Only show processes with errors:** Limits the process history display to processes with errors. The setting does not effect how the process history is recorded, only how it is displayed.

# Creating and using Job queue filters

Use filters if you frequently run specific search queries in the Job queue (JobQueue table). You can create your own (private) filters or public filters.

**NOTE:** To create, edit, and delete public filters, the user requires the **Option to define, modify, and delete public filters.** program function. (Common\_PublicFilterDefine).

### To create Job Queue Info filters

1. In the Job Queue Info, select the **Filter > Define filter** menu item.
2. In the **Define a filter** dialog, in the **Filter method** pane, select your preferred filter method. Custom filters allow you to run the following searches:
  - **Wildcard:** Search for a string using wildcards.
  - **SQL:** Search for entries with a SQL condition.
3. In the **Filter parameter** pane, define the search pattern.
  - Enter the search pattern for the **Wildcard** filter method. You can use \* as a wildcard in the search pattern.

#### Example:

Pattern\* - searches for all entries whose display value starts with the Pattern string

\*Pattern - searches for all entries whose display value ends with the Pattern string

\*Pattern\* - searches for all entries whose display value contains the Pattern string

Pattern - searches for all entries whose display value matches the Pattern string

- Enter a condition for the **SQL** filter method. Enter the condition as a valid database query WHERE clause. You can enter the database queries as a SQL query directly or compile the database queries with a wizard. Use the **Expert view** or **Simple view** button to switch to the appropriate view.
4. To save the filter, enter a name and a description for the search filter in the **Save filter** pane and click **Save**.
  5. (Optional) To make the filter available to all users
    - a. Click **Publish**.
    - b. Confirm the security prompt with **Yes**.
  6. To apply a filter, click **Filter**.

#### **To use a saved filter in the Job Queue Info**

1. In the Job Queue Info, select the **Filter > Define filter** menu item.
2. Double-click the search filter in the **Saved filters** pane.
3. Click **Filter**.

#### **To publish a saved filter in the Job Queue Info**

1. In the Job Queue Info, select the **Filter > Define filter** menu item.
2. Double-click the search filter in the **Saved filters** section.
3. Click **Publish**.

## Helpful keyboard shortcuts for working with Job Queue Info

The following keyboard shortcuts are helpful for the daily work with Job Queue Info. Some of the keyboard shortcuts are available only if the logged-in user has the corresponding program functionality.

**Table 2: Shortcuts in Job Queue Info**

<b>Shortcut</b>	<b>Usage</b>
F5	Reload the data.
Shift + Select or Ctrl + Select	Multi-select process steps.

Shortcut	Usage
Ctrl + F2	Mark individual process steps with a bookmark.
F2 or Shift + F2	Switch between the marked process steps.
Ctrl + C	Copy selected data to the clipboard.
Ctrl + T	Set new start time for a process step.
Ctrl + R	Change the number of retries for a process step.
Ctrl + S	Reset start time of a process step.
Ctrl + P	Increase the priority of a process step.

## Monitoring process handling in the Job Queue Info

You can allow monitoring of individual processes. The process information is updated regularly. The progress of a process step is displayed in the font color.

### To monitor process information

- In the Job Queue Info, in the **Job queue** view or the **Base objects** view, select a process and select the **Monitor process** context menu entry.

The process information is updated regularly.

#### TIP:

- To monitor the entire Job queue, select the **Monitor job queue** context menu item in the **Job queue** view.

The context menu entry is only available if the logged-in user has the **Option to monitor the Job queue in Job Queue Info** (JobQueue\_Monitor) program function.

- To display the objects affected by a process step, use the **Affected objects** view.

**Table 3: Job queue display - meaning of the colors**

Color	Meaning	Progress state
Orange	This process step is being processed.	Processing
Yellow	This process step is loaded for processing.	Loaded
Green	This process step is ready for processing.	True
Blue	This process step has already been processed.	Finished

Color	Meaning	Progress state
Black	This process step is not ready for processing.	False
Red	The process step being dealt with cannot be processed. You can re-enable process steps with <b>Frozen</b> status and therefore set them again for processing. The error message is shown in a tooltip.	Frozen
Purple	The process step being dealt with cannot be processed. You can re-enable process steps with <b>Overlimit</b> status and therefore set them again for processing. The error message is shown in a tooltip.	Overlimit
Light purple	The process step cannot be found.	Missing

## Related topics



- [Re-enabling process steps and processes](#) on page 18
- [Initializing One Identity Manager Service queues](#) on page 22
- [Helpful keyboard shortcuts for working with Job Queue Info](#) on page 11

## Details about process handling

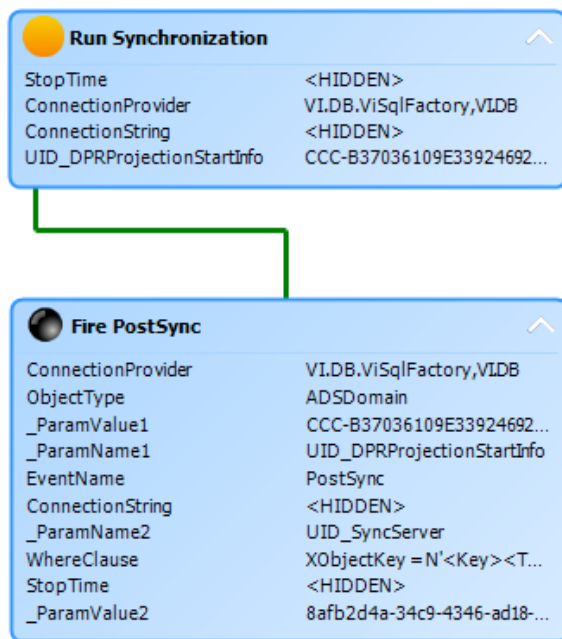
This view gives an overview of how process steps are linked within a process. In this way, the handling sequence of individual process steps for large processes can be monitored better.

### *To display details of the process handling*

- In Job Queue Info, select a process and select the **View > Process** menu item.  
All the process steps of the selected process are displayed.

The process step and its properties are displayed through a special control element. The process step name is displayed in the control's header. The progress state of the process step is clarified by the use of a color icon (●). All other entries represent the parameters for this process step. You can hide or show the parameter list by clicking on the   icons in the header of the control element.

**Figure 1: The process view**



Each control element entry has a tooltip.

The process step's tooltip displays the following information:

- Name of the running queue
- Name of the process component
- Name of the process task name
- Progress state
- Start time of the process step
- Error Message

A parameter's tooltip show the following information:

- Parameter name
- Parameter value

**Table 4: Displaying process's process steps - meaning of the colors**

Color	Meaning	Progress state
Orange	This process step is being processed.	Processing
Yellow	This process step is loaded for processing.	Loaded
Green	This process step is ready for processing.	True
Blue	This process step has already been processed.	Finished

Color	Meaning	Progress state
Black	This process step is not ready for processing.	False
Red	The process step being dealt with cannot be processed. You can re-enable process steps that have <b>Frozen</b> or <b>Overlimit</b> status and therefore queue them again for processing.	Frozen/Overlimit/unknown

## Details about handling process steps

In this view detailed information is displayed for each process step. The view shows the data structure for a process step at compilation time. After selecting a process step, specific information from the Job queue is mapped as well as each parameter of the selected process step with its values.

### To display details of the process step handling



- In Job Queue Info, select a process step and select the **View > Process step** menu.

**Figure 2: Process step view**

Parameter	Value
UID_Job	0A4349C3-454A-46F7-BD32-0238629I
BasisObjectKey	
ComponentAssembly	HandleObjectComponent
ComponentClass	VI.JobService.JobComponents.Hand
DeferOnError	False
ErrorNotify	False
ErrorMessage	
ExecutionType	INTERNAL
GenProcID	7BA8D4CF-568F-4EF0-8FBB-E01E587
IgnoreErrors	False
IsRootJob	True
IsSplitOnly	False
IsToFreezeOnError	False
JobChainName	Proc: vid_InsertForHandleObject Obj
LimitationCount	0
MaxInstance	0
MinutesToDefer	0
NotifyAddress	
NotifyAddressSuccess	
NotifyBody	
NotifvRndvSuccess	

**Table 5: Process step view - meaning of icons**

Icon	Meaning
	Selection of a process step and its parameters.

Icon	Meaning
	Displays a column from the Jobqueue table and the value.
	Displays a process step parameter and the value.

**TIP:** You can copy the data currently selected in the view into the clipboard by pressing **Ctrl + C**. The data format is column name value.

## Details of a process step's parameters

After selecting a process step, the passing parameters of the process step are displayed with their names and their values. If the selected node does not represent a process step, the parameter view is cleared.

### To display process step parameters

- In the Job Queue Info, select a process step and select the **View > Parameter** menu.

This shows all the parameters of the selected process step.

Parameters that contain an object key are displayed as a link.

