Quest® InTrust 11.6.0

Technical Insight

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

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site (https://www.quest.com) for regional and international office information.

Patents

Quest Software 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 https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. All other trademarks and registered trademarks are property of their respective owners.

Legend

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

IMPORTANT, **NOTE**, **TIP**, **MOBILE**, or **VIDEO**: An information icon indicates supporting information.

InTrust Technical Insight
Updated - November 2023
Version - 11.6.0

Contents

InTrust Technical Overview **5**

InTrust Services **6**

Quest InTrust Server Service 6

Quest InTrust Real-Time Monitoring Service 6

Quest InTrust Agent Service 7

Quest InTrust Agent Installer Service 7

Remote Agents **8**

Remote Agent Installation 8

Manual Installation of Remote Agents 8

Semi-Automatic Installation of Remote Agents 9

Automatic Installation of Remote Agents 10

Connection Between Remote Agents 10

Tracking Remote Agent State 11

Remote Agent Uninstallation 11

Semi-Automatic Uninstallation of Remote Agents 12

Automatic Uninstallation of Remote Agents 12

Sites and Site Enumeration **13**

InTrust Sites and Site Objects 13

Site Enumeration Process 13

Site Enumeration Scripts 15

Licensing **17**

Licensing Errors 17

Real-Time Monitoring **18**

Agent-Side Rule Matching 18

Server-Side Rule Matching 18

Real-Time Event Collection **20**

Implicit Configuration 20

What Happens on the Agent Side 20

What Happens on the Server Side 21

Technical Imperfections 21

Workflow **22**

Tasks 22

Workflow Communication Process ... 22
Sample Scenario ... 22
Workflow Communication During Task Processing ... 23

Gathering ... 24
Agent-Based Gathering ... 24
Agent-Side Audit Log Backup (Cache) ... 25
Agent-Side Audit Log Backup Concepts ... 25
How It Works ... 26
Agent Data Location ... 26
Agent Log Structure ... 26
Agent Log Retention and Cleanup ... 27
Agent Log Controls and Settings ... 28
Agentless Gathering ... 29

Import ... 30

Consolidation ... 31

Repository Indexing ... 32
Indexing Configuration ... 32

Reporting ... 34
Reporting Jobs ... 34
Reporting Process ... 34

Cleanup Jobs ... 35

Notification, Scheduled Task and Application Jobs ... 36

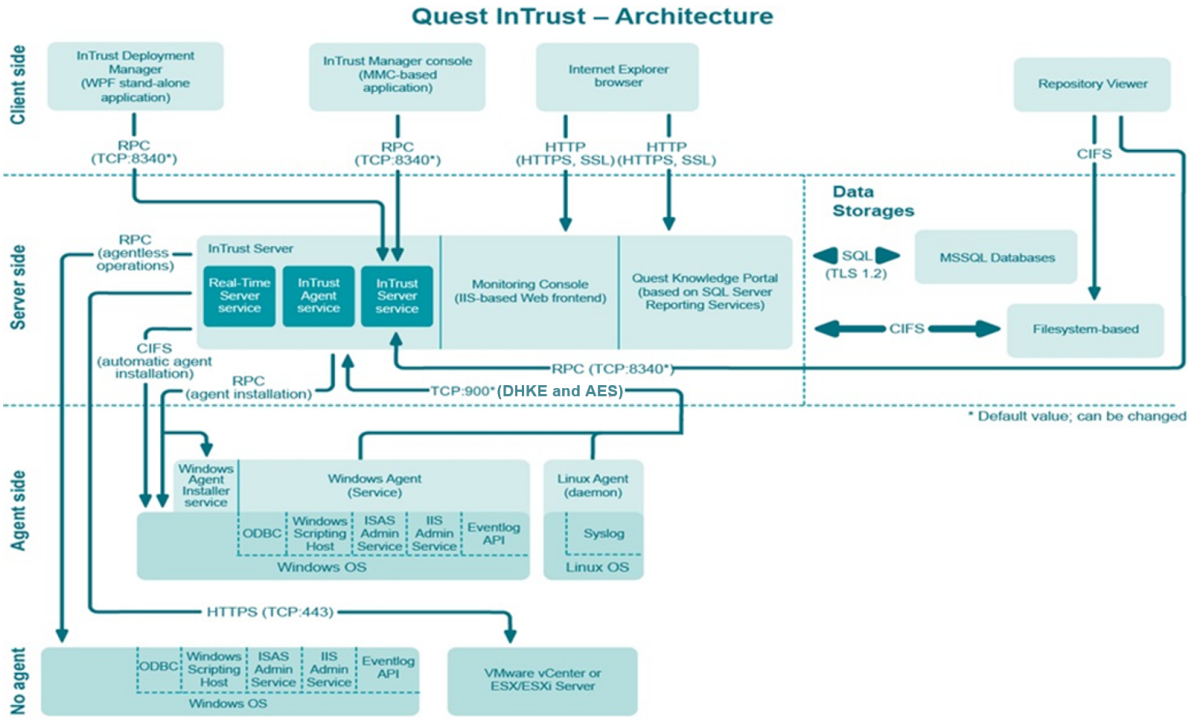
InTrust Server Failover and Rollback ... 37
Server Failure Detection ... 37
Failover ... 37
Rollback ... 38

About us ... 39
Contacting Quest ... 39
Technical support resources ... 39

InTrust Technical Overview

This document is intended for IT specialists who are involved in InTrust deployment, configuration, and maintenance. It provides a technical insight of the product components operation and processes flow.

The main components, services, and communication facilities of InTrust are shown in the figure below.



InTrust Services

The InTrust framework comprises four main services:

- Quest InTrust Server Service
- Quest InTrust Real-Time Monitoring Service
- Quest InTrust Agent Service
- Quest InTrust Agent Installer Service

Quest InTrust Server Service

This service runs in the **adcrpcs.exe** executable and performs the following operations:

- Maintenance of InTrust sites for real-time monitoring and gathering processes.
- InTrust agent deployment, checking (it checks if the agent is installed; detects the agent version and upgrades it, if necessary) and recovery
- Start of InTrust tasks (launched either manually or on schedule)
- Support for report-driven data import functionality
- Provisioning of access to the configuration database: all other components read or write data to that database, using the InTrust Server Service

Quest InTrust Real-Time Monitoring Service

This service runs in the **itrt_svc.exe** executable and performs the following operations:

- Checks configuration changes to real-time monitoring rules/polices
- Configures agents (supports for configuration of monitoring-related items only)
- Receives real-time monitoring alerts/events from remote agents
- Saves real-time monitoring alerts/events in the Alerts database
- Processes server-side rules, performing rule matching on the server side)
- Executes server-side response actions (emailing, scheduled task launch, etc.)
- Controls agent-side log backup (cache)
- Checks the license

Quest InTrust Agent Service

This service runs in the **adcscm.nt_intel.exe** executable (for Windows platform; executable name varies depending on the platform). Actually, the same service runs on the InTrust Servers and the clients (that is, they share the code base).

