Service Desk

User Guide

Version R93

English

February 9, 2017

Copyright Agreement

The purchase and use of all Software and Services is subject to the Agreement as defined in Kaseya's "Click-Accept" EULATOS as updated from time to time by Kaseya at http://www.kaseya.com/legal.aspx. If Customer does not agree with the Agreement, please do not install, use or purchase any Software and Services from Kaseya as continued use of the Software or Services indicates Customer's acceptance of the Agreement."

Contents

Service Desk Overview

Welcome to Service Desk!

The **Service Desk** module manages an organization's response to a support issue. A support issue may be a question, a problem, a request for service, or a suggestion for a future enhancement.

Before you begin your implementation of **Service Desk** we recommend you review all the overview topics listed in this section, to orient yourself to the **Service Desk** structure and design.

Additional Resources

Service Desk demos, training, documentation, and Kaseya Professional Services consulting are all available to help quickstart your implementation of **Service Desk**.

Service Desk Module Minimum Requirements

Kaseya Server

- The Service Desk R93 module requires VSA R93.

> **Note:** See general **System Requirements** *(http://help.kaseya.com/WebHelp/EN/VSA/9030000/reqs/index.asp#home.htm).*

Planning for Service Desk

Here are some issues to consider when planning the implementation of **Service Desk**:

- Who are the users and what are their roles?
- What resource pools should you define? Resource pools are groups of technicians allocated to different areas of specialization.
- What is the life cycle, or workflow, of a ticket? Ticket workflows are defined using stages.
- Should tasks be used to track assignments required to close a ticket?
- What ticket properties are required? What values are allowed for each property? What are their default values?
- What ticket property access permissions should be assigned, by role and by user?
- What ticket list view preferences should be defined? Which columns should display? Conditional formatting enables you to set color shading and highlighting based on ticket creation, modification, or due date. What predefined filters should be defined?
- What are your service goals: response time, time to resolution, etc.?
- Will you provide more than one service level?
- What are your ticket sources: manual creation, incoming email, monitoring, etc.
- What are your addresses for incoming email?
- What is your criteria for detecting duplicate monitoring alerts?

- What automation would be beneficial for different events and conditions: ticket created, ticket changed, stage entry/exit, escalation based on elapsed time in current stage and ticket age?
- What level of end-user communication is desired?
- What templates would be useful for email notifications and inserting notes into the ticket history log?
- What are your knowledge base requirements?
- What reporting and metrics are desired?
- Are you tracking time spent on tickets for billing purposes?
- At what point should you **activate** *(page 51)* the creation of **Service Desk** tickets from alarms instead of **Ticketing** tickets.

Automation in Service Desk

Service Desk provides automation for the processing of tickets as follows:

- **Automated Processing Using Service Procedures** *(page 2)*
- **Automated Processing Using Policies** *(page 4)*
- **Automated Processing Using Inbound Emails and Alerts** *(page 5)*

Additional automation is available using one or more **integration options** *(page 9)*.

Automated Processing Using Service Procedures

Automatic processing of tickets in **Service Desk** can be triggered by the creation of the ticket record itself or the transition of the ticket to a "stage" in its lifecycle. Automatic processing can be triggered each time a ticket progresses to *any* stage. This includes the *first* stage, when a ticket is created. Automatic processing can also be triggered if a ticket remains in a stage beyond some time limit that you define. Stages are defined in Service Desk > **Desk Definition** *(page 32)*. **Service Desk** procedures and can also trigger the running of an agent procedure on a managed machine using the **scheduleProcedure()** *(page 94)* step.

See **Procedures Definition** *(page 64)* for more information about procedures.

How Stages Work

Stages are designed to model the flow of work a business performs to process support issues. They are defined by desk definition. One stage is defined as a **Begin** stage and multiple stages can be defined as an **End** stage. There are usually several **Middle** stages.

Stages are *sequenced* by selecting one or more "to" stages. The sequence of stages is user-defined. It depends on the policies defined by the service organization and the judgments of individual users using the system. A sequence of stages is graphically represented by the **View** *(page 39)* tab within a desk definition.

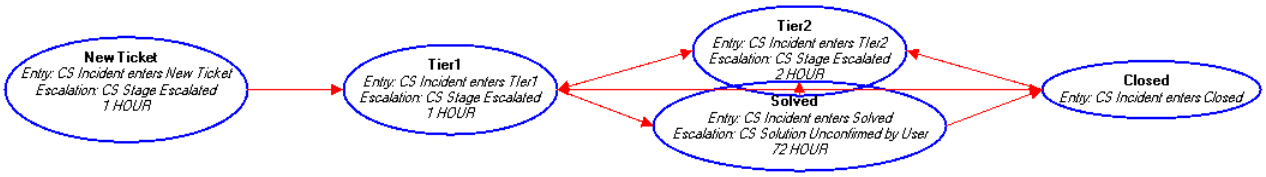
An **event procedure** runs as soon as a ticket transitions from one stage to the next. A **scheduled procedure**, such as an escalation procedure, is triggered to run based on how long a ticket remains in a stage. For example, escalation procedures typically run when a ticket has remained too long in a stage, instead of being resolved and moved to another stage.

When a desk definition is created, stage sequences are populated by the *template* used to create the desk definition. Once a desk definition is created based on a template, these default stages can be modified to suit your business requirements.

Example - The Stages of Customer_SD_Automation

The description of each stage below assumes the VSA users discussed are *not* **Service Desk** *administrators*, and so have their views of tickets *limited by their user role and scope.*

Note: See the **Desk Templates** *(page 42)* topic for instructions on how to create this and other predefined service desks automatically.



New Ticket - A newly created ticket in `Customer_SD_Automation` starts out in the `New Ticket` stage.

- A `CS Incident enters New Ticket` stage entry procedure assigns the ticket to a `Tier1Support` pool of service ticket technicians. VSA users see any tickets assigned to them, plus any tickets assigned to pools they belong to.
- A VSA message and an email message is created for these same users. The submitter of the ticket is also sent an acknowledgment email that a ticket has been created.
- An escalation procedure is linked to this stage. If the ticket has not been moved out of the `CS Incident enters New Ticket` stage within 1 hour, the escalation procedure runs. The escalation procedure notifies the assignee of the ticket and the `Tier1Support` pool of users. The intent of the escalation procedure is to ensure that all newly created tickets are reviewed immediately.

Two other factors can automate the behavior of tickets at any stage:

- **Automatic Ticket Assignment from a Pool** - The first time the ticket is opened by any member of the `Tier1Pool`, that member is assigned the ticket. Other members of the `Tier1Pool` no longer see the ticket in their view of tickets.
- **Ticket is Changed** - This not a stage, but a **Ticket Change** *(page 76)* procedure. The same `CS Incident is Changed` procedure is run each time the ticket is saved, even if a minor change to the ticket does not include switching the stage. This ticket change procedure tests for the following:
  - If the ticket is new, the ticket is immediately moved to the `Tier1` stage.
  - If the ticket has an assignee, the assignee also becomes the "owner" of the ticket. An owner of a ticket typically continues to be responsible for the ticket, even if the ticket is assigned to a series of different assignees over the course of its life cycle. Ticket views can be filtered by owner, enabling owners to manage tickets that may have been forgotten about by their current assignees.
  - If the ticket has been solved, the ticket is closed.
  - If a ticket was already closed, but a customer sends an email referencing the ticket, the ticket is reopened.

Tier1 - The assignee of the ticket is now tasked with working with the ticket in the `Tier1` stage.

- The assignee is sent an email message.
- An escalation procedure is also linked to this stage.
- A ticket can be manually assigned to either the `Tier2` stage or the `Solved` stage, as indicated by the two red arrows coming out of the `Tier1` bubble in the graphic above.

Tier2 - Moving a ticket to the `Tier2` stage typically means the ticket requires a person with more experience to resolve it, perhaps providing a special area of expertise.

- A `CS Incident Enters Tier2` stage entry procedure automatically assigns the ticket to a `Tier2Pool` of users, so that `Tier1` personnel don't have to guess who to assign it to.
- A VSA message and an email message is created for these same users. The submitter of the ticket is also sent an acknowledgment email that the ticket has been assigned to the `Tier2Pool` for further investigation.
- An escalation procedure is also linked to this stage.

Solved - Resolved tickets are sent to this stage to wait for acknowledgment back from the submitter after being notified.

- The `CS Incident Enters Solved` stage sends the submitter an email notification that the ticket has been resolved.
- Solved tickets are either re-opened when the customer sends back an email reply, or the customer is never heard from again. Eventually the ticket is manually closed.

Closed - Resolved tickets are set to the `Closed` stage.

- The `CS Incident Enters Closed` stage entry procedure notifies the submitter of the ticket that the ticket is closed.

Automated Processing Using Policies

You can apply different values to the fields within a ticket and automate the processing of that ticket differently, based on the **policy** assigned to that ticket. *Each desk definition can be associated with any number of different policies.*

> **Note:** The **Desk Templates** *(page 42)* page provides instructions on how to create an ITIL-based `Incident` service desk. The `Incident` service desk incorporates the automation features described in this topic.

Configuring Policies

A policy can be assigned to a ticket at the time a ticket is created manually, or automatically based on an **inbound email or alert** *(page 5)*. The policy selected can be based on rules set up for matching the organization name, organization type or machine group of a ticket using Desk Definition > Processing > **Associated Policies** *(page 40)*.

The policy can also be assigned to a ticket at any time using a procedure.

The automatic processing of new tickets is based on four interlocking tables of information:

- Procedure Variables
- Policies
- Policy Variables
- Associated Policies

Variables

A variable name and default value must be defined first, before it can be used as a **policy variable**.

Policies

A selected policy defines two types of information: **hour of coverage** and **policy variables**. The system includes hours of coverage when calculating escalation and goal times.

Policy Variables

Policy variables represent an important link in automating the processing of new tickets using policies. Each policy can be assigned one more variables with a default value. Once a policy is selected and applied to a newly created ticket, the values of any of the policy's variables can be interpreted by a procedure.

Associated Policies

The assignment of a policy to a ticket is based on the **Associated Policies** table defined in a desk definition. A single record in the **Associated Policies** table represents an association of *one* policy and any *one* of the following types of information: an **organization**, an **organization type**, or a **combination of machine group and operating system**. Multiple policy associations can be defined in this table. As a ticket is created, based on a specified desk definition, one or more of these three types of information can be entered *before the ticket is saved the first time*. Based on the three types of information entered, a corresponding policy "rule" is selected from the **Associated Policies** table and applied to the ticket. A default policy is selected if none of the three types of information is entered with the new ticket. This default is defined using the Desk Definition > Edit > **Standard Field Defaults** *(page 35)* tab. If there is no default policy for a definition, the system assumes a coverage schedule of 7x24.

For example, assume many different organizations are all generating tickets *using the same desk definition.* Several organizations can be defined that have all purchased `Diamond` service coverage. A second set of individual organizations have purchased `Gold` service coverage, and so on. Alternatively, these same policies could be selected and applied to tickets using organization types or machine group classifications instead.

Once each new ticket is assigned the appropriate policy, based on the **Associated Policies** table, procedures can be run that automatically act on the new ticket.

Procedures

Recall that one or more policy variables can be associated with each policy. IF-ELSE procedures can be written that *interpret the value of a policy variable assigned to a ticket*. Doing so enables a procedure to customize the settings of that ticket. Typically, this procedure would be run as a **Stage Entry or Exit** procedure, and be associated with the first stage of a ticket, which is usually called `New`.

For example, assume that a `Diamond` policy is associated with the organization `Walmart` and should always be prioritized as `High`. Instead of requiring service representatives to enter this information manually, you can create a procedure that always ensures this important customer receives the highest possible priority. Additional IF-ELSE branches can automatically set the properties of any of your other new tickets, including **Status**, **Priority**, **Severity**, **Category** and **Resolution**. The same procedure can also trigger outbound email and message notifications, add notes to the ticket, and even execute additional procedures as appropriate.

Note: Automatic processing of manually created tickets occurs only after you save the ticket.

Automated Processing Using Alerts and Inbound Emails

You can configure and automate the processing of new tickets *using alerts and inbound emails*.

Creating New Tickets Automatically from Alerts

Almost all alerts throughout the VSA include the option of automatically creating tickets. In addition there are links in various functions that enable you to manually create a ticket. To process these ticket-creation links into **Service Desk** tickets:

- The **Service Desk** module must be installed.
- The **Service Desk** module must be *activated* (page 51).
- Incoming Email and Alarm Settings must be *enabled* (page 56).

Creating New Tickets Automatically from Inbound Emails

Automatic creation and processing of tickets can be based on inbound emails. Service Desk > **Incoming Email and Alarm Settings** *(page 56)* specifies an email account to periodically poll. The body of the incoming email can contain a number of special fields of the form `~field="value"` that can be used to set various properties of the incoming ticket.

Email messages retrieved from the email server can be classified using a **Ticket Request Mapping** *(page 77)* procedure and converted into different types of tickets. The **Ticket Request Mapping** procedure uses IF-ELSE logic to interpret the email sender's *domain* or *domain/username* and determine how a newly created ticket should be processed. This includes selecting the appropriate desk definition, creating outbound emails and messages, adding notes, and executing additional procedures as appropriate.

Service Desk User Security

Note: When you create a pre-defined service desk using the Service Desk > **Desk Template** *(page 42)* page, much of the user security issues described in this topic are configured for you.

Note: See System > **User Security** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4575.htm)* for an introduction to VSA user security concepts mentioned in this topic.

Access to desk definitions, tickets, and KB articles comprise five levels of user security:

- Role Types
- User Roles
- Users
- Scopes
- Field Permissions

Note: Machine roles and **Service Desk** are discussed in a separate topic, **Integrating Service Desk, Live Connect, and Portal Access** *(page 9)*.

Role Types

Kaseya licensing is purchased by role type. There are separate role types for licensing users by *user role type* and licensing machines by *machine role type*. Each role type enables selected functions listed in the User Roles > Access Rights tab and Machine Roles > Access Rights tab. The number of role type licenses purchased displays in the System > License Manager > Role Type tab. Each role type license specifies the number of *named users* and *concurrent users* allowed.

User licensing for the **Service Desk** module is purchased and managed using two *user role types*:

- `Service Desk Administrators` - Equivalent to a `Master` role user within the **Service Desk** module only.
- `Service Desk Technicians` - A user who works with tickets and KB articles.

Note: The `Master` role is assigned the `Service Desk Administrators` role type by default.

User Roles

Three user roles are created when the **Service Desk** module is installed. These user roles provide three types of function access typically required by **Service Desk** users. You can use them as is, or modify them if you like, or use them as models for creating your own new user roles.

- `SD Admin` – A **Service Desk** administrator who has access to all **Service Desk** functions and tickets, regardless of scope. An `SD Admin` can create and edit desk definitions, configure **Service Desk** support tables and **Service Desk** procedures, and perform all actions on tickets. Only `SD Admin` users have access to advanced functions in the Service Desk > Tickets table such as **Delete** and **Unlock**. Like a `Master` role user, an `SD Admin` user is not limited by *field permissions*, described below. This user role is a member of the `Service Desk Administrators` role type.
- `SD User` – A **Service Desk** user who works with **Tickets**, **Archived Tickets**, **Search All** and **User Preferences**. This role does not permit access to desk definitions, procedures or any other support tables. This user role can only *view* published KB articles, but cannot create or edit KB articles. This user role is a member of the `Service Desk Technicians` role type.
- `KB Admin` – A **Service Desk** administrator who creates, edits and manages KB articles. The `KB Admin` user has access to all **Service Desk** functions. This user role is a member of both the `Service Desk Administrators` and `Service Desk Technicians` role type.

Users

A VSA user only has access to the **Service Desk** module and functions by assigning that VSA user to a user role using the `Service Desk Administrators` or `Service Desk Technicians` role type.

Scopes

Scopes and Service Desks

The following applies to a VSA user using a role that is linked to the `Service Desk Technicians` *role type and that same role is* not *linked to the* `Service Desk Administrators` *role type.*

- Assigning a desk definition to a scope using System > **Scopes** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4578.htm)* provides:
  - Visibility and selection of the service desk in drop down lists in **Service Desk**.
  - Visibility and selection of service desk tickets in ticket tables.
- Scope access only provides *visibility* of tickets. Further access to *editing* tickets is determined by role *field permissions*.

Note: See **Visibility of Service Desk Tickets by a Staff Member** *(page 8)* for an alternate method of making tickets visible to staff members.

Scopes, User Roles and Knowledge Base Desks

The following applies to a VSA user using a role that is linked to the `Service Desk Technicians` *role type and that same role is* not *linked to the* `Service Desk Administrators` *role type.*

- Knowledge base desks do not need to be added to any user role or any scope for KB articles to be visible to VSA users using those roles and scopes.
- If you are using the pre-configured `KnowledgeBase` desk, any KB articles set to the `Published` stage are visible and viewable for all service desk users and machine users in **Live Connect**, regardless of user role or scope. The same is true for any knowledge base desk created from scratch, so long as the KB article is set to the `End` stage, whatever the name of that `End` stage.
- If you want non-service desk administrators to be able to create a new KB article and edit the KB article, but don't want those same users complete service desk administrator access, select or create a user role associated with the `Service Desk Technicians` role type. Then associate the knowledge base with the user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab. Then assign users to that user role. The `KB Admin` user role can be used for this purpose. The `KB Admin` is already associated with the `KnowledgeBase` desk. You only need to remove the `Service Desk Administrator` roletype from the `KB Admin` user role.

