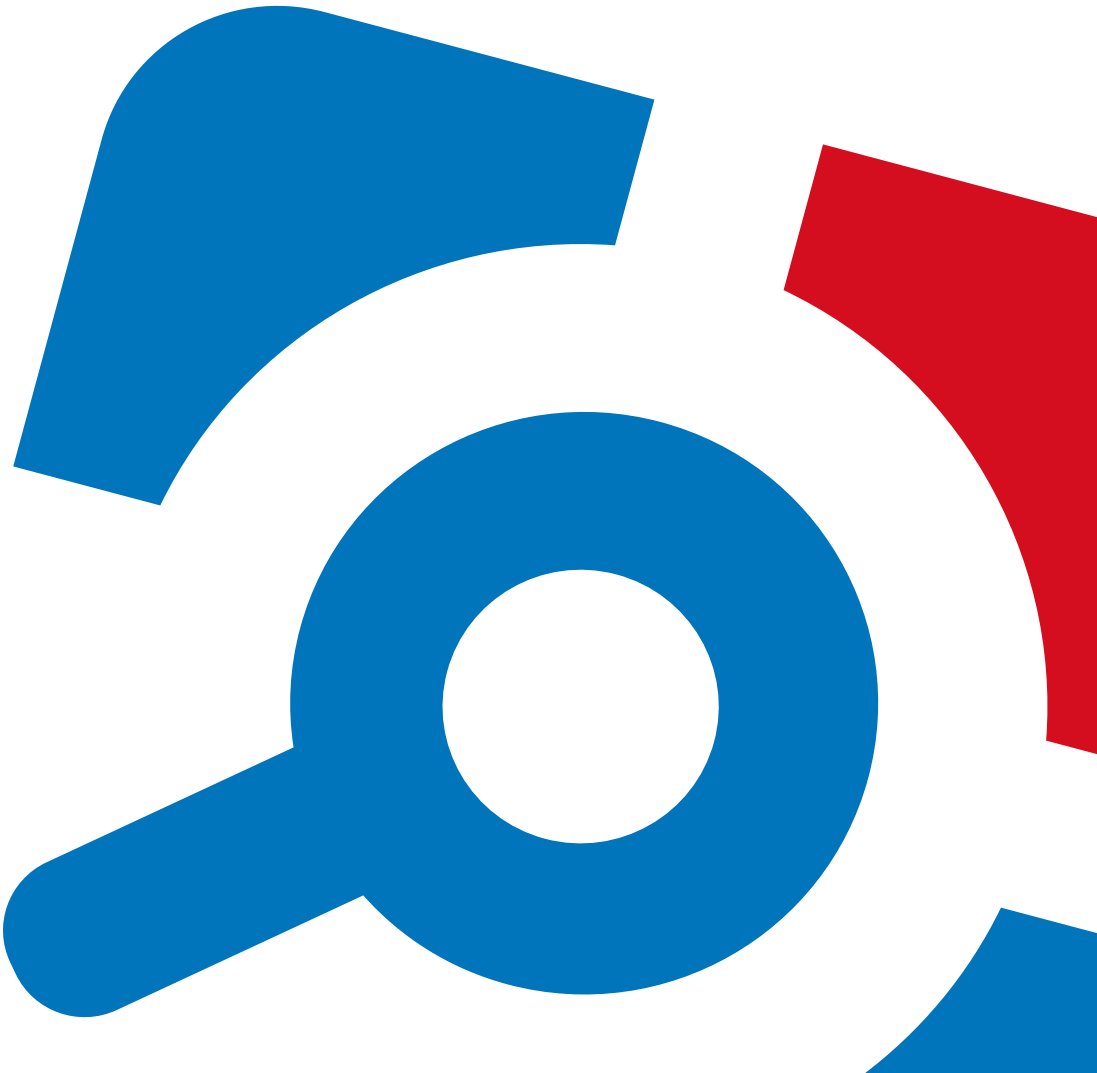
netwrix

Netwrix Auditor Add-on for ConnectWise Manage

Quick-Start Guide

Version: 9.8
3/28/2019



Legal Notice

The information in this publication is furnished for information use only, and does not constitute a commitment from Netwrix Corporation of any features or functions, as this publication may describe features or functionality not applicable to the product release or version you are using. Netwrix makes no representations or warranties about the Software beyond what is provided in the License Agreement. Netwrix Corporation assumes no responsibility or liability for the accuracy of the information presented, which is subject to change without notice. If you believe there is an error in this publication, please report it to us in writing.

Netwrix is a registered trademark of Netwrix Corporation. The Netwrix logo and all other Netwrix product or service names and slogans are registered trademarks or trademarks of Netwrix Corporation. Microsoft, Active Directory, Exchange, Exchange Online, Office 365, SharePoint, SQL Server, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks and registered trademarks are property of their respective owners.

Disclaimers

This document may contain information regarding the use and installation of non-Netwrix products. Please note that this information is provided as a courtesy to assist you. While Netwrix tries to ensure that this information accurately reflects the information provided by the supplier, please refer to the materials provided with any non-Netwrix product and contact the supplier for confirmation. Netwrix Corporation assumes no responsibility or liability for incorrect or incomplete information provided about non-Netwrix products.

Table of Contents

1. About This Document

This guide is intended for the first-time users of Netwrix Audit add-on for ConnectWise Manage. It can be used for evaluation purposes, therefore, it is recommended to read it sequentially, and follow the instructions in the order they are provided. After reading this guide you will be able to:

- Install the add-on
- Configure its parameters
- Use the add-on

NOTE: The add-on works only in combination with Netwrix Auditor so this guide covers a basic procedure for running the add-on and assumes that you have Netwrix Auditor installed and configured in your environment. For installation scenarios, data collection options, as well as detailed information on Integration API, refer to the Online Help Center and product documentation:

- Netwrix Auditor Online Help Center
- Netwrix Auditor Installation and Configuration Guide
- Netwrix Auditor Integration API Guide

2. Solution Overview

Managed Service Providers (MSP) need to effectively utilize and standardize IT service management tools. Those who use for that purpose the ConnectWise Manage solution usually have similar processes in place:

- When an incident or a problem occurs in the IT environment, managed client sends (usually by email) a request to the MSP's service desk. A service ticket is then created manually or automatically in ConnectWise Manage.
- Each ticket is assigned to authorized personnel for investigation and resolution in accordance with the existing workflow.
- To control ticket handling and report on statistics, ConnectWise service boards are used.