The agent that runs on the clients is known as the remote agent; it performs the following operations:

1. Establishing a communication channel to the InTrust Server(s)
2. Sending a keep-alive packet to the InTrust Server(s) every 2 minutes (configurable), so that agent status can be tracked
3. Real-time monitoring:
    - Processing of agent-side rules using real-time monitoring data providers
    - Execution of agent-side response actions
    - Pre-filtering server-side rules and forwarding matched events to the InTrust Real-Time Monitoring Service
4. Maintaining the event cache (agent-side log backup) if it is enabled
5. Collecting events (either from the log or the cache) and sending the data to the InTrust server
6. Deployment of distributable modules (they appear on the corresponding tab of monitoring rules' and data sources' Properties) on target computers

The agent that runs on the InTrust Servers is known as the local agent. This agent performs the same operations as remote agent; besides, it provides communication between remote agents and all other InTrust services/components

> **NOTE:** Remote agents only communicate with the local agent, they do not communicate directly with other InTrust services/components.

Quest InTrust Agent Installer Service

This service only runs on the Windows platform in the **adcscm.nt_intel.exe** executable. It provides for agent installation, checking and removal.

The InTrust Server Service communicates with the Agent Installer service using remote Service Control Manager (SCM) commands (with special parameters to control Agent Installer behavior). It installs agents by:

- Unpacking the agent files (copied by the InTrust Server Service)
- Installing the agent to the **\\<*agent_host_name*>\Admin$\ADCAgent** directory (configurable)

Remote Agents

- Remote Agents Installation
- Connection Between Remote Agents
- Tracking Remote Agent State
- Remote Agents Uninstallation

Remote Agent Installation

Remote agents can be installed on the client computers (those from which event data should be gathered); installation procedure can be performed manually, semi-automatically or automatically.

When performing automatic or semi-automatic installation, **Agent_InstallFolderInShare** configuration parameter is used together with the **Agent_InstallShare** parameter to set the agent installation folder. **Agent_InstallFolderInShare** specifies the local folder where the agent is installed, relative to the shared folder set by **Agent_InstallShare** parameter. Changes to this parameter affect subsequently installed agents. The default location is the **ADCAgent** folder in the **ADMIN$** share. To change the values:

1. Under InTrust Manager **Configuration** node, select the InTrust Server the agents will report to.
2. From its shortcut menu, select **Properties**.
3. Go to the **Parameters** tab, select the parameter and click **Edit**.

Manual Installation of Remote Agents

Remote agent installation should be performed manually in the following cases:

- The InTrust server and the remote computers are connected by unreliable and slow links. Agent installation fails when the packet drop rate is higher than 5%.
- The remote computers are behind a firewall (and the Server Windows service is unavailable for InTrust server).
- The remote computers are Linux-based or Unix-based

Agent installation is typically performed using TCP/IP and only requires one port (900) to be open for connection from the remote computer to the InTrust Server. This is a configurable setting, as described in the Connection Between Remote Agents topic.

To install an agent, an administrator should follow the instructions provided in Installing Agents Manually.

A list of commands for agent deployment is provided below:

Command	Description
-install	Installs the agent.
-add	Establishes a connection between the agent and the specified InTrust server

Command	Description
	using the specified port. After an agent is added, it will be named as follows: • If installed on Windows computer: NetBios name FQDN for each IP address • If installed on Unix computer: FQDN for each IP address. The list of names that can be used to access the agent will also include the name given to site-processing agent (available in the site properties). The display name will be the one with the highest priority. The priorities are assigned as follows (the highest is 10 - for the name of site-processing agent): ADC_AGENTNAME_RANGE_REGISTERED 10 ADC_AGENTNAME_RANGE_SERVER 20 ADC_AGENTNAME_RANGE_SERVER_DNS 23 ADC_AGENTNAME_RANGE_SERVER_NETBIOS 26 ADC_AGENTNAME_RANGE_DNS 30 ADC_AGENTNAME_RANGE_NETBIOS 40
-register	Registers (adds) an alias for the computer on the specified InTrust server (adds the alias to the list of names that can be used to access the agent & if the display name has not explicitly been set, then it updates the display name used by InTrust Manager to be the alias).
-reauth	Forces the agent to re-authenticate with the specified InTrust server.
-upgrade	Upgrades the agent to a newer version.
-uninstall_for_upgrade	Removes the agent but does not delete the agent configuration

Semi-Automatic Installation of Remote Agents

The installation procedure is initiated by InTrust administrator and performed automatically on Windows Platform only. It requires TCP port 900 (configurable) to be open on InTrust server for agent-server communication, ICMP echo request to be allowed between InTrust server and agent, and the Server Windows service on agent to be available for InTrust server (see Microsoft KB article 832017 Service overview and network port requirements for the Windows Server system for details).

An administrator initiates the agent installation process by right-clicking on a site and selecting the **Install Agents** menu item.

The InTrust Server Service completes the installation by:

1. Copying the agent package to the target computer.
2. Installing the Quest InTrust Agent Installer service, using the Service Control Manager.
3. Instructing the Quest InTrust Agent Installer service (using the Service Control Manager) to install and start the agent.

Automatic Installation of Remote Agents

Automatic agent installation is performed in the same way as semi-automatic installation, except that the process is initiated not by an administrator but by any of the following:

- A gathering job
- A change to real-time monitoring configuration (due to which the monitoring of new sites starts)
- A new computer being discovered during the periodic site enumeration process (for a site involved in monitoring policy)

Connection Between Remote Agents

Connections between remote agents and local agents are ALWAYS established by the remote agent; local agents never establish the connection

The default port for a connection between remote agents and local agent on InTrust Server is TCP port 900. This setting can be modified during InTrust Server setup, or by taking the following steps:

1. Modify Agent_Port advanced parameter for the required InTrust Server.
2. Restart InTrust services on the InTrust Server computer for changes to take effect (gathering will not operate properly until the restart).
3. Agents' states will change to "Not Responding", and then to "Lost"; after that, agents will be recovered and re-configured by InTrust Server Service.
   Communication will be then restored automatically for the agents that meet both of the following requirements:
   a. An agent belongs to site with automatic agent deployment allowed (i.e., "Prohibit automatic agent deployment on site computers" option cleared)
   b. The accounts specified for agent configuration and operation have sufficient rights

   For other agents, the **-add** command should be used to restore communication (see Installing Agents Manually for details)
4. Agents start using the new port.

When the connection has been established, both the remote and local agents use the connection to send/receive data (there is no out-of-band communication).