Field Permissions

Field permissions are set by role. For VSA users using roles linked to the `Service Desk Technicians` role type, **field permissions** *determine what fields a user can view or edit within the ticket editor or KB article editor.* Typical field permissions include: `Editable`, `View Only`, `Hidden`, or `Required`. Default field permissions are set by **editing template** *(page 125)*.

Note: VSA users using a role linked to the `Service Desk Administrators` role type can see and work with any field in any ticket editor or KB article editor. `Master` role users also always have complete field permission access, regardless of roletype assignment.

Editing Templates

An **editing template** serves three purposes:

1. The editing template defines the layout of the dialog used to edit a ticket or KB article.
2. An editing template may *mask* selected fields, even though the fields are defined by the desk definition. *Using an editing template to mask a field overrides whatever field permission is set for that field.*
3. The editing templates also sets default **field permissions** *(page 126)* for editing a ticket or KB article. Whether assigned by role or by user, you can override the default field permissions set by the editing template to suit your business requirements.

An editing template is applied to a combination of desk definition and user role (or machine role) using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab. An editing template

can also be applied to a combination of desk definition and user using **User Preferences** *(page 55)*. User Preferences has precedence over Role Preferences. The default editing template for all roles and all users working with a service desk is specified in the Service Desk > Desk Definition > New or Edit > **General Info** *(page 34)* tab.

Default Field Permissions

Portal Access users (machine users) *(page 9)* use ticket field level permissions defined for the `Default` machine role. The `Default` machine role also applies to VSA users using a user role that does not include either the `Service Desk Administrator` or `Service Desk Technician` roletype. When a VSA user is using the `Default` machine role to view or edit a ticket, a `Default Permissions Apply` message displays at the top of a service desk ticket. If even the `Default` machine role does not provide access to a ticket, then an error message tells the user their role does not permit access to the ticket.

Unassigned Tickets

Tickets can be created without assigning an organization to the ticket. The Desk Definition > **General Info tab** *(page 34)* > **Only Masters see Unassigned Tickets** checkbox determines whether master users or all users can see unassigned tickets.

Visibility of Service Desk Tickets by a Staff Member

If a VSA user name is associated with the staff member record of an organization, then that VSA user has visibility of tickets associated with that staff member record *even if the VSA user's scope does not allow it*. Any tickets created by that VSA user are automatically associated with their staff member record and organization. This method primarily supports machine users using Portal Access to create and manage their own tickets. Machine users expect to have access to all the tickets they create and to any tickets created on their behalf, but may have no scope privileges defined for them. If a scope does exist for a VSA user associated with a staff member, checking the checkbox called **View all tickets** in the staff member record provides visibility of those additional tickets by scope.

Example: Dale is the main customer contact for the XYZ organization. He is provided a scope that allows him to see all tickets related to his organization, even tickets not created by him, so the **View all tickets** checkbox is enabled. Brandon from the XYZ organization contacts the service desk to submit a ticket as well. Initially it's unclear whether Brandon should have access to any other tickets beyond the tickets he himself creates, so the **View all tickets** is left unchecked. Later, if Dale okays greater access for Brandon, the service desk provider can assign a scope to Brandon and check the **View all tickets** checkbox.

Sharing User-Owned Objects in Service Desk

Service Desk Procedures - Unlike agent procedures, **Service Desk** procedures are always created in *shared* folders.

Named Filters - Named filters in **Tickets** *(page 15)*, **Knowledge Base** *(page 30)* and **Search All** *(page 31)* always start as private, but can be shared.

Note: See Agent Procedures > **Folder Rights** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4673.htm)* for an introduction to sharing rules in folder trees.

Note: See System > **Sharing User-Owned Objects** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#5537.htm)* for an introduction to sharing user-owned objects.

Integration Options in Service Desk

Service Desk can be integrated with several other Kaseya modules. The following are standard options.

- **Migrating Tickets from the Ticketing Module** *(page 9)*
- **Integrating Service Desk and Live Connect** *(page 9)*
- **Time Tracking and Service Desk** *(page 11)*
- **Task Information** *(page 14)* - Enables the creation and management of *tasks* within a **Service Desk** ticket.
- **Service Billing Integration** *(page 14)* - Enables hours worked in a **Service Desk** ticket to be *billed* in the **Service Billing** module.

Note: The **Service Billing Integration** and **Task Information** options cannot be used together within the same service desk.

Migrating Tickets from the Ticketing Module

You can migrate tickets from the **Ticketing** module, as well as import **Service Desk** tickets into the **Ticketing** module.

Migrating Tickets from Ticketing into Service Desk

The paging area of **Migrate Tickets** displays all the tickets visible to you in the Ticketing > View Summary page.

1. Select the tickets you want to migrate in the paging area. Click **Select All** to select all tickets.
2. Click **Migrate** to migrate all the selected tickets into **Service Desk**.

Importing Service Desk Tickets into Ticketing

1. Export selected tickets in **Service Desk** to an XML file on your local machine or network, using System > **Import Center** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6963.htm)*.
2. Click **Import** in Ticketing > **Migrate Tickets** and select the XML file you created in step 1 above.

Integrating Service Desk, Live Connect and Kaseya User Portal

One of the functions provided by **Live Connect** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#33845.htm)* or **Kaseya User Portal** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#33858.htm)* is the ability to see all tickets associated with a selected machine. By default, the tickets displayed by **Live Connect** and **Kaseya User Portal** are for the **Service Desk** module only. If **Service Desk** is **de-activated** *(page 51)*, the **Ticketing** function in **Live Connect** and **Kaseya User Portal** displays **Service Desk** tickets only.

When **Live Connect** and **Kaseya User Portal** are integrated with **Service Desk**, the **Ticketing** function displays two tabs: one tab for tickets and a second tab for KB articles. Working with tickets and KB articles within **Live Connect** and **Kaseya User Portal** is similar to working with them using the **Service Desk** module in the VSA.

Prerequisites

- The **Service Desk** module is installed.
- Agents are already installed on managed machines.

Configuration

1. Create a desk definition and knowledge base definition. The easiest way to do this is to use Service Desk > **Desk Templates** *(page 42)* to create sample definitions and modify them to suit your preferences.
2. Create a ticket in **Service Desk** associated with a specific machine ID account you can use to test with.
3. Create a KB article. KB articles are not associated with specific machines.
4. For KB articles created using the pre-configured `KnowledgeBase` desk, set any KB articles to the `Published` stage to make them visible to **Live Connect** and **Kaseya User Portal** users . If you created your knowledge base definition from scratch and want to display its KB articles to **Live Connect** and **Kaseya User Portal** users, then those KB articles must be set to the `End` stage, whatever the name of that `End` stage.
5. Configure the `Default` machine role.
   - Ensure your created service desk and knowledge base definitions are assigned to the `Default` machine role using Service Desk > **Role Preferences** *(page 52)*.
   - Ensure machine IDs using **Live Connect** and **Kaseya User Portal** are assigned to the `Default` machine role using System > **Machine Roles** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4827.htm)*.
   - Ensure the **Function Access** tab for the `Default` machine role shows the **Service Desk** functions enabled.

   > **Note:** If you create an additional machine role for machine users, repeat step 5 for this machine role.
6. Both the service desk and the organization or machine must be a member of the `Anonymous` **scope** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4578.htm)* to display **Service Desk** tickets in **Live Connect** and **Kaseya User Portal** or **Live Connect (Classic)** and **Portal Access (Classic)**.
7. Ensure **Service Desk** is **activated** *(page 51)* **Service Desk** using Service Desk > **Global Settings** *(page 51)*.

Using Live Connect with Service Desk as a VSA User

1. Click the agent check-in icon next to a machine ID to display the **Live Connect** page.
2. Click the **Ticketing** function to display the **Service Desk** tab and **Knowledge Base** tab. You should see any tickets and KB articles created within the VSA that are associated with this machine ID.
3. Create one or more tickets within **Live Connect**. You cannot create KB articles as a **Live Kaseya User Portal** machine user.
4. As a VSA user in the **Service Desk** module, you should see the tickets created as a **Live Connect** user for this machine ID.

Using Kaseya User Portal and Service Desk as a Machine User

1. Click the agent icon in the system tray of the managed machine to display the **Kaseya User Portal** page.

   > **Note:** When a machine user uses **Live Connect**, it is called the **Kaseya User Portal** page.
2. Click the **Ticketing** function to display the **Service Desk** tab and **Knowledge Base** tab. You should see any tickets and KB articles created within the VSA that are associated with this machine ID.
3. Create one or more tickets within **Kaseya User Portal**. You cannot create KB articles as a **Kaseya User Portal** machine user.
4. As a VSA user in the **Service Desk** module, you should see the tickets created by the **Kaseya User Portal** user of this machine ID.

Time Tracking and Service Desk

You can optionally report hours worked and add notes to **Service Desk**:

- **Using Session Timers in the Ticket Editor** *(page 11)* - Requires "session timers" be enabled by desk definition and by user role.
- **Using System Timers with Service Desk** *(page 12)* - Requires timers be enabled. See Time Tracking > **Configuring Time Tracking** *(http://help.kaseya.com/webhelp/EN/KTT/9030000/index.asp#7164.htm)*.
- **Using Timesheets with Service Desk** *(page 13)* - Requires timesheets be enabled. See Time Tracking > **Configuring Time Tracking** *(http://help.kaseya.com/webhelp/EN/KTT/9030000/index.asp#7164.htm)*.

Using Session Timers In the Ticket Editor

Enabling "Session Timers"

1. Ensure **Enable Session Timers** is checked for a service desk using Service Desk > Desk Definition > New / Edit > **General Info** *(page 39)* tab.
2. Ensure **Enable Session Timers** is checked for a user role using Service Desk > **Role Preferences** *(page 52)*.

> **Note:** You can always use timesheet entries and the timers displayed in the upper-right corner of the VSA to add time entries to a **Service Desk** ticket, regardless of whether "session timers" are displayed within the ticket editor.

Using "Session Timers" in the Ticket Editor

When enabled, the following "session timer" displays in the ticket editor.

- **Start Timer** - Starts a timer associated with the ticket.
  - The ticket does not have to be saved to keep the timer going.
  - You can close the ticket, log out of the VSA, then log back in later. The timer continues to track elapsed time.
  - If a ticket timer is already running, the **Pause Timer** and **Apply Timer** buttons display. You can now pause or apply the elapsed time.
  - When you start a timer in the ticket timer, it displays as one of the standard timers in the upper-right hand corner of the VSA.
    - ✓ A timer created using the **Start Timer** button in the ticket editor always displays the number of the ticket it is linked to, for example, `Ticket CS009345`.
    - ✓ You can pause and apply a timer linked to a ticket using the timer controls in the upper-right hand corner of the VSA and the time will be applied to the ticket just as if you did so from within the ticket editor.

      > **Note:** If you happen to create additional timers *linked to the same ticket number* using the timer control panel in the upper-right hand corner of the VSA, the timer furthest to the left is the one that is always selected if the ticket is re-opened in the ticket editor. Typically this is the timer that has been running the longest, since timers are created from left to right.

- **Pause Timer / Resume Timer** - Displays once the **Start Timer** button is clicked. Toggles between pausing the timer and resuming.
- **Apply Timer Time** - Displays once the **Start Timer** button is clicked. Populates the **Hours Worked** field with the elapsed time. You can override the time displayed in the **Hours Worked** field before you save the ticket.

Using System Timers with Service Desk

If timers have been **configured** *(http://help.kaseya.com/webhelp/EN/KTT/9030000/index.asp#7164.htm)* for your VSA logon use the following procedure.

Create a **Service Desk Ticket** timer entry to add the time to the entire **Service Desk** ticket. You can optionally link the ticket to **Service Billing** records or to a task, if either one of these features is enabled. Time entered displays as a note in the ticket.

1. Click the new timer  icon to add a new timer. Timers are located in the upper right hand corner of the VSA window. The **New Timer** dialog displays. Enter or select values for the following fields.
2. Select a unique **Timer Color**. You can define multiple timers concurrently so it helps to assign them different colors.
3. Enter a **Label** for your timer. The label displays whenever the timer icon is selected and added as a note to any time entry you create from the timer. If blank, the timer is labeled by the work type you select.
4. If **Start of Save** is checked, the time starts running as soon as you save this dialog.
5. Select the **Service Desk Ticket work type** *(page 127)* option.
6. Select the **Service Desk** definition.
7. Optionally select a **Status Filter** to limit the list of tickets displayed in the **Ticket** drop-down list.
8. Select either:
   - **Show All Tickets**
   - **Show My Tickets**
9. Select a **Ticket**.
10. The following fields only display if a service desk is integrated with **Service Billing**. See **Creating Billing Entries using Service Desk** *(page 14)*.
    - **Work Order** - Display only. Displays only if a work order is associated with the ticket on the General tab of the ticket editor.
    - **Work Order Item** - The work order line to associate with the hours worked. Displays only if a work order is associated with the ticket on the **General** *(page 22)* tab of the ticket editor.
    - **Activity Type** - Labor entries are grouped by **activity type** to analyze the cost and revenue of labor. The classification of activity types typically reflects the accounting requirements of a company. Labor entries are classified by both activity type and **resource type** *(page 127)*. Not editable if a detailed work order is selected.
    - **Resource Type** - A **resource type** sets a default rate for a billable labor item or entry. Typically a resource type represents a skill performed by a staff member. A billing rate and standard cost is defined for each resource type required to perform the service. The rate can be overridden when selected. The classification of resource types typically reflects the production requirements of a company. Labor entries are classified by both resource type and by activity type. Not editable if a detailed work order is selected.
    - **Rate** - The default billing rate for the selected resource type. Display only.
    - **Override Rate** - A manually entered rate that overrides the default billing rate for a selected resource type. Does not display if a detailed work order is selected.
    - **Show Note on Invoice** - If checked, the note is displayed on the printed invoice.
11. Select a **Task**. This field only displays if **Tasks** *(page 14)* are enabled for the service desk.
12. Optionally add a **Note**.
13. Optionally make the note a **Hidden Note** in the ticket.
14. **Billable** - If checked, the entry is billable. If **Service Billing** is not installed, the **Billable** checkbox is for reference purposes only. If **Service Billing** is installed the entry is forwarded to **Service Billing**. If timesheets require approval, the timesheet containing this entry must be approved before the entry is forwarded to **Service Billing**.

15. Click **Save** to close this dialog. The new timer clocks begins recording the time for this activity.
16. Complete the activity being timed by this timer.
17. Click the Checkmark ✔ icon to display the **Apply Time** window. You can edit your time entry, including the elapsed time, and either:
    - **Apply and Remove** - Apply your time entry to your timesheet and remove the timer.
18. **Apply and Reset** - Apply your time entry to your timesheet and reset the timer to 0.

Using Timesheets with Service Desk

If timesheets have been **configured** *(http://help.kaseya.com/webhelp/EN/KTT/9030000/index.asp#7164.htm)* for your VSA logon use the following procedure.

Create a **Service Desk Ticket** timesheet entry to add the time to the entire **Service Desk** ticket. You can optionally link the ticket to **Service Billing** records or to a task, if either one of these features is enabled. Time entered displays as a note in the ticket.

1. Select a timesheet using Time Tracking > **My Timesheets**.
2. Add a new entry to the timesheet by clicking **Add Entry**. The **New Timesheet Entry** dialog displays.
3. Enter a date and time for the timesheet entry.
4. Select the **Service Desk Ticket work type** *(page 127)* option.
5. Select the **Service Desk** definition.
6. Optionally select a **Status Filter** to limit the list of tickets displayed in the **Ticket** drop-down list.display.
7. Select either:
   - **Show All Tickets**
   - **Show My Tickets**
8. Select a **Ticket**.
9. The following fields display only if the **Service Billing** is installed and integrated with **Service Desk**. See **Creating Billing Entries using Service Desk** *(page 14)*.
   - **Work Order** - Display only. Displays only if a work order is associated with the ticket on the General tab of the ticket editor.
   - **Work Order Item** - The work order line to associate with the hours worked. Displays only if a work order is associated with the ticket on the **General** *(page 22)* tab of the ticket editor.
   - **Activity Type** - Labor entries are grouped by **activity type** to analyze the cost and revenue of labor. The classification of activity types typically reflects the accounting requirements of a company. Labor entries are classified by both activity type and **resource type** *(page 127)*. Not editable if a detailed work order is selected.
   - **Resource Type** - A **resource type** sets a default rate for a billable labor item or entry. Typically a resource type represents a skill performed by a staff member. A billing rate and standard cost is defined for each resource type required to perform the service. The rate can be overridden when selected. The classification of resource types typically reflects the production requirements of a company. Labor entries are classified by both resource type and by activity type. Not editable if a detailed work order is selected.
   - **Rate** - The default billing rate for the selected resource type. Display only.
   - **Override Rate** - A manually entered rate that overrides the default billing rate for a selected resource type. Does not display if a detailed work order is selected.
   - **Show Note on Invoice** - If checked, the note is displayed on the printed invoice.
10. Select a **Task**. This field only displays if **Tasks** *(page 14)* are enabled for the service desk.
11. Optionally add a **Note**.
12. Optionally make the note a **Hidden Note** in the ticket.