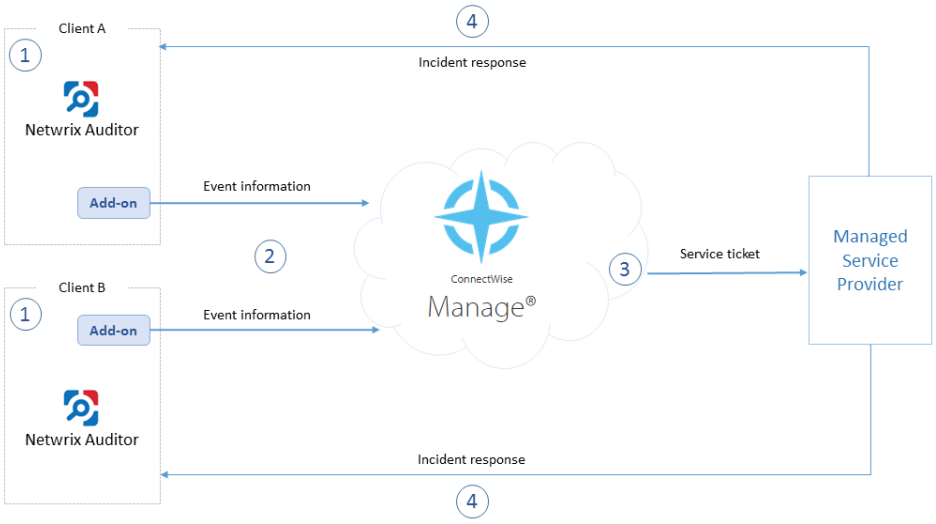
Netwrix has built a ready-to-use add-on that automates incident management, automatically creating service tickets for security alerts triggered by Netwrix Auditor. This integration brings in the following benefits:

- Seamless integration with the existing MSP service process
- Speeding up the process of restoring secure, normal business service
- Minimizing the gap between incident detection and the start of a resolution process
- Automating ticket handling and reducing human errors that could impact its quality
- Meeting or exceeding service level agreements (SLAs) while saving time and effort

To implement the solution, Managed Service Provider does the following on the client side:

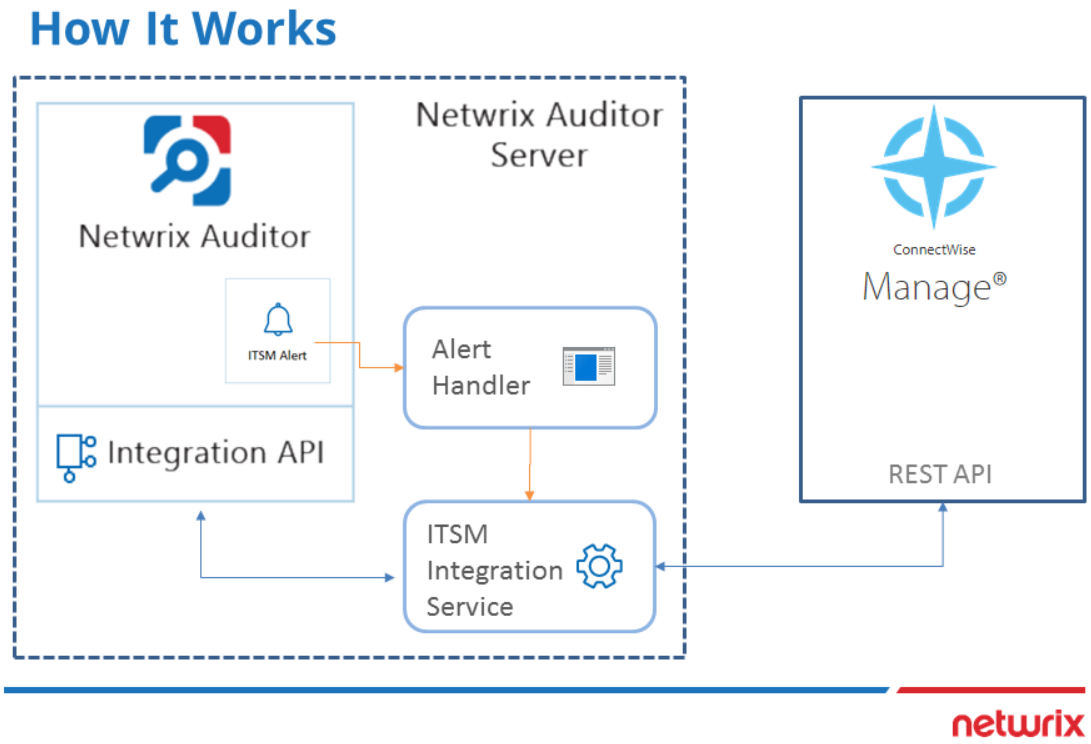
1. Deploys and maintains Netwrix Auditor that monitors users' activity and configuration changes
2. Installs and configures integration solution (add-on) on Netwrix Auditor server
3. Controls ticket resolution and corrective measures

On a high level, the workflow is as follows:



1. Managed Service Provider installs and configures the add-on on Netwrix Auditor server. MSP also enables the necessary alerts in Netwrix Auditor, specifying add-on launch as the response action in the alert settings.
2. Whenever the alert is triggered, the add-on uses Netwrix Auditor Integration API to retrieve activity record for the original event from the audit store. An activity record contains the user account, action, time, and other details. The add-on creates a service ticket in ConnectWise Manage, populates it with data from the activity record, and assigns Impact, Priority and SLA status to the ticket.
3. The designated service team performs data analysis and root cause detection to resolve the ticket; MSP is notified of the results and possible response actions to take on the client side.
4. MSP performs actions for incident response.

Solution architecture and key components are shown in the figure below:



- **Alert Handler (Netwrix.ITSM.AlertResponseAction.exe)** — the executable that is specified in the Netwrix Auditor alerts as the response action. Alert Handler:
  1. Receives the IDs of the alert and associated activity record.
  2. Forwards them to the Netwrix Auditor ConnectWise Manage Integration Service over RPC, putting the alert into the service queue.

NOTE: For details on the alert response action, see Create Alert.

- **Netwrix Auditor ConnectWise Manage Integration Service (Netwrix.ITSM.IntegrationServiceCW.exe)** — the main component of the solution, implemented as Windows service. It does the following:
  1. Interacts with Netwrix Auditor via its Integration API to retrieve the activity records from the Audit Database by record ID.
  2. Forwards activity record data to ConnectWise Manage via its REST API, creates a new service ticket and populates its properties, as specified by user in the add-on configuration.

3. Prerequisites

Before running Netwrix Auditor Add-on for ConnectWise Manage, ensure that all the necessary components and policies are configured as follows:

Location	Prerequisites
Netwrix Auditor Server	• The add-on requires Netwrix Auditor version 9.7 or higher. • It will run on the computer where Netwrix Auditor Server works. • Unless specified, the **Netwrix.ITSM.IntegrationServiceCW.exe** Windows service (main add-on component) will run under the **LocalSystem** account. Also, ensure that: • The Audit Database settings are configured in Netwrix Auditor Server. • The TCP 9699 port (default Netwrix Auditor Integration API port) is open for inbound connections. • Monitoring plans are configured to store data to the Audit Database. • The account that will be used by **Netwrix.ITSM.IntegrationServiceCW.exe** component to access Netwrix Auditor Server is granted the **Global administrator** role in Netwrix Auditor. -OR- is a member of the **Netwrix Auditor Administrators** group. • The add-on package is copied to a computer where Netwrix Auditor Server resides.
ConnectWise Manage	• By default, the add-on connects to the latest version of the ConnectWise Manage application (**v4_6_release**). • To connect to ConnectWise Manage via its REST API, you will require an **API Member** account — it is needed to log in to ConnectWise Manage. See this article for details. **NOTE:** It is recommended to use a restricted security role for the **API Member** account that will only allow access to the Service Desk module (to create and update service tickets).