The connection remains open until it is closed by:

- Stopping the remote agent (either manually or during a reboot)
- Stopping the local agent
- A network problem
- Using the -unregister command, or the Uninstall menu item in InTrust Manager
- Agent uninstallation

Tracking Remote Agent State

Remote agent state is tracked using “keep-alive” packets. Each remote agent sends a ‘keep alive’ packet to InTrust Server every 2 minutes (this setting can be edited with **Comm_AgentSendLifeMarkInterval** organization parameter). The InTrust Server Service records the latest 'keep-alive' arrival time in the **LastTimeMark** column of the **ADCInstalledAgent** table in the configuration database, and then a corresponding state is assigned considering the difference between CurrentTime and LastTimeMark (i.e., current time and last ping received).

Currenttime-lastTimeMark	State	Action	Org Parameter
< 4 mins	Running	None	
+ 4 mins	Not responding	If the InTrust server is not using the agent (gathering etc.), then the InTrust server waits to see if the agent starts sending keep-alives. If a gathering job is run or the Install Agents on Site option is selected, InTrust tries to recover the agent.	Comm_ AgentSendLifeMarkInterval
+ 10 mins	Lost	InTrust tries to recover the agent	Comm_AgentLostAfter
n/a	Awaiting Authorization		An Administrator must reconnect (add) the agent to the InTrust server

To view current agent’s state, in InTrust Manager, select **Configuration | InTrust Servers**, expand your InTrust Server’s node, and click **Agents**.

Remote Agent Uninstallation

Remote agents can be uninstalled manually, semi-automatically or automatically. For more details, see the following topics:

- Semi-Automatic Uninstallation of Remote Agents
- Automatic Uninstallation of Remote Agents

To uninstall an agent that was installed from an ***.msi** package, an administrator can use the Add/Remove Programs facility. However, if the agent was installed through the command prompt, an administrator should execute the **adcscm.nt_intel.exe –uninstall** command on the remote agent computer (executable name varies depending on platform). This command does the following:

1. Removes the connection between the remote computer and the InTrust Server.
2. Stops the remote agent and uninstalls it from the remote agent computer

Command	Description
-uninstall	Uninstalls the agent
-remove	Removes a connection between the agent and the specified InTrust server on the specified port
-unregister	Unregisters (removes) an alias for the computer on the specified InTrust server (removes the alias from the list of names that can be used to access the agent and updates the display name used by InTrust Manager)
-shutdown	Stops the agent
-uninstall_for_upgrade	Removes the agent, but does not delete the agent configuration

Semi-Automatic Uninstallation of Remote Agents

To uninstall the remote agent, an administrator has to right-click on the agent in InTrust Manager and select Uninstall. The uninstall behavior varies depending on the remote agent state and the number of InTrust servers it has a connection to:

Agent State	Number of InTrust Servers Connected	Action
Any state except Running	Any	Connection between the agent and the InTrust server is removed.
Running	1	Connection between the agent and the InTrust server is removed; the remote agent is stopped and then uninstalled.
Running	>1	Connection between the agent and the InTrust server is removed (the parent InTrust server of the agent object selected in the InTrust Manager console)

Automatic Uninstallation of Remote Agents

If a remote agent cannot connect to an InTrust server for more than 7 days (set by the **AgentUninstallTimeout** organization parameter), the remote agent uninstalls itself. If an agent communicates to several InTrust servers, this timeout should occur for each server the agent works with.

Sites and Site Enumeration

- InTrust Sites and Site Objects
- Site Enumeration Process

InTrust Sites and Site Objects

Sites are collections of objects used as a target for InTrust operations (Gathering and Real-Time Monitoring). There are 2 types of sites:

- Microsoft Windows Network
- Unix Network

Site objects are resolved to computer names or IP Addresses. The sites are accessed through the MSNNSiteProvider (Windows sites) and SolarisSiteProvider (Unix sites) components

For Microsoft Windows Network sites, you can do the following:

- Specify the security credentials which are used to access computers in the site
- Prevent the automatic installation of agents
- Specify the enumeration options (Active Directory, Browser service)

A list of objects you can include in a site of the certain type is provided below.

Objects	Microsoft Windows Site	Unix Site
Whole Network	Yes	No
Domains	Yes	No
Computers	Yes	Yes
IP address range	Yes	Yes
Computer list	Yes	Yes
Organizational unit	Yes	No
Active Directory site	Yes	No
All DCs in the domain	Yes	No
All DCs in AD site	Yes	No

Site Enumeration Process

Site enumeration can be initiated in the following ways:

- Automatically—by a gathering job or scheduled enumeration for real-time monitoring (configured using site property)
- Manually—by clicking the **Refresh** button in the Details pane of the site properties (on the **Enumeration** tab)

This process is executed using 2 threads running in parallel: one thread resolves computer names/IP addresses, and the other collects information from the resolved objects.

The enumeration process continues without waiting for the whole result set to be returned, so enumeration results are processed as soon as they are available, in the order they are returned.

This process uses various internal lists when determining the actual site membership (the individual computers and IP addresses). There is not one definitive list—real-time monitoring and each gathering job maintain their own lists.

Enumeration process determines the actual site membership in the following way:

1. Reads the site objects from the configuration database
2. Processes the object list, expanding site objects as necessary (see the table below for details)
3. Checks if an agent is already installed on the computer:
   - If it is installed, enumeration process queries the agent for the default property set (OS, computer type)
   - If it is not installed, enumeration process queries the client directly for the default property set
4. Verifies the site filters and, if necessary, retrieves additional information from the client computer, using the agent (if it is already installed) or directly querying the client (if the agent is not installed)
5. If a computer matches the site filter, then it is added to the final site membership list.
6. If a site filter cannot be checked (client is not available/accessible), it is still added to the final site membership list.

Site Object	Enumeration
Whole Network	First the domain list is obtained using the method selected for enumeration (Computer Browser or Active Directory). Active Directory domains are enumerated by the **DsEnumerateDomainTrusts** API function; the return value is a list of domains that trust the current domain. Next, each domain is enumerated individually.
Domains	Enumerated according to the domain enumeration method. Enumeration through Active Directory uses the following LDAP query: **"(objectCategory=computer)"**. **Important:** Accounts are not filtered by age.
Computers	No enumeration required.
IP address range	Only expanded to individual IP addresses.
Computer list	Computer names/IP addresses are read from the file.
Organizational Unit	Enumerated using the following LDAP query: **"(objectCategory=computer)"** The OU is the root of the search.
Active Directory Site	1. Get a list of forest subnets that belong to the specified site. 2. Enumerate the domains in the forest.