- You can display the objects by clicking on the link or using the **Show object properties** context menu.
- Use the **Open in Object Browser** context menu item to start the Object Browser and display the object.
- For object keys that refer to a synchronization project, use the **Open in Synchronization Editor** context menu item to start the Synchronization Editor and load the synchronization project.

**TIP:** You can copy the data currently selected in the view into the clipboard using **Ctrl + C**. The data format is column name value.

## OUT parameters

Parameters of the **OUT** or **INOUT** type are parameters that a process component can use to output a value. This value is then available in all subsequent process steps in the process and can be used as a value for parameters of the **IN** type.

The Job Queue Info program is not technically capable of determining the point at which or for which process step these parameters are valid. For this reason, OUT parameters are added to the list of parameters of a process step and are highlighted in blue.

They cannot be seen in the view of the process step under <ParamIN> of a process step because this view presents the data structure of every process step at compilation time. However, the OUT parameters are created in the context of the process.



The time at which the process is loaded into the Job Queue Info is important. If a parameter is overwritten several times, only the state at the time of data query is displayed.

### Example:

Step 1	OUT parameter: X=1
Step 2	IN parameter: X=1
	Value changes: X=2
	OUT parameter: X=2
Step 3	IN parameter: X=2

If the process is loaded into the Job Queue Info before step 2 is processed, the Job Queue Info displays the **X=1** value for the OUT parameter. If the process is loaded after step 2 is processed, the **X=2** value is displayed for the OUT parameter.

For more information about each process step and how the parameters are filled, see the One Identity Manager Service log file.

### Related topics

- [Displaying the One Identity Manager Service log file](#) on page 33
- [Output of extended return values from individual process components](#) on page 54

## Show hidden parameters

Parameters in the One Identity Manager Service log file and in the Job Queue Info program that are not to be displayed are labeled with the **Hidden** option. Values for hidden parameters are shown as <HIDDEN>.

The following users can view hidden parameters in Job Queue Info.

- Administrative users
- In the Job Queue Info, users with the **Option to see the values of hidden parameters in Job Queue Info** program (JobQueue\_ShowHiddenParameters) function

### Related topics

- [Which system user is the current user using?](#) on page 36
- [Which program functions are available to the current user?](#) on page 37

# Re-enabling process steps and processes

The maximum number of times a process can appear in the Job queue can be limited in order to prevent mass modifications.

If the limit is exceeded, the process steps are set to **Overlimit** status and are therefore no longer collected for processing. You can enable these process steps to be run again.

Critical process steps that have failed to be processed are given **Frozen** status. You can also re-enable these processes after correcting the error.

## **To re-enable process steps**

- Select the process step in the Job Queue Info and select the **Re-enable process step** context menu item.

**NOTE:** Use **Shift + select** or **Ctrl + select** to select and re-enable multiple process steps.

## **To re-enable a process step**

- Select the process in the Job Queue Info and select the **Restart process** context menu item.

**IMPORTANT:** When you restart a process, all process steps are processed again. All previously handled processes up to the point at which the error occurred are run again. This can lead to data inconsistencies in certain circumstances.

Sometimes a rerun of the failed process step is not desired. This might occur when the action to be carried out by the process has been carried out manually, for example, an expected directory has been manually added in the meantime. Even so, it may just happen that the process should be rerun even though the error has not been fixed, for example, for a rollback of already processed steps. In this case, to continue with the process, the next process step in the success or failure branch can be handled.

## **To run the subsequent process step**

- Select the failed process step and select the **End with success** or the **End with error** context menu item.

### **NOTE:**

- Both context menu entries are only viewable if there is an error/success successor and the process step has the **Frozen** status.
- Use **Shift+ select** or **Ctrl + select** to select multiple process steps and start further processing.

# Enabling and disabling extended logging of process steps

Success and error messages from process handling are written to the One Identity Manager Service log file. In order to test your processes, you can enable logging mode for process steps in the Job Queue Info. In this case, the processing messages of the processing step are written along with the **Debug** level of information into a separate log. You can display the log in the Job Queue Info as well as in the log file of the One Identity Manager Service itself.

**NOTE:** The log mode is only available if the logged in user has permissions for the **Option to selectively set the logging mode of process steps in the Job queue in Job Queue Info** program function (JobQueue\_LogMode).

## To enable process step logging mode

- To log the messages on success and on failure, select the process step in the **Job queue** view in Job Queue Info and select the **Processing log > Create always** context menu.
- To log the messages on failure only, in the Job Queue Info, select the process step in the **Job queue** view and select the **Processing log > On Error** context menu item.

**NOTE:** You can set the log mode by default for separate process steps. To do this, edit the process step in the Designer in Process Editor. For detailed information about editing processes and process steps, see the *One Identity Manager Configuration Guide*.

## To display the log in Job Queue Info

- In the **Job queues** view in Job Queue Info, select the process step and select the **Processing log > Display** context menu.

This displays the log in a separate window. If a process step was run more than once, for example, if it is re-enabled more than once, several logs are displayed.

## To display the log in the One Identity Manager Service log file

- In the Job Queue Info, select the **Server status** view on the **Job server** tab and select the **Show in browser** context menu item.
- The log is marked with a link entry Log written to Job\_<UID\_Job>\_<yyyymmdd>\_<Timestamp>.log. Click the link to display the log.

The files are stored in the One Identity Manager Service log directory.

Repository structure:

```
<Log directory>\JobLogs\<First 4 digits of the UID_Job>\Job_<UID_Job>_<yyyymmdd>_<Timestamp>.log
```

### To end log mode

- In Job Queue Info, in the **Job queue** view, select the process step and select the **Processing log > Disable** context menu item.

### Related topics

- [Displaying the One Identity Manager Service log file](#) on page 33
- [Which program functions are available to the current user?](#) on page 37

## Determining Job server and web server status

This view gives you a quick overview of Job server availability.

**NOTE:** Set the HTTP port to be queried and the maximum response time in the program settings.

One Identity Manager Service configurations of each Job server stored in the database are used to get more detailed results of Job server status queries. This is especially required if the HTTP server port has been set individually or a Job server processes several queues.

**NOTE:** In the Designer, configure and enable the **Get configuration file from the Job server and write in the Job server configuration** schedule to import the One Identity Manager Service configuration of the Job server into the database. For more information, see the *One Identity Manager Configuration Guide*.

### To query the status of all the existing Job servers in the database

- In the Job Queue Info, open the **View > Server status** view, select the **Job server** tab and press **F5**.

### To query the status of a single Job server

1. In the Job Queue Info, select the **View > Server state** menu item.
2. On the **Job server** tab, select the Job server and then the **Get status** context menu item.

**NOTE:** Use the **Enter credentials** context menu item to enter a user and password to request the server status. You can select more than one Job server. The user information is kept until the next time the Job Queue Info starts.

If the server responds, the system time, the One Identity Manager Service version and the One Identity Manager Service account name are determined and displayed. The software update status as well as the current version of the software are also displayed.

**TIP:** Use **Refresh server list** or **F6** to reload the list of servers.

### **To display a Job server's services**

1. In the Job Queue Info, select the **View > Server state** menu item.
2. On the **Job server** tab, select the Job server and select the **Show in browser** context menu item.

The One Identity Manager Service HTTP server for the Job server is queried and the varying One Identity Manager Service services are displayed.

### **To show the status of a web server**

1. In the Job Queue Info, select the **View > Server state** menu item.
2. On the **Web server** tab, select the web server and select the **Show in browser** context menu item.

| **TIP:** Use **Refresh server list** or **F6** to reload the list of servers.

### **Related topics**

- [Changing program settings in the Job Queue Info](#) on page 9
- [Prerequisites for displaying the One Identity Manager Service log file](#) on page 47
- [Configure process collection check for Job servers](#) on page 21

## **Configure process collection check for Job servers**

On the Job server, the **Last fetch time** property logs when a process step was retrieved by the One Identity Manager Service. If no completion message for this process step has been returned within the time specified in the **Common | Jobservice | LoadedJobsTimeOut** configuration parameter, the One Identity Manager Service runs a check. The time of the last check for loaded process steps (**Last timeout check** property) is set to the current time.

To configure the process collection check, adjust the following settings in the Designer.

- **Common | Jobservice | LoadedJobsTimeOut** configuration parameter  
The configuration parameter contains the time in minutes within which a process should be reported back before a check is run. Default value is **15** minutes.  
| **NOTE:** If there are Job server that have exceeded the time limit, the **Last timeout check** column is displayed in red.
- **Common | MailNotification | NotifyAboutRequestStall** configuration parameter  
Use the configuration parameter to specify whether an email notification should be sent when the One Identity Manager Service stops running requests. The configuration parameter is not set by default.

- **Send notification when Job server is not requesting processes** schedule  
The schedule checks whether the One Identity Manager Service regularly asks for processes from a queue. If a One Identity Manager Service stops making process requests, notification is sent by email.