13. **Billable** - If checked, the entry is billable. If **Service Billing** is not installed, the **Billable** checkbox is for reference purposes only. If **Service Billing** is installed the entry is forwarded to **Service Billing**. If timesheets require approval, the timesheet containing this entry must be approved before the entry is forwarded to **Service Billing**.
14. Click **Save** to close this dialog. The new entry displays in the timesheet.
15. Click **Save** to save your changes to the timesheet.

Task Information

Tasks organize a **Service Desk** ticket into a series of steps. Each task can be assigned a different assignee and tracked individually for completion. You can add notes and hours worked by individual task. All tasks for all tickets can be listed, sorted, filtered, searched and viewed using the **Tasks Associated with Tickets** *(page 29)* page. If enabled in the ticket editor in the **Tickets** page, tasks are accessed using a separate **Tasks tab** *(page 26)*.

Note: The **Service Billing Integration** and **Task Information** options cannot be used together within the same service desk.

To enable the display of tasks in **Service Desk**:

1. Ensure **Enable Task Information** is checked in the Desk Definition > New / Edit / **General Info** *(page 34)* tab if you want to make this the default setting for all new user roles created from this point forward.
2. Ensure **Show Task Information** is checked in the Service Desk > Role Preferences > **Service Desk** *(page 54)* tab for each combination of role and service desk, as necessary.
3. Ensure task values are defined for the desk definition, using the Desk Definition > Properties > **Task Status Values** *(page 38)* tab.
4. Create or edit a ticket. The **Tasks tab** *(page 26)* will display.

For more information see:

- **Tasks Associated With Tickets** *(page 29)*
- **Adding / Editing a Task** *(page 30)*
- **The Add Note Dialog** *(page 27)*

Creating Billing Entries using Service Desk

Time entered in tickets for **Service Desk** can be included as billable entries in **Service Billing**. Tickets associated with a **Service Billing** customer are listed in the Customers > Tickets tab.

Note: The **Service Billing Integration** and **Task Information** options cannot be used together within the same service desk.

Creating Billable Entries Using Service Desk

1. Ensure the customer in **Service Billing** is also an organization in the VSA.
2. Ensure **Enable Service Billing Integration** is checked for the service desk using Service Desk > Desk Definition > New / Edit > **General Info** *(page 39)* tab.
3. Ensure **Submit Time Entry Data to Service Desk** is checked using Time Tracking > **Settings**. This allows time entered in **Service Desk** tickets to appear in **Service Billing**.
4. Ensure the following fields are checked in Service Desk > Role Preferences > **Service Desk** *(page 54)* tab for each combination of role and service desk, as necessary.
   - Work Order
   - Activity
   - Resource Type

- Standard Rate
- Override Rate

5. Associate a ticket with an organization in the **General** tab of the **Service Desk** ticket editor.
6. Optionally associate the ticket with a selected work order in the **General** tab of the ticket editor. The work order field only displays after the organization is selected.
7. Create an billable or non-billable entry in **Service Desk** that can be forwarded to **Service Billing** using one of three procedures:
   - Enter a entry in the **Notes tab** *(page 24)* of the ticket editor.
   - Add a note using the **The Add Note Dialog** *(page 27)*.
   - Create a time entry using **timesheets or timers** *(page 11)*.
8. Bill for the billing entries you submit in Service Billing > **Pending Items**. If timesheets require approval, a timesheet containing a billing entry must be approved before the billing entry is forwarded to **Service Billing**.

Operations

The **Operations** folder contains the items you use most frequently in the **Service Desk** module:

- **Tickets** *(page 15)* are records of your organization's responses to support issues.
- **Organization Tickets** *(page 28)* display tickets by organization.
- **Tasks Associated with Tickets** *(page 29)* displays all ticket tasks by task. See **Task Information** *(page 14)*.
- **Archive Tickets** *(page 30)* are closed tickets that you can continue to reference.
- **Search All** *(page 31)* searches for text in any ticket or KB article you have access to.
- **Knowledge Base** *(page 30)* articles comprise an ever-growing reference to known issues about your products or services.

Tickets

Service Desk > Operations > Tickets

A **ticket** tracks the detection, reporting, and resolution of an issue. Every communication and action taken on behalf of the issue can be recorded in the ticket record.

See the following subtopics for **Tickets**:

The Tickets Table

Service Desk > Operations > Tickets

The **Tickets** page displays all tickets a VSA user is authorized to see in a table format. See the following

additional topics about using the **Tickets** table.

- **Action Buttons** *(page 17)*
- **Filtering Tickets** *(page 18)*

Upper and Lower Panes

The page can be optionally divided into upper and lower panes, by service desk, using the Desk Definition > **General Info tab** *(page 34)* > **Show Incident Notes Pane** checkbox. If enabled, the table is in the upper pane and the *description* and *notes* of a selected ticket display in two separate tabs in the lower pane.

Page Selection and Ticket Counts

A page selection bar displays just above the column headings of the **Tickets** table.

- **Page Select** - When more rows of data are selected than can be displayed on a single page, click the < and > buttons to display the previous and next page. The drop-down list alphabetically lists the first record of each page of data using the *sort order of the selected column on that page*.
- **Selected** - The number of rows selected in the table. Includes selections on multiple pages.
- **Viewing** - The number of rows in the current page and the total number of rows in all pages.

Customizing Columns

This table supports **selectable columns, column sorting, column filtering and flexible columns widths** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6875.htm)*.

Standard Fields and Custom Fields

Standard Fields *(page 68)* refer to built-in fields provided by a service desk. They display as columns in the ticket table and in **field permission** *(page 54)* dialogs. A standard field is not necessarily available for all combinations of desk definition and user role. Nor does the same standard field necessarily display in both the ticket table and in field permission dialogs. **Custom fields** *(page 38)*—if defined for a service desk—also display as columns in the ticket table and in field permission dialogs.

Tickets Icons

Tickets rows display one of the following icons:

- - The ticket has been *read* since it was last changed.
- - The ticket is *unread* since it was last changed. When a note is added to a ticket, the ticket is reset to *unread* for other users.
- - **Email notification is pending** *(page 127)* since the ticket was last changed. Select the ticket and click **Stop Notification** to prevent notification.
- - The ticket is *locked* for editing.

Previewing Tickets

To display the details of a ticket in a *new window* click the new window icon. Hovering the mouse cursor over the icon of a ticket displays a preview window of the latest notes for that ticket. Use this to quickly review tickets in your queue. The number of milliseconds the cursor has to hover can be specified using System > **Preferences** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#503.htm)*.

Color Coding of Ticket Rows

Ticket rows are color-coded by their age, from a selected type of date: `Creation Date`, `Due Date`, `Promised Date`, or `Date Last Modified.` Colors are assigned to:

- Tickets older than a specified number of days from the selected type of date.
- Tickets younger than a specified number of days from the selected type of date.
- Tickets in between these two age groups.

Color coding is defined using **Role Preferences** *(page 52)* and **User Preferences** *(page 55)*. **User**

Preferences has precedence over **Role Preferences**.

Action Buttons

Service Desk > Operations > Tickets

The following actions are provided to manage selected tickets. In some cases you have to select one or more rows to enable these action buttons.

- **New Ticket** - Creates a new ticket.

  > **Note:** You can customize the **New Ticket** forms displayed when adding and editing tickets.

- **Add Note** - Enters notes and reports hours worked for a **selected** ticket. See **The Add Note Dialog** *(page 27)*.
- **Goto** - Enter a known ticket ID to edit or view the ticket.
- **View** - Displays a ticket without allowing any edit changes. Selected users may only have access to view a ticket using this option. You can select multiple tickets and page from one ticket to the next in **View** mode.
- **Edit** - Edits a selected ticket.
- **Delete** - Deletes selected tickets.
- **Mark** - Displays the following options.
  - **Read** - Marks selected unviewed tickets as having been previously viewed.
  - **Unread** - Unmarks selected viewed tickets as having not yet been viewed.
- **Group Updates** - Updates multiple tickets at the same time. For example, you can set a group of tickets to `Closed`. All tickets must be members of the same service desk.
- **Ticket** - Displays the following options.
  - **Stop Notification** - Pending email notification of selected tickets can be stopped using **Stop Notification** in **Tickets** *(page 15)*. This option is used in conjunction with **Email Send Delay** in **Role Preferences** *(page 52)* and **User Preferences** *(page 55)*. For example, if **Email Send Delay** is set to `5` minutes for the role you are using, then you have `5` minutes to cancel any emails that are pending for a ticket using **Stop Notification**. Applies to ticket rows displaying a  icon in the icon column of the ticket table. **Stop Notification** only stops currently pending email.
  - **Archive** - Archives selected tickets. See **Archived Tickets** *(page 30)* to review archived tickets or to unarchive tickets.
  - **Copy** - **Copies a selected ticket** *(page 18)* to a target service desk or knowledge base desk.
  - **Link** - Links two or more selected items. Linked items display in the **Related Tickets** tab of a ticket or KB articles. Use **Search All** *(page 31)* to select and link items from different desks.
  - **Unlink Related Items** - Unlinks items related to a single selected item.
  - **Print** - Prints a single ticket in PDF, Excel, or HTML output, with or without notes. You can optionally hide hidden notes and custom fields.
  - **Unlock** - Unlocks selected items. An item locks each time you edit it, preventing other users from editing it at the same time. If a user has left an item open and you need to maintain it, you can unlock the item to gain access to it. Doing so prevents the user who originally opened the item from saving his or her changes. Tickets locked by expired or non-existent sessions are automatically unlocked when **reapply schema** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#248.htm)* runs.
  - **Merge** - Merges multiple tickets into a single ticket. Presents differences between tickets as options. You must select which option you want to keep after the merge. A single ticket ID is selected from the tickets begin merged as the ticket ID for the merged ticket. All other merged tickets are closed. Notification is suppressed for tickets closed as a result of a merge. Use **Search All** *(page 31)* to select and merge tickets from different service desks.

  - **Undo** - Reverts changes made during the last save to a ticket's status, priority, stage, category, subcategory, resolution, last edit time and any note made during the last save. All other fields are unchanged. Procedures are not run as a result of an **Undo**.
  - **Re-Open** - Changes an End-stage (for example, `Closed`) ticket to a Begin-stage (for example, `New`) ticket.
  - **Remove HTML** - Removes HTML tagging from the Description field and all notes in the ticket.
- **Statistics** - Displays **Service Desk** statistics, by desk definition and date range.
-  - Click this icon to display a list of standard options.
  - **Export** - Generates a CSV file of user data provided by the page that you can download. Options include `All`, `Selected`, `Current Page`.
  - **Refresh** - Refreshes the page.
  - **Reset** - Clears any filtered set on this page.

Copying Tickets and KB Articles

`Service Desk > Tickets > Ticket > Copy`
`Service Desk > Knowledge Base > Copy an Article`
`Service Desk > Search All > Copy`

The fastest way of creating a new item is to create it from another item. In the **Service Desk** module there are four buttons that support copying tickets or KB articles.

- The Service Desk > Tickets > Ticket > **Copy** button *copies tickets to either a new ticket or a KB article*.
- The Service Desk > Tickets > Edit > **Copy Ticket** button *copies the currently open ticket to either a new ticket or a KB article*.
- The Service Desk > Knowledge Base > **Copy an Article** button *only copies a KB article to a KB article.*
- The Service Desk > Search All > **Copy** button *copies either tickets or KB articles to either tickets or KB articles.*

Options

- **Link Ticket** - If checked, links source and target items as related items.
- **Summary** - A one line summary description of the ticket.
- **(Field Values)** - Changes the field values of the target item being created.
- **Note Handling** - Inserts a hidden note in the target item with the source note information.
  - **Don't Copy Notes** - Notes are not copied to the target item.
  - **Copy Notes** - Notes are copied to the target item without changing them.
- **Action to take on original ticket**
  - **Close** - Makes a copy of the original item, then closes the original item.
  - **Close and Archive** - Makes a copy of the original item, then closes and archives the original item.
  - **Delete** - Makes a copy of the original item, then deletes the original item permanently. In effect, the original item is "moved" from one desk to another.
  - **Do Nothing** - Takes no action on the original item.

Filtering Tickets

`Service Desk > Operations > Tickets`

There are multiple ways of filtering the rows displayed in the **Tickets** table. **All methods of filtering can be combined at any time.**

- **Service Desk Selection** *(page 19)*

- **Searching for Tickets** *(page 19)*
- **Column Filtering** *(page 20)*
- **Named Filters** *(page 20)*

Clear Filtering

Click the  icon in the upper right corner of the **Tickets** table and select **Reset** to remove all filtering on the Tickets table.

Service Desk Selection

The tickets displayed in the **Tickets** table depend on the **Service Desk** you select at the top of the table.

> **Note:** Custom columns are defined by service desk and display only if you select the service desk they were defined for. Custom fields can be selected like any other column, sorted on and used to filter items.

Searching Tickets

Enter a search string in the **Search** edit box. Only rows containing the search string in the **Ticket Id**, **Summary**, **Description** or **Notes** fields are displayed. You can also use the **Goto** *(page 17)* action button to search for a ticket by ticket ID.

Advanced Searches

Users can enter both words and phrases when using the **Search** field in **Tickets** *(page 15)* and **Search All** *(page 31)*. Searches are not case sensitive.

	Sample Data

	`agent only`
	`agent and machine group`
	`machine only`
	`agent machine`

Example	**Returns**

`agent`	`agent only`
	`agent and machine group`
	`agent machine`

`"agent machine"`	`agent machine`

Search replaces common words—such as `the`, `on`, `when`, `for`, `is`—with `and`. You can see the entire list by running the following SQL query:
`select * from sys.fulltext_system_stopwords where language_id = 0`

Example	**Returns**

`an agent for you`	`agent only`
	`agent and machine group`
	`agent machine`

`an agent for you on the machine` (`An agent for you on the machine` is interpreted as `agent and machine`. The words `an`, `for`, `you`, `on`, `the` are removed.)	`agent and machine group` `agent machine`

Users may use `*`, but must at least have the starting word. `*` at the end returns nothing.

Example	Returns

`agen*`	`agent and machine group`
	`agent machine`
	`agent only`

`*gent`	(nothing)

Users can use `and`, `or` and `not` words.

Example	Returns
`agent and machine`	`agent and machine group` `agent machine`

`agent or machine`	`agent only`
	`agent and machine group`
	`machine only`
	`agent machine`

`agent and not machine`	`agent only`

Column Filtering

Click the column drop-down arrow to enter a filter value for that column. For example enter `NS` to find all rows that start with `NS` in that column. Enter `NS%2` to find all rows that start with **NS** and end with `2` in that column. You can filter by multiple column filters if you like.

Named Filters

Named filters are "fixed" combinations of filtering criteria, based on many of the columns you see in the table. They are selected using the **View** drop-down list. Named filters are created, edited, saved, shared and deleted by users. Additional options include:

- **This field can be a column filter** - This extra checkbox displays for many of the filter columns shown in the **Edit a Filter** dialog.
  - If checked, the filter selections you've defined for that column can be changed dynamically within the table, just like any other column not specified by the named filter.
  - If blank, the filter selections you've defined for that column cannot be modified in the table. The column filter option doesn't display for this column in the table, preventing you from making any changes to it. *Leaving this checkbox blank reduces confusion*, since you can be confident that the named filter settings have not been modified by any additional column filtering you may have performed.
- **Similar to/Like** - The text entered occurs *anywhere* within the target cell. Entering wild card characters is not necessary. For example, filtering a column by text that contains `age` will display records with the following cell values: `age`, `agent`, `storage`, etc.
- **Not Like/Is Not Assigned** - The text entered occurs *nowhere* within the target cell.
- **Assignees and Pools** - Filtering by assignee includes all pools an assignee is a member of. Filtering by `Any Assignee` shows all tickets assigned to either an assignee or pool. Filtering by `No Assignee` shows unassigned tickets.
- **Sharing** - See **Sharing User-Owned Objects** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#5537.htm)*.

Named Filters and Service Desk Selection

- If you have a named filter that specifies a service desk, then the service desk for that filter is selected in the service desk drop-down list.

- If you have a named filter that does not specify a service desk, then the named filter still applies when you switch service desks using the drop-down list.
- If you have a named filter that specifies a service desk, and then you change the service desk in the service desk drop-down list, the other filter criteria for that named filter still apply, but the **View** combo box shows `No Filter`.

The Ticket Editor

Service Desk > Operations > Tickets > New Ticket or Edit

> **Note:** The ticket layout described below is for the standard `Fixed_Width_Tabbed` **editing template** *(page 125)* used to create a ticket based on the `ITIL Incident` **definition template** *(page 44)*. You may see a different order and selection of fields if a different editing template and desk definition is used.

Adding / Editing a Ticket

Click **New Ticket** to display the **Create a New Ticket** window. To edit an existing ticket, click a row then click the **Edit** button, or double-click a row.

Ticket Editor Action Buttons

- **Save and Close** - Save and close the ticket.
- **Save and New** - Save and close the ticket, then open a new ticket.
- **Cancel** - Close the ticket without saving any changes.

> **Note:** See **Time Tracking and Service Desk** *(page 11)* for an introduction to timers in **Service Desk**.

