ivanti

Event Auditing Guide

Ivanti UWM Recommended Practices

Document Revision History

Date	Author	Revision	Change Reference
12/27/2011	Jason Cherrington	0.1	Initial Release
1/15/2012	Keith Hart	0.2	Initial review
8/2/2012	Richard Thompson	0.3	Technical review
8/3/2012	Keith Hart	1.0	Approved for release
1/23/2013	Paul Gartland	1.1	Rebranded
10/19/2021	Randy Barger	2.0	Rebranded and overhauled

Contents

1. Introduction / Purpose

The purpose of this document is to provide an overview of how auditing works within the Ivanti User Workspace Management (UWM) software, and provide recommended practices regarding both the implementation of auditing, as well as suggested events to audit.

Auditing is a method of discovering what the UWM agents are doing on the endpoints without having to enable detailed (debug) logging. Many things can be monitored and raised as events. These events can be collected in a variety of ways including being written to a local CSV/XML file, written into the local event logs, and captured by the Management Center server.

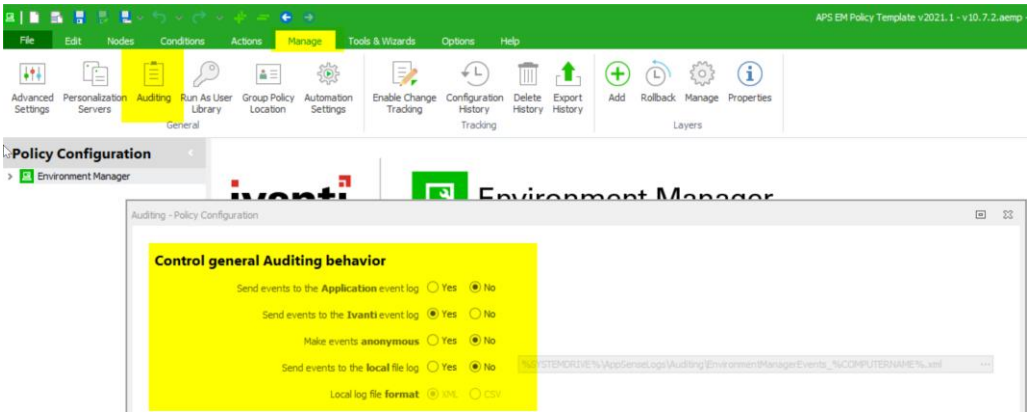
There are two methods of auditing with UWM: Policy Auditing (using local event logs) and Enterprise Auditing (using Management Center and the Deployment Agent).

2. Policy Auditing

Policy Auditing is the capture and storage of auditing events to the **local** (endpoint) event viewer or designated log file. This type of auditing must be enabled in the UWM agent configuration policy.

2.1. Environment Manager

In the Environment Manager policy, click on the Auditing button from the Manage menu. Events can be sent locally to the Application event log, the Ivanti (or AppSense) event log, or to a local CSV or XML file of your choosing. The file option is often used in conjunction with third-party audit collection tools. Standard recommended practice is to send events to the Ivanti event log.



The bottom half of this screen is used to determine which events are desired to be audited. Click on the “Log Locally” checkbox for each event desired.

For Environment Manager policies, it is especially useful to log all trigger actions; both success and fail. When reviewing event logs, it can quickly be determined whether an action succeeded or failed. Also,

if neither a success or fail log is present, it can be deduced that the action never executed. This could be because the trigger never fired, or because some prerequisite condition failed.

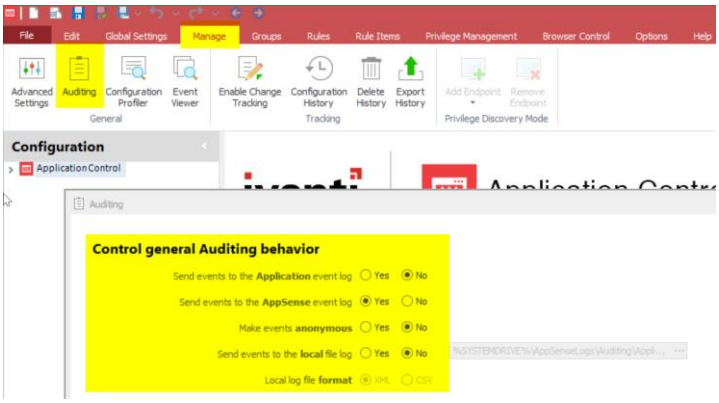
A full listing of available events can be found here: https://help.ivanti.com/ap/help/en_US/em/2021/Content/Environment_Manager/Auditing.htm. Below is a table of Ivanti recommended events.