Enable the schedule. The interval should be set as in the **Common | Jobservice | LoadedJobsTimeout** configuration parameter.

## Initializing One Identity Manager Service queues

A queue is initialized when the One Identity Manager Service starts. The One Identity Manager Service queries the Job queue to see which processes are waiting for its own queue. During the initialization phase, no processes are handled and it may take a long time, particularly if the Job queue is very full.

In Job Queue Info, in the **Progress** view, a warning shows you the queue being initialized. Click the message to get more detail.

### Related topics

- [Displaying Job queue progress](#) on page 23

## Displaying DBQueue processing

Within One Identity Manager, changes to inheritance-relevant data, such as changes to assignments or changes to specific system data, such as changes to the user interface for a system user, necessitate recalculation of the resulting data. These calculations are queued in the DBQueue and processed by the DBQueue Processor.

### *To display DBQueue entries*

- In Job Queue Info, select the **View > DBQueue** menu.

Calculation tasks in the DialogDBQueue table used for DBQueue Processor processing are displayed in this view. The number, sort order and name of the queued requests are displayed. The display is updated at fixed time intervals of 2 seconds.

# Displaying Job queue progress

## *To display the Job queue sequence*

- In Job Queue Info, select the **View > History** menu.

This queries the number of entries in the Job queue. The current value is represented by a number and inserted, at the same time, into a bar graph. The process step progress state is shown in different colors. The display is updated every **5** seconds. The tooltip shows the timestamp and the number of process steps in the Job queue at this point.

**Table 6: Progress view - meaning of the colors**

<b>Color</b>	<b>Meaning</b>	<b>Progress state</b>
Black	Number of process steps that are not read for processing.	False
Green	Number of process steps ready for processing.	True
Yellow	Number of process steps loaded for processing.	Loaded
Blue	Number of process step that have completed processing	Finished
Red	Number of process steps with an unknown progress state	Frozen/Overlimit/Missing

## Emergency stop

In certain circumstances, situations can occur in the system that require processing by One Identity Manager Service and processing of tasks by the DBQueue Processor to be stopped. For example, changes in One Identity Manager can sometimes cause the system to become overloaded by making mass entries in the Job queue or the DBQueue.

To analyze this situation and to take the necessary steps to solve the problem where necessary, in the Job Queue Info program, you can stop the system and restart it once the problem has been fixed.

### *To temporarily halt process handling of a single Job server*

1. In Job Queue Info, select the **View > Server state** menu item.
2. On the **Job server** tab, select the Job server and select the **Stop processing** context menu item.

**NOTE:** After solving the problem, you can use the **Start processing** context menu item to restart processing.

### To stop processing entirely

1. In the Job Queue Info, select the **Help > Emergency stop** menu.
2. To stop DBQueue processing, click the **DBQueue Processor** button.

From this point on no new calculations are carried out in the database.

**NOTE:** After the problem is eliminated, you can click the button to restart the DBQueue Processor.

3. To stop collection of process steps for every One Identity Manager Service, click the **One Identity Manager Service** button.

Process steps that have already been collected are still processed by the services but no new process steps are sent to the services.

**NOTE:** After the problem is eliminated, you can click the button to restart the running of services.

The following icons are displayed in the status bar of all administration tools to inform the user that DBQueue Processor processing and services have been stopped.

**Table 7: Special icon in the status bar for system stop**

Icon	Meaning
	The DBQueue Processor has been stopped.
	The server services have been stopped.



# Support for error localization in One Identity Manager

This section explains various possibilities for error localization within One Identity Manager.

## Detailed information about this topic

- [Overview of the system configuration and transport history](#) on page 25
- [Displaying error messages in the One Identity Manager tools](#) on page 26
- [Displaying the One Identity Manager Service log file](#) on page 33
- [Querying One Identity Manager Service availability](#) on page 34
- [Displaying an application server's status](#) on page 35
- [Which authentication module is the current user using?](#) on page 35
- [Which system user is the current user using?](#) on page 36
- [Which permissions apply to the current user?](#) on page 37
- [Which program functions are available to the current user?](#) on page 37
- [Which permission level does the user use?](#) on page 38

## Overview of the system configuration and transport history

### *To obtain an overview of the system configuration*

- Start the Designer or the Manager and select the **Help > Info** menu item.  
The **System information** tab provides an overview of your current system administration and the installed modules with their versions.

**IMPORTANT:** You will need to provide this information if you contact the Support Team.

**NOTE:** If you have enabled vendor notification, this report is sent once a month to One Identity.

During a schema installation or schema update using the Configuration Wizard, the migration date and migration version are recorded in the database transport history.

When you import a transport package with the Database Transporter, the import date and description, the database version, and the transport package name are recorded in the transport history of the target database.

### ***To display transport history***

- Start the Designer and select the **Help > Transport history** menu item.

## **Displaying error messages in the One Identity Manager tools**

The One Identity Manager tools offer various possible ways to display error messages.

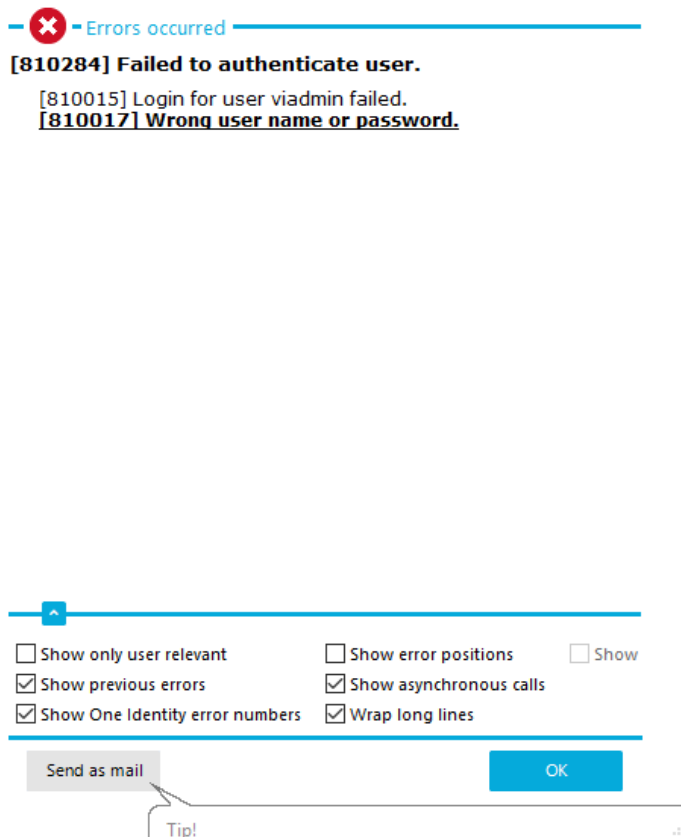
### **Detailed information about this topic**

- [The error message window in One Identity Manager tools](#) on page 26
- [Displaying error log messages](#) on page 28
- [Displaying system journal messages](#) on page 31

## **The error message window in One Identity Manager tools**

Error messages for the One Identity Manager tools are shown in a separate window. In addition, a more detailed description of the error is displayed.

**Figure 3: Error message window**




- To send the messages, click on the **Send as mail** button.  
This creates a new email message in the default mail program and copies over the error text.
- To copy the messages to the clipboard, open the context menu for the **Send as mail** button and click on **Copy to clipboard**.
- To record the steps taken that gave the error, start the Windows Steps Recorder.
  - Open the context menu for the **Send as mail** button and click on **Create problem report**.
  - Confirm the security prompt with **OK**.

You can now start recording the individual steps. Detailed information about recording the steps taken to reproduce a problem using the Windows Steps Recorder can be found in the [Microsoft documentation](#).

Configure the amount of information to be displayed using the options in the error message window.

### To change options

- Open the configuration view for the error messages window with the  button and enable or disable the options you want.

**Table 8: Options for displaying error messages**


Option	Meaning
Show previous errors	Specifies whether all previous errors that lead to the current error, should also be shown.
Show One Identity error numbers	Specifies whether internal error numbers are shown.
Show error positions	Specifies whether error position is also shown in the program code.
Wrap long lines	Specifies whether long error messages are wrapped.
Show only user relevant	Specifies whether all error messages are to be displayed or only those error messages that are classified as user relevant.
Show asynchronous calls	Specifies whether error messages in asynchronous method calls are shown.
Show crash report	Specifies whether error messages from the crash recorder are shown.

### Related topics


- [Enabling the crash recorder](#) on page 55


## Displaying error log messages


A program's error log, as in the Manager for example, displays all the messages, such as error messages and warnings, that have occurred since the program started. The error log is reinitialized when the program is restarted.

**NOTE:** In the Manager, the  icon in the program's status bar indicates new messages in the error log. Double-click the icon to open the error log.