- **Start Timer** - Starts a timer associated with the ticket.
  - The ticket does not have to be saved to keep the timer going.
  - You can close the ticket, log out of the VSA, then log back in later. The timer continues to track elapsed time.
  - If a ticket timer is already running, the **Pause Timer** and **Apply Timer** buttons display. You can now pause or apply the elapsed time.
  - When you start a timer in the ticket timer, it displays as one of the standard timers in the upper-right hand corner of the VSA.
    - ✓ A timer created using the **Start Timer** button in the ticket editor always displays the number of the ticket it is linked to, for example, `Ticket CS009345`.
    - ✓ You can pause and apply a timer linked to a ticket using the timer controls in the upper-right hand corner of the VSA and the time will be applied to the ticket just as if you did so from within the ticket editor.

    > **Note:** If you happen to create additional timers *linked to the same ticket number* using the timer control panel in the upper-right hand corner of the VSA, the timer furthest to the left is the one that is always selected if the ticket is re-opened in the ticket editor. Typically this is the timer that has been running the longest, since timers are created from left to right.

- **Pause Timer / Resume Timer** - Displays once the **Start Timer** button is clicked. Toggles between pausing the timer and resuming.
- **Apply Timer Time** - Displays once the **Start Timer** button is clicked. Populates the **Hours Worked** field with the elapsed time. You can override the time displayed in the **Hours Worked** field before you save the ticket.

Tickets are added and edited using the following tabs in the ticket editor dialog.

- **General tab** *(page 22)*
- **Notes tab** *(page 24)* - Displays only when editing a ticket.
- **Related Items tab** *(page 26)*
- **Tasks tab** *(page 26)*

Note: Some of the fields in the ticket editor may be hidden or view only, based on settings in **Role Preferences**.

General tab

Service Desk > Operations > Tickets > Add/Edit > General tab

Summary Information

- **Service Desk** - *Can only be changed if the ticket is new.* Select a desk definition for the new ticket. The desk definition populates a new ticket with pre-defined values and determines what values are possible for you to select in standard and custom fields. Automated processing of tickets is determined by the set of procedures linked to that ticket's service desk. You can select a different service desk provided you've been given access to more than one service desk. Defaults to the **service desk selected in the Tickets table** *(page 19)*.
- **Ticket Number** - Blank if the ticket is new. The unique identifier for the ticket. This value cannot be changed.
- **Summary** - A one line summary description of the ticket.

User Associations

- **Submitter Name** - The name of the person submitting the ticket. Usually this is the name of the user calling or emailing about a service issue.
- **Submitter Email** - The email address of the submitter.
- **Assigned to** - The VSA user or **pool** *(page 126)* the ticket is assigned to. The list of assignees available to select includes *any* member of *any* non-master scope you belong to, providing that scope is assigned the same service desk you are using. A scope does not have to be currently selected for the **Assigned to** field to include its members in the list.
- **Owner** - This field only displays after a ticket is created. The VSA user ultimately responsible for resolving the ticket.

Asset and Organization Associations

A ticket can be associated with other types of data defined in other modules of the VSA. There are two ways of associating a ticket: by specific *inventory asset* or more generally by *organization*. Selecting a specific inventory asset automatically selects the asset's corresponding organization. Associating a ticket with an asset or organization:

- Makes the ticket visible to users if their *user role* uses the `Service Desk Technicians` role type and the related asset or related organization is within their *scope*.
- Can determine the *policy* assigned to this ticket *if the association is created while the ticket is new*. See Service Desk > Desk Definition > Processing > **Associated Policies** *(page 40)* for more information.

Use the following fields to create or reference associations with a ticket.

- **Inventory Asset** - A machine ID or **mobile device** *(http://help.kaseya.com/webhelp/EN/KMDM/9030000/index.asp#6904.htm)* installed with agent—or a device discovered on a network using the Discovery module and **promoted to an asset** *(http://help.kaseya.com/webhelp/EN/KDIS/9030000/index.asp#10804.htm)* and displaying on the Audit > **View Assets** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#10649.htm)* page.
- **Organization**, **Contact**, **Phone Number** - Associates a ticket with an organization by using the magnifier icon  next to any of these fields. A contact is a staff member of an organization.

- **Organization ID** - If an organization has been associated with a ticket, an Organization ID hyperlink displays beneath the organization. Click the hyperlink to display information about the organization, including general information, custom fields, machine groups, departments, staff, and secured data. This same data is available using **Organization Tickets** *(page 28)*.
- **Contact Email** - Displays a contact email address, if an organization's staff member is associated with the ticket and the email address is defined in the staff member record.

Service Billing Associations

- **Work Order** - Displays only if **Enable Service Billing Integration** is enabled in Desk Definition > **General Info** *(page 34)* tab. Associates the ticket with a work order in Service Billing. See **Creating Billing Entries using Service Desk** *(page 14)*. If displayed, additional fields related to **Service Desk** display on the **Notes tab** *(page 24)*.

Date/Time Fields

- **Created** - The date/time the ticket was created.
- **Last Edit** - The date/time the ticket was last edited.
- **Closed** - The date/time the ticket was closed by setting the **Status** field to **Closed**.
- **Due** - The due date/time for the ticket. A value only displays in this field if a goal procedure is linked to the desk definition. The due date can be modified by the `pauseTicketGoal()` (page 94), `resumeTicketGoal()` (page 94) and `setTicketGoalTime()` (page 95) service desk procedure commands.
- **Promised** - The date/time the customer has been promised resolution of the ticket. This can be manually changed by the call taker.
- **Escalation** - The date/time an escalation counter was last set for this ticket and the amount of time before the escalation procedure is triggered.
- **Work Performed On** - The date and time work was performed on the ticket. This value is independent of the entries entered for hours worked.

Desk Definition Properties

The following fields classify a ticket by values defined using the **Properties tab** *(page 36)* in **Desk Definition**.

- **Status** - Status codes represent the state a ticket is in, regardless of any other classification.
- **Priority** - Prioritizes service requests.
- **Stage** - The stage the ticket is in.
- **Category** - This section, when collapsed, only displays the lowest level category used to classify the type of service requested. When expanded this section displays up to 5 levels of categories, if these levels have been defined in the service desk definition. Use these levels to select a different category for the ticket. You must select a parent level to access a lower-level category. See **Category tab** *(page 37)* for more information.
- **Severity** - Classifies how critical a service request is to an organization's operations.
- **Policy** - The **policy** *(page 4)* in effect for this ticket. Policies are assigned by the desk definition used to create the ticket.
- **Custom Fields** - Additional custom fields may display and are available for data entry, depending on the requirements of your service organization.

Description

- Use the edit toolbar to add images and special formatting to the text. *Images must be uploaded rather than copied and pasted in.* Toggle full screen mode to edit or view the description full screen.

- - Hyperlink selected text. You may need to reset links copied and pasted from another source.
- - Insert a table.
- - Insert a horizontal line as a percentage of the width, or set a fixed width in pixels.
- - Indent text.
- - Outdent text.
- - Remove formatting.
- - Insert a symbol.
- - Insert an emoticon.
- - Preview the display of text and images.
- - Upload a file or image.
- - Set selected text to subscript.
- - Set selected text to superscript.
- - Toggle full screen mode for editing and viewing.

Resolution

- **Resolution** - Select a value that classifies how an issue was resolved.
- **Resolution Description** - Enter a description of how the issue was resolved. Use the edit toolbar buttons described for the **Description** field to add special formatting to the text.

Notes tab

Service Desk > Operations > Tickets > Add/Edit > Notes tab

- Displays only when editing the ticket.

The **Notes** tab enables you to enter notes and report hours worked for the entire ticket. Additional fields are used to classify the type of work performed or to submit the hours worked to **Service Billing** for billing purposes. When a note is added to a ticket, the ticket is reset to **Unread** for other users.

Add Note

- **Note Time** - Displays only if **Auto Save Clock** is unchecked in in the **General Info** *(page 34)* tab in **Desk Definition**. Specifies the date and time of a note for backdating purposes. If hours are reported when the note is added, then the hours are posted in the **Time Tracking** module using the date and time specified. Only supported using the `Fixed_Width_Tabbed` and `Fixed_Width_No_Tabs` editing template.
- **Hours Worked** - Enter the number of hours you've worked on this ticket. Typically when you enter hours worked, you also enter a note describing the work you've done. This field is hidden if the Service Desk > Desk Definition > **General** *(page 34)* > **Auto Save Clock** checkbox is checked.
- **Note Template** - Append a note template to the ticket notes. Note templates are predefined blocks of text. They are maintained using Service Desk > **Note Templates** *(page 48)*.
- **Add as Hidden** - If checked, do not notify the submitter by email and hide the note from machine users viewing the ticket using **Portal Access**. If unchecked, the added note is public for all users.
- **Suppress Notification** - If checked, suppresses email and message notification, both internal and external, including any other changes included with this note.

Service Billing Fields

- **Billable** - If checked, the entry is billable. If **Service Billing** is not installed, the **Billable** checkbox is for reference purposes only. If **Service Billing** is installed the entry is forwarded to **Service Billing**. If timesheets require approval, the timesheet containing this entry must be approved before the entry is forwarded to **Service Billing**.

The following fields reference **Service Desk** integration with **Service Billing**. See **Creating Billing**

Entries using Service Desk *(page 14)*.

- **Work Order Item** - The work order line to associate with the hours worked. Displays only if a work order is associated with the ticket on the **General** *(page 22)* tab of the ticket editor.
- **Activity Type** - Labor entries are grouped by **activity type** to analyze the cost and revenue of labor. The classification of activity types typically reflects the accounting requirements of a company. Labor entries are classified by both activity type and **resource type** *(page 127)*. Not editable if a detailed work order is selected.
- **Resource Type** - A **resource type** sets a default rate for a billable labor item or entry. Typically a resource type represents a skill performed by a staff member. A billing rate and standard cost is defined for each resource type required to perform the service. The rate can be overridden when selected. The classification of resource types typically reflects the production requirements of a company. Labor entries are classified by both resource type and by activity type. Not editable if a detailed work order is selected.
- **Rate** - The default billing rate for the selected resource type. Display only.
- **Override Rate** - A manually entered rate that overrides the default billing rate for a selected resource type. Does not display if a detailed work order is selected.
- **Show Note on Invoice** - If checked, the note is displayed on the printed invoice.

Note

- **Note** - Enter text in the note pane.
- **Toolbar** - Use the edit toolbar to add images and special formatting to the text. *Images must be uploaded rather than copied and pasted in.*

  - - Hyperlink selected text. You may need to reset links copied and pasted from another source.
  - - Insert a table.
  - - Insert a horizontal line as a percentage of the width, or set a fixed width in pixels.
  - - Indent text.
  - - Outdent text.
  - - Remove formatting.
  - - Insert a symbol.
  - - Insert an emoticon.
  - - Preview the display of text and images.
  - - Upload a file or image.
  - - Set selected text to subscript.
  - - Set selected text to superscript.
  - - Toggle full screen mode for editing and viewing.

(Note History)

- **(Edit)** - Double click a note to edit the text of the note.
- **Split** - Splits the current ticket and move all notes *after the selected note's date* and all current settings of the current ticket to a new ticket.
- **Delete** - Deletes a selected note. This option only displays if **Allow Deleting Notes** is unchecked in in the **General Info** *(page 34)* tab in **Desk Definition**.
- **Hidden** - Select a note and check it to mark the note as internal-only, meaning hidden from external users, such as submitters being notified by email, or machine users viewing the ticket using **Portal Access**. Uncheck the note to make it public to all users. Yellow is the default background color for hidden notes.
- **Task Notes** - The background colors of types of notes are set using **Global Settings** *(page 51)*.

Related Items tab

Service Desk > Operations > Tickets > Add/Edit > Related Times tab

Click the **Related Items** tab to link items to your ticket and view a list of existing linked items to your ticket. This includes other tickets as well as KB articles.

- You can *unlink* items using the Ticket > **Unlink Related Items** *(page 17)* action button.
- You can add tickets to the **Related Tickets** table without opening a ticket, by selecting two or more rows in the ticket table and clicking the Ticket > **Link** *(page 17)* button.
- A service desk procedure can automate the creation of related tickets using the **linkTickets()** *(page 93)* step. A service desk procedure can also return a list of related tickets using two property variables: `[$RelatedTicketsAsBulletList$]` and `[$RelatedTicketsAsString$]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables.

Tab Sections

- **(Related Items)** - The list of tickets and KB articles related to the current ticket. Usually blank when a ticket is first created.
- **Related Ticket Details** - Displays the related item contents in the bottom pane, including **Priority**, **Description** and **Notes**.

Actions

- **Link Related Item** - Click this icon to display a search window of all other tickets and KB articles you are authorized to view. You can add one item at a time to the **Related Items** list. When the search window opens you can click the column drop-down arrow to hide or show columns and filter the list of items in the search list.

Tasks tab

Service Desk > Operations > Tickets > Add/Edit > Tasks tab

The **Tasks** tab creates and maintains a list of tasks by ticket. **Tasks** organize a **Service Desk** ticket into a series of steps. Each task can be assigned a different assignee and tracked individually for completion. You can add notes and hours worked by individual task. All tasks for all tickets can be listed, sorted, filtered, searched and viewed using the **Tasks Associated with Tickets** *(page 29)* page.

> **Note:** *See* **Task Information** *(page 14).*

Actions

- **Add Task** - Creates a new task for ticket. See **Adding / Editing a Task.** *(page 30)*
- **View** - Displays a task without allowing any edit changes. Selected users may only have access to viewing a task, depending on their role.
- **Edit** - Edits a selected task.
- **Delete** - Deletes a selected task.
- **Add Note** - Enters notes and reports hours worked for a selected tasks. See **The Add Note Dialog** *(page 27)*.

Table Columns

- **Task No** - A number assigned to the task when you add the task.
- **Description** - A one line description of the task.
- **Estimated** - The number of hours estimated to complete the task.
- **Status** - The **status** *(page 38)* of the task.
- **Percent Complete** - The percent complete for the task.
- **Assignee** - The VSA assigned to the task.

The Add Note Dialog

Service Desk > Tickets > New > Add Note

- You must select an *existing* ticket to enable the Add Note button in the Tickets table.

Service Desk > Organization Tickets > Tickets tab > Add Note

Service Desk > Tasks Associated with Tickets > Add Note

Service Desk > Tickets > Task tab > Add Note

You can *enter notes and report hours worked* for a **selected** ticket, without necessarily having to open that ticket. A similar add note form displays for adding a note to the entire ticket or to a specific task within a ticket.

Note: The ticket layout described below is for the standard `Fixed_Width_Tabbed` **editing template** *(page 125)* used to create a ticket based on the `ITIL Incident` **definition template** *(page 44)*. You may see a different order and selection of fields if a different editing template and desk definition is used.

Task Enabled Fields

The following field displays if **Tasks** *(page 14)* are enabled for the desk definition.

- **Task** - Selects or displays the associated task, if applicable.

The following fields only display if a **Task** is selected.

- **Status** - The **status** *(page 38)* of the task.
- **% Complete** - The percent complete for the task.

Add Note Fields

- **Date Time Started** - Displays only if **Auto Save Clock** is unchecked in in the **General Info** *(page 34)* tab in **Desk Definition**. Specifies the date and time of a note for backdating purposes. If hours are reported when the note is added, then the hours are posted in the **Time Tracking** module using the date and time specified. Only supported using the `Fixed_Width_Tabbed` and `Fixed_Width_No_Tabs` editing template.
- **Hours Worked** - Enter the number of hours you've worked on this ticket. Typically when you enter hours worked, you also enter a note describing the work you've done. This field is hidden if the Service Desk > Desk Definition > **General** *(page 34)* > **Auto Save Clock** checkbox is checked.
- **Note Template** - Append a note template to the ticket notes. Note templates are predefined blocks of text. They are maintained using Service Desk > **Note Templates** *(page 48)*.
- **Add as Hidden** - If checked, do not notify the submitter by email and hide the note from machine users viewing the ticket using **Live Connect**. If unchecked, the added note is public for all users.
- **Suppress Notification** - If checked, suppresses email and message notification, both internal and external, including any other changes included with this note.

Service Billing

The following fields only display if a desk definition is integrated with **Service Billing**. See **Creating Billing Entries using Service Desk** *(page 14)*.

- **Work Order** - Display only. Displays only if a work order is associated with the ticket on the General tab of the ticket editor.
- **Work Order Item** - The work order line to associate with the hours worked. Displays only if a work order is associated with the ticket on the **General** *(page 22)* tab of the ticket editor.
- **Activity Type** - Labor entries are grouped by **activity type** to analyze the cost and revenue of labor. The classification of activity types typically reflects the accounting requirements of a company. Labor entries are classified by both activity type and **resource type** *(page 127)*. Not editable if a detailed work order is selected.

- **Resource Type** - A **resource type** sets a default rate for a billable labor item or entry. Typically a resource type represents a skill performed by a staff member. A billing rate and standard cost is defined for each resource type required to perform the service. The rate can be overridden when selected. The classification of resource types typically reflects the production requirements of a company. Labor entries are classified by both resource type and by activity type. Not editable if a detailed work order is selected.
- **Rate** - The default billing rate for the selected resource type. Display only.
- **Override Rate** - A manually entered rate that overrides the default billing rate for a selected resource type. Does not display if a detailed work order is selected.
- **Show Note on Invoice** - If checked, the note is displayed on the printed invoice.

Billing

- **Billable** - If checked and the time entered is associated with a work order line, the hours worked are billable.

> **Note:** The **Billable** checkbox is for reference purposes only in timesheets, if **Service Billing** is not installed.