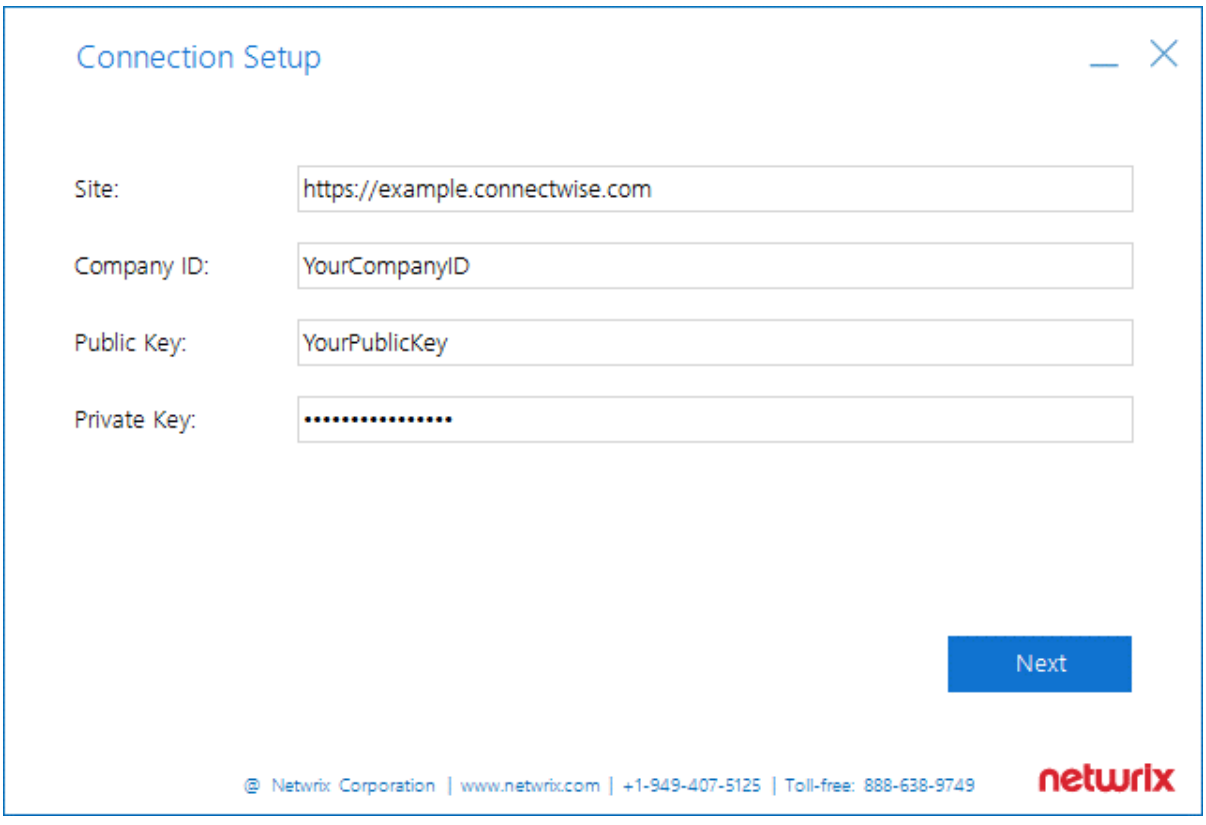
4. Deploying the Add-on

1. Prepare Netwrix Auditor for using the add-on:
   a. In Netwrix Auditor settings, enable Integration API and specify connection port. See Configure Integration API Settings.
   b. Make sure your monitoring plans set up in Netwrix Auditor are using Audit Databases to store collected data. See Audit Database.
2. Download the add-on package and copy it to the computer where Netwrix Auditor Server resides.
3. Unpack the ZIP archive to a folder of your choice; by default, it will be unpacked to **Netwrix Auditor Add-On for ConnectWise Manage** folder.
4. Run the **install.cmd** file. It will deploy and enable the **Netwrix Auditor ConnectWise Manage Integration Service**.
5. Run the **ConfigureConnection.exe** and follow the steps of the wizard to configure connection and ticketing settings for ConectWise Manage. See Configuring ConnectWise Manage Connection and Ticketing Settings.
6. (optional) To adjust the add-on operation and data flow settings, edit the **ITSMSettings.xml** file. See Appendix B. Operational Settings for more information.
7. In Netwrix Auditor, go to **Alerts**, select the required alerts, click **Edit**, and in the **Response Action** section of the alert properties specify the full path to **Netwrix.ITSM.AlertResponseAction.exe** file (the add-on component responsible for alert handling), for example, *C:\Addon\ITSM_CW\Netwrix.ITSM.AlertResponseAction.exe*.

5. Configuring ConnectWise Manage Connection and Ticketing Settings

This section describes how to configure settings of the main add-on component — Netwrix Auditor ConnectWise Manage Integration Service — required for connection to ConnectWise Manage and service ticket creation.