Event ID	Event Name
9399	Software is not licensed
9405	User logon action success
9406	User logon action fail
9407	User logoff action success
9408	User logoff action fail
9409	Computer startup action success
9410	Computer startup action fail
9413	Computer network available
9414	Computer network available action fail
9420	User session reconnect action success
9421	User session reconnect action fail
9422	User session disconnect action success
9423	User session disconnect action fail
9424	User session locked action success
9425	User session locked action fail
9426	User session unlocked action success
9427	User session unlocked action fail
9428	Process start action success
9429	Process start action fail
9430	Process stopped action success
9431	Process stopped action fail
9432	Network connection action success
9433	Network connection action fail

9434	Network disconnected action success
9435	Network disconnected action fail
9436	User logon (pre-session) action success
9437	User logon (pre-session) action fail
9438	User logon (pre-desktop) action success
9439	User logon (pre-desktop) action fail
9440	User logon (desktop created) action success
9441	User logon (desktop created) action fail
9496	Configuration unsupported
9652	Personalization load error
9653	Personalization save error
9660	Personalization failed
9661	Timeout Communicating with Personalization Server
9662	Trigger Action Times

2.2. Application Control

In the Application Control policy, click on the Auditing button from the Manage menu. Events can be sent locally to the Application event log, the Ivanti (or AppSense) event log, or to a local CSV or XML file of your choosing. The file option is often used in conjunction with third-party audit collection tools. Standard recommended practice is to send events to the Ivanti event log.



The bottom half of this screen is used to determine which events are desired to be audited. Click on the “Log Locally” checkbox for each event desired.

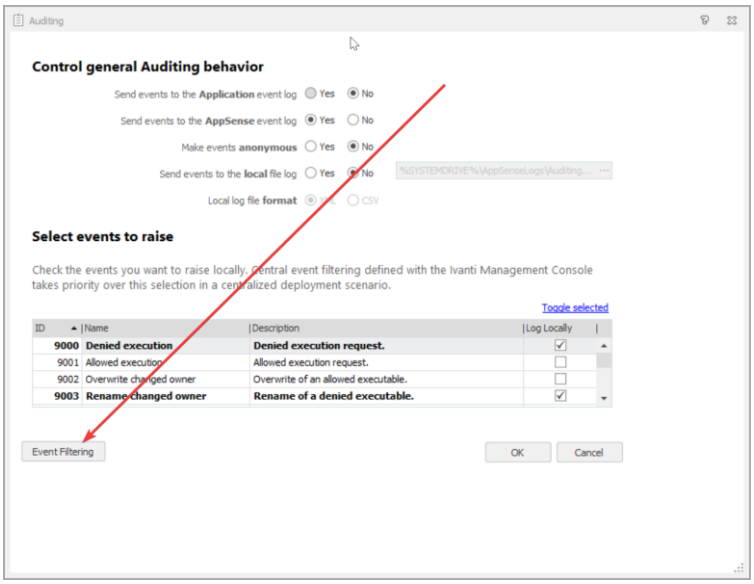
For Application Control policies, it is useful to log all deny and privilege events.

A full listing of available events can be found here: https://help.ivanti.com/ap/help/en_US/am/2021/Content/Application_Manager/Auditing.htm. Below is a table of Ivanti recommended events.