Notes

- **Notes** - Enter text in the note pane.
- **Toolbar** - Use the edit toolbar to add images and special formatting to the text. *Images must be uploaded rather than copied and pasted in.*

  - - Hyperlink selected text. You may need to reset links copied and pasted from another source.
  - - Insert a table.
  - - Insert a horizontal line as a percentage of the width, or set a fixed width in pixels.
  - - Indent text.
  - - Outdent text.
  - - Remove formatting.
  - - Insert a symbol.
  - - Insert an emoticon.
  - - Preview the display of text and images.
  - - Upload a file or image.
  - - Set selected text to subscript.
  - - Set selected text to superscript.
  - - Toggle full screen mode for editing and viewing.

Organization Tickets

Service Desk > Operations > Organization Tickets

The **Organization Tickets** page displays all tickets associated with a selected organization.

Group by Columns

Right-clicking on a column heading in the ticket table of the **Organization Tickets** page displays two options not available on the **Tickets** page.

- **Group by this field** - Selects the "group by" column used to display tickets in the ticket table.

- **Show in Groups** - If checked, displays tickets in groups using the selected "group by" column.

Actions

- **Search** - Enter a search string in the **Search** edit box. Rows containing the search string in any of the columns in the upper panel are displayed: **ID**, **Organization**, **Contact Name**, **Phone Number**, **Email Address**, **Tickets**
- **View** - Filters the list of organizations displayed in the upper panel. Similar to **named filters** *(page 20)* in the **Tickets** table.

The details of a selected organization are displayed using the following tabs:

- **General** - Displays general information about an organization.
- **Custom Fields** - Displays **custom fields** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#7184.htm)* and their values defined for an organization.
- **Machine Groups** - Displays the machine groups of an organization.
- **Departments** - Displays the departments of an organization.
- **Staff** - Displays the staff members of an organization.
- **Secure Data** - Stores data that remains encrypted in the database. For example, you could store the password necessary to connect to a customer's remote website.

> **Note:** Secure data entered here is the same data displayed in **Secured Data** *(http://help.kaseya.com/webhelp/EN/KSB/9030000/index.asp#7116.htm)* in **Service Billing** and in the **Service Desk** ticket editor when the hyperlink of an organization ID is selected on the **General** tab.

- **Tickets** - Lists and accesses tickets associated with an organization. See **Action Buttons** *(page 17)* for more information about each of these buttons.
  - **New** - Creates a new ticket.
  - **View** - Displays a ticket without allowing any edit changes. Selected users may only have access to view a ticket using this option. You can select multiple tickets and page from one ticket to the next in **View** mode.
  - **Edit** - Edits a selected ticket.
  - **Add Note** - Enters notes and reports hours worked for a selected ticket. See **The Add Note Dialog** *(page 27)*.

Tasks Associated With Tickets

Service Desk > Operations > Tasks Associated With Tickets

The **Task Associated with Tickets** page lists all tasks you are authorized to see, displayed as a list of rows in a table. Tasks are identified by ticket and task number, separated by a period. For example `IN006178.3` indicates the third task in ticket `IN006178`. See **Task Information** *(page 14)*.

Actions

- **New** - Creates a new task for ticket. See **Adding / Editing a Task.** *(page 30)*
- **Add Note** - Enters notes and reports hours worked for a selected tasks. See **The Add Note Dialog** *(page 27)*.
- **View** - Displays a task without allowing any edit changes. Selected users may only have access to view a task, depending on their role. You can select multiple tasks and page from one task to the next in **View** mode.
- **Edit** - Edits a selected task.

Adding / Editing a Task

Service Desk > Tickets > Add or Edit > Tasks > Add or Edit
Service Desk > Tasks Associated With Tickets > Add or Edit

Click **Add** or **Edit** to specify a task.

- **Description** - Enter a one line description of the task.
- **Assignee** - The VSA user assigned this task.
- **Estimated Hours** - The estimated hours to complete this task.
- **Projected Date** - The expected date to complete this task.

Archived Tickets

Service Desk > Operations > Archived Tickets

Archived Tickets are tickets that have been archived. Archived tickets can be displayed but not edited. They can also be unarchived so that you can continue editing them. The layout and navigation of an archived ticket is exactly the same as a ticket in the **Tickets** *(page 15)* table.

> **Note:** Tickets entering the *End* stage, which by default is called `Closed`, are automatically archived after a specified number of days, using the Auto Archive Tickets (days) setting in the Desk Definition > **General Info tab** *(page 34)*.

Select a single row before clicking either of these action buttons.

- **View** - Displays a ticket without allowing any edit changes.
- **UnArchive** - Returns selected archived tickets to the **Tickets** *(page 15)* table.
- **Print Ticket** - Prints a single ticket in PDF, Excel, or HTML output, with or without notes.

Knowledge Base

Service Desk > Operations > Knowledge Base

The **Knowledge Base** table comprises all *KB articles* you have access to view. KB articles represent known solutions to issues that are reported by your customers.

- Search by keyword across multiple KB articles, similar to **searching for tickets** *(page 19)* in the Tickets table.

  > **Note: Search All** *(page 31)* enables you to search by keyword across *all tickets and all KB articles*.

- Associate multiple tickets with the same KB article, enabling you to *manage tickets using a shared problem/solution business process.* If a problem is shared by many customers but the solution is still in development, once you have the solution, you can contact each customer affected by the problem by identifying all the customer tickets related to the KB article.
- **Create a KB article by copying it from a ticket** *(page 18)*. This is the fastest way to create a KB article because the originating ticket often contains much of the detailed information required by the KB article.
- Create a KB article manually, in the same way you create a ticket.
- Ensure knowledge base articles display in **Live Connect** and **Portal Access**. See **Integrating Service Desk, Live Connect, Portal Access** *(page 9)*.

Knowledge Base Table

All KB articles you are authorized to see are displayed as a list of rows in a table. The following actions

are provided to manage KB articles. In some cases you have to select one or more rows to enable an action button.

- **New** - Creates a new KB article. Similar to **adding / editing a ticket** *(page 21)*.
- **View** - Displays a KB article without allowing any edit changes. Selected users may only have access to view a KB article, depending on their role.
- **Edit** - Edits a selected KB article. Similar to **adding / editing a ticket** *(page 21)*.
- **Delete** - Deletes one or more selected KB articles.
- **Copy an Article** - Copies a selected KB article to another KB article. See **Copying Tickets and KB Articles** *(page 18)*.
- **Link** - Links two or more selected items. Linked items display in the **Related Tickets** tab of a ticket or KB articles. Use **Search All** *(page 31)* to select and link items from different desks.
- **Unlink Related Items** - Unlinks items related to a single selected item.
- **Unlock Articles** - Unlocks selected items. An item locks each time you edit it, preventing other users from editing it at the same time. If a user has left an item open and you need to maintain it, you can unlock the item to gain access to it. Doing so prevents the user who originally opened the item from saving his or her changes.
- **Re-Open** - Changes an End-stage (for example, `Closed`) article to a Begin-stage (for example, `New`) article.
- **Print an Article** - Prints a single KB article in PDF, Excel, or HTML output, with or without notes.

Knowledge Base Definitions

Service Desk comes with a knowledge base definition already created for you called `KnowledgeBase`. You can begin to add KB articles immediately or create a new desk that matches your preferences. See **Setup Knowledge Base** *(page 47)*.

Knowledge Base Templates

Two types of templates apply to knowledge base desks and KB articles.

- **Definition Template** - `Blank KnowledgeBase.xml` - *This template is mandatory to create a new KB article definition.* This is the template the `KnowledgeBase` definition is created from. A knowledge base definition supports the creation of KB articles rather than tickets. Once created, this definition can only be selected using the **Knowledge Base** function. It isn't selectable using the **Tickets** function. The word `Blank` in the xml filename indicates that standard field values are not populated when you create a definition based on this template. See Service Desk > **Desk Definition** *(page 32)* for detailed instructions about editing a definition.
- **Editing Template** - `Knowledge Base Article.xml` - *An editing template determines the layout of fields in the ticket editor or KB article editor.* The default `KnowledgeBase` definition uses an **editing template** *(page 125)* called `Knowledge Base Article.xml` that hides many of the standard ticket fields that are visible when you edit a ticket. The assumption is that a KB article doesn't require users to maintain all these extra fields. The editing template used to edit a ticket or KB article is optional. For example, you can use the `Default Ticket Entry.xml` to display a KB article editor that looks just like the standard ticket editor. Editing templates are assigned to desks using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

Search All

Service Desk > Operations > Search All

The **Search All** page displays *all tickets* and all *KB articles* a VSA user is authorized to see in a table format. You can use this specialized table to search for text in **Description**, **Notes**, or **Summary** fields, whether that text or value is used in a ticket or KB article. You can also use the **Search All** table to link any combination of tickets or KB articles.

Actions

The following actions are provided to manage items in the **Search All** table. In some cases you have to select one or more rows to enable an action buttons. The functionality is similar to the **Tickets** *(page 15)* page for tickets and the **Knowledge Base** *(page 30)* page for knowledge base articles.

- **New** - Creates a new ticket.
- **View** - Displays an item without allowing any edit changes. You can select multiple items and page from one ticket to the next in **View** mode.
- **Edit** - Edits a selected ticket or KB article.
- **Delete** - Deletes selected tickets.
- **Group Updates** - Updates multiple tickets at the same time. For example, you can set a group of tickets to `Closed`. All tickets must be members of the same service desk.
- **Merge** - Merges multiple tickets into a single ticket. Tickets can be members of different service desks.
- **Copy** - Copies a selected ticket or KB article to either a ticket or a KB article. See **Copying Tickets and KB Articles** *(page 18)*.
- **Link** - Links two or more selected items. Linked items display in the **Related Tickets** tab of a ticket or KB articles.
- **Unlink Related Items** - Unlinks items related to a single selected item.
- **Unlock** - Unlocks selected items. An item locks each time you edit it, preventing other users from editing it at the same time. If a user has left an item open and you need to maintain it, you can unlock the item to gain access to it. Doing so prevents the user who originally opened the item from saving his or her changes.
- **Print Ticket** - Prints a single ticket or KB article in PDF, Excel, or HTML output, with or without notes.
- **Refresh** - Refreshes the page.

Desk Definition

Service Desk > Desk Definition

Desk Definition determines how tickets and KB articles are managed and tracked. When created, tickets are populated with properties and values from a desk definition.

Actions

The following action buttons are provided:

- **New** - Creates the general settings of a new desk definition.
- **Edit** - Edits the general settings of a selected desk definition.
- **Rename** - Renames a selected desk definition.
- **Delete** - Deletes a selected desk definition.
- **Set Default** - Sets a selected desk definition as the default. Inbound email and alarms create tickets using the default desk definition, if no other methods override this default. See Desk Definition > New or Edit > **Standard Field Defaults** *(page 35)* > Email.
- **Import** - Imports a desk definition. When you import a desk definition, if that desk definition includes field level permissions for a role that does not exist on the target system, the role is created. No users are associated with this new role, but the correct role types are associated with it.

> **Note:** Desk definitions, message templates and tickets can be imported and exported using the System > **Import Center** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6963.htm)*. A desk definition can be exported as a template or as desk. Exporting a desk definition can include all the procedures that support it. Exporting a ticket includes the desk that supports it.

- **Export** - Exports a selected desk definition as an XML file.

Configuring a Desk Definition

Click any of the following topics for details on how to configure a desk definition.

General Settings

Service Desk > Desk Definition > New or Edit

General settings for existing desk definitions are displayed as a series of columns in the upper panel of the **Desk Definition** page.

Creating a New Desk Definition

1. Click the **New** button at the top of the Service Desk > **Desk Definition** page. The **Add Service Desk Definition** dialog displays.
2. Select one of two desk template options:
   - **Use selected template** - Displays the **Definition Template** drop-down list so that a template can be selected to create a desk definition.
   - **Import template from file** - Displays the **Upload an XML file for Import** field so that a new desk definition *template* can be imported. After the import, the template displays with the other templates and can be used to create a new desk definition.
3. **Definition Template** - Select a desk definition template.
4. Enter a **Name** and **Description** for your new desk definition.

5. Enter a **Prefix** to assign to tickets IDs based on your new desk definition. You can include hyphen (-) and underscore (_) characters.
6. Select an **Editing Template** *(page 125)*. An editing template sets the default field permissions for editing a ticket or KB article.
7. Select the **Desk Administrator** who should be notified if there is a problem with the desk definition. For reference purposes only.
8. Optionally select a default **Policy** *(page 4)*.
9. Optionally select an **email reader** *(page 57)* to associate with the desk definition in the **Email** field. Emails received by this reader are converted into tickets based on this service desk by default. You can override this default association using a **Ticket Request Mapping** procedure.
10. Click the **Save** button.

Editing a Desk Definition

1. Select the row of a desk definition in the upper pane.
2. Click the **Edit** button at the top of the **Desk Definition** page. To edit an existing desk definition, select a desk definition, then click **Edit** or double-click the desk definition to open the desk immediately.
3. When editing a desk definition general settings are displayed in a dialog with three tabs.
   - **General Info tab** *(page 34)*
   - **Standard Field Defaults tab** *(page 35)*
   - **Procedures tab** *(page 36)*

General Info tab

Service Desk > Desk Definition > Edit > General Info tab

Options on the **General Info** tab determine how the ticket editor or KB article editor behaves globally. In some cases these settings can be overridden by **Role Preferences** *(page 52)* or **User Preferences** *(page 55)*.

> **Note:** See **General Settings** *(page 33)* for *adding* a new desk definition.

- **Name** - The name of the service desk. Display only in Edit mode.
- **Description** - A brief description of the service desk.
- **Prefix** - The prefix is added to each ticket created based on this service desk. Display only in Edit mode. Hyphen (-) and underscore (_) characters are allowed.
- **Definition Template** - The template used to create the desk definition. Display only in Edit mode.
- **Editing Template** - The default **editing template** *(page 125)* used to maintain tickets based on this service desk. Can be overridden by role using Desk Definition > Access > **Roles** *(page 41)* or by **Role Preferences** *(page 52)*.
- **Desk Administrator** - The person responsible for maintaining the service desk. This user is sent messages for procedures that fail to execute properly.
- **Default Service Desk Definition** - If checked, sets this desk definition as the default. Inbound email and alarms create tickets using the default desk definition, if no other method specifies the desk definition to use. Can also be set using the **Set Default** *(page 32)* action button.
- **Display Machine Info** - If checked, displays machine info in the ticket editor.
- **Require Machine Info** - If checked, requires a machine ID to be entered in the ticket editor.
- **Display Organization Info** - If checked, displays the organization name in the ticket editor.
- **Require Organization Info** - If checked, requires the organization to be entered in the ticket editor.
- **Require Time** - If checked, requires the time spent working on a ticket to be entered each time the ticket is modified in the ticket editor.
- **Auto Save Clock** - If checked, automatically saves the time spent editing a ticket. Overrides the **Require Time** checkbox and **Show Session Timer** checkbox and hides the **Hours Worked** field and **Timer**

buttons in the ticket editor, regardless of **role** *(page 41)* permissions settings. If unchecked, the **Note Time** field displays in the **Add Note** *(page 27)* dialog and when adding a note inside the ticket editor.

- **Enable Incident Notes Panes** - If checked, an extra pane with two tabs displays below the **tickets table** *(page 15)* showing the selected ticket's **Description** and **Notes**.
- **Enable Service Billing Integration** - if checked, billable entries can be created using the **Service Billing** module and billed in **Service Desk**. See **Creating Billing Entries using Service Desk** *(page 14)*.
- **Enable Session Timers** - If checked and **Auto Save Clock** is not checked, timer buttons display in the **ticket editor** *(page 21)*. See **Time Tracking and Service Desk** *(page 11)*.
- **Enable Task Information** - If checked, the **Task** tab displays in the ticket editor. See **Task Information** *(page 14)*.
- **Allow Deleting Notes** - If checked, notes can be deleted in the ticket editor. This setting can be overridden in **Role Preferences - General tab - Default Columns tab** *(page 53)*.
- **Log Ticket Changes**
  - `Log Changes` - Generate a note when a field is changed that is visible to users.
  - `Log Changes in Hidden Notes` - Generate a note when a field is changed that is visible to service desk users but hidden from submitters submitting tickets by email and machine users using **Portal Access** *(page 9)*.
  - `Do Not Log Changes`
- **Procedure Variable Time Zone** - Sets the time zone used by date/time procedure variables such as `[$CreateDateTime$]`, `[$ClosedDateTime$]`, and `[$EscalationDateTime$]`. This time zone also applies to procedure-generated date/time *text* embedded in the **Note** column of the notes history of the ticket and date/time *text* embedded in messages and emails generated by procedures. (The dates/times displayed in the **Time** column of the notes history continue to use the time zone specified for the VSA user in System > **Preferences** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#503.htm)*.)
- **Email Display Name** - Enter the display name of the FROM address sent with outgoing emails from **Service Desk**.

> **Note:** The name should be a string of letters unlikely to be found in the subject text or body text of emails. It is used as a delimiter when an email is sent in reply to an outgoing **Service Desk** email.

- **Auto Archive Tickets (days)** - Specifies the number of days to wait after a ticket is assigned the *End* stage, which by default is called the `Closed` stage.
- **Only Masters see tickets that are not assigned to an organization** - Tickets can be created without assigning an organization to the ticket. If checked, only **master users** *(page 5)* see unassigned tickets that are not assigned to an organization. If blank, all users can see a ticket unassigned to an organization.