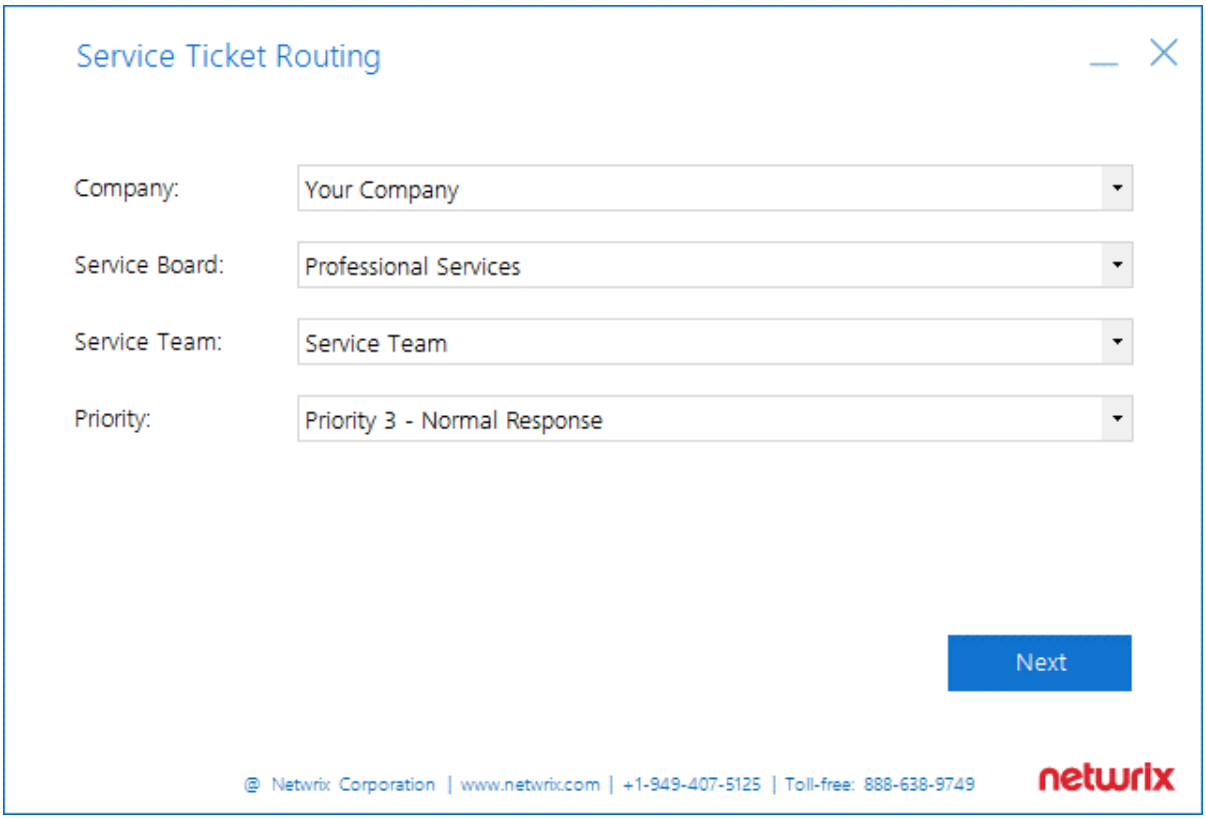
1. To connect to ConnectWise Manage REST API, the API keys will be required. To obtain them, you will need an API Member account. See this article for details.
2. Navigate to the add-on folder and run **ConfigureConnection.exe**. Follow the steps of the wizard to configure connection to ConnectWise Manage and ticketing options. At the **Connection Setup** step, specify the following:



Parameter	Description
Site	URL of ConnectWise Manage system.

Parameter	Description
Company ID	The ID of ConnectWise Manage subscriber (Managed Service Provider).
PublicKey	Public key you obtained for the API Member — it will be used to access ConnectWise REST API.
PrivateKey	Private key you obtained for the API Member — it will be used to access ConnectWise REST API.

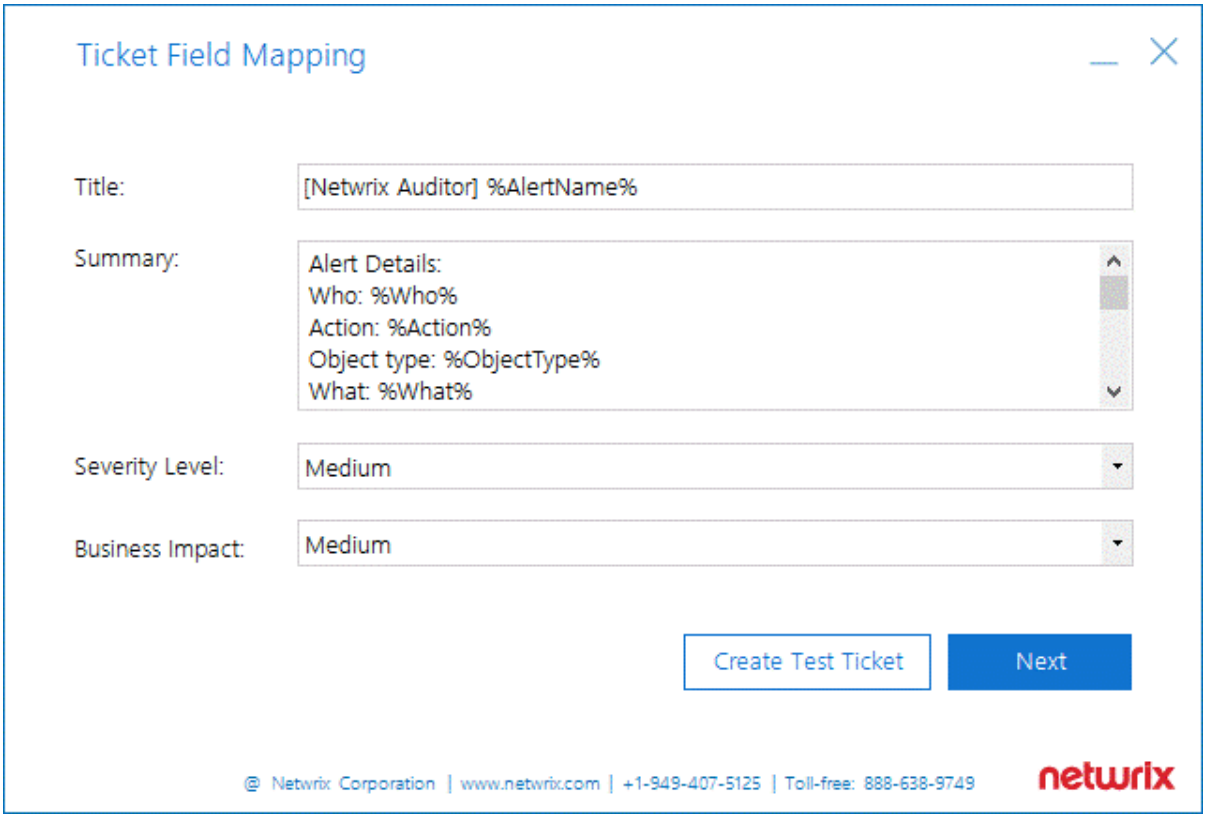
3. At the **Service Ticket Routing** step, specify the following:



Parameter	Description
Company	Organization that will be recorded as ticket originator — this can be a company or MSP's managed client.
Service Board	Service board where the tickets will be processed. NOTE: Service tickets created by the add-on will be assigned the default ticket status for the selected service board.

Parameter	Description
Service Team	Service team that will be responsible for tickets handling.
Priority	Priority for ticket handling. Default is *Priority 3 — Normal Response*.