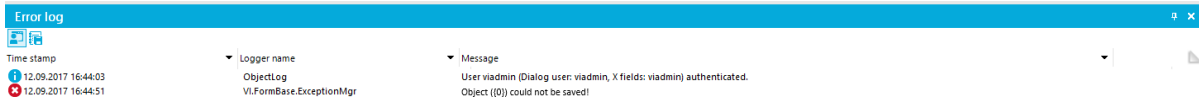
### To display items from the Manager error log

1. In the Manager, select the **View > Error log** menu item.
2. Enable the  view in the error log toolbar.















You can configure how the messages are displayed in the error log. To do this, switch the error log to advanced mode by clicking  on the right of the column headers. Here you have the possibility to debug individual actions.





**TIP:** You can apply different filters to limit the information being displayed. Click the arrow in the column header and select a filter. The  icon in the log toolbar shows whether a filter is active.

**Figure 4: Simple error log (above) and advanced error log (below)**



**Table 9: Meaning of icons in the log**

Icon	Meaning
	Logs all critical error messages. (Info level <b>Fatal</b> )
	Logs all information. (Info level <b>Info</b> )
	Logs all warnings. (Info level <b>Warning</b> )
	Logs all error messages. (Info level <b>Error</b> )
	Logs debugger output. This setting should only be used for testing. (Info level <b>Debug</b> )
	Logs highly detailed information. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome. (Info level <b>Trace</b> )
	Adds a custom filter condition.
	Deletes filter condition.
	Searches for term.
	Searches next term.
	Marks all messages with a specific term.
Buffer size	Sets the message buffer size. The buffer's level is displayed next to the field.
	Deletes the buffer contents.
	Stops logging.
	Starts logging.

Icon	Meaning
	Saves log to file.
	Specifies which column are displayed in the error log.
	Copies selected messages to the clipboard.
	Opens the error log with a text editor.

The following information is displayed about a message. The range of information depends on the severity level of a message.

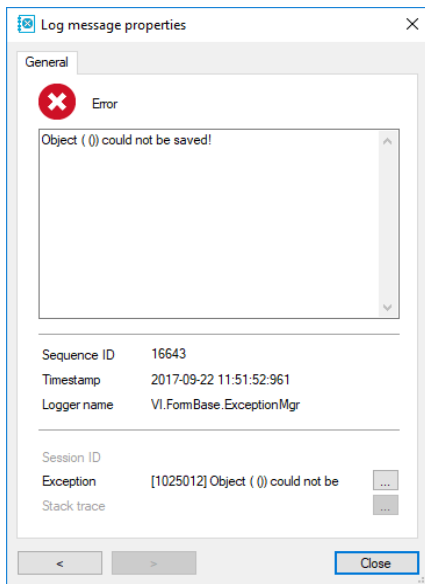
**Table 10: Information about a message**

Detail	Description
Severity code	Level of information supplied for the message.
Timestamp	Time and date of the log entry.
Logger name	One Identity Manager component from which the message was sent.
Message	Logged message.
Error Message	Detailed error message.
Data	Additional data about the message.
Sequence ID	Number of the line in the error log.
Stack trace	Complete stack trace for the error message.
Session ID	Session identification number.

**NOTE:** If there is a filter set on the session ID, only the messages for this session are displayed, for example, loading collections and single objects. If the filter is not set, actions outside of the connection, such as loading of table definitions or configuration parameters, are also displayed.

**TIP:** Double-click a message to display detailed information.

**Figure 5: Detailed information about a message**




## Related topics

- [Logging the One Identity Manager components](#) on page 45

## Displaying system journal messages


The system journal is used to store information, warning, and error messages from different components of One Identity Manager, for example, DBQueue Processor, Configuration Wizard, or One Identity Manager Service. Actions in the Job Queue Info program, such as reactivating process steps, are also recorded in the system journal.

### ***To display system journal entries in the Manager***




1. In the Manager, select the **View > Error log** menu item.
2. Enable the  view in the error log toolbar.

### ***To display system journal entries in the Job Queue Info***

- In the Job Queue Info, select the **View > System journal** menu item.

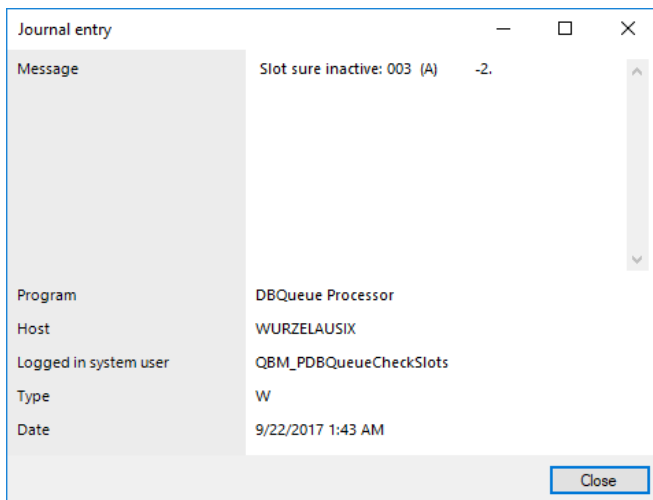
**TIP:** You can apply different filters to limit the information being displayed. Click the arrow in the column header and select a filter. The  icon in the log toolbar shows if a filter is active.

**Table 11: Displaying messages in the system journal**

Icon	Meaning
	Information is written to the error log/system journal.
	A warning has been written to the system journal.
	An error has been written to the system journal.

**TIP:** Double-click a message to display detailed information.

**Figure 6: Detailed information about a message**



The following information is displayed about a message. The range of information depends on the type of message.

**Table 12: Information about a message**

Detail	Description
Message	Logged message.
Program	One Identity Manager component from which the message was sent.
Host	Computer from which the action was started.
Logged in system user	System user that triggered the action.
Type	Type of message. Following values are possible: <ul style="list-style-type: none"> <li>• <b>Warning:</b> This is a warning (<b>Warning</b> info level).</li> <li>• <b>Information:</b> This is information (<b>info</b> info level).</li> <li>• <b>Error:</b> This is an error (<b>Error</b> info level).</li> </ul>



Detail	Description
	<ul style="list-style-type: none"> <li>• <b>Debug:</b> This is a debug message (<b>debug</b> info level).</li> <li>• <b>Trace:</b> This is an output with more detailed information (<b>Trace</b> info level).</li> </ul>
Date	Time and date of the log entry.

### Related topics

- [Logging process handling errors in the system journal](#) on page 41
- [Logging logins and logouts in the system journal](#) on page 42
- [Configuring retention times of messages in the system journal](#) on page 40

## Displaying the One Identity Manager Service log file

You can use a browser front-end to display the One Identity Manager Service log file.

You call up the log file with the appropriate URL:

```
http://<server name>:<port number>
```

The default value is port 1880.

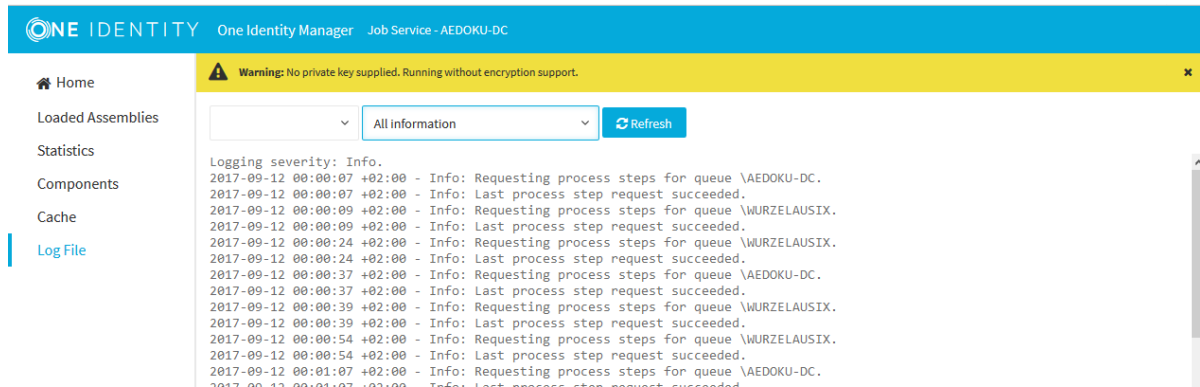
### **To open the One Identity Manager Service log file in the Job Queue Info**

1. Start the Job Queue Info program.
2. In the **Server state** view, select the Job server and select the **Open in browser** context menu item.

The One Identity Manager Service HTTP server for the Job server is queried and the various One Identity Manager Service services are displayed.

3. To display the contents of the log file, select **Log File** in the navigation view.

**Figure 7: The One Identity Manager Service log file**



The messages to be displayed on the web page can be filtered interactively. There is a menu on the website for this. Only text contained in the log file can be displayed in this case. For example, if the message type is **Warning**, messages with the **Info** message type cannot also be displayed if the relevant filter is selected.