Standard Field Defaults tab

Service Desk > Desk Definition > New or Edit > Standard Field Defaults tab

The **Standard Field Defaults** tab defines default settings for any new ticket created based on this desk definition. If you create a new value for a standard field that you want to serve as the default value, return to this tab and select it.

Select the default values for the standard fields of a ticket based on the selected desk definition.

- **Status** - The default **status** *(page 36)*.
- **Priority** - The default **priority** *(page 37)*.
- **Category** - The default **category** *(page 37)*.
- **Severity** - The default **severity** *(page 37)*.
- **Resolution** - The default **resolution** *(page 37)* type.
- **Policy** - The default **policy** *(page 40)*.

- **Email** - The **email reader** *(page 57)* associated with a service desk. Emails received by this reader are converted into tickets based on this service desk. Overrides the Default Desk Definition setting in the Desk Definition > **General Info** *(page 34)* tab. Can be overridden using a **Ticket Request Mapping** *(page 56)* procedure.

Procedures tab

Service Desk > Desk Definition > New or Edit > Procedures tab

These procedures run immediately after a new ticket is created or are scheduled to be run a specified time after the ticket is created. If you create a new procedure that you want to run or schedule when a ticket is created, return to this tab to select it.

- **Ticket Change Procedure** - The default **Ticket Change** procedure. A **Ticket Change** *(page 76)* procedure runs each time a ticket is changed. Typically **Ticket Change** procedures notify the owner, assignee or submitter that a ticket has been updated.
- **Goal Procedure** - The default **Ticket Goals** procedure. A **Ticket Goals** *(page 77)* procedure runs after a specified time period, measured from the time a ticket was created. A ticket goal procedure usually records whether or not a ticket goal was met and sets the due date for closing the ticket.

  Note: The elapsed time counter for a ticket goal can be **paused** *(page 94)* and **resumed** *(page 94)* and the due date can be **set** *(page 95)* in a procedure.

  Note: Each stage of a ticket can also specify and run a *stage* (page 39) goal procedure.

- **Goal Time** - The number of time-periods to wait before running the goal procedure.
- **Goal Time Units** - The time-period unit of measure.

Properties tab

Service Desk > Desk Definition > Edit > Properties

The **Properties** tab displays tabs for each type of property that can be assigned to a desk definition.

- **Status tab** *(page 36)*
- **Priority tab** *(page 37)*
- **Category tab** *(page 37)*
- **Severity tab** *(page 37)*
- **Resolution tab** *(page 37)*
- **Custom Fields tab** *(page 38)*
- **Task Status Values tab** *(page 38)*

Status tab

Service Desk > Desk Definition > Edit > Properties > Status tab

Define an identifier and description for a status code. Status codes represent the state a ticket is in, regardless of any other classification. Possible types of status codes include: `Open`, `Closed`, `On Hold`, `Awaiting User Feedback`, `Under Investigation`, etc.

- **Parent ID** - The name of the desk definition.
- **ID** - The record identifier.
- **Description** - A brief description of this ID.

Priority tab

Service Desk > Desk Definition > Edit > Properties > Priority tab

Define an identifier and description for a priority code. Priority codes are defined for each desk definition and enable you to prioritize service requests.

- **Parent ID** - The name of the desk definition.
- **ID** - The record identifier.
- **Description** - A brief description of this ID.

Category tab

Service Desk > Desk Definition > Edit > Properties > Category tab

Categories can be defined for each desk definition and enable you to classify the type of service requested.

Categories can be defined five levels deep. The number of category levels defined by a service desk determines the number of categories that can be selected in the ticket editor. The **Category** column of the **Tickets table** *(page 15)* displays a selected category value in *parentheses ()*, followed by the hierarchy of categories used to select it. For example:

```
(Harddisk) - Equipment - Hardware Error - Harddisk
```

Configuration

Define categories and descriptions at each of the five levels. A lower level category requires selecting a parent directory before you add or edit a category at that lower level.

- **Service Desk ID** - The name of the desk definition.
- **Categories** - For each of the five levels you can:
  - **Select** - Select an existing category from the drop-down list.
  - **Add** - Add a category and description.
  - **Edit** - Edit a selected category and description.
  - **Delete** - Delete a selected category

Service Procedures

Categories can be tested or set in several service procedure commands. For setting or testing the value of a **Category** property, use two bar characters `||` to delimit levels of categories. For example: `Hardware||CPU||Overheating`

Severity tab

Service Desk > Desk Definition > Edit > Properties > Severity tab

Define an identifier and description for a severity code. Severity codes are defined for each desk definition and enable you to classify how critical a service request is to an organization's operations.

- **Parent ID** - The name of the desk definition.
- **ID** - The record identifier.
- **Description** - A brief description of this ID.

Resolution tab

Service Desk > Desk Definition > Edit > Properties > Resolution tab

Define an identifier and description for a resolution. Resolutions are defined for each desk definition and enable you to classify a service request by how the ticket was resolved.

- **Parent ID** - The name of the desk definition.
- **ID** - The record identifier.
- **Description** - A brief description of this ID.

Custom Fields tab

Service Desk > Desk Definition > Edit > Properties > Custom Fields tab

Add custom fields to this desk definition. When a ticket is created, these custom fields are available for data entry. A corresponding **property variable** *(page 68)* is also created for each custom field you define. You can order the display of custom fields in the ticket editor by using the **Move Up** and **Move Down** buttons. Custom reports do not support more than 40 custom fields.

- **ID** - The record identifier. Allows the hyphen (-) and underscore character (_).
- **Screen Caption** - The label for the custom field in windows.
- **Report Title** - The label for the custom field in reports.
- **Field Format** - The data type for the custom field. Includes:
  - `Amt` - Positive or negative number, up to 2 decimal places.
  - `Boolean` - Displays a **Default Value** checkbox. If checked, the default value is true. If unchecked, the default value is false. A boolean custom field remains `editable` if set to `required` in an **editing template** *(page 125)*.
  - `Date` - Any date.
  - `Datetime` - Any date and time.
  - `Integer` - Positive or negative number, whole number only.
  - `List` - A list of string values. If selected, these additional fields display.
    - ✓ **Value to Add to the List** - Enter a value to add to the list.
    - ✓ **Add Item** - Click to add an entered value to the list.
    - ✓ **Values Added to the List** - Displays a list of values.
    - ✓ **Delete Item** - Click to delete a selected value.
  - `MultiLine String` - Displays an edit box that supports multiple lines of text.
  - `Quantity` - Positive or negative number, up to 5 decimal places.
  - `Rate` - Positive or negative number, up to 5 decimal places.
  - `Rich Text` - Displays an edit box that supports rich text entry.
  - `String` - Any alphanumeric string value.
- **Default Value** - The default value for the custom field.

Task Status Values tab

Service Desk > Desk Definition > Edit > Properties > Task Status Values tab

Define an identifier and description for a task status code. Task status codes represent the state a task is in, regardless of any other classification. Possible types of task status codes include: `Open`, `Closed`, `On Hold`, `Awaiting User Feedback`, `Under Investigation`, etc.

- **Parent ID** - The name of the desk definition.
- **ID** - The record identifier.
- **Description** - A brief description of this ID.

Processing tab

Service Desk > Desk Definition > Edit > Processing

The **Processing** tab provides three tabs that display the *automated behavior of a ticket*, based on its desk definition.

- **View tab** *(page 39)*
- **Stage tab** *(page 39)*
- **Associated Policies tab** *(page 40)*

View tab

Service Desk > Desk Definition > Edit > Processing > View

The **View** tab provides a visual display of all stage sequences defined for the desk definition. Stages progress from left to right, branching to multiple stages as defined in the Processing > **Stage** *(page 39)* tab.

Stage tab

Service Desk > Desk Definition > Edit > Processing > Stage

Stages are designed to model the flow of work a business performs to process support issues. They are defined by desk definition. One stage is defined as a **Begin** stage and multiple stages can be defined as an **End** stage. There are usually several **Middle** stages.

Stages are *sequenced* by selecting one or more "to" stages. The sequence of stages is user-defined. It depends on the policies defined by the service organization and the judgments of individual users using the system. A sequence of stages is graphically represented by the **View** *(page 39)* tab within a desk definition.

An **event procedure** runs as soon as a ticket transitions from one stage to the next. A **scheduled procedure**, such as an escalation procedure, is triggered to run based on how long a ticket remains in a stage. For example, escalation procedures typically run when a ticket has remained too long in a stage, instead of being resolved and moved to another stage.

When a desk definition is created, stage sequences are populated by the *template* used to create the desk definition. Once a desk definition is created based on a template, these default stages can be modified to suit your business requirements.

Adding / Editing a Stage

Click **New** to add a new stage or select a stage and click **Edit** to edit a stage. You can also double-click a stage to edit it.

There are two tabs in a stage dialog.

- **General Info** *(page 39)*
- **Procedures** *(page 40)*

Processing > Stage > General Info tab

Enter the following attributes:

- **Name** - The name of the stage.
- **Description** - A brief description of this stage.
- **Stage Type**
  - `Begin` - The first stage in the life cycle of a response to an ticket. You can only have one `Begin` stage.

  - `Middle` - An interim stage.
  - `End` - The last stage. You can have multiple `End` stages.
- **Goal Procedure** - This **goal** *(page 77)* procedure runs once the **Goal Time** has been exceeded for this stage.
- **Goal Time** - The number of **Goal Time Units** to wait before running the goal procedure, based on setting a ticket to this stage. This default value can be overridden by a **Stage Entry** procedure.
- **Goal Time Units** - The time period unit of measure.
- **To Stage** - Select one or more stages that this stage can transition to.

> **Recommendation:** If you're creating an entirely new sequence of stages, create `Middle` stages from an `End` stage back to the `Begin` stage. This enables you select the appropriate **To Stages** at the same time you create each stage.

Processing > Stage > Procedures tab

- **Stage Entry** - This **stage entry** *(page 76)* procedure runs whenever a ticket enters this stage.
- **Escalation Procedure** - This **escalation** *(page 78)* procedure runs once the **Escalation Time** has been exceeded.
- **Escalation Time** - The number of **Escalation Time Units** to wait before running the **Escalation** *(page 78)* procedure, based on setting a ticket to this stage. This default value can be overridden by a **Stage Entry** procedure.
- **Escalation Time Units** - The time period unit of measure.
- **Stage Exit** - This **stage exit** *(page 76)* procedure runs whenever a ticket exits this stage.

Associated Policies tab

Service Desk > Desk Definition > Edit > Processing > Associated Policies

Policies can be optionally assigned to a ticket *when a new ticket is saved for the first time*. Assigning a policy determines the **hours of coverage** *(page 62)*, the manner of contact accepted by a service desk and the assignment of policy-specific variables. Policy variables can affect how a procedure is run on a ticket.

The **Associated Policies** tab provides a method of automatically assigning one of several policies to a *new* ticket. Each policy added to the **Associated Policies** tab of a desk definition is distinguished by its association with an *organization*, *organization type,* or *machine group and operating system combination.*

When a *new* ticket is associated with a machine ID, organization, contact or phone number *and saved for the first time*, a policy can be assigned to the new ticket by looking up *the first policy that matches the ticket associations* listed in the **Associated Policies** tab—*in alphanumeric order of the* **Policy** *name column*.

- Selecting a contact name or phone number in a ticket has the effect of selecting that contact's organization.
- Selecting an organization in a ticket has the effect of selecting an organization type, if the organization is a member of the organization type.
- Selecting a machine ID and operating system combination in a ticket has the effect of selecting a machine group, operating system, and organization.

> **Note:** A default policy can be assigned to a new ticket if no other method is used to assign a policy. The default policy is set in Service Desk > Desk Definition > Properties > General > **Standard Field Defaults** *(page 35)*.
>
> **Note:** See **Automated Processing Using Policies** *(page 4)*.

Enter the following attributes:

- **Parent ID** - Display only. The desk definition name selected in the *middle* pane.
- **Policy** - Select a policy in the list box. The list box is empty if no policies are defined, or if all policies are already added to the desk definition.
- **Type** - Specifies the record type you are associating with this policy.
  - `Machine Group` - Select a machine group using the **Machine Group** field.
  - `Organization` - Select an organization using the the **Org Name** field.
  - `Organization Type` - Select an organization type using the **Org Type** field.
- **Operating System** - Displays only if `Machine Group` is selected. Select one of several classes of operating systems.

Access tab

Service Desk > Desk Definition > Edit > Processing > Access

The **Access** tab specifies the roles and pools of users that has access to the desk definition and its tickets.

- **Roles tab** *(page 41)*
- **Pools tab** *(page 41)*

Roles tab

Service Desk > Desk Definition > Edit > Processing > Access > Roles
Service Desk > Desk Definition > Role Preferences > Service Desks tab

Field permissions are set by role. For VSA users using roles linked to the `Service Desk Technicians` role type, **field permissions** *determine what fields a user can view or edit within the ticket editor or KB article editor.* Typical field permissions include: `Editable`, `View Only`, `Hidden`, or `Required`. Default field permissions are set by **editing template** *(page 125)*.

Note: VSA users using a role linked to the `Service Desk Administrators` role type can see and work with any field in any ticket editor or KB article editor. `Master` role users also always have complete field permission access, regardless of roletype assignment.

Actions

- **Assign** - Associate a role with the desk definition and edit field permissions for each field.
- **Change** - Edit the field permissions for each field in the desk definition.
- **Remove** - Remove the association between the role and the desk definition.

See **Role Preferences - Service Desks tab** *(page 54)* for more information.

Pools tab

Service Desk > Desk Definition > Edit > Processing > Access > Pools

Users can be assigned to **pools** of users, regardless of the roles they belong to. Tickets can be assigned to pools using the **Assigned To** field in the editor, or by procedure using the `assignTicketProperty() > Pool` command. When a user in that pool opens the ticket, the ticket is automatically *reassigned* to the individual user.

Actions

- **New** - Adds a new pool of users.
  - **ID** - The identifier for the pool of users.

  - **Description** - A brief description of the pool.
  - **Filter by Role** - Filters the users in the left hand pane by user role.
  - Use the **>** button to added selected users in the left side list *to* the pool.
  - Use the **<** button to remove selected users in the right side list *from* the pool.
- **Edit** - Edits a selected pool of users.
- **Delete** - Deletes a pool of users.

Templates

The **Templates** folder contains functions that create and maintain templates for desk definitions, notes and messages.

- **Desk Templates** *(page 42)* - Provides sample **desk templates** to help you get started using **Service Desk** as quickly as possible.
- **Note Templates** *(page 48)* - Defines blocks of static text that can be appended to a note while editing a ticket.
- **Message Templates** *(page 49)* - Defines reusable, standard messages you can send as emails or display as messages in the **Inbox** of other users.

Desk Templates

Service Desk > Desk Templates

The **Desk Templates** page provides the following sample **desk templates** to help you get started using **Service Desk** as quickly as possible. Each desk template is configured to support one of several business processes required by IT organizations. You can modify any of these sample desk definitions and put them into production, or install them just to learn how to create your own desk definitions from scratch. Once installed, none of the changes you make will be overwritten by subsequent updates or hotfixes.

- **Basic Customer Service Desk** *(page 42)*
- **Automated Customer Service Desk** *(page 43)*
- **Knowledge Base** *(page 47)*
- **Incident Management Desk** *(page 44)*
- **Problem Management Desk** *(page 46)*
- **Change Request Desk** *(page 47)*

Setup Customer_SD_Basic

The `Customer_SD_Basic` is a basic ticket support desk with sample tickets. *This service desk requires no other configuration to get started.* If you need to get started as quickly as possible—without making use of **Service Desk**'s automation features—this is the one to use.

Configuration

1. **Create the Desk** - Click the **Install Desk** button for the `Basic Customer Service Desk` using Service Desk > **Desk Templates**.
2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - *This service desk is automatically associated with the* `SD User` *role.* If you assign users to a user role that uses the `Service Desk Technicians` role type, you

must associate this desk with that user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.

4. **Assign the Desk to Scopes** - Assign this desk to the *scopes* of users who use roles—such as `SD User`—that use the `Service Desk Technicians` role type.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.

5. **Verify Outbound Email is Enabled** - Ensure that *outbound* emails are enabled using System > **Outbound Email**.
6. **Create an Inbound Email Reader** - Create an email reader using the Service Desk > Incoming Email and Alarm Settings > **Readers (page 57) tab**. This enables tickets to be created when *inbound* emails are received. Ensure emails received by this reader are converted into tickets for this service desk by linking the email reader to the service desk using the Desk Definition > New or Edit > General > **Standard Field Defaults** *(page 35)* > **Email** field. You can override this default association using a **Ticket Request Mapping** *(page 77)* procedure.
7. **Activate Service Desk** - Alarm generated tickets can be configured to create tickets using this desk. Use Service Desk > Incoming Email and Alarm Settings > **General** *(page 56)* tab to specify `Customer_SD_Basic` as the service desk to use when tickets are created from alarms. Then use Service Desk > Global Settings to **activate** *(page 51)* **Service Desk**.

Setup Customer_SD_Automation

The `Customer_SD_Automation` service desk provides basic automation with a simplified workflow. Automation in this service desk demonstrates email and message notification to users and user pools when a ticket is created, starts a new stage or is escalated. **Minimal configuration is required to use this service desk.**

> **Note:** See **How Stages Work** *(page 2)* for a description of each stage in the `Customer_SD_Automation` service desk.