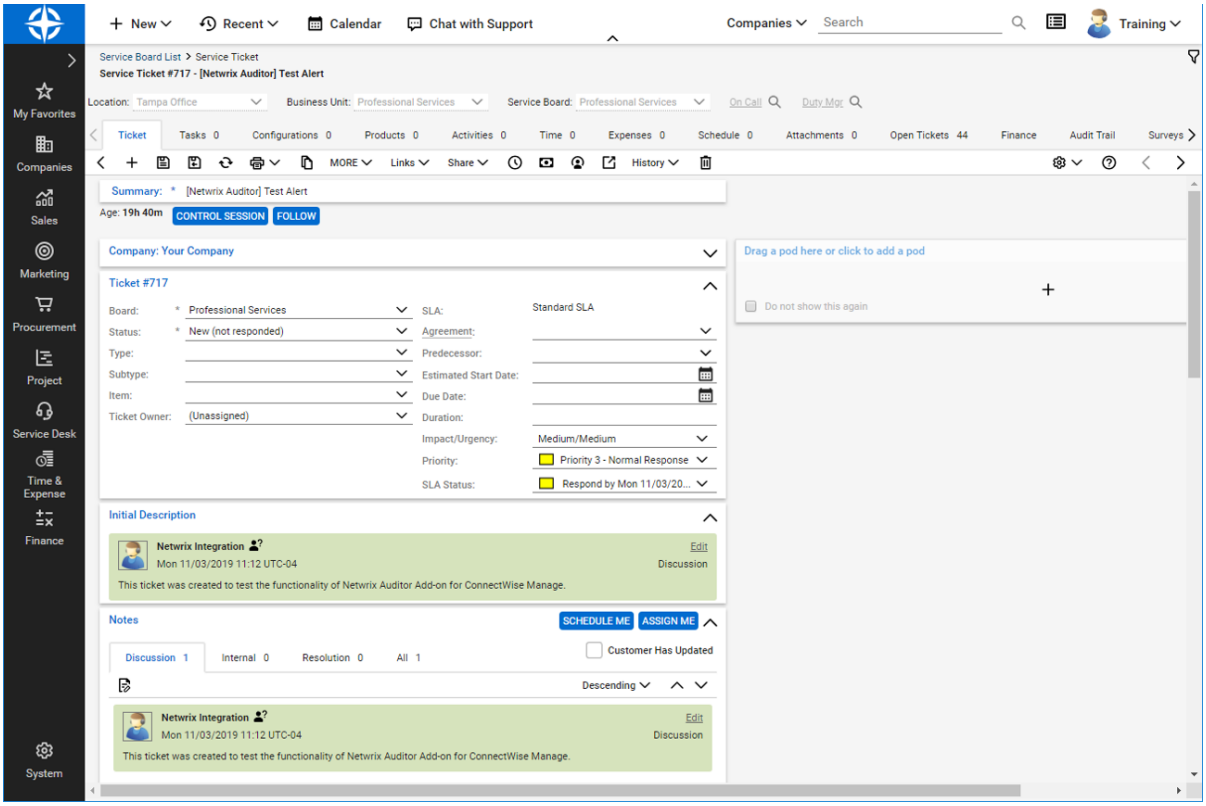
4. Next, configure how Netwrix Auditor activity record fields will be mapped with ConnectWise Manage ticket fields.



Parameter	Description
Title	Specify how the **Title** field of the service ticket will be filled in. Default: **[Netwrix Auditor] %AlertName%** That is, the **Title** field for tickets originating from Netwrix alerts will include the alert name with [**Netwrix Auditor**] prefix (e.g., *[Netwrix Auditor] Password Reset)*.
Summary	Specify how the **Summary** field of the service ticket will be filled in.

Parameter	Description
	By default, it will contain the following detailed information received from the corresponding Netwrix Auditor alert and activity record: Alert Details: Who: %Who% Action: %Action% Object type: %ObjectType% What: %What% When: %When% Where: %Where% Workstation: %Workstation% Details: %Details% Data source: %DataSource% Monitoring plan: %MonitoringPlanName% Item: %Item% Sent by Netwrix Auditor from %Computer%
Severity Level	Specify what severity level will be assigned to the service tickets. Default is **Medium**.
Business Impact	Specify what business impact level will be assigned to the service tickets. Default is **Medium**.

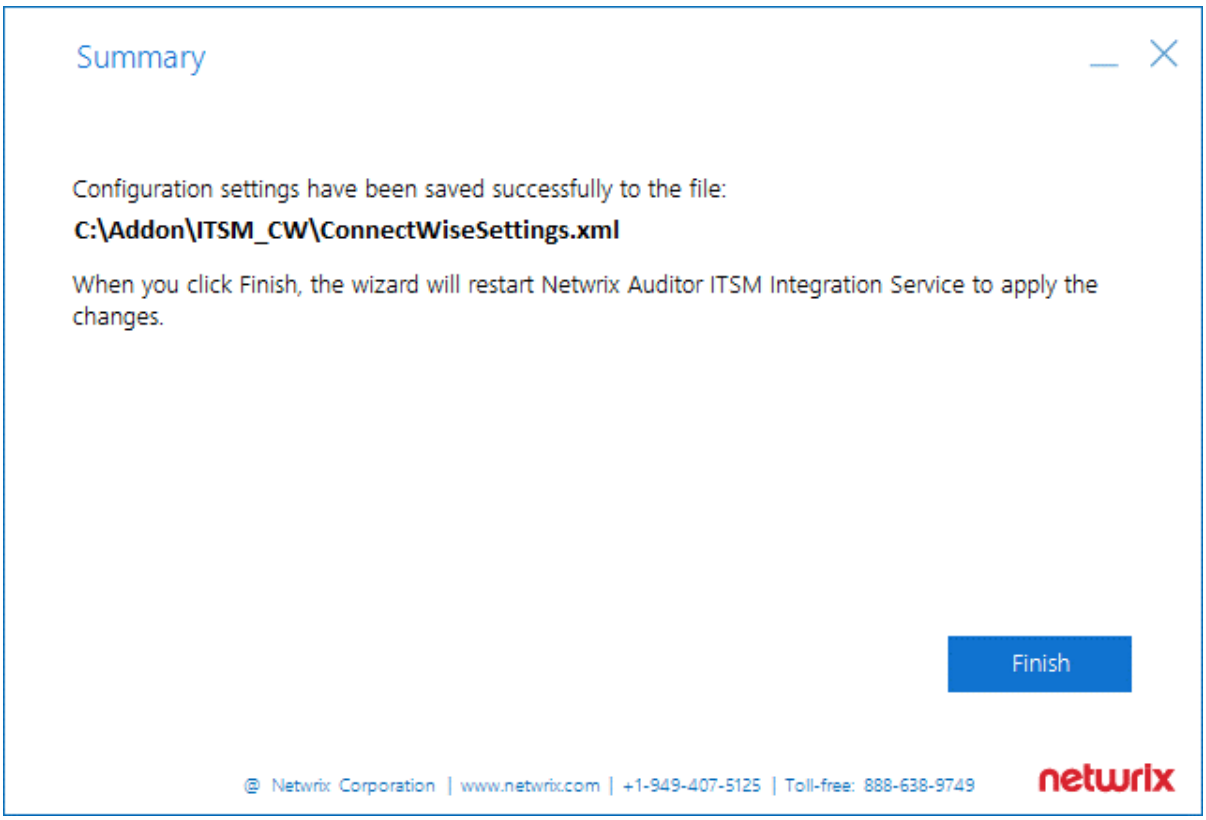
Optionally, you can click the **Create Test Ticket** button — then a test ticket will be created in ConnectWise Manage to help you verify the connection and ticketing settings you configured. Its **Summary** field will contain *[Netwrix Auditor] Test Alert*; its **Initial Description** field will contain *This ticket was created to test the functionality of Netwrix Auditor Add-on for ConnectWise Manage*.