The log output is color-coded to make it easier to identify.

**Table 13: Log file color code**

Color	Meaning
Green	Processing successful
Yellow	Warnings occurred during processing
Red	Fatal errors occurred during processing

**NOTE:** If you want to retain the color information to send by email, you need to save the complete web page.

**Related topics**

- [Configuring One Identity Manager Service logging](#) on page 46
- [Prerequisites for displaying the One Identity Manager Service log file](#) on page 47
- [Querying One Identity Manager Service availability](#) on page 34

## Querying One Identity Manager Service availability

The availability of a One Identity Manager Service can be tested over /alive.

Example call: `http://<server name>:<port number>/alive`

Only success (HTTP 200 with **True** as content) or fail (HTTP 500) is returned.

# Displaying an application server's status

You can access the application server from a browser.

Use the appropriate URL for this:

`http://<server name>/<application name>`

`https://<server>/<application name>`

**TIP:** You can open the web server's status display in the Job Queue Info. In the Job Queue Info, select **View > Server state** in the menu and, on the **Web servers** tab, open the web server status display from the **Open in browser** context menu.

You will see different status information. Status information for the application server is displayed as performance indicators. Users with the **Enables log display in the application server** program function (AppServer\_Logs) can see the log.

In addition, API documentation is available here. To access the REST API on the application server, the user required the **Enables access to the REST API on the application server** (AppServer\_API). For more information about the REST API, see the *One Identity Manager REST API Reference Guide*

## Related topics


- [Determining Job server and web server status](#) on page 20

# Which authentication module is the current user using?

One Identity Manager uses different authentication modules for logging in to administration tools. Authentication modules identify the system users to be used and load the user interface and database resource editing permissions depending on their permission group memberships.

For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.

## To identify the current authentication module for the current user

- To display user information, double-click the  icon in the program status bar. The **System user** tab displays the following information about the authentication module used:

- **Authenticated by:** Name of the authentication module used for logging in.
- **Employee UID (UserUID):** Unique ID for the current user's employee if an employee related authentication module is used to log in.

### Related topics


- [Which system user is the current user using?](#) on page 36
- [Which permissions apply to the current user?](#) on page 37
- [Which program functions are available to the current user?](#) on page 37
- [Which permission level does the user use?](#) on page 38

## Which system user is the current user using?

Users log in to the running administration tool using a system ID. Permitted system user IDs are determined by the authentication module you select. When the system user logs in to the One Identity Manager administration tools, the user interface is displayed and editing permissions are assigned depending on the permissions groups to which the user belongs.

For more information about the One Identity Manager authentication modules and system users, see the *One Identity Manager Authorization and Authentication Guide*.

### To identify the current system user for the current user:

- To display user information, double-click the  icon in the program status bar. The **System user** tab displays the following information about the system user.
  - **System user:** Name of the current system user.
  - **Dynamic user:** Specifies whether the logged in user is using a dynamic system user. Dynamic system users are applied when a role-based authentication module is used.
  - **Administrative user:** Specifies whether the current user is using a system user.
  - **Remarks:** Detailed description about the system user.

### Related topics

- [Which authentication module is the current user using?](#) on page 35
- [Which permissions apply to the current user?](#) on page 37
- [Which program functions are available to the current user?](#) on page 37
- [Which permission level does the user use?](#) on page 38

# Which permissions apply to the current user?

Depending on the membership of the system user in permissions groups, the user interface and the authorizations for objects are made available to the current user.

For more information about permissions in One Identity Manager, see the *One Identity Manager Authorization and Authentication Guide*.

## **To identify the current permissions groups for the current user**

- To display user information, double-click the  icon in the program status bar.

The **Permissions groups** tab lists the user's permissions groups.

**NOTE:** On the **System users** tab, the **Read-only** option specifies whether the system user being used has read-only permissions. If so, the user is not permitted to change data.

## **To identify which permissions are assigned to the current user for an object:**

- Select the object for which you want to see the permissions.
- Select the **Properties** context menu.

On the **Permissions** tab, based on the permissions groups, you see what permissions apply to an object.

## **Related topics**


- [Which authentication module is the current user using?](#) on page 35
- [Which system user is the current user using?](#) on page 36
- [Which program functions are available to the current user?](#) on page 37
- [Which permission level does the user use?](#) on page 38

# Which program functions are available to the current user?

Some functions in One Identity Manager tools are available only if the program functions are assigned to the current user. For example, this includes data export from the Manager, calling the SQL Editor in the Designer or showing DBQueue Processor information in all programs.

For detailed information about program functions in One Identity Manager, see the *One Identity Manager Authorization and Authentication Guide*.

### **To identify the program functions available to the current user:**

- To display user information, double-click the  icon in the program status bar. The **Program functions** tab shows the program functions that are available.

### **Related topics**

- [Which authentication module is the current user using?](#) on page 35
- [Which system user is the current user using?](#) on page 36
- [Which permissions apply to the current user?](#) on page 37
- [Which permission level does the user use?](#) on page 38

## **Which permission level does the user use?**

To implement a One Identity Manager database or a One Identity Manager History Database on a SQL Server or a managed instance in Azure SQL Database, you are provided with SQL Server logins and database users for administrative users, configuration users and end users. Permissions at server and database level are matched to suit the user's tasks.


For detailed information about users and their permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Data Archiving Administration Guide*.

For more information about minimum access level for One Identity Manager tools, see the *One Identity Manager Authorization and Authentication Guide*.

#### **NOTE:**

- If you select an existing database connection in the connections dialog, the access level of the login to be used is shown in a tooltip.
- Some user interfaces expect configuration user permissions at least. Logging in as an end user is not possible in this case.

### **To find the access level of the logged in user**

- To display user information, double-click the  icon in the program status bar. On the **System user** tab, in the **SQL access level** field, you will see the access level for the current login. The access levels displayed are **End user**, **Configuration user**, **Administrative user**, **System administrator**, and **Unknown**.

### **Related topics**

- [Which authentication module is the current user using?](#) on page 35
- [Which system user is the current user using?](#) on page 36

- [Which permissions apply to the current user?](#) on page 37
- [Which program functions are available to the current user?](#) on page 37

# Configuring logs in One Identity Manager

One Identity Manager provides various options for extending its log. The log can be configured for each One Identity Manager component.

## Detailed information about this topic

- [Configuring retention times of messages in the system journal](#) on page 40
- [Logging process handling errors in the system journal](#) on page 41
- [Logging logins and logouts in the system journal](#) on page 42
- [Logging information about OAuth 2.0/OpenID Connect authentication](#) on page 42
- [Global configuration of logging with NLog](#) on page 43
- [Logging the One Identity Manager components](#) on page 45
- [Configuring One Identity Manager Service logging](#) on page 46
- [Protokollierung von Meldungen Zielsystemkonnektoren](#)
- [Output of extended return values from individual process components](#) on page 54
- [Configuring notification behavior for DBQueue Processor initialization](#) on page 55
- [Enabling the crash recorder](#) on page 55

## Configuring retention times of messages in the system journal

**Table 14: Configuration parameters for logging in the system journal**

Configuration parameter	Meaning
Common   Journal	General parameter for configuring the system journal.



Configuration parameter	Meaning
Common   Journal   LifeTime	Maximum retention time in days for a system journal entry in the database. Older entries are deleted from the database.
Common   Journal   LifeTime   D	Retention time in days for messages with the <b>Debug</b> information level.
Common   Journal   LifeTime   E	Retention time in days for messages with the information <b>Error</b> level.
Common   Journal   LifeTime   I	Retention time in days for messages with the <b>Info</b> information level.
Common   Journal   LifeTime   T	Retention time in days for messages with the <b>Trace</b> information level.
Common   Journal   LifeTime   W	Retention time in days for messages with the <b>Warning</b> information level.
Common   Journal   Delete	Configuration of deletion behavior for system messages.
Common   Journal   Delete   BulkCount	Number of entries to be deleted in any operation.
Common   Journal   Delete   TotalCount	Total number of entries to be deleted in any processing run.

Messages in the system journal are regularly deleted by the DBQueue Processor.

### ***To delete log entries in the system journal***

- In the Designer, enable the **Common | Journal | LifeTime** configuration parameter and enter the maximum retention period for the entries in the system journal. Use the configuration subparameters to specify the retention time for each information level.
- If there is a large amount of data, you can specify the number of objects to delete per DBQueue Processor operation and run in order to improve performance. To do this, use the **Common | Journal | Delete | BulkCount** and **Common | Journal | Delete | TotalCount** configuration parameters.

## **Logging process handing errors in the system journal**

For more information about editing processes and process steps, see the *One Identity Manager Configuration Guide*.

### ***To log error in process handling in the system journal***

- At the process steps in the Designer, enable the **Log errors to journal** option.