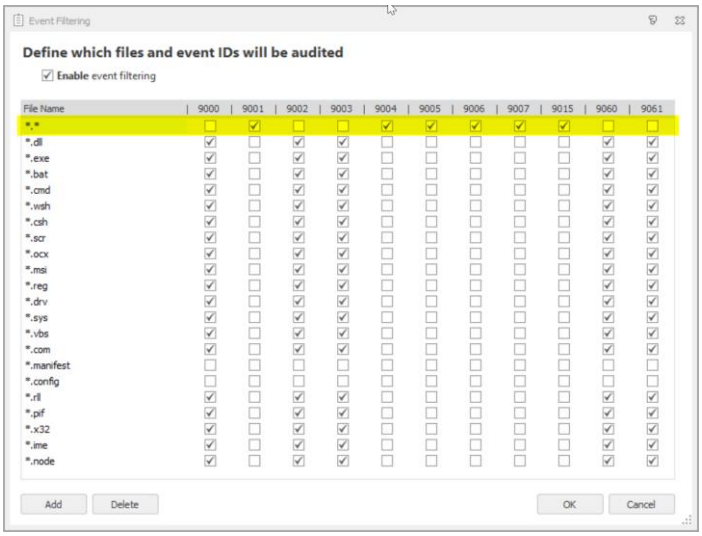
Event ID	Event Name
9006	Self-Authorization
9007	Self-Authorized allow
9009	Scripted Rule Timeout
9010	Scripted Rule Fail
9012	Trusted Vendor Denial
9013	Network Item denied
9017	Application Termination
9018	Application User Privileges Changed
9019	Web Installation allowed
9020	Web Installation restricted
9021	Web Installation restricted
9022	Web Installation fail
9023	Self-Elevation allowed
9024	URL Redirection
9051	Policy Change granted
9052	Policy Change invalid response code
9053	User-requested allow
9054	User-requested elevate
9060	Denied execution (Trusted Ownership)
9061*	Denied execution (Rule Policy)
9062*	Admin process started event
9099	Agent not licensed

* These two events combined will display the same events as event ID 9000. Older agents did not have the 9061/9062 events, so event ID 9000 should be captured instead for older environments.

Event Filtering is another option worth configuring. This button/option is found on the same Auditing screen.



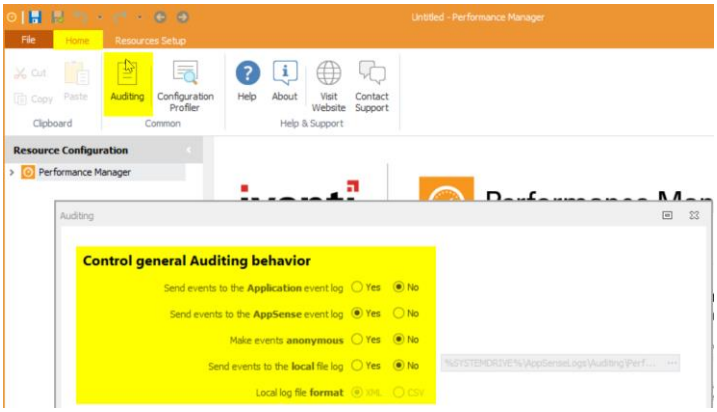
High volume events, such as 9000, 9002, 9003, 9060, and 9061 may generate more event log entries than desired. When the *.* wildcard is used, all events will be captured. For 9000, 9060, and 9061, these may include rename/overwrite events for many other file types, which isn't ideal from a troubleshooting perspective. For these events, deselect the wildcard, and only select the file types that are desired to be audited.



Note: *While Event Filtering is defined in the policy auditing settings, the filters also affect events collected and uploaded to Management Center (Enterprise Auditing).*

2.3. Performance Manager

In the Performance Manager policy, click on the Auditing button from the Home menu. Events can be sent locally to the Application event log, the Ivanti (or AppSense) event log, or to a local CSV or XML file of your choosing. The file option is often used in conjunction with third-party audit collection tools. Standard recommended practice is to send events to the Ivanti event log.



The bottom half of this screen is used to determine which events are desired to be audited. Click on the "Log Locally" checkbox for each event desired.

A full listing of available events can be found here: https://help.ivanti.com/ap/help/en_US/pm/2020/Content/Performance_Manager/Audit.htm. Below is a table of Ivanti recommended events. *However, **customers are advised to only audit those events which are useful to them**. Many of these events can be quite verbose, filling up event logs for no reason.*

Event ID	Event Name
9104	Thread Throttling Clamping On
9105	Thread Throttling Clamping Off
9106	Application CPU Usage clamping On
9107	Per Application Memory Usage Exceeded
9108	Per Application Memory Usage Reduced
9109	Per Application Memory Usage Terminated
9110	Application CPU Usage Clamping Off
9115	Working set trimmed
9116	CPU Affinity changed

9119	Per Application Hard Memory Limit Reached
9120	Thread Throttling - Clamped Processes
9121	Application CPU Soft Limit - Started
9122	Application CPU Soft Limit - Stopped
9123	Application CPU Reservation Applied
9199	Valid License Not Found
9236	System Memory Exceeded Threshold Warning
9237	System Memory Exceeded Threshold Lapse

2.4. File Director

The File Director agent does not utilize a policy configuration file. Rather, settings for Windows agents are manipulated via registry or group policy settings.