Configuration

1. **Create the Desk** - Click the **Install Desk** button for `Automated Customer Service Desk` using Service Desk > **Desk Template**.
2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - *This service desk is automatically associated with the* `SD User` *role.* If you assign users to a user role that uses the `Service Desk Technicians` role type, you must associate this desk with that user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.

4. **Assign the Desk to Scopes** - Assign this desk to the *scopes* of users who use roles—such as `SD User`—that use the `Service Desk Technicians` role type.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.

5. **Assign Users to Pools** - Assign users to the following user pools within the `Customer_SD_Automation` service desk. You assign users to user pools using Service Desk > Desk Definition > Access > **Pools** *(page 41)*.
   - `Tier1Support`

- Tier2Support

6. **Configure Procedure Variables** - Enter email addresses, as appropriate, in the value fields for the following procedure variables using Service Desk > **Procedure Variables** *(page 60)*.
   - Email_Tier1Support - These email addresses correspond to the same users added to the Tier1Support pool.
   - Email_Tier2Support - These email addresses correspond to the same users added to the Tier2Support pool.
   - Email_KaseyaServer - Enter a single "From" email address.

   Multiple email recipients should be separated by semi-colons, for example: jsmith@acme.com;fgorham@acme.com;twinters@consultants.com.
7. **Verify Outbound Email is Enabled** - Ensure that *outbound* emails are enabled using System > **Outbound Email**.
8. **Create an Inbound Email Reader** - Create an email reader using the Service Desk > Incoming Email and Alarm Settings > **Readers (page 57) tab**. This enables tickets to be created when *inbound* emails are received. Ensure emails received by this reader are converted into tickets for this service desk by linking the email reader to the service desk using the Desk Definition > New or Edit > General > **Standard Field Defaults** *(page 35)* > **Email** field. You can override this default association using a **Ticket Request Mapping** *(page 77)* procedure.
9. **Activate Service Desk** - Alarm generated tickets can be configured to create tickets using this desk. Use Service Desk > Incoming Email and Alarm Settings > **General** *(page 56)* tab to specify Customer_SD_Automation as the desk definition to use when tickets are created from alarms. Then use Service Desk > Global Settings to **activate** *(page 51)* **Service Desk**.
10. **Modify CS Message Templates** - You may wish to update the message templates used by the Customer_SD_Automation using Service Desk > **Message Templates** *(page 49)*. These message templates are all identified with a CS prefix.

Setup Incident Service Desk

The Incident service desk, based on ITIL guidelines, is a more advanced service desk. The **Incident** service desk demonstrates the benefits of *automating* the processing of support tickets by including sample stage entry procedures, escalation procedures, and goal procedures. This service desk integrates with pre-defined sample user roles, user pools, organization types, procedure variables, message templates, policies, coverage schedules, and holidays. **Additional configuration is required to use this service desk.**

> **Note:** *See* **Service Desk Priority Calculations** *(page 79)* for an explanation of how the **Incident** service desk sets priorities using a sub-procedure.

Prerequisites

You may or may not have the following data already defined in your VSA. If not, you'll have to create them.

- **Create a Review Team** - Identify, or if necessary create, a small group of users to act as your initial *"review team"* for the **Incident** sample service desk. You'll need to specify these names in certain steps below. After you complete the configuration, you can repeat the user-specific steps to add more users to your review team.
- **Create Organizations** - Identify, or if necessary create, a small set of organizations and agent machines within those organizations. You'll need at least 3 organizations, one for each of the 3 different organization types described below. This is necessary to demonstrate how to associate a ticket with an organization and organization type or with an agent machine. Organizations are maintained using System > **Manage**.
- **Create Contacts** - Identify, or if necessary create, at least one department within an organization and at least one staff member within that department. This is necessary to demonstrate how to

associate a ticket with the staff member of an organization. Contacts are maintained using System > **Manage**.

Configuration

1. **Create the Desk** - Click the **Install Desk** button for the `Incident Management Desk` using Service Desk > **Desk Templates**.
2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - *This service desk is automatically associated with the* `SD User` *role.* If you assign users to a user role that uses the `Service Desk Technicians` role type, you must associate this desk with that user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
4. **Assign the Desk to Scopes** - Assign this desk to the *scopes* of users who use roles—such as `SD User`—that use the `Service Desk Technicians` role type.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
5. **Assign Organization to Organization Types** - Assign customer organizations to one of the following organization types using System > Orgs/Groups/Depts > **Manage** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4017.htm)*.
   - `Gold_SLA`
   - `Silver_SLA`
   - `Bronze_SLA`
6. **Assign Users to Pools** - Assign *all* review team members to *each* of the following user pools within the `Incident` desk. This allows your review team to see each step in the life cycle of a ticket. Once your review team is familiar with the entire life cycle, you can assign different users to different user pools. You assign users to user pools using Service Desk > Desk Definition > Access > **Pools** *(page 41)*. *Your review team members can expect to be continually notified by VSA inbox messages about the status of each incident as it progresses through its life cycle.*
   - `Tier1Support`
   - `Tier2Support`
   - `Tier3Support`
   - `SupportManagement`
7. **Configure Procedure Variables** - Change the value of the `Email_ReviewTeam` procedure variable from its default generic value to a series of review team email addresses. Procedure variables are maintained using Service Desk > **Procedure Variables** *(page 60)*.

   For example, you could change the value to `jsmith@acme.com;fgorham@acme.com;twinters@consultants.com`. This procedure variable is referenced by other procedure variables. It sets the **To:** address of all **sendEmail()** commands in all procedures run by the `Incident` sample service desk. *Your review team members can expect to be continually notified by email about the status of each incident as it progresses through its life cycle.*

   Once your review team is familiar with the entire life cycle of an incident, you can re-assign different values to the following procedure variables:
   - `Email_Tier1Support` - These email addresses correspond to the same users added to the `Tier1Support` pool.
   - `Email_Tier2Support` - These email addresses correspond to the same users added to the `Tier2Support` pool.

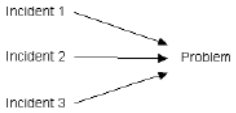
- `Email_Specialists` - These email addresses correspond to the same users added to the `Tier3Support` pool.
- `Email_SupportManagement` - These email addresses correspond to the same users added to the `SupportManager` pool.
- `Email_KaseyaServer` - Enter a single "From" email address.

8. **Verify Outbound Email is Enabled** - Ensure that *outbound* emails are enabled using System > **Outbound Email**.
9. **Create an Inbound Email Reader** - Create an email reader using the Service Desk > Incoming Email and Alarm Settings > **Readers (page 57) tab**. This enables tickets to be created when *inbound* emails are received. Ensure emails received by this reader are converted into tickets for this service desk by linking the email reader to the service desk using the Desk Definition > New or Edit > General > **Standard Field Defaults** *(page 35)* > **Email** field. You can override this default association using a **Ticket Request Mapping** *(page 77)* procedure.
10. **Activate Service Desk** - Alarm generated tickets can be configured to create tickets using this desk. Use Service Desk > Incoming Email and Alarm Settings > **General** *(page 56)* tab to specify `Incident` as the desk to use when tickets are created from alarms. Then use Service Desk > Global Settings > to **activate** *(page 51)* the **Service Desk**.
11. **Modify Message Templates** - You may wish to update the message templates used by the `Incident` using Service Desk > **Message Templates** *(page 49)*. These message templates do *not* have a `CS` prefix.

Setup Problem Service Desk

The `Problem` desk, based on ITIL guidelines, demonstrates the benefits of analyzing and resolving broadly-based product or service problems separately from immediate customer support issues. Problem tickets can be associated with multiple issues. When a solution is identified, customers with related issues can be notified.

When multiple organization/contacts have reported a similar type of incident, it can be easier to manage a response to those incidents by linking them to a single problem ticket. This can apply regardless of whether the incidents are questions/known errors, suggestions, or unknown errors. The problem ticket restates the reported incidents in a generic way that applies to anyone faced with the same set of conditions. For example the title of a problem ticket could be: `Cannot apply printer drivers to mapped drives.`



Configuration

1. **Create the Desk** - Click the **Install Desk** button for the `Problem Management Desk` using Service Desk > **Desk Templates**.
2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - *This service desk is automatically associated with the* `SD User` *role.* If you assign users to a user role that uses the `Service Desk Technicians` role type, you must associate this desk with that user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
4. **Assign the Desk to Scopes** - Assign this desk to the *scopes* of users who use roles—such as `SD User`—that use the `Service Desk Technicians` role type.

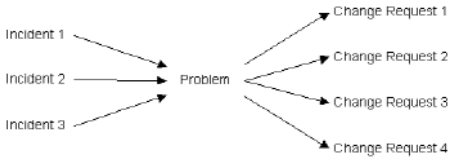
This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.

5. **Verify Outbound Email is Enabled** - Ensure that *outbound* emails are enabled using System > **Outbound Email**.

Setup Change Requests Service Desk

The `Change Request` desk is based on ITIL guidelines. Change requests have their own life-cycle, not necessarily related to customer support issues. Change requests are approved or rejected. If approved, then the change request is tracked through to successful completion. Change requests can be associated with multiple incidents reported by customers. When the change request is implemented, customers with related incidents can be notified.

A change request ticket describes a change to a product or service. They don't involve organization/contacts or problems directly. Instead they describe, in technical detail, exactly what aspect of the product or service has to be changed. Change requests can be created internally, or can be associated with a problem ticket. Frequently a single problem ticket can be the source of multiple change requests.



Configuration

1. **Create the Desk** - Click the **Install Desk** button for the `Change Request Desk` sample service desk using Service Desk > **Desk Templates**.
2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - *This service desk is not automatically associated with any user role.* If you assign users to a user role that uses the `Service Desk Technicians` role type, you must associate this desk with that user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
4. **Assign the Desk to Scopes** - Assign this desk to the *scopes* of users who use roles—such as `SD User`—that use the `Service Desk Technicians` role type.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
5. **Verify Outbound Email is Enabled** - Ensure that *outbound* emails are enabled using System > **Outbound Email**.

Setup Knowledge Base

The `Knowledge Base` desk, based on ITIL guidelines, serves as a repository of known errors and recommended solutions. The **Search All** *(page 31)* feature in Service Desk lets you perform a single search to find matching text in any service desk ticket and any KB article.

Configuration

1. **Create the Desk** - Click the **Install Desk** button for the `Knowledge Base` desk using Service Desk > **Desk Templates**.

2. **Assign Users to User Roles** - Assign users requiring access to the **Service Desk** module to user roles—such as `SD User` or `SD Admin`—that use the `Service Desk Administrators` or `Service Desk Technicians` role type.
3. **Associate the Desk with User Roles** - If users only need to have view-only access to KB articles that are in the `Published` stage, this step is not required for any user role. If you want non-service desk administrators to be able to create a new KB article and edit the KB article, but don't want those same users complete service desk administrator access, select or create a user role associated with the `Service Desk Technicians` role type. Then associate the knowledge base with the user role using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab. Then assign users to that user role. The `KB Admin` user role can be used for this purpose. The `KB Admin` is already associated with the `KnowledgeBase` desk. You only need to remove the `Service Desk Administrator` roletype from the `KB Admin` user role.

   This step is not necessary for users assigned to user roles—such as `SD Admin`—that use the `Service Desk Administrators` role type.
4. **Create Knowledge Base Articles** - See **Knowledge Base** *(page 30)* for instructions on *creating KB articles*.

   > **Note:** Without any further configuration, all users using non-administrator roles—such as `SD User` that use the `Service Desk Technicians` role type—can see any `KnowledgeBase` article set to the `Published` stage.
5. **Display Knowledge Base Articles in Live Connect** - See **Integrating Service Desk, Live Connect, and Portal Access** *(page 9)* for instructions on displaying KB articles in **Live Connect** and **Portal Access**.

Note Templates

Service Desk > Configure > Note Templates

The **Note Templates** page defines blocks of static text that can be appended to a note while editing a ticket. After creating the note template, you can select it from the ticket editor **Note Template** drop-down list in the **Add Note** section of the **Notes** tab, or when adding a note using the **Add Note** *(page 27)* dialog.

Actions

- **New** - Adds a new note template.
- **Edit** - Edits a selected note template.
- **Rename** - Renames a selected note template.
- **Delete** - Deletes a selected note template.

Adding / Editing a Note Template

- **Name** - The name of the note template.
- **Prompt** - A brief description of the note template.
- **Response** - Enter the block of static text that will be entered into a note when you select this note template in a ticket. Use the following toolbar buttons to add special formatting to the text:
  - - Hyperlink selected text. You may need to reset links copied and pasted from another source.
  - - Insert a table.
  - - Insert a horizontal line as a percentage of the width, or set a fixed width in pixels.
  - - Indent text.
  - - Outdent text.

- - Remove formatting.
- - Insert a symbol.
- - Insert an emoticon.
- - Preview the display of text and images.
- - Upload a file or image.
- - Set selected text to subscript.
- - Set selected text to superscript.
- - Toggle full screen mode for editing and viewing.

Note: You can not add an attachment to a note template.

Message Templates

Service Desk > Configure > Message Templates

The **Message Templates** page defines reusable, standard messages you can send as emails or display as messages in the **Inbox** of other users. These message templates can be selected when you specify a **sendEmail()** step or **sendMessage()** step within *any* service desk procedure. Message templates can also include ticket **property variables** *(page 68)* and **procedure variables** *(page 60)*.

Note: **Service Desk** installs pre-defined message templates. These pre-defined message templates are used by one or more pre-configured service desks and procedures installed using the **Desk Templates** *(page 42)* page. You can change their contents and your changes will be preserved. But if you rename or delete these pre-defined message templates they will be re-created during the next **maintenance cycle** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#248.htm)* of the VSA.

Actions

- **New** - Adds a new message template.
- **Edit** - Edits a selected message template.
- **Rename** - Renames a selected message template.
- **Delete** - Deletes a selected message template.
- **Refresh** - Refreshes the page.

Adding / Editing Message Templates

- **ID** - The name of the message template.
- **Description** - A brief description of this ID.

Note: Any value you enter in the following four fields can be overridden by a value in the corresponding fields of a **sendMessage()** or **sendEmail()** step in a procedure.

- **Subject** - The subject of the message template.
- **To** - The recipients of the message template. Multiple recipients are delimited by commas or semicolons.

Note: Any user name included in the **To** field of a message template is automatically converted into that user's corresponding email address. For example, if the user `jsmith` has a corresponding email address of `jsmith@acme.com`, then the same message template—with `jsmith` in the `To` field—can be used by a **sendMessage()** step to send a message to `jsmith` and also used by a **sendEmail()** step to send an email to `jsmith@acme.com`. Email addresses for each user are maintained using **System > Users**.

- **From** - The sender of the message template. The From address of an outbound email is specified using the following order of precedence:
  1. If there is a From address in the **sendEmail()** step of a procedure, then that address is used.
  2. Else the sendEmail() step uses the From address provided by a linked Service Desk > **Message Template**, if the link exists and a From address is specified.
  3. Else the sendEmail() step uses the **Reply Email Address** of the Service Desk > **Incoming Email and Alarm Settings** > email reader linked to the service desk. This link between the email reader and the service desk is set using the Service Desk > Desk Definition > Properties > General > **Standard Field Defaults** *(page 35)* > Email field.
  4. Else the **Default Sender Email** address set in System > **Outbound Email** is used.
- **Body** - Enter the body text of the message template. Use the following toolbar buttons to add special formatting to the text:
  - - Hyperlink selected text. You may need to reset links copied and pasted from another source.
  - - Insert a table.
  - - Insert a horizontal line as a percentage of the width, or set a fixed width in pixels.
  - - Indent text.
  - - Outdent text.
  - - Remove formatting.
  - Ω - Insert a symbol.
  - - Insert an emoticon.
  - - Preview the display of text and images.
  - - Upload a file or image.
  - - Set selected text to subscript.
  - - Set selected text to superscript.
  - - Toggle full screen mode for editing and viewing.

Common Configuration

Service Desk > Configure

The **Configure** folder contains a set of user defined support tables that *can be used by multiple desk definitions and their tickets.*

> **Note:** Many of the following configuration tables are populated when **Service Desk** is installed, or when you create a service desk using the **Desk Templates** *(page 42)* page.

- **Global Settings** *(page 51)* - Defines global settings that apply to the entire **Service Desk** module.
- **Role Preferences** *(page 52)* - Defines ticket editing preferences and field permissions by *user role*.
- **User Preferences** *(page 55)* - Defines ticket editing preferences and field permissions by *user*.
- **Incoming Email and Alarm Settings** *(page 56)* - Creates email readers used to retrieve email messages from email servers and convert them into tickets.
- **Procedure Variables** *(page 60)* - Defines procedure variables, including variables associated with service desk policies.
- **Policies** *(page 61)*, **Coverage Schedules** *(page 62)* and **Holidays** *(page 63)* - Defines service policies.

Global Settings

Service Desk > Common Configuration > Global Settings

The **Global Settings** page sets options that apply to the entire **Service Desk** module.