### **Related topics**

- [Displaying system journal messages](#) on page 31

## **Logging logins and logouts in the system journal**

One Identity Manager logins and One Identity Manager logoffs can be recorded in the system journal.

| **NOTE:** Logins and logoffs are recorded in the QBM\_VDialogJournalLoginAudit view.

### ***To record successful One Identity Manager logins***

- In the Designer, set the **Common | Journal | LoginAudit** configuration parameter.

### ***To record One Identity Manager logoffs***

- In the Designer, set the **Common | Journal | LogoffAudit** configuration parameter.

### **Related topics**

- [Displaying system journal messages](#) on page 31

## **Logging information about OAuth 2.0/OpenID Connect authentication**

To support troubleshooting in OAuth 2.0/OpenID Connect authentication you can log personal login data, such as information about tokens or issuers. The log is written to the object log file (<appName>\_object.log) of the respective One Identity Manager component.

### ***To log authentication data***

- In the Designer, set the **QBM | DebugMode | OAuth2 | LogPersonalInfoOnException** configuration parameter.

# Global configuration of logging with NLog

Configuration setting for logging messages are made by NLog in `Globallog.config`. For an exact description and functionality of NLog, see the online help (<http://nlog-project.org/>).

`Globallog.config` is referenced in the One Identity Manager component's configuration files.

**IMPORTANT:** The settings in `globallog.config` apply globally to all One Identity Manager components. Use the application specific `*.exe.config` configuration file to customize individual components.

**NOTE:** The default settings of the `globallog.config` file assume that `%localappdata%` has write access.

If an `*.exe` does not have the correct permissions, by changing the `logBaseDir` variable in `globallog.config` or by introducing a special log configuration in the application-specific `*.exe.config` or `Web.config` configuration file, you can write the log to a directory with write access.

Use variables to define names, output path and layout of the log files. The variable `appName` is defined in the One Identity Manager component's configuration files.

The `targets` section defines the output targets for the messages. NLog already has predefined targets that you can use in the configuration file.

The `rules` section is used to define rules for logging the messages.

By providing `logger name`, you specify for which One Identity Manager components messages are logged. Messages are logged for all components with the default setting `logger name="*"`. To limit logs to certain components, use the name contained in the log.

**Table 15: Logger names of components**

Logger name	Description
FrontendLog	Logs actions in front-ends.
JobGenLog	Logs during process generation.
Jobservice	Logs One Identity Manager Service messages.
ObjectLog	Logs object actions through the object level.
ProjectorEngine	Logs messages from the synchronization engine.
SqlLog	Logs database queries
StopWatch	Logs timings.
SystemConnection	Detailed logging of data communication with the system connection during synchronization, including system configuration and system connectors' data communication.

Logger name	Description
SystemConnector	Logs system connector data communication during synchronization.
Update	Logs update handling.
WebLog	Logs Web service actions.
DebugLogObserver	Logs performance data from the synchronization engine.

You can enter the severity level through:

- `minlevel=` Messages are logged from this severity level. The `LogFileLevel` variable can overwrite the severity level in a custom configuration file.
- `level=` Message are logged which have exactly this severity level. The `eventLogLevel` variable can overwrite the severity level in a custom configuration file.

**Table 16: Permitted severity levels**

Severity Level	Description
Trace	Logs highly detailed information. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
Debug	Logs debug steps. This setting should only be used for testing.
Info	Logs all information.
Warning	Logs all warnings.
Error	Logs all error messages.
Fatal	Logs all critical error messages.

The following files are defined for custom extensions.

```
<include file="{basedir}/custom-log-variables.config" ignoreErrors="true"/>
<include file="{basedir}/custom-log-targets.config" ignoreErrors="true"/>
```

#### Example: Structure of `globallog.config`

```
<nlog autoReload="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <variable name="companyName" value="One Identity"/>
  <variable name="productTitle" value="One Identity Manager"/>
  <variable name="logBaseDir"
```

```

value="${specialfolder:LocalApplicationData}/${companyName}/${productTitle}
}/${appName}"/>
<variable name="layout" value="${longdate} ${level:upperCase=true}
(${logger} ${event-context:item=SessionId}) : ${event-
context:item=Indentation}${message} ${exception:format=ToString,StackTrace}"
/>
<targets async="true">
    <default-wrapper xsi:type="BufferingWrapper" bufferSize="256"
flushTimeout="2000" />
    <target name="logfile" xsi:type="File"
fileName="${logBaseDir}/${appName}.log" layout="${layout}"
encoding="utf-8"
archiveFileName="${logBaseDir}/${appName}.#{#.log"
maxArchiveFiles="7" archiveEvery="Day" archiveNumbering="Rolling"/>
</targets>
<targets>
    <target name="eventLog" xsi:type="EventLog" source="${companyName}
${productTitle} ${appName}"
layout="${message}${newline}${exception:format=toString}"/>
</targets>
<rules>
    <logger name="*" minlevel="${logFileLevel}" writeTo="logfile"/>
    <logger name="*" level="${eventLogLevel}" writeTo="eventLog"/>
</rules>
</nlog>

```

## Logging the One Identity Manager components

In the One Identity Manager default installation, the log files are written to the %LocalAppData%\One Identity\One Identity Manager\<>appName> under the name <appName>.log directory, where appName is the name of the One Identity Manager component.

All messages with a minimum information level of **Info** are recorded in the <appName>.log file. The files are kept for 7 days and backed up daily.

In addition, all messages with a severity level of **Fatal** are recorded in the event log for the **One Identity Manager <appName>** source.

Each One Identity Manager component supports message logging using the integrated NLog functionality. For an exact description and functionality, see the online help (<http://nlog-project.org/>).

The configuration files of the One Identity Manager component (\*.exe.config) contain the nlog section, in which settings for logging by means of NLog are entered. Use the appName variable to pass One Identity Manager component names.

The configuration of the logs is defined in the globallog.config global configuration file. This file is referenced in the configuration files of the One Identity Manager components.

### Example: Referencing NLog logging in an application-specific configuration file

```
<configuration>
  <configSections>
    ...
    <section name="nlog" type="NLog.Config.ConfigSectionHandler, NLog"/>
  </configSections>
  ...
  <nlog autoReload="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <variable name="appName" value="Manager"/>
    <include file="{basedir}/globallog.config" ignoreErrors="true"/>
  </nlog>
  ...
</configuration>
```

### Related topics

- [Global configuration of logging with NLog on page 43](#)
- [Application-specific configuration files on page 57](#)

## Configuring One Identity Manager Service logging

Success and error messages from process handling are written to the One Identity Manager Service log file. Messages can also be written to a server's event log. A severity level can be configured for output to this log file.

You can create most of the settings in the One Identity Manager Service configuration file. Use the Job Service Configuration program to do this. For more information about working with Job Service Configuration and configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Detailed information about this topic

- [Prerequisites for displaying the One Identity Manager Service log file](#) on page 47
- [Advanced logging in the One Identity Manager Service](#) on page 50
- [Extended debugging in One Identity Manager Service](#) on page 50
- [Outputting custom messages in the One Identity Manager Service log file](#) on page 51
- [Logging One Identity Manager Service messages in the event view](#) on page 52
- [HTTPLogPlugins log file](#) on page 53
- [Global configuration of logging with NLog](#) on page 43

# Prerequisites for displaying the One Identity Manager Service log file

The One Identity Manager Service log files can be displayed using a HTTP server (`http://<server name>:<port number>`).

- Users require permission to open an HTTP server. The administrator must grant URL approval to the user to do this. This can be run with the following command line call:

```
netsh http add urlacl url=http://*:<port number>/ user=<domain>\<user name>
```

If the One Identity Manager Service has to run under the Network Service's user account (**NT Authority\NetworkService**), explicit permissions for the internal web service must be granted. This can be run with the following command line call:

```
netsh http add urlacl url=http://<IP address>:<port number>/ user="NT AUTHORITY\NETWORKSERVICE"
```

You can check the result with the following command line call:

```
netsh http show urlacl
```

To display the One Identity Manager Service log file, configure the following modules in the One Identity Manager Service configuration file:

- **FileLogWriter** module  
Create the log file settings in this module.
- **Configuration** module  
Configure the port for displaying the services. The default value is port 1880.
- **HTTP authentication** module  
Set up an authentication method to display the log file.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

### Detailed information about this topic

- [Configuring the One Identity Manager Service log file](#) on page 48
- [Authentication method for displaying the One Identity Manager Service log file](#) on page 49

## Configuring the One Identity Manager Service log file

To generate the log file, customize the **FileLogWriter** module in the One Identity Manager Service configuration file for each One Identity Manager Service.