Site Object	Enumeration
	3. Enumerate each domain as a domain object. 4. For each resulting computer, get the IP address and check that it belongs to one of the subnets obtained on step 1. If the subnet list is empty, all computers in the forest are discovered.
All domain controllers in domain	Enumerated according to the domain enumeration method. If enumeration through Active Directory is used, first all computers in the domain are discovered using the following LDAP query: **"(objectCategory=computer)"** Next, computers with an empty **Server-Reference-BL** attribute are filtered out. Domain controllers are discovered among the remaining computers.
All domain controllers in Active Directory site	Always enumerated through Active Directory. The list of domain controllers in obtained from the configuration namespace by running the following LDAP query against the site: **"(objectClass=server)"**

The final site membership list will contain the original object name (the name used when the object was added to the site) and the name that InTrust can use to connect to the client with. Gathering and real-time monitoring currently respond differently to the site filter check:

- A gathering job will gather audit data, even if InTrust cannot check the site filters.
- Real-time monitoring will not generate an alert on a computer if InTrust cannot check the site filters.

If a site is being enumerated for real-time monitoring, the Quest InTrust Real-Time Monitoring service does the following:

1. Checks whether the computer was already in the site (against the previous version of its final site membership list). If it was not, then it pushes the real-time monitoring configuration to the agent.
2. Checks whether the computer was removed from the site and if it was, then it removes the real-time monitoring configuration from the agent.

Site Enumeration Scripts

If you want to specify your own algorithm for the enumeration of objects in the site, you can use the **Enumeration Script** option, which prompts you for a script that will perform the enumeration. This option is available:

- During site creation: on the Site Objects step
- For an existing site: from the context menu, or in the site properties on the **Objects** tab

Selecting Enumeration Script prompts you for the script you want to use. The scripts are located in the **Quest InTrust Manager | Configuration | Advanced | Scripts** container node.

InTrust comes with the example "Enumeration script: LDAP query" script for this purpose. For your sites, you can use this script, copies of it, or your own scripts.

The "Enumeration script: LDAP query" script has the following parameters, which you can customize without modifying the script itself:

Parameter	Meaning
Attribute Name	Name of the attribute that will be used as the object name in the list of site objects.
Bind String	ADSI bind string; for example, “GC:” means that the entire AD forest will be searched, “LDAP:” specifies the current domain.
Filter	LDAP filter, such as **"(objectCategory=serviceConnectionPoint)"**
Need Deep Search	What to do if the search in the entire forest finds objects whose names (specified by the Attribute Name parameter) cannot be read: • 0—do nothing; the matching object is not included in the site • 1—try searching in individual domains and reading the attributes again This parameter is considered only if the Bind String begins with “GC:”.
Search Scope	Search scope in LDAP terms, with the following values: • 0—base • 1—one level • 2—subtree

Licensing

The InTrust license is stored in the configuration database. License verification is performed by the InTrust Real-Time Monitoring Service on InTrust servers every three hours. If a license check fails due to the configuration database being offline, attempts are repeated until a connection is established.

Licensing Errors

Licensing errors occur in the following situations:

- No license has been installed.
- The license has expired.

If either of these situations occurs:

- InTrust operation is suspended.
- Event 8207 is written to the InTrust Server event log.
- Until a valid license is installed, the same event is written every three hours, and license checks are performed every three minutes.

If you attempt to gather audit data without a license or with an expired license, corresponding error messages will appear in the sessions. No events are written to the InTrust Server log for such attempts.

Real-Time Monitoring

A real-time monitoring rule can be configured so that matching and response actions are executed either on the agent side or on the server side.

Rule matching and response action execution on remote agent mean that:

- Agent matches rule (body only)
- Agent executes response action (under security context of the agent)

Rule matching on remote agent and response action execution on server mean that:

- Agent matches rule (body only)
- Server executes response action (under security context of the RTM service)

Rule matching and response action execution on server mean that:

- Agent matched rule (pre-filter only)
- Server matches rule (both pre-filter and body)
- Server executes response action (under security context of the RTM service)

If rule matching is configured to be done by server and response action execution—on remote agent, then rule is rejected, and an event is logged in the InTrust log.

Agent-Side Rule Matching

If a rule is configured to be matched on the agent side, the real-time monitoring process goes as follows:

1. An event is matched.
2. A remote agent adds the alert to its outbound buffer (a 10MB file buffer).
3. The remote agent removes the alert from the outbound buffer and sends it to the local agent.
4. A local (server-side) agent places the alert in a per-agent buffer (a 1MB file buffer):
    - When the per-agent buffer is full, the local agent logs a buffer overflow“” event and stops accepting alerts from that remote agent until 50% of the buffer is free.
    - When the buffer drops to 30% full, a “buffer overflow resolved” event is logged.
    - The remote agent continues trying to re-send the alert.
5. The Quest InTrust Real-Time Monitoring service reads the alert from the per-agent buffer and places it in another buffer (“4k” buffer), which can store about 4000 alerts and resides in the Quest InTrust Real-Time Monitoring service memory.
6. The alert is written to the alert database.

Server-Side Rule Matching

If a rule is configured to be matched on the server side, the real-time monitoring process goes as follows:

1. An event occurs on the agent side.
2. The remote agent matches the rule's pre-filter.
3. The remote agent forwards the event to the local agent (using the process described for agent-side rule processing).
4. The Real-Time Monitoring rule matches the complete rule (both pre-filter and body).
5. The RTM Service stores the alert in the "4k" buffer and executes the response action. If the "4k" buffer is full:
    - The response action is not executed.
    - The alert does not get stored in the alert database.

Real-Time Event Collection

Real-time event collection is based on real-time monitoring technology but uses InTrust repositories as target data stores. During real-time collection, an agent essentially performs rule matching on incoming audit data. However, instead of processing this data to see if an alert should be raised or an action should be taken, the agent puts the matching data in a short-term local store and sends it to the server at short intervals. The hidden rules for this purpose are set up automatically, and they use a catch-all matching condition so that no data is skipped.

Implicit Configuration

From the user's perspective, setting up real-time event collection involves creating collections of Windows computers in InTrust Deployment Manager and associating some data sources and a repository with each collection. Internally, InTrust translates this type of configuration into a different one. This is needed to adapt it to the real-time monitoring system, of which this type of gathering is an extension. As with regular real-time monitoring, the set of configuration objects for real-time event collection includes the following:

- Site
  The site contains the same computers as the corresponding collection in InTrust Deployment Manager.
- Real-time monitoring rule
  The rule matches everything.
- Data source
  The rule is associated with the data source that is specified for the collection in InTrust Deployment Manager. The data source defines the log to capture.
- Real-time monitoring policy
  The policy applies the rule to the site, so that agents in the site know when to activate real-time audit data transfer.

An agent has one special-purpose rule for each repository – data source pair that real-time collection is set up for. All of these objects are handled automatically, and none of them is visible in InTrust Manager. No alerting or response actions are configured for rules that perform real-time collection.

What Happens on the Agent Side

The agent uses a single send queue for both real-time monitoring results and real-time collection results, even though they use different data formats. The real-time collection component of the agent periodically takes items from the queue and submits them to the server, as follows:

- On server editions of Windows, real-time collection data is submitted every minute.
- On workstation editions of Windows, real-time collection data is submitted every seven minutes.
- To prevent slowdowns, the data can be submitted sooner if it exceeds internally defined limits on size and number of events.