Actions

- **Edit** - Edits global settings for the **Service Desk** module.
  - **Activate Service Desk** - Check to *activate (page 51)* **Service Desk**. Uncheck to *deactivate* **Service Desk**.
  - Use the color picker to select the background color for each type of note.
    - ✓ **Hidden Note Color** - Hidden ticket notes do not notify the submitter by email and hide the note from machine users viewing the ticket using **Portal Access (Classic)** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#438.htm)* or Kaseya User Portal.
    - ✓ **System Note Color** - A note created by a system event, such as the start of a service procedure.
    - ✓ **Task Note Color** - A note created for a task within a ticket.
    - ✓ **Hidden Task Note Color** - Hidden task notes do not notify the submitter by email and hide the note from machine users viewing the ticket using **Portal Access**.
    - ✓ **Normal Note Color** - The color displayed by public notes for the entire ticket.

Activation

Service Desk > Common Configuration > Global Settings

Activation of **Service Desk** enables system events—such as alarms—to create *new* tickets in **Service Desk** instead of the legacy **Ticketing** module. Activation also causes **Live Connect**, **Portal Access**, **Monitoring** and other pages that show *existing* tickets to list **Service Desk** tickets instead of **Ticketing** module tickets. Other modules in the VSA are either integrated with the **Ticketing** module or the **Service Desk** module, but cannot be integrated with both modules at the same time.

Activation of **Service Desk** integration can be done at any time before or after **Service Desk** configuration. Typically **Service Desk** is configured before activation.

You must perform all three steps below to enable system events to create new tickets in Service Desk.

> **Note:** See the Incoming Email and Alarm Settings > **Readers tab** *(page 57)* to create tickets based on *inbound email*.

1. The **Service Desk** module must be **activated** *(page 51)* to process ticket requests from system events using Service Desk > Common Configuration > **Global Settings** *(page 51)*.
2. **Incoming Email and Alarm Settings** must be *enabled*.

   > **Note:** No email readers need be defined to create **Service Desk** tickets from alarms.

3. Specify the desk definition used to create the ticket, in any *one of the following three ways*:
   a. Optionally set the **Default** desk definition in Service Desk > **Desk Definition** *(page 32)*.
   b. Optionally set the **Alarm Default Service Desk** in Service Desk > Incoming Email and Alarm Settings. Has precedence over the **Default** desk definition.
   c. Optionally specify a unique **Ticket Request De-Dup** *(page 77)* or **Ticket Request Mapping** *(page 77)* procedure to process ticket requests from system events. Set using the Service Desk > Incoming Email and Alarm Settings > **General** *(page 56)* tab. Has precedence over the **Default** desk definition and the **Alarm Default Service Desk** setting. See the **testTicketProperty()** *(page 87)* command for details on testing the property values of ticket

requests, including the *source* system event of the ticket request. This IF command is only available in a **Ticket Request Mapping** procedure.

Deactivation

Once activated, **Service Desk** can be *deactivated*. Deactivation redirects **Live Connect**, **Monitoring** and **Alarms** to integrate with the **Ticketing** module instead of the **Service Desk** module. Use Service Desk > Global Settings to deactivate **Service Desk** integration.

Disabling the Ticketing Email Reader

After **Service Desk** activation, the **Ticketing** email reader will still convert inbound emails into **Ticketing** module tickets, so you may want to disable the **Ticketing** email reader manually before **Service Desk** is activated. If both the **Ticketing** and **Service Desk** email readers are to be run concurrently, they should poll different email servers.

Role Preferences

Service Desk > Configure > Role Preferences

The **Role Preferences** page defines ticket preferences by role. **Role Preferences** can be overridden by **User Preferences** *(page 55)*. **Role Preferences** includes:

- Email send delay
- Color coding of tickets in the tickets selector table.
- Columns displayed by default in the tickets selector table.
- Field permissions in the ticket editor by user role and desk definition.
- Ticket grid refresh rate.

Role Preferences tabs are organized as follows.

- **General tab** *(page 52)*
  - **Role Properties tab** *(page 52)*
  - **Default Columns tab** *(page 53)*
- **Service Desks tab** *(page 54)*

Role Preferences - General tab

Service Desk > Role Preferences > General tab

Click a row in the middle panel, then click **Edit** to display the **Change Ticket Preferences for Role** window. This dialog has two tabs.

- **Role Properties tab** *(page 52)*
- **Default Columns tab** *(page 53)*

General tab > Role Properties tab

General

- **Role Name** - The name of the role.
- **Preference Type** - If `Assigned`, this preference record has been edited. If `Default`, no edits have been made.
- **Email Send Delay (Mins)** - Pending email notification of selected tickets can be stopped using **Stop Notification** in **Tickets** *(page 15)*. This option is used in conjunction with **Email Send Delay** in **Role Preferences** *(page 52)* and **User Preferences** *(page 55)*. For example, if **Email Send Delay** is set to `5` minutes for the role you are using, then you have `5` minutes to cancel any emails that are pending

for a ticket using **Stop Notification**. Applies to ticket rows displaying a  icon in the icon column of the ticket table. **Stop Notification** only stops currently pending email.

- **Ticket Grid Refresh Interval** - Sets the interval for refreshing the **Tickets** table in minutes.

Color Preferences

Set the color codes for ticket rows displayed in tickets tables, based on a **Date Range** *benchmark*. Available benchmarks include:

- **Creation Date** - The creation date of the ticket.
- **Due Date** - The due date of the *current stage* of a ticket. The due date for the current stage of a ticket can only be used if a goal procedure is linked to that stage using the Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab.
- **Last Date Modified** - The date the ticket was last modified.
- **Last Public Update** - The last time a non-hidden note was added.
- **Promised Date** - The promise date of the ticket.

Ticket Age

For the selected date range benchmark you can select three colors:

- **Older than** - Enter the number of days *before* the selected date range benchmark to color code tickets. Enter a negative number to represent days *after* the select date range benchmark. Use the color picker to select a color. For example, you might want to color code the oldest tickets as *red*.
- **Between** - Use the color picker to select a color. For example, you might want to color intermediate tickets as *yellow*.
- **Newer than** - Enter the number of days *after* the selected date range benchmark to color code tickets. Enter a negative number to represent days *before* the select date range benchmark. Use the color picker to select a color. For example, you might want to color recent tickets as *green*.

General tab > Default Columns tab

Select the columns displayed in the ticket selector table and the order they are displayed.

1. Use `Ctrl+Click` or `Shift+Click` to select multiple items in the left-hand listbox.
2. Click the right arrow button to move them to the right hand listbox.
3. Re-order the display of columns in the right-hand listbox by clicking the up arrow or down arrow buttons.
4. Click **Save** to save your default column preferences.

> **Note:** Adhoc changes you make to column selections in the **Tickets** *(page 15)* table are always remembered, even between sessions. To reset column selections back to **Role Preferences** settings, click the **Defaults** button in System > **Preferences**. This resets *all* your user preferences throughout the VSA back to their default settings.

Ticket Grid Column Filters

- **Assignee Filter is selectable** - When checked (default setting), the filter option for the **Assignee** column in the **Tickets** table displays a picker list. If blank, the user must enter in text to filter the column.
- **Category Filter is selectable** - When checked (default setting), the filter option for the **Category** column in the **Tickets** table displays a picker list. If blank, the user must enter in text to filter the column.

Role Preferences - Service Desks tab

Service Desk > Role Preferences > Service Desks tab
Service Desk > Desk Definition > Edit > Processing > Access > Roles

Click a row in the middle panel, then click the **Service Desk** tab to assign, change or remove the association between a desk definition and a role.

Configuring this tab involves setting three types of options

- Setting the Editing Template

> **Note:** This preference can be overridden by the editing template selected in **User Preferences** *(page 55)*.

- Setting Enable/Disable Options
- Setting Field Permissions

Setting the Editing Template

An **editing template** serves three purposes:

1. The editing template defines the layout of the dialog used to edit a ticket or KB article.
2. An editing template may *mask* selected fields, even though the fields are defined by the desk definition. *Using an editing template to mask a field overrides whatever field permission is set for that field.*
3. The editing templates also sets default **field permissions** *(page 126)* for editing a ticket or KB article. Whether assigned by role or by user, you can override the default field permissions set by the editing template to suit your business requirements.

An editing template is applied to a combination of desk definition and user role (or machine role) using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab. An editing template can also be applied to a combination of desk definition and user using **User Preferences** *(page 55)*. User Preferences has precedence over Role Preferences. The default editing template for all roles and all users working with a service desk is specified in the Service Desk > Desk Definition > New or Edit > **General Info** *(page 34)* tab.

Setting Enable/Disable Options

You can can override the following Desk Definition > **General Info** *(page 34)* settings for any combination of role and desk definition.

- **Enable Session Timers** - If checked and **Auto Save Clock** is not checked, a Timer button displays in the **ticket editor** *(page 21)*. See **Time Tracking and Service Desk** *(page 11)*.
- **Show Task Information** - If checked, the **Task** tab displays in the ticket editor. See **Task Information** *(page 14)*.
- **Allow Deleting Notes** - If checked, notes can be deleted in the ticket editor.

Setting Field Permissions

Field permissions are set by role. For VSA users using roles linked to the `Service Desk Technicians` role type, **field permissions** *determine what fields a user can view or edit within the ticket editor or KB article editor.* Typical field permissions include: `Editable`, `View Only`, `Hidden`, or `Required`. Default field permissions are set by **editing template** *(page 125)*.

> **Note:** VSA users using a role linked to the `Service Desk Administrators` role type can see and work with any field in any ticket editor or KB article editor. `Master` role users also always have complete field permission access, regardless of roletype assignment.

When an editing template is selected using this tab, the fields defined for this desk definition display in the dialog, showing the default permission set by the editing template. You may override the default for

this combination of role and desk definition.

Standard Fields and Custom Fields

Standard Fields *(page 68)* refer to built-in fields provided by a service desk. They display as columns in the ticket table and in **field permission** *(page 54)* dialogs. A standard field is not necessarily available for all combinations of desk definition and user role. Nor does the same standard field necessarily display in both the ticket table and in field permission dialogs. **Custom fields** *(page 38)*—if defined for a service desk—also display as columns in the ticket table and in field permission dialogs.

User Preferences

Service Desk > Configure > User Preferences

The **User Preferences** page defines user-specific preferences that override **Role Preferences** *(page 52)*. Each user has access to their own **User Preferences** page. **User Preferences** include:

- Email send delay
- Color coding of tickets in the tickets selector table.
- The user's preferred ticket editing template, one for each service desk.

Preferences are set using two tabs on this page:

- General tab
- Service Desks tab

General tab

Click **Edit** to edit General tab information.

General

- **User Name** - The currently logged in VSA user.
- **Preference Type** - If `Assigned`, this preference record has been edited. If `Default`, no edits have been made.
- **Email Send Delay (mins)** - Pending email notification of selected tickets can be stopped using **Stop Notification** in **Tickets** *(page 15)*. This option is used in conjunction with **Email Send Delay** in **Role Preferences** *(page 52)* and **User Preferences** *(page 55)*. For example, if **Email Send Delay** is set to `5` minutes for the role you are using, then you have `5` minutes to cancel any emails that are pending for a ticket using **Stop Notification**. Applies to ticket rows displaying a  icon in the icon column of the ticket table. **Stop Notification** only stops currently pending email.
- **Show Incident Notes Panes** - If checked, an extra pane with two tabs displays below the **tickets table** *(page 15)* showing the selected ticket's **Description** and **Notes**.
- **Ticket Grid Refresh Interval** - Sets the interval for refreshing the **Tickets** table in minutes.

Color Preferences

Set the color codes for ticket rows displayed in tickets tables, based on a **Date Range** *benchmark*. Available benchmarks include:

- **Creation Date** - The creation date of the ticket.
- **Due Date** - The due date of the *current stage* of a ticket. The due date for the current stage of a ticket can only be used if a goal procedure is linked to that stage using the Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab.
- **Last Date Modified** - The date the ticket was last modified.
- **Last Public Update** - The last time a non-hidden note was added.
- **Promised Date** - The promise date of the ticket.

Ticket Age

For the selected date range benchmark you can select three colors:

- **Older than** - Enter the number of days *before* the selected date range benchmark to color code tickets. Enter a negative number to represent days *after* the select date range benchmark. Use the color picker to select a color. For example, you might want to color code the oldest tickets as *red*.
- **Between** - Use the color picker to select a color. For example, you might want to color intermediate tickets as *yellow*.
- **Newer than** - Enter the number of days *after* the selected date range benchmark to color code tickets. Enter a negative number to represent days *before* the select date range benchmark. Use the color picker to select a color. For example, you might want to color recent tickets as *green*.

Service Desks tab

Use this tab to select your "favorite" ticket editing template. You can specify a favorite editing template for each desk definition. This selection overrides the editing template preference set using the Service Desk > **Role Preferences - Service Desks tab** *(page 54)*.

Incoming Email and Alarm Settings

Service Desk > Configure > Incoming Email and Alarm Settings

The **Incoming Email and Alarm Settings** page specifies how to process ticket requests in **Service Desk**. **Ticket requests** are created from either inbound emails or system events, such as alert conditions. **Incoming Email and Alarm Settings** *(page 56)* and linked service desk procedures determine how ticket requests are processed into tickets, what service desks are used to create the tickets, and whether ticket requests are canceled.

There are two tabs on this page:

- **General** *(page 56)* - Specifies general settings for email readers and alarm settings.
- **Readers** *(page 57)* - Specifies one or more readers to periodically poll email accounts. Email messages are downloaded and used to create ticket requests.

Disabling the Ticketing Email Reader

After **Service Desk** activation, the **Ticketing** email reader will still convert inbound emails into **Ticketing** module tickets, so you may want to disable the **Ticketing** email reader manually before **Service Desk** is activated. If both the **Ticketing** and **Service Desk** email readers are to be run concurrently, they should poll different email servers.

General tab

Service Desk > Configure > Incoming Email and Alarm Settings > General tab

The following settings *apply to all email readers* defined in the **Readers** tab.

Actions

The following action buttons apply to all email readers defined in the **Readers** tab.

- **Edit** - Edits general settings for all email readers and alarm settings.
- **Connect Now** - Polls all emails readers immediately.
- **Enable** - Enables polling by all email readers. **Also enables alarm integration if Service Desk is activated (page 51).**
- **Disable** - Disables polling of all email readers **and prevents alarm integration.**

> **Note:** If one or more email readers are failing to connect, click **Disable**, then click **Enable** to re-initialize polling.

Table Columns

- **Poll Interval** - Specify how frequently all email readers connect to email accounts. The default connection frequency is 2 minutes.
- **Ignore emails with these subjects**. - Enter text to ignore inbound emails containing this text *in the subject line*. Matching is case insensitive. *Quotes and wildcard characters such as * and ? are interpreted literally as part of the string content.*

  Create multiple filters using multiple lines. Multiple filters act as an OR statement. Surround whole words with spaces on both sides of each word. Example:

  ```
  Undeliverable
  Do not reply
  ```

  Bracket subject text with single bar | characters to ignore emails with entire subjects that exactly match the ignore text. Example: `|Your timesheet is overdue|`

  This same ignore list can be maintained in the Ticketing > **Email Reader** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#434.htm)* page and the Service Desk > Incoming Email and Alarm Settings > **General tab** *(page 56)*. This list can also be maintained manually by editing the `<Kaseya_Installation_Directory>\Kaseya\KServer\ignoreSubject.txt` file.

  Current Reader - The last reader polled.

  Current Host - The host name of the last reader polled.

The following settings apply to *alarm requests only and are unrelated to email readers.* See **Activation** *(page 51)* and **Source Properties** *(page 73)*.

- **Alarm Mapping Procedure** - The **Ticket Request Mapping** procedure to run when a ticket is created from a system event, such as an alert condition. A **Ticket Request Mapping** *(page 77)* procedure sets the attributes of a ticket request, just before the ticket is created, including the desk definition used to create the ticket.
- **Alarm De-Dup Procedure** - The **Ticket Request De-Dup** procedure to run when a ticket is created from a system event, such as an alert condition. A **Ticket Request De-Dup** *(page 77)* procedure compares a ticket request with existing tickets to determine if the ticket request is a duplicate. If a ticket request is a duplicate of existing tickets, the ticket request is canceled.
- **Alarm Default Service Desk** - The desk definition to use to create tickets from alarms. Can be overridden by the **Alarm Mapping Procedure**.

Readers tab

Service Desk > Configure > Incoming Email and Alarm Settings > Readers tab

Inbound Email Configuration

You must perform all five steps below to enable inbound emails to create new tickets in Service Desk.

1. Add/edit an email reader as described below.
2. Ensure POP or IMAP, as specified in the email reader, is enabled in the email server being polled.
3. Specify the desk definition used to create the ticket, in any *one of the following three ways*:
   a. Optionally set the **Default** desk definition in Service Desk > **Desk Definition** *(page 32)*.
   b. Optionally associate an email reader with a service desk using the Desk Definition > Service Desk > Desk Definition > New or Edit > Standard Field Defaults tab. Has precedence over the **Default** desk definition.
   c. Optionally specify a service desk by associating an email reader with a unique **Ticket Request De-Dup** *(page 77)* or **Ticket Request Mapping** *(page 77)* procedure. Has precedence over the **Default** desk definition and the Desk Definition > **Email** setting.

4. Enable **Incoming Email and Alarm Settings** using the Service Desk > Common Configuration > Incoming Email and Alarm Settings > General tab.
5. Ensure **Ignore emails with these subjects** in the Incoming Email and Alarm Settings > **