**Table 17: FileLogWriter parameters**

Parameters	Description
Log file (OutputFile)	Name of the log file, including the directory name. Log information for the One Identity Manager Service is written to this file.  <b>IMPORTANT:</b> The directory specified for the file must exist. If the file cannot be created, no error output is possible. Error messages then appear under Windows operating systems in the event log or under Linux operating systems in /var/log/messages.
Log rename interval (LogLifeTime)	In order to avoid unnecessarily large log files, the module supports the functionality of exchanging the log file with a history list. The LogLifeTime specifies the maximum life of a log file before it is renamed as backup. If the log file has reached its maximum age, the file is renamed (for example, as JobService.log_20040819-083554) and a new log file is started.  Timeout format: day.hour:minutes:seconds
Process step log lifetime ( JobLogLifeTime )	Retention time for process step logs. After this expires, the logs are deleted.  Timeout format: day.hour:minutes:seconds  For test purposes, you can enable logging of individual process steps in the Job Queue Info. The processing messages of the process step is written to a separate log with the <b>Debug</b> NLog severity. The files are stored in the log directory.  Repository structure:



Parameters	Description
	<Log directory>\JobLogs\<<First 4 digits of the UID_Job>\Job_<UID_Job>_<yyyymmdd>_<Timestamp>.log
Number of history logs (HistorySize)	Maximum number of log files. If several log files exist, the oldest backup file is deleted when a new log file is created so that the limit is not exceeded.
Max. log file size (MB) (MaxLogSize)	Maximum size in MB of the log file. Once the log file has reached the limit, it is renamed as a backup file and a new log file is created.
Max. length of parameters ( ParamMaxLength )	Maximum number of characters allowed in a process step parameter so that they are written to the log file.
LogSeverity	Severity levels of the logged messages. Permitted values are: <ul style="list-style-type: none"> <li>• <b>Info:</b> All messages are written to the event log. The event log quickly becomes large and confusing.</li> <li>• <b>Warning:</b> Only warnings and exception errors are written to the event log (default).</li> <li>• <b>Serious:</b> Only exception messages are written to the event log.</li> </ul>
Add server name ( AddServerName)	Specifies whether the server name is to be added to the log entries.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Authentication method for displaying the One Identity Manager Service log file

Use the HTTP authentication module to specify how authentication on the HTTP server works to access the services, for example, to display the log file or status display.

The following module types may be selected:

- BasicHttpAuthentication  
Use this authentication type to specify a user account and the corresponding password for accessing the HTTP server.
- WindowsHttpAuthentication

Use this authentication type to specify an Active Directory group, whose users can be authenticated on the HTTP server. A security ID (SID) or the Active Directory group name in the domain of the Job server can be specified. If Active Directory is not located in the domain of the Job server, the SID must be used.

**NOTE:** If a module is not specified, authentication is not required. In this case, all users can access the services.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Advanced logging in the One Identity Manager Service

To use advanced logging for the One Identity Manager Service, configure the storage of log files in the One Identity Manager Service configuration file in the **Connection** module.

**NOTE:** The given directories must exist and the One Identity Manager Service user account must have write permissions to the directory.

Following parameters are available:

- Process generation log directory (JobGenLogDir)  
Log files are created in the specified directory to log the process generation instructions generated by One Identity Manager Service.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Extended debugging in One Identity Manager Service

The **Configuration** module of the One Identity Manager Service configuration file provides two parameters for advanced debugging:

- DebugMode
- ComponentDebugMode

If the **Debug mode** (DebugMode) parameter is enabled, the One Identity Manager Service writes more extensive information into the log file, such as all parameters transferred to a component and the results of the process handling and their Out parameters.

Individual One Identity Manager Service process components can output additional process data to the One Identity Manager Service log file. For this purpose, you can enable the **Component debug mode** (ComponentDebugMode) parameter in the configuration module. Use this debug mode only for localizing errors because the effect on performance means that it is not recommended for normal use.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Outputting custom messages in the One Identity Manager Service log file

**IMPORTANT:** You should never use the VB.NET functions `Msgbox` and `Inputbox` on servers. Use the functions `VID_Write2Log`, `RaiseMessage` or `AppData.Instance.RaiseMessage`.

You can use the `RaiseMessage` and `AppData.Instance.RaiseMessage` script engine tasks from within process steps to write custom messages to the One Identity Manager Service log file. Use the `ScriptComponent` process component to run the scripts.

The messages in the log file are marked in color depending on the specified severity (`MsgSeverity` parameter).

**Figure 8: Example output of custom messages to the One Identity Manager Service log file**

```
2007-08-10 12:48:58 - Warning: Example warning message
2007-08-10 12:48:58 - Info: Example Info message
2007-08-10 12:48:58 - Serious: Example error message
```

### RaiseMessage

The output is consolidated with other messages and logged at the end of processing the process step.

Syntax:

```
RaiseMessage (MsgSeverity, "string")
```

#### Example:

```
RaiseMessage (MsgSeverity.Warning, "Example warning message")
RaiseMessage (MsgSeverity.Info, "Example Info message")
RaiseMessage (MsgSeverity.Serious, "Example error marked message")
```

### AppData.Instance.RaiseMessage

The output is issued immediately during processing regardless of whether processing of the process step has ended.

Syntax:

```
AppData.Instance.RaiseMessage (MsgSeverity, "string")
```

### Example:

```
AppData.Instance.RaiseMessage (MsgSeverity.Warning, "Example warning message")
AppData.Instance.RaiseMessage (MsgSeverity.Info, "Example Info message")
AppData.Instance.RaiseMessage (MsgSeverity.Serious, "Example error marked
message")
```

For more examples of One Identity Manager Service log file output, see the script example on the installation medium in the directory QBM\dvd\AddOn\SDK\ScriptSamples.

## Logging One Identity Manager Service messages in the event view

To log One Identity Manager Service messages in the server's event log, modify the **EventLogLogWriter** module in the One Identity Manager Service's configuration file. To view the event log, you can use the results display in the Microsoft Management Console, for example.

**Table 18: EventLogLogWriter parameters**

Parameters	Description
EventLog	Name of the event log to which the messages are written. The messages are written to the application log with <b>Application</b> as the default value. <b>NOTE:</b> If more than one One Identity Manager Service write event logs on a server, make sure that the first eight letters in the log name are unique on the server.
LogSeverity	Severity levels of the logged messages. Permitted values are: <ul style="list-style-type: none"><li>• <b>Info:</b> All messages are written to the event log. The event log quickly becomes large and confusing.</li><li>• <b>Warning:</b> Only warnings and exception errors are written to the event log (default).</li><li>• <b>Serious:</b> Only exception messages are written to the event log.</li></ul>
EventID	The ID of the messages written to the event log.
Category	The category of the messages written to the event log.
Source	The name of the source of the messages written to the event log.

Process handling errors can also be written to a server's result log. To do this, use the LogComponent process component.

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Related topics

- [Changing the event log for the One Identity Manager Service](#) on page 53

# Changing the event log for the One Identity Manager Service

By default, the One Identity Manager Service only logs messages in the event log **Application**.

## To use an event log with a different name

1. On the Job server, manually add the file for the One Identity Manager Service to write to. You can use Windows PowerShell, for example, to do this.
  - a. Run Windows PowerShell as administrator on the Job server.
  - b. Run the following CmdLet:

```
New-EventLog -Source "Foobar" -LogName "<file name>"
```
2. Enter this file name in the One Identity Manager Service configuration file, in the module **EventLogWriter** as the name for the event log.
3. Restart the computer.
4. Restart the One Identity Manager Service.

# HTTPLogPlugins log file

If the **HTTPLogPlugin** plugin is configured in the One Identity Manager Service configuration file, a log file is generated with the HTTP queries of the One Identity Manager Service. The file is written in Apache HTTP Server Combined Log Format.

### Example:

```
172.19.2.18 - - [03/Feb/2005:14:55:48 +0100] "GET /resources/JobService.css
HTTP/1.x" OK - "http://<server name>:<port>/status/LogWriter/Config"Mozilla/5.0
(Windows; U; 5.1; en-US; rv:1.7.5) Gecko/20041108Firefox/1.0"
```

**Table 19: Meaning of each entry**

<b>Entry</b>	<b>Meaning</b>
172.19.2.18	IP address that sent the request.
-	Client user name using IDENT protocol (RFC 1413)-
-	User name of the client according to HTTP authentication.
[03/Feb/2005:14:55:48 +0100]	Time that the request is processed on the server
GET /resources/JobService.css HTTP/1.x	Request
OK	Status code-
-	Size of data sent back to the browser
"http://<server name>:<port>/status/LogWriter/Config"	URL from which the page can be accessed
"Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE; rv:1.7.5) Gecko/20041108Firefox/1.0"	Browser name

For more information about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

## Output of extended return values from individual process components

Individual process components have process tasks with parameters that supply extended return values (OUT).

The entire output of the parameter is written to the One Identity Manager Service log file when a error occurs. For example, the output text of the listed commands or programs can be returned when you run a command or a program using the CommandComponent process component.