In file system terms, the queue is located in the **<*agent_data_path*>\tasks\<*GUID*>\tpvc** folder on the agent computer. The path contains the GUID of the real-time portion of the agent. On the agent computer, there are as many such GUIDs as there are InTrust servers that the agent is talking to.

The total size of this queue is controlled by the **ITRT_CommAgentMaxSizeCommunicationQueue** organization parameter. This size is about 1GB by default. The maximum size is 1.7GB, but setting it this high is not recommended.

The queue works as a circular buffer, meaning that when this size is reached, the agent begins overwriting the oldest data with the most recent data.

NOTE: The queue is not a caching mechanism, and scheduled task-based gathering does not use it in any way. It is unrelated to the log backup described in Agent-Side Audit Log Backup (Cache).

When the agent submits data to the server, it says what kind of rule produced that data: a regular real-time monitoring rule or a real-time collection rule.

What Happens on the Server Side

The server looks at the type of rule that produced the incoming data. If it is a real-time collection rule, the data is treated as repository files, and it goes into the repository indicated by the rule. Otherwise, it is treated as alert records and goes to the alert database.

To prevent denial of service, the server has a limit on how much data it will accept from a single agent. The limit is 10MB of server-side queued data per agent. If the queue of unprocessed data grows this large for an agent, the server suspends reception of data from that agent. It resumes the reception only after it has processed (meaning, written to the alert database or repositories) enough of the accumulated data to make the queue half-empty.

Technical Imperfections

The current implementation has a significant drawback in that both real-time monitoring and real-time collection running on the same server stop working if any of the repositories used for real-time collection suddenly become unavailable. Intuitively, real-time monitoring should not be affected by real-time collection problems, but in reality, it is.

NOTE: The reverse problem does not exist: if an alert database becomes unavailable to an InTrust server, it doesn't affect the real-time collection activity on that server.

Such disruptions resolve themselves only when the repository becomes available again or if you delete it from InTrust configuration. If the repository stays offline for a long time, gathered data can be lost, because it will be overwritten by more recent data on the agent side.

The recommended way to address this is to configure at least two InTrust servers: one for real-time monitoring, and one for real-time collection. Separating these duties among individual InTrust servers removes the dependency that can lead to the problem.

Workflow

InTrust workflow is based on tasks that can be started in schedule or on demand. Each task comprises one or more jobs.

For more details, see the related topics:

- Tasks
- Workflow Communication Process

Tasks

InTrust tasks run in the **WorkflowManager.exe** executable. When a task is started, the InTrust Server service creates an instance of **WorkflowManager.exe** (the **WorkflowManager** component).

Tasks are started by all InTrust servers that are responsible for running one or more jobs within the task. The **Task Scheduler** component is responsible for maintaining a list of scheduled task; it gets notified of changes to scheduled tasks no later than every minute. Usually, tasks run under the InTrust Server service account.

Jobs within a task are processed by InTrust Servers as follows:

- InTrust servers start the jobs that are assigned to them; they are also responsible for synchronizing jobs within the same task when the jobs are running on different InTrust servers
- If job start or job synchronization fails, error is written in the InTrust log and to the corresponding session.
- Communication between InTrust Servers is handled by the InTrust Server Service; port TCP 8340 (this is the default value; the port number can be configured during setup) must be open on these servers.

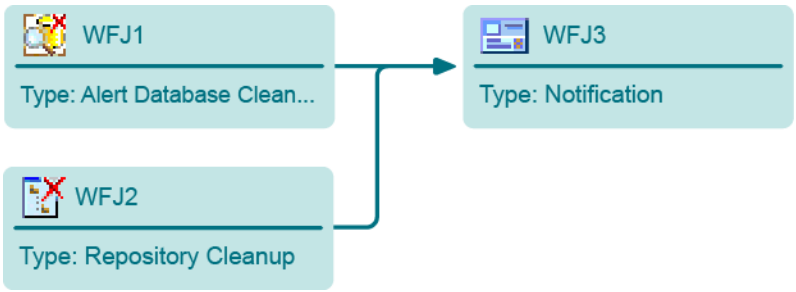
Workflow Communication Process

- Sample Scenario
- Workflow Communication During Task Processing

Sample Scenario

In this example, InTrust workflow is configured in the following way:

- There are 3 InTrust servers (IT01, IT02 & IT03) in the InTrust Organization
- One Workflow Task (WFT1) was configured, containing 3 Workflow Jobs (WFJ1, WFJ2, WFJ3)
- Workflow jobs (WFJ1, WFJ2, and WFJ3) are run by IT01, IT02, and IT03 InTrust Servers, respectively
- Jobs WFJ1 and WFJ2 have no job dependencies
- Job WFJ3 depends on both WFJ1 and WFJ2



Workflow Communication During Task Processing

Workflow communication during task processing is described below:

1. All 3 InTrust servers create their own instance of the WorkflowManager (WM) component and start the task.
2. The WM components on InTrust Servers IT01 and IT02 start their respective jobs (WFJ1and WFJ2) as these jobs do not have any dependencies
3. The WM component on InTrust Server IT03 communicates through port 8340 with InTrust Servers IT01 and IT02 to check the status of the WFJ1 andWFJ2 jobs, as WFJ3 depends on both of these jobs
    - If IT03 cannot connect to all InTrust servers running predecessor jobs (IT01 & IT02 in this example), then it starts WFJ3.
    - If IT03 can connect, but the predecessor job is not running, then it waits for 10 minutes
    - If IT03 can connect, but the predecessor job is not running for more than 10 minutes, it logs an error and then starts WFJ3
    - If IT03 can connect and the predecessor job is running, it waits for 10 minutes or until a notification is received (within the 10 minute period), then it again polls the InTrust servers (IT01 & IT02) for the status of the predecessor jobs.
4. When a predecessor job is complete (WFJ1 for example), the WM on the InTrust server (IT01) notifies the other InTrust Servers that are responsible for jobs within the task (IT02 & IT03), that the job has been completed.

Gathering

Event data gathering runs in the **GatheringEngine.exe** executable; a separate instance of this executable is created for each gathering job.

If required, gathering pre-empts agent installation. In case of using agent-based gathering, the agent configuration is updated (via local agent), regardless of whether the configuration has changed. GatheringEngine receives data from the remote agent via local (server-side) agent.

Agentless gathering collects data directly from the remote computer. This type of gathering is only supported for logs from Windows based computers and custom text logs from non-Windows based computers using SAMBA shares.

The collected data is stored in the repository and/or audit database; data uniqueness is determined by combination of Date/Time, Record Number, Computer and Log (audit database only).