A full listing of events can be found here: https://help.ivanti.com/ap/help/en_US/fd/2021/Content/FileDirector/Admin/Clients/Windows_Client/Client_Side_Auditing.htm. ***This list is not configurable; all events will be stored in the Windows Application event logs.***

3. Enterprise Auditing

Enterprise Auditing is the capture and storage of auditing events in the **Management Center database**. This type of auditing must be enabled in the Management Center console, for each deployment group.

The Deployment Agent (aka Client Communications Agent or CCA) is responsible for collecting events on the endpoints, and then uploading the events to Management Center at the appropriate interval.

A detailed knowledgebase article exists, describing the configuration and data flow of Enterprise Auditing:

How Enterprise Auditing Works - https://forums.ivanti.com/s/article/How-Enterprise-Auditing-Works

A full listing of available events can be found here: https://help.ivanti.com/ap/help/en_US/mc/2021/Content/Management_Center/Deployment_Group_Settings_Auditing.htm. Below is a table of Ivanti recommended events.

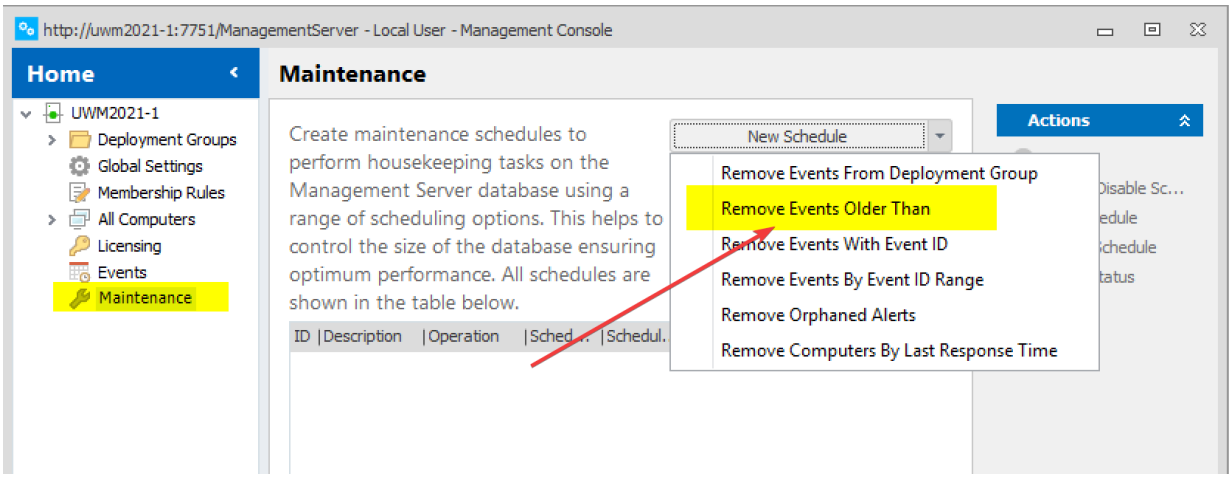
Application Control

Event ID	Event Name
9006	Self-Authorization
9007	Self-Authorized allow
9023	Self-Elevation
9051	Policy Change granted
9060	Denied execution (Trusted Ownership)
9061	Denied execution (Rule Policy)
9062	Application started elevated
	Environment Manager
Event ID	Event Name
9660	Personalization failed
9661	Timeout Communicating with Personalization Server
	Performance Manager
Event ID	Event Name
	Customers are advised to only audit those events which are useful to them
	Management Center
Event ID	Event Name
9090	Service Ended Unexpectedly (Application Control)
9190	Service Ended Unexpectedly (Performance Manager)
9390	Service Ended Unexpectedly (Environment Manager)
9704	Priority Event Failure
9705	Event Upload Failure
9711	Package Installation Failure
9713	Failover Change URL
9755	BITS Error
9790	Service ended unexpectedly
	File Director
Event ID	Event Name
9812	ILS Configuration Failure
9816	Low Disk Space
9821	Cache fully in-sync at logoff
9825	File not in-sync
9834	Summary of Outside the Profile data scan
	User Personalization Manager
Event ID	Event Name
	None recommended