5. Finally, at the **Summary** step, review the location of configuration file with the settings you specified: *C:\Addon\ITSM_CW\ConnectWiseSettings.xml*.

   If needed, you can edit the configuration file manually. All parameters are listed in the Appendix A. Connection and Ticketing Settings.

   Click **Finish** to restart the add-on service so that the changes can take effect.



5.1. Transferring Configuration

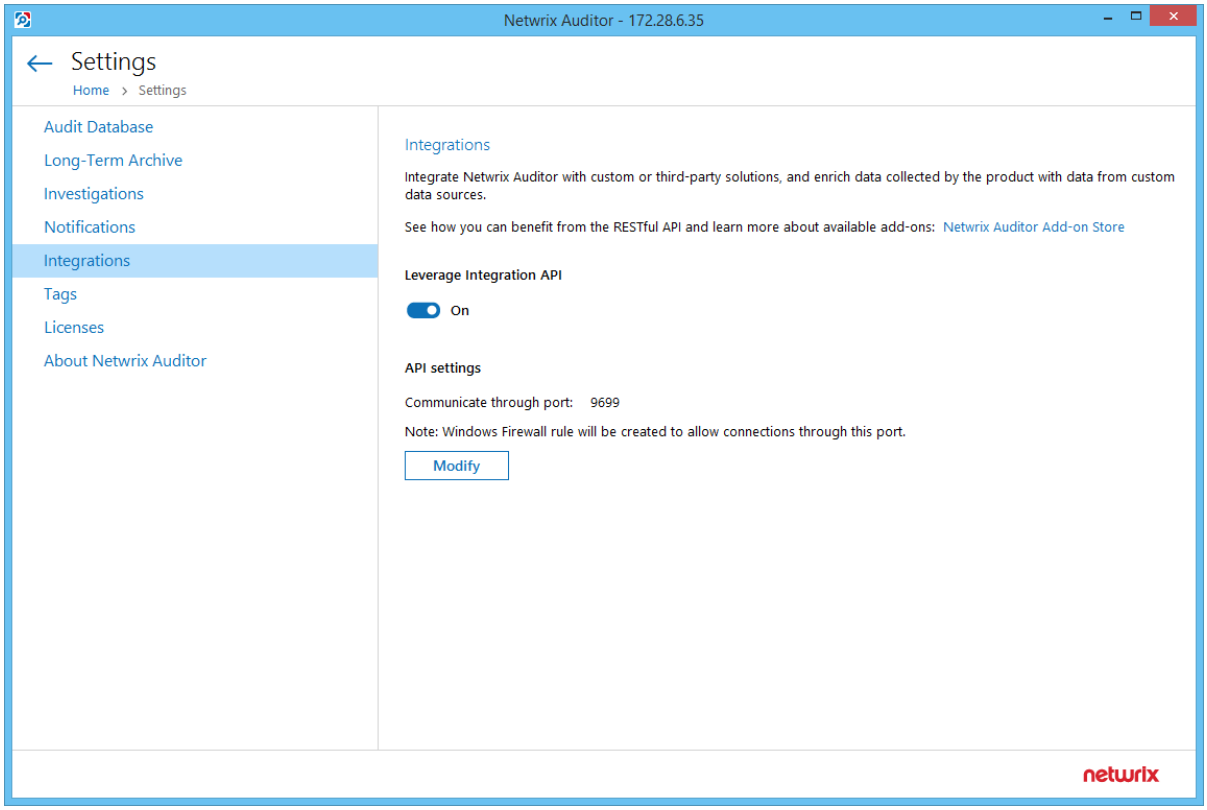
If necessary, you can use configuration file created with this wizard as a template for multiple managed clients. For that:

1. Open the file path provided at the **Summary** step of the wizard.
2. Locate the **ConnectWiseSettings.xml** file and copy it to the add-on folder on another client's server.
3. Then run **ConfigureConnection.exe** on that server to launch the configuration wizard and specify the necessary settings — for example, provide the managed client company name at the **Service Ticket Routing** step, and so on.

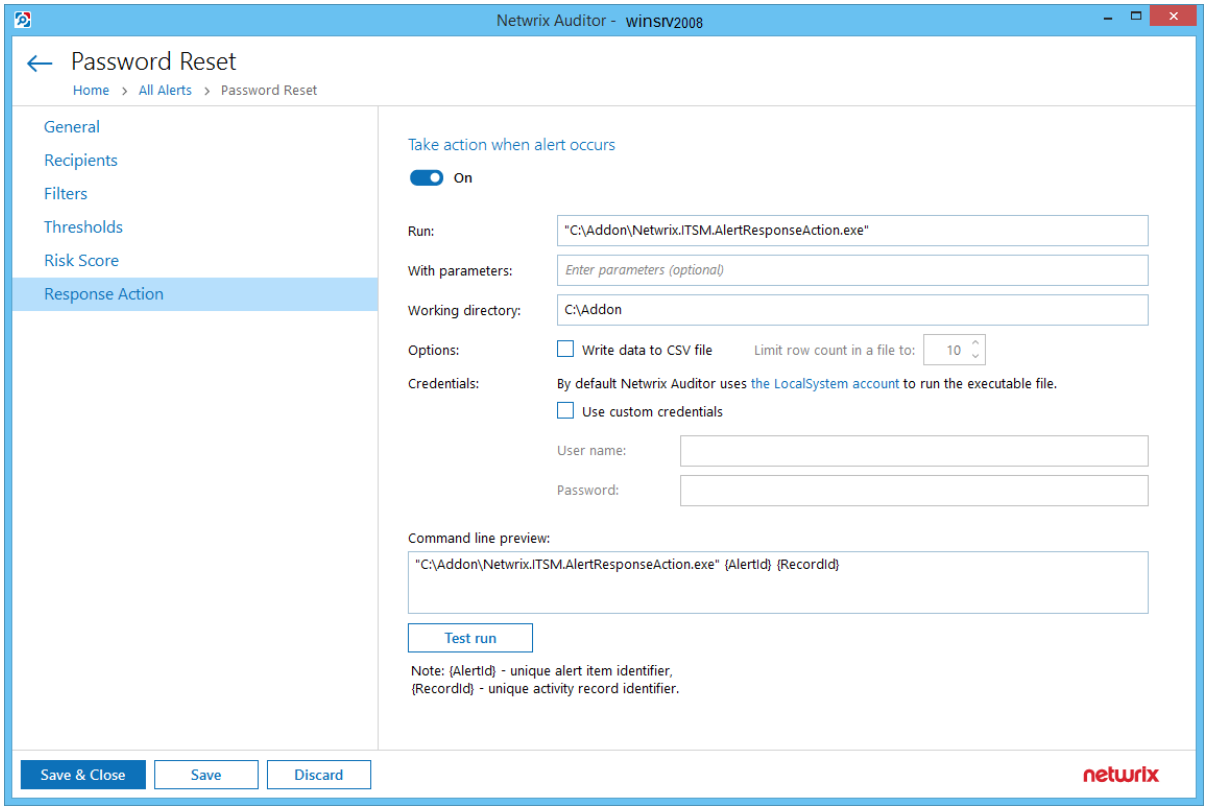
6. Usage Example

Consider a situation when a password is reset for a user, computer, or inetOrgPerson account.