The GatheringEngine works, as follows:

1. The GatheringEngine (GE) component reads the configuration from the configuration database, then it initiates the site enumeration process
2. Next, it queries the InTrust Server Service for agent availability (if gathering with agent):
    - If the remote agent is not installed, then it initiates agent installation
    - Agents are only installed if the **Prohibit automatic agent deployment on site computers** option is not selected
    - If the agent is already installed, and gathering job is configured to use agents, then the option takes no effect, and agent-based gathering takes place.

Agent-Based Gathering

In the case of agent-based gathering, the process runs in the following way:

1. The GatheringEngine sends the configuration to the remote agent
2. The GatheringEngine asks the remote agent for data and waits to receive it
3. The remote agent reads the data either from the event log or the client-side cache (agent-side log backup) (depending on the caching settings)
4. The remote agent applies filters and writes temporary files in the ‘Repository File’ format:
    - At least 1 ‘Repository File’ for gathering to a repository
    - At least 1 ‘Repository File’ for gathering to an audit database
    - The number of ‘Repository Files’ is dependent on the size of the data (Max 8MB uncompressed - configurable)

5. The temporary files are communicated to GatheringEngine:
    - When a buffer size reaches 8MB (configurable) of event data, the 'Repository File' is ready to delivery
    - The remote agent waits for acknowledgement from GatheringEngine (i.e., GatheringEngine is ready to receive new data)
    - The remote agent sends data to GatheringEngine. Note that new data will not be written to a 'Repository File' if the previous file is not sent; also, the agent must be notified by GatheringEngine about data successfully put into a storage
6. The GatheringEngine writes the temporary ‘Repository File’ into a temporary folder on the InTrust server
    - If data is being gathered to a repository, the ‘Repository File’ is moved to the repository
    - If data is being gathered to an audit database, the data is extracted from the ‘Repository File’ and imported into the audit database

Agent-Side Audit Log Backup (Cache)

- Agent-Side Audit Log Backup Concepts
- How It Works

Agent-Side Audit Log Backup Concepts

Agent-side cache, referred to in the product GUI as "agent-side audit log backup", is an InTrust proprietary technology primarily aimed at protection of audit data from being lost even if an accidental or malicious log cleanup occurs on the target machine. Another benefit of this technology is that it allows InTrust to never access the same audit trail (most commonly, event log) more than once for one and the same event, even if multiple gathering jobs applied to the local agent by one or more InTrust Server(s) require the same events.

Agent-side log backup is used only with scheduled task-based event log gathering and doesn't affect real-time monitoring or real-time event collection in any way. It is also optional, so gathering can be performed without enabling it.

NOTE: Number of data sources processed by gathering jobs defines the number of accesses to native log.

The cache is implemented as a set of folders and files on the agent machine. The structure and names of files and folders follows a specific pattern. Cached audit data is stored on disk in compressed form in order to provide for keeping a large amount of events for the period of time sufficient for InTrust to gather them all according to its gathering schedule, regardless of the original audit trail retention settings. That is, even if the events in their native log are overwritten (or the log is purged) by the time when InTrust comes for them, those events still can be gathered from the agent cache.

An agent with caching enabled receives events being logged to an audit trail specified for scheduled gathering in real time, as they are written to the native log.

NOTE: InTrust agent does not interfere with the native event logging workflow but listens to the events in parallel with the original event logging. The native process of writing events to the original audit trail is not interrupted when agent-side caching is stopped or something unexpected happens to it (except Syslog processing that can be restarted when caching is turned on/off).

Since an agent may be engaged in gathering events from the same log with multiple gathering policies, it caches all the events in the log regardless of filters imposed by settings of a specific policy or InTrust data source the policy is based on. The elementary entity to be specified for caching is an audit trail (log) underlying the InTrust data source.

It is a specific gathering job where any filtering is done, exactly as in the case of gathering non-cached events from a native log.

Cached events remain stored on disk and available for further gathering when the agent machine is rebooted or the agent is temporarily stopped. When the agent is started again, it proceeds with caching events generated since the agent start. For Windows NT data source type, however, the following takes place:

The data source of Windows NT type keeps the position of the last monitored event. If that event is not found in the log at the agent start, then the log data (for the specified period) will be re-read from the earliest records.

Events from a particular audit trail (log) remain stored in the agent cache until one of the following happens:

- The cache folder size limit is exceeded
- The retention period defined for an InTrust data source based on that log expires
- A gathering job with the runs for events from this specific log and clears events after gathering

The effect each of these conditions has on events stored in agent cache is different and will be described in more detail further in this document.

How It Works

Agent Data Location

Agent cache files are located in the **proxy_manager** subfolder of InTrust agent data folder. The location of the agent data folder is specified by the value of the adc_data_path variable defined in the **agent.ini** file.

The value of **adc_data_path** is defined in **agent.ini** through another variable, **%ADC_INSTALLPATH%**. The value of this variable is read from the local registry at **HKEY_LOCAL_MACHINE\SOFTWARE\Aelita\ADC\ADCCoreInstallPath** (for agents on Windows-based computers).

By default, the data folder is an immediate subfolder of the agent installation folder named **data**.

You can also use InTrust Manager to change the agent-side cache location for one or more agents selected (see the **General** tab of the agent's properties dialog box).

Agent Log Structure

An agent writes data to be cached into a file. When it reaches the **ITRT_RTCacheFileLimitSize** value (see *Agent Log Controls and Settings* below), the agent stops adding data to it, compresses the file and starts writing to a new one. Another thing the agent does at this point is create a new entry in the index file **cfg.1** mapping the name (GUID) of the new archived file to the timestamps of the first and the last event stored in it. Use of this index file allows the agent to quickly retrieve events when a gathering time comes, with no necessity to search through all the compressed files in attempt to locate events matching the scope of the particular gathering job. This index also makes it easier for the agent to quickly clean up its cache since outdated event data is deleted on the per-file basis, and files to be removed from the cache are spotted based on the map from **cfg.1**.

As stated above, the most granular entity to be specified for caching is an audit trail (log) underlying the InTrust data source. That is, in terms of InTrust configuration objects, a data source. Therefore folders named after data sources look like the most obvious way to organize the cache files in, since each data source can be uniquely identified by its GUID organization-wide. But an agent may work for multiple InTrust Servers that not necessarily belong to one and the same InTrust organization, while a data source GUID is recognized within one organization only. For this reason, data source subfolders are located on the second hierarchy level, making way for folders named by InTrust organization GUIDs. An individual index files is maintained for each data source and is located in corresponding folder.