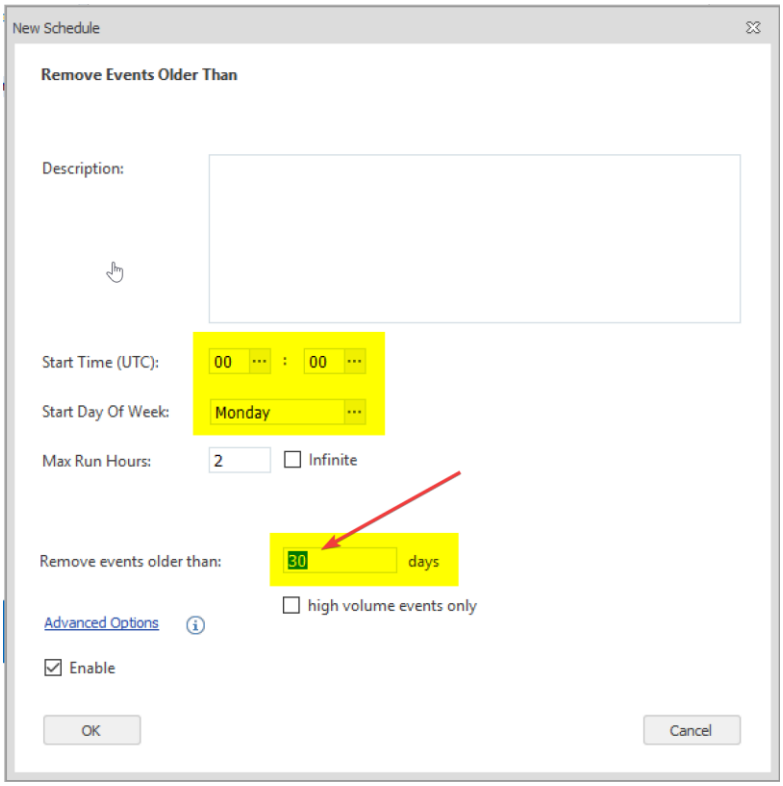
Note: *Events 9060, 9061, and 9062 are high volume events. These events are essential for auditing the environment. However, if auditing information is not needed, these events should be deselected.*

3.1. Event Maintenance

All events are stored in the Management Center database. By default, events are stored indefinitely. It is highly recommended to configure Maintenance to remove old events.



Select the day of week and start time to run, and the number of days to retain events.



3.2. Configuring Enterprise Auditing for All Deployment Groups

Enterprise Auditing is configured on a per deployment group basis. There is no global setting. If it's desired to have the same auditing settings for ALL deployments groups, each deployment group must be configured manually to match. There is, however, a SQL script that may be executed directly against the database to update all of the deployment group auditing settings at once. ***Note:*** *This script is only valid for Management Center* ***2021.1 or later****. All scripts should be run against a test environment before being applied to production. Ensure full backups are available before running any scripts.*

```
SET NOCOUNT ON

--Define event definition IDs to deselect
CREATE TABLE #EventIDsToDelete (a int)
INSERT INTO #EventIDsToDelete VALUES(9096),(9495),(9496),(9600),(9601),(9602)

--Define event definition IDs to select
CREATE TABLE #EventIDsToAdd (a int)
INSERT INTO #EventIDsToAdd
VALUES(9006),(9007),(9023),(9051),(9060),(9061),(9062),(9660),(9661),(9090),(9190),(939
0),(9704),(9705),(9711),(9713),(9755),(9790),(9812),(9816),(9821),(9825),(9834)

SET NOCOUNT OFF

--Unassign event definition from all groups
DELETE FROM [GroupEventFilter] WHERE EventDefinitionFK IN (SELECT a FROM
#EventIDsToDelete)
PRINT ' DELETED GROUP EVENT FILTERS'

--Disable Alert rules
UPDATE [AlertRules] SET [Enabled] = 0, [ModifiedTime] = GETUTCDATE() WHERE [Enabled] =
1 AND [EventQuery] NOT LIKE '%|%' AND CONVERT(int,[EventQuery]) IN (SELECT a FROM
#EventIDsToDelete)
PRINT ' DISABLED ALERT RULES'

--Assign event definitions to all groups
MERGE [GroupEventFilter] AS gef
USING (
       SELECT g.[GroupPK], e.[a]
       FROM [Groups] g
       JOIN #EventIDsToAdd e ON 0 = 0
) AS s  ON gef.GroupFK = s.GroupPK AND gef.EventDefinitionFK = s.a
WHEN NOT MATCHED BY TARGET THEN
       INSERT (GroupFK, EventDefinitionFK)
       VALUES (s.[GroupPK], s.[a]);
PRINT ' ADDED EVENT IDS'

--Clean up tables
DROP TABLE #EventIDsToAdd
DROP TABLE #EventIDsToDelete
```