After deploying and configuring the add-on, as described in this guide, the MSP staff member enabled Netwrix Auditor integration feature:

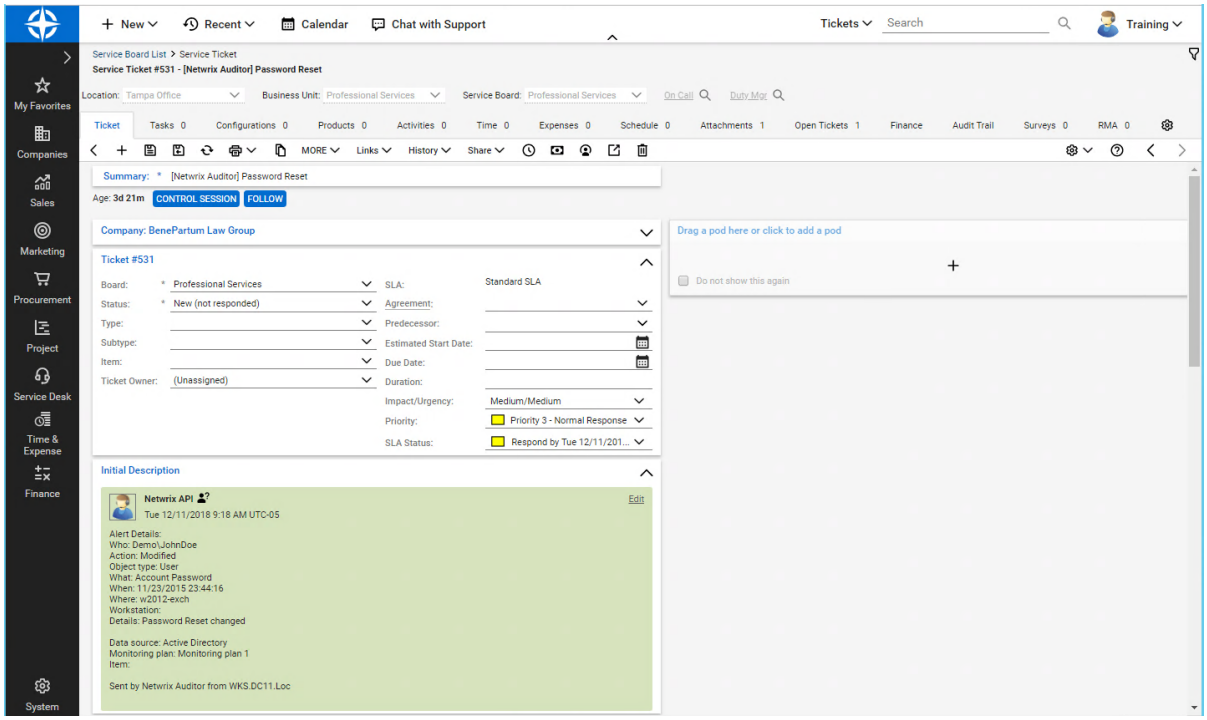


Also, she enabled the 'Password Reset' alert from the Netwrix Auditor predefined set of alerts and specified the add-on launch as response action.



Then a new ticket is automatically created shortly after any account password is reset.

All necessary details about the case are automatically entered into the ConnectWise ticket (*Initial Description* field), including the name of the workstation, the name of the account in question, and the time when the event occurred:



7. Appendix A. Connection and Ticketing Settings

It is recommended that you use configuration wizard to specify connection and ticketing settings. However, you can adjust them manually, using the information provided in this section.

7.1. Settings for ConnectWise Ticket Creation

Specify how data arriving from Netwrix Auditor should be used to fill in ConnectWise ticket fields. For that, review `<TicketParameters>`section of the **ConnectWiseSettings.xml** file. The parameters inside this section correspond to ConnectWise ticket fields and use the same naming (e.g., priority, urgency).

Each `<TicketParameter>` includes the `<Name></Name>` and `<Value></Value>` pair that defines a ConnectWise ticket field and a value that will be assigned to it. For most parameters, default values are provided. Add more ticket parameters or update values if necessary.

<Name>	<Value>	Description
Summary	[Netwrix Auditor] %AlertName%	Instructs the system to fill in the **Summary** ticket field with the Netwrix Auditor alert name (e.g., *[Netwrix Auditor] Password Reset)*.
InitialDescription	Alert Details: Who: %Who% Action: %Action% Object type: %ObjectType% What: %What% When: %When% Where: %Where% Workstation: %Workstation% Details: %Details% Data source: %DataSource% Monitoring plan: %MonitoringPlanName% Item: %Item%	Instructs the system to fill in the **InitialDescription** ticket field with the Netwrix Auditor activity record data. To read more about activity records, refer to this documentation section. You may need to fill in the internal description intended for use by MSP only (this description will not be visible to managed clients), you can do the following: 1. Run the configuration wizard (or modify *ConnectWiseSettings.xml*) to specify the settings you need. 2. Then open *ConnectWiseSettings.xml* for edit. 3. Locate the **InitialDescription** parameter and change the **Name** attribute to *initialInternalAnalysis*.

<Name>	<Value>	Description
	Sent by Netwrix Auditor from %Computer%	
Impact/Urgency	Medium	Instructs the system to set ticket **Impact/Urgency** to *Medium*.

7.2. Parameters for Handling Related Tickets

Review the `<CorrelationTicketFormat>` section. It shows what information about related tickets will be included in your current ticket. Update the template if necessary.

CorrelationTicketFormat	Description
Previous incident for the same alert type: Id: %id%	The service will automatically substitute parameters from this section with values from a related ticket.

7.3. Parameters for Reopening Tickets

Review the `<ReopenTicketOptions>` section. It defines the tickets the add-on can reopen automatically.

Name	Description
ClosedTicketStates TicketState	Lists ticket statuses. Only tickets with this status can be reopened. By default, resolved, closed, and canceled tickets can be reopened. To specify a new status, provide its ID in the <TicketState> tag (e.g., 8 for canceled).
NewState	Defines a ticket status once it is reopened. By default, new. To specify another status, provide its ID in the <NewState> tag (e.g., 1 for new).

When finished, save your changes to configuration file.

NOTE: Remember to restart the add-on service every time you update any of configuration files.

7.4. Review Other Parameters

Review the `<TicketParameterRefs>` section. It shows information related to ConnectWise Manage objects. You can update these parameters with your own values if necessary; however, it is recommended that you contact Netwrix before modifying this section.

Example:

```
<TicketParameterRefs>
```

```
<TicketParameterRef>
	<Name>company</Name>
	<TicketParameters>
		<TicketParameter>
			<Name>id</Name>
			<!--My.Sample.Company-->
			<Value>42</Value>
		</TicketParameter>
	</TicketParameters>
</TicketParameterRef>
<TicketParameterRef>
	<Name>board</Name>
	<TicketParameters>
		<TicketParameter>
			<Name>id</Name>
			<!--Professional Services-->
			<Value>1</Value>
		</TicketParameter>
	</TicketParameters>
</TicketParameterRef>
<TicketParameterRef>
	<Name>priority</Name>
	<TicketParameters>
		<TicketParameter>
			<Name>id</Name>
			<!--Priority 3 - Normal Response-->
			<Value>4</Value>
		</TicketParameter>
	</TicketParameters>
</TicketParameterRef>
<TicketParameterRef>
	<Name>team</Name>
	<TicketParameters>
		<TicketParameter>
			<Name>id</Name>
			<!--Service Team-->
			<Value>25</Value>
		</TicketParameter>
	</TicketParameters>
```

- enter ID of the company-ticket originator

- enter ID of the service board for the tickets

- enter ID of the service team responsible for resolution

```
        </TicketParameterRef>
    </TicketParameterRefs>
```

8. Appendix B. Operational Settings

This section describes how to configure settings of the main add-on component — Netwrix Auditor ConnectWise Manage Integration Service — required for its operation, including connection to Netwrix Auditor Server, activity records processing, queuing and forwarding, ticket creation, and so on.

1. Navigate to add-on folder and select **ITSMSettings.xml**.
2. Define operational parameters such as Netwrix Auditor connection settings, the number of tickets the service can create per hour, ability to reopen closed tickets, etc. For most parameters, default values are provided. You can adjust them depending on your execution scenario and security policies. Use the following format: `<parameter>value</parameter>`.

Parameter	Default value	Description
NetwrixAuditorHost	https://localhost:9699	The add-on runs on the computer where Netwrix Auditor Server resides and uses the default Integration API port (TCP port 9699). To specify a non-default port, provide a new port number (e.g., *https://localhost:8788*). **NOTE:** The add-on must always run locally, on the computer where Netwrix Auditor Server resides.
NetwrixAuditorUserName	—	Unless specified, the **Netwrix Auditor ConnectWise Manage Integration Service** runs under the **LocalSystem** account. If you want this service to use another account to connect to Netwrix Auditor Server, specify the account name in the *DOMAIN\username* format in this parameter value. **NOTE:** The user account for

Parameter	Default value	Description
		running the service and connecting to Netwrix Auditor Server must be granted the **Global administrator** role in Netwrix Auditor or be a member of the **Netwrix Auditor Administrators** group. It must also have sufficient permissions to create files on the local computer.
NetwrixAuditorPassword	—	Provide a password for the account. Unless an account is specified, the service runs under the **LocalSystem** account and does not require a password.
TicketFloodLimit	10	Specify the maximum number of standalone tickets the service can create during **TicketFloodInterval**. If a ticket flood limit is reached, the service writes all new alerts into a single ticket.
TicketFloodInterval	3600	Specify the time period, in seconds. During this time period, the service can create as many tickets as specified in **TicketFloodLimit**. The default value is 3600 seconds, i.e., 1 hour.
ConsolidationInterval	900	Specify the time period, in seconds. During this time period, the service does not process similar alerts as they happen but consolidates them before updating open tickets. The default values is 900 seconds, i.e., 15 minutes. This option works in combination

Parameter	Default value	Description
		with **UpdateTicketOnRepetitiveAlerts** and is helpful if you want to reduce the number of ticket updates on ConnectWise Manage side. I.e., this option defines the maximum delay for processing alerts and updating existing tickets. Tickets for new alert types are created immediately. For example, a new alert is triggered—the service opens a new ticket. The alert keeps firing 20 times more within 10 minutes. Instead of updating the ticket every time, the service consolidates alerts for 15 minutes, and then updates a ticket just once with all collected data.
CheckAlertQueueInterval	5	Internal parameter. Check and process the alert queue every N seconds; in seconds.
UpdateTicketOnRepetitiveAlerts	true	Instead of creating a new ticket, update an existing active ticket if a similar alert occurs within **UpdateInterval**. To open a new ticket for every alert, set the parameter to *"false"*.
ReopenTicketOnRepetitiveAlerts	true	Instead of creating a new ticket, reopen an existing ticket that is in a closed state (be default, closed, canceled, and resolved) if a similar alert occurs within **UpdateInterval**. This option works only when **UpdateTicketOnRepetitiveAlerts** is set to *"true"*. **NOTE:** If you want to reopen

Parameter	Default value	Description
		closed tickets, you must be granted the right to perform **Write** operations on inactive tickets.
UpdateInterval	86400	Specify the time period, in seconds. If a similar alert occurs in less than N seconds, it is treated as a part of an existing ticket. The default value is 86400 seconds, i.e., 24 hours. If an alerts is triggered after the **UpdateInterval** is over, a new ticket is created.
EnableTicketCorrelation	true	Review history and complement new tickets with information about similar tickets created previously. This information is written to the **Description** field. This option is helpful if you want to see if there is any correlation between past tickets (from the last month, by default) and a current ticket.
CorrelationInterval	2592000	Specify the time period, in seconds. During this time period, the service treats similar tickets as related and complements a new ticket with data from a previous ticket. The default value is 2592000 seconds, i.e., 1 month. Information on alerts that are older than 1 month is removed from internal service storage.
ProcessActivityRecord QueueInterval	5	Internal parameter. Process activity record queue every N seconds; in seconds.
DisplayOnlyFirstActivityRecord	true	Add only the first activity record in the work notes, activity records

Parameter	Default value	Description
		that update this ticket will be added as attachments to this ticket. If false, all activity records will be displayed in the ticket work notes.
ActivityRecordRequestsRetention		
RequestLimit	5000	Internal parameter. The maximum number of activity record requests the service can store in its internal memory. Once the limit is reached, the service clears activity record requests starting with older ones.
RequestLimitInterval	604800	Internal parameter. The service can store the activity record requests not older than N seconds; in seconds. Older activity record requests are cleared.
ActivityRecordWebRequests		
RequestLimit	200	Internal parameter. The maximum number of activity records the service can retrieve in a single request.
RequestTimeout	180	Internal parameter. By default, 3 minutes. Defines the connection timeout.
TicketRequestsRetention		
RequestLimit	300000	Internal parameter. The maximum number of ticket requests the service can store in its internal memory. Once the limit is reached, the service clears ticket requests starting with older ones.
RequestLimitInterval	604800	Internal parameter. The service can store the ticket requests not older than N seconds; in seconds. Older tickets requests are cleared.

NOTE: Restart the service every time you update **ITSMSettings.xml** configuration file.