As a result, the tree of folders and files eventually follows the following pattern:
<Agent_data_folder>\proxy_manager (folder)

- Organization 1 folder GUID (folder)
    a. data source 1 GUID (folder)
        - current cache file (uncompressed)
        - cfg.1 (index file, uncompressed)
        - archived cache file 1 (compressed)
        - archived cache file 2 (compressed)
        - archived cache file 3 (compressed)
        - archived cache file 4 (compressed)
        - <...>

    b. data source 2 GUID (folder)
        - current cache file (uncompressed)
        - cfg.1 (index file, uncompressed)
        - archived cache file 1 (compressed)
        - archived cache file 2 (compressed)
        - archived cache file 3 (compressed)
        - archived cache file 4 (compressed)
        - <...>

    c. data source 3 GUID (folder)
        - <...>

    d. <...>
        - Organization 2 folder GUID (folder)

    e. data source 1 GUID (folder)
        - <...>

    f. data source 2 GUID (folder)
        - <...>

    g. <...>
        - <...>

    h. <...>

Agent Log Retention and Cleanup

When the total size of cache files in all data source subfolders of an organization folder reaches the value defined by the **ITRT_RTCacheFolderLimitSize** organization parameter, the agent deletes the compressed file that stores the oldest events from the biggest data source folder to free up space for new compressed files.

If the new compressed file still does not fit into **ITRT_RTCacheFolderLimitSize**, the agent removes one more file using the same logic, and so on.

The condition imposed by the **Agent-side audit log backup retention period (days)** data source setting is checked every 24 hours (hardcoded) starting from the agent start time. Upon the condition verification, every file, compressed or uncompressed, with no event in it younger than the retention criterion is deleted from the data source folder.

When a gathering job runs against the agent computer and clears the log after gathering, all compressed cache files are deleted from the folder(s) matching the data source(s) included into the job.

Every time an agent creates or removes a cache file, it adds or removes an entry to or from the **cfg.1** index file in the appropriate data source folder.

Every time a cache-enabled agent starts, it evaluates all the files in each data source folder against records in the **cfg.1** file stored there. Every file found in a folder with no matching entry in the local index file is considered unrelated and deleted unconditionally.

When a last task with a gathering job in it that applies to an agent is unscheduled or deleted from an InTrust organization configuration, and caching of some (but not all, within an organization) data sources is switched off on this agent, folder with files for those data sources remains in cache for 10 more days (this value is hardcoded and does not depend on the data source retention period settings). The same is true if caching is stopped for all the data sources on the agent that remains engaged in any real-time monitoring for at least one InTrust Server in the same organization the cached data sources belong to.

If agent-side caching for all data sources belonging to an organization is stopped and the agent does not do any real-time monitoring for any InTrust Server in that organization, cached data under the organization GUID will remain on disk for 1 more day + 3 hours of uninterrupted agent uptime (defaults values, controlled by **ITRT_ ZombieTimeout** and **ITRT_ZombieLastChanceTimeout** organization parameters respectively).

Agent Log Controls and Settings

The **Agent-side audit log backup retention period (days)** parameter is specified in properties of each particular InTrust data source. The default value is 2 days for data sources based on the Security event log and 8 days for other data sources.

The following settings related to cache behavior are defined as the organization parameters:

Parameter Name	Default Value	Maximum Value	Description
ITRT_ZombieTimeout	86400	2147483647	Period of time, in seconds, after stopping all the real-time monitoring and event caching activity on an agent, during which it does not delete from disk any data related to its former activity (**tasks** and **proxy_manager** subfolders of the agent data folder).
ITRT_ ZombieLastChanceTimeout	10800	2147483647	A period, in seconds, of uninterrupted uptime for an agent with the ITRT_ZombieTimeout period expired, during which the agent waits for appropriate InTrust Server to resume this agent's real-time or event caching activity before it permanently deletes the tasks and proxy_manager subfolders of the agent data folder from disk.

NOTE: Each of the organization parameters can be defined at either the organization (default) or InTrust Server or agent level.

If an agent is caching events from the same data sources for scheduled gathering job on multiple InTrust servers, and effective parameter settings differ from of those Servers to another, the most strict (the least) of the conflicting parameter values within one organization becomes effective for the whole agent cache for that organization.

In addition, the following settings are defined in **agent.ini**:

Parameter Name	Default Value	Maximum Value	Description
ITRT_ RTCacheFileLimitSize	64	1796	Maximum size, in megabytes, of a single cached data file. When a file reaches this size, it is closed and saved in a compressed form, and a new file is started.
ITRT_ RTCacheFolderLimitSize	1024	2147483647 (or -1 for 'unlimited')	Maximum total size, in megabytes, of all cached data files. When a cache reaches this size, the oldest cached data file is irrevocably deleted and a new file is started.
ITRT_RTCacheLockFiles	1	—	When set to '1', the InTrust Agent service keeps each cached data file open while it works. This makes cache files locked by the agent process and protects them from being deleted by a user or some other application. Opening each file, however, takes time and may result in a long time required for an agent to start when its local cache contains a large number of cached data files. To disable cache file locking, set the value of this parameter to '0'.

Agentless Gathering

In case of agent-less gathering, the GatheringEngine connects to the remote computer via RPC, SMB or ODBC (depending on data source type).

During gathering to a repository, the following takes place:

- GatheringEngine collects the data, applying filters as necessary, and writes the data into a “repository file” in a temporary folder on the InTrust server.
- GatheringEngine moves the “repository file” to the repository.

During gathering to an audit database, GatheringEngine collects the data and stores it directly in the audit database.

Import

Import process runs in the **GatheringEngine.exe** executable; a separate instance of **GatheringEngine.exe** is created for each import job. The process extracts data from the repository files, applying the appropriate filters, and stores the data in the audit database: **GatheringEngine.exe** opens a connection to the specified audit database, and imports the data using the **BULK INSERT** statement. Up to 4000 events are imported per transaction.

Consolidation

Consolidation runs in the **GatheringEngine.exe** executable; a separate instance of **GatheringEngine.exe** is created for each consolidation job.

Consolidation process checks that the source repository actually contains the appropriate data for the data source (s) defined in the consolidation policy, and for the timeframe specified in the consolidation job.

- • If no relevant data exists in the source repository, GatheringEngine logs a warning, and the consolidation job exits.
- • If relevant data exists in the source repository, the needed repository files are copied from the source repository to the target repository (whole files are copied).

You can consolidate audit data from a repository that is located on an InTrust server behind a firewall. To do it, first find out the repository path on the InTrust server behind the firewall and the password of the InTrust organization behind the firewall. Then take the following steps:

1. Create a new repository. For that, right-click **Configuration | Data Stores | Repositories** and select **New Repository**.
   Consider giving the new repository a name that indicates it is located behind the firewall.
2. On the Repository Location step of the New Repository Wizard, supply a UNC repository path that is also valid for the other repository behind the firewall. Complete the wizard.
   The path you specify is not verified. The repository object you created on step 1 is just a representation of the source repository. The actual repository will be found as long as the path is correct.
3. Right-click the necessary task and select **New Job**; start creating a consolidation job.
4. On the Select Repositories step, do the following:
   - Select the source repository
   - Select **Use this server to manage source repository** and specify the InTrust server that hosts the repository.
   - Specify the port over which your firewall allows communication. By default, port 900 is used.
   - Specify and confirm the password of the InTrust organization that the InTrust server behind the firewall belongs to.
   - Select the destination repository.
5. Complete the wizard.