### To log return values

- In the Designer, enable the **Common | Jobservice | DoReturnOutput** configuration parameter.

### Related topics

- [OUT parameters](#) on page 16

## Configuring notification behavior for DBQueue Processor initialization

If errors occur during initialization of the DBQueue Processor, messages are written to the application log. You can use the results display in the Microsoft Management Console, for example, to view the application log.

Use the **QBM | DBServerAgent | CreateNotification** configuration parameter to configure in which cases error messages are written to the application log. In the Designer, you can modify the configuration parameter as required.

Permitted values are:

- **0**: No logging.
- **1**: Only success messages are logged.
- **2**: Only error messages are logged.
- **3**: All messages are logged.

For more information about the DBQueue Processor, see the *One Identity Manager Configuration Guide*.

## Enabling the crash recorder

The crash recorder saves the previous **128** messages starting at **Debug** level and issues these in the error message window. You can configure the crash recorder using the configuration files for the One Identity Manager tools (\*.exe.config).

- If the variable CrashRecorderBuffer is set to the value **0**, the crash record functionality is disabled.
- Permitted values for CrashRecorderLevel are **Debug, Error, Fatal, Info, Off, Trace** and **Warn**.

### Example: Enabling the crash recorder in the configuration file

```
<configuration>
  <configSections>
    ...
    <section name="connectionbehaviour" type="System.Configuration.
      NameValueSectionHandler" />
  </configSections>
  ...
  <appSettings>
    <add key="CrashRecorderBuffer" value="128" />
    <add key="CrashRecorderLevel" value="Error" />
  </appSettings>
  <connectionbehaviour>
    ...
  </connectionbehaviour>
  ...
</configuration>
```

### Related topics

- [The error message window in One Identity Manager tools on page 26](#)



## One Identity Manager configuration files

General configuration settings can be preset in a configuration file. The configuration file is kept in the program directory. Each administration tool can take its settings from a configuration file in .NET .exe format. Valid global configuration settings can also be defined through a configuration file in One Identity Manager's own format.

### Detailed information about this topic

- [Application-specific configuration files](#) on page 57
- [Global configuration file for One Identity Manager tools](#) on page 59

## Application-specific configuration files

**NOTE:** Use the `globallog.config` configuration file to define global settings that apply to all One Identity Manager components.

One Identity Manager components, such as the Manager or the Designer, have a configuration file for .NET exe's with a predefined format for this. There is a configuration section in the file for each of the different modules of a One Identity Manager component.

The root in the XML file is always called `configuration`. All other sections of the configuration file must be in the mandatory `configSections` section and their type must be defined.

**NOTE:** Entries are case-sensitive.

### Format of the configuration file using `.exe.config` as an example

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
```

```

    <section name="formprovider"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="formarchives"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="vicontrols"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="connectionbehaviour"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="dialogplugins"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="consistencychecks"
    type="System.Configuration.NameValueSectionHandler" />
    <section name="nlog" type="NLog.Config.ConfigSectionHandler, NLog"/>
</configSections>
<dialogplugins>
    <add key="ComplianceRuleSimulation"
    value="VI.DialogEngine.Plugins.ComplianceRuleSimulation,
    AE.DialogEngine.Plugins" />
    <add key="ComplianceRuleSimulationSummary"
    value="VI.DialogEngine.Plugins.ComplianceRuleSimulationSummary,
    AE.DialogEngine.Plugins" />
</dialogplugins>
<consistencychecks>
    <add key="AE" value="VI.ConsistencyChecks.AE.dll" />
    <add key="Common" value="VI.ConsistencyChecks.Common.dll" />
</consistencychecks>
<formarchives>
    <add key="Forms" value="archive:.\???.Forms*.vif;10" />
    <add key="CustomForms" value="archive:.\AE.CustomForms*.vif;5" />
    <add key="CommonForms" value="archive:.\Common.Forms*.vif;5" />
</formarchives>
<vicontrols>
    <add key="defaultcontroldesign" value="System" />
</vicontrols>
<nlog autoReload="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <variable name="appName" value="Manager"/>
    <include file="{basedir}/globallog.config" ignoreErrors="true"/>
</nlog>
</configuration>

```

## Related topics

- [Global configuration of logging with NLog on page 43](#)
- [Global configuration file for One Identity Manager tools on page 59](#)

# Global configuration file for One Identity Manager tools

The Global.cfg is an XML configuration file in One Identity Manager's own simplified format. The advantage of this file is that run-time loading is supported. Each of the different modules has its own section allocated within the file.

You can find an example of a configuration file on the installation medium in the QBM\dvd\AddOn\SDK\ConfigSample directory. If the file Global.cfg is in the program directory, it is used when the One Identity Manager tools start up.

The root in the XML file is always called configuration. Each configuration file module and its values are defined in a section category respectively.

**NOTE:** Entries are case-sensitive. Both the sections and the names of the values must be written in lower case.

## Format of global.cfg

```
<configuration>
  <category name="settings">
    <value name="language">English</value>
    <value name="autoupdateenabled">>true</value>
    <value name="connectiontimeout">15</value>
  </category>
  <category name="connections">
    <value name="database display 1">ConnectionString</value>
    <value name="database display 2">ConnectionString</value>
  </category>
</configuration>
```

**TIP:** To generate the (ConnectionString) connection parameters, use the Config Encryptor program. You will find this program on the installation medium in the directory QBM\dvd\AddOn\ConfigEncryptor.

## Related topics

- [Application-specific configuration files on page 57](#)

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

## Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit <https://www.oneidentity.com/company/contact-us.aspx>.

## Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at <https://support.oneidentity.com/>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at [www.YouTube.com/OneIdentity](http://www.YouTube.com/OneIdentity)
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

## \*

\*.exe.config 57

## A

application server  
    status display 35

## C

combined log format 53  
configurations  
    ComponentDebugMode 50  
    DebugMode 50  
connection  
    JobGenLogDir 50  
Crashrecorder 55  
custom-log-targets.config 43  
custom-log-variables.config 43

## D

database journal  
    delete 40  
    display 31  
    log login 42  
    log logoff 42  
    retention period 40  
DBQueue  
    display 22  
    stop processing 23  
DBQueue Processor  
    stop 23

## E

emergency stop 23  
error log 28  
error message window 26  
EventLogLogWriter  
    EventLog 52  
    LogSeverity 52

## F

FileLogWriter 48  
    HistorySize 48  
    JobLogLifeTime 48  
    LogLifeTime 48  
    LogSeverity 48  
    MaxLogSize 48  
    OutPutFile 48  
    ParamMaxLength 48

## G

Global.cfg 59  
Globallog.config 43

## H

HTTP authentication  
    BasicHttpAuthentication 49  
    WindowsHttpAuthentication 49  
HTTPLogPlugin  
    log file 53

## J

- Job queue
  - initialization 22
  - progress 23
  - stop processing 23
- Job Queue Info 6
  - column configuration 8
  - database journal 31
  - emergency stop 23
  - filter
    - apply 10
    - create 10
    - publish 10
  - HTTP port 9
  - language 9
  - One Identity Manager Service
    - log file 33
  - polling interval 9
  - process history 9
  - program setting 9
  - stop system 23
  - timeout 9
  - update 8
- Job server
  - continue processing 23
  - find state 20
  - stop processing 23

## L

- Logger name
  - FrontendLog 43
  - JobGenLog 43
  - Jobservice 43

- ObjectLog 43
- ProjectorEngine 43
- SqlLog 43
- StopWatch 43
- SyncLog 43
- SystemConnection 43
- SystemConnector 43
- update 43
- WebLog 43

## LogWriter

- FileLogWriter 48

## N

- NLog 45
  - Logger name 43
  - severity level 43

## O

- One Identity Manager Service
  - ComponentDebugMode 50
  - DebugMode 50
  - event log 52-53
  - FileLogWriter 48
  - generation log 50
  - HTTP Server 47
  - log file 19, 48, 50
    - display 33, 47
  - log file (HTTPLogPlugin) 53
  - NSProviderTrace.log 50
  - out parameter 54
  - RaiseMessage 51
  - services 47
  - stop 23

## P

### process

- frozen 18
- monitor 12
- over limit 18
- reenable 18
- restart 18

### process component

- ComponentDebugMode 50
- return value 54

### process handling

- monitor 6

### process step

- details 13, 15
- end on failure 18
- end on success 18
- frozen 18
- log error 41
- logging
  - deactivate 19
  - enable 19
- over limit 18
- parameter 16
  - hidden 17
  - out parameter 16, 54
- processing log 19
- processing state 12-13
- reenable 18

## S

### script

- RaiseMessage 51

### server state 20

### system

- stop 23

### system configurations

- report 25

## U

### user

- access level 38
- authentication module 35
- dynamic 36
- permissions group 37
- program function 37
- system user 36

## W

### Web server

- find state 20