Repository Indexing

The indexes of InTrust repositories are created and maintained by the **IndexingTool.exe** utility, which resides in the **InTrust\Server** subfolder of the InTrust installation folder on the InTrust server and on Windows computers with the InTrust agent installed. This utility is designed to be launched automatically by InTrust Server or agent, not manually. However, manual operation is also supported for situations described in Repository Indexing for Advanced Search Capabilities.

Depending on the configuration of the repository, the indexing workload can be managed in one of the following ways:

- All of the work is done by the **IndexingTool.exe** utility on the specified InTrust server.
- The index is processed by **IndexingTool.exe** utility instances on computers in the specified InTrust site.

The workload distribution is managed by the **RemoteIndexLauncher.exe** utility on the InTrust server. This utility handles both server-side and distributed indexing. The utility cannot be launched manually.

Indexing Configuration

In addition to repository properties, indexing is configured by the following organization parameters:

Organization Parameter	Meaning
IDX_ IndexAccessCheckMode	Specifies how to check index security configuration. The values are as follows: • • 0 Check only that the indexing account is a member of the computer local **AMS Readers** group on the InTrust server. • • 1 Test whether the indexing account can actually read and write to repository files. However, if the indexing account is specified explicitly in the repository properties, it is only checked that the account is a member of the computer local **AMS Readers** group on the InTrust server. The default value is 0.
IDX_IndexingThreadCount	Sets how many threads indexing will use. The values are as follows: • 0 or blank Use as many threads as there are CPU cores. • -1 Use one less threads than there are CPU cores, but no less one thread. • Any positive integer Use this many threads. Values less that -1 will cause errors. The default value is -1.

For more information, see Repository Indexing for Advanced Search Capabilities.

Reporting

- Reporting Jobs
- Reporting Process

Reporting Jobs

InTrust Reporting Jobs use Microsoft SQL Server Reporting Services for report generation. Reporting jobs run in the **Quest.Shotgun.InTrust.ReportingJob.exe** executable; a separate instance of **Quest.Shotgun.InTrust.ReportingJob.exe** is created for each job of that type. InTrust Reporting jobs allow reports to be run against any InTrust database (Configuration, Alerts, Audit) configured as a data source for reporting:

- A reporting job can use the data source specified in the report configuration.
- To access the database, a reporting job requires that the credentials are specified correctly.
- A reporting job creates a temporary data source for report generation.

Reporting Process

The reporting process in InTrust involves the following:

1. Reads the reporting job configuration from the configuration database.
2. Establishes a connection to the Report Server.
3. Instructs the Report Server to generate the reports (reports are generated sequentially) and deliver the reports (as specified in the delivery options).
4. Sends a notification (if appropriate).

If a reporting job is configured to use report-driven data import, then data required for the report(s) is retrieved from the repository and placed into the database prior to report generation. The following accounts are used during the reporting job that has data import enabled:

- Reporting job account—the one under which the reporting job is launched. Reporting job account is specified in the job properties on the **General** tab.
- Import job account—the one under which data is imported from the source repository to the audit database. This is either the account inherited from the task, or (if this is a hidden import job that InTrust creates for Knowledge Portal) the InTrust Server service account.
- Database connection account—the one under which the audit database is accessed to import data and report on it.

For more information on report-driven data import, refer to the Reporting Job topic.

Cleanup Jobs

Two executables are used for data cleanup purposes:

- **itrt_retention.exe**, which is used by alert database cleanup jobs to remove the specified alerts from the alerts database.
- **RetentionJob.exe**, which is used by audit database cleanup and repository cleanup jobs.

Repository cleanup jobs delete whole files from the repository; if a repository file spans a date/time boundary, the file is not deleted.

Audit database cleanup jobs remove data from the following tables:

- Events
- EventsStrings
- EventsData
- EventsDescriptions

The TRUNCATE statement, which requires dbo rights, is used when the delete ALL option selected.

RetentionJob.exe does not delete position data, so InTrust knows the last gathered/imported events. Position data for imported events is stored in the **GatheredEvents** table; position data for gathered events is stored in the repository, in files that contain a NULL GUID in the filename.

Notification, Scheduled Task and Application Jobs

Notification jobs, scheduled task jobs and application jobs run in the **TaskProxy.exe** executable, which does the following:

1. Reads the job configuration from the configuration database.
2. Loads the appropriate DLLs or starts the external executable which then executes the job.

InTrust Server Failover and Rollback

- Server Failure Detection
- Failover
- Rollback

Server Failure Detection

Every minute, all InTrust servers update the **LastServerPing** column of the **ADCServer** table of the configuration database with the current date/time.

An **UPDATE** trigger on the **LastServerPing** column executes a Stored Procedure which checks the date/time in the LastServerPing column against the current date/time:

- If the time difference is less than the **Unresponsive Period**, it sets the **Status** column in the **ADCServer** table to 1 (that is, server is “alive”)
- If the time difference is more than the **Unresponsive Period**, it sets the **Status** column in the **ADCServer** table to 0 (that is, server is down)

This stored procedure also closes any sessions that are open (running) on the failed InTrust server - this is to prevent jobs managed by other InTrust servers waiting for a job managed by the failed server to complete.

Every 60 seconds, a specially intended **InTrust Server Failure Detection** data source executes a script which checks the status of each InTrust server and generates an event for each InTrust server that is down (with status of 0).

The **InTrust Server is Down** real-time monitoring rule detects these event(s) and invokes the response action (once per event).

Failover

After an InTrust server failure has been detected, the failover process switches from the failed server to the standby server, as follows:

1. It checks all sites and jobs (except gathering jobs) managed by the active (failed) InTrust server and fails them over to the standby InTrust server.
2. All agents that were automatically deployed by the active (failed) InTrust server are automatically registered with the standby server.
3. All agents that were manually installed do not get automatically registered with the standby server—they need to be registered manually (prior to failover).

For details about failover configuration, see Backup Procedures for InTrust.

Rollback

After restoring the InTrust server, you can roll back this switching session (switch sites and jobs back to the server initially responsible for their processing). Rollback is a manual process, performed as follows:

1. Start the Rollback Wizard by selecting **Failover | Roll Back** from the restored server's shortcut menu.
2. Select the session to roll back.
3. Commit the changes after finishing the wizard.

Rollback history is stored in the configuration database. Once the active (failed) InTrust server has been restored and all job/sites failed back, the only communication between remote agents and the standby server is the keep-alive packet.

After the failure has been resolved, you might have to manually change the indexing servers in the properties of your repositories. These settings are not rolled back by the wizard.

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit www.quest.com.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.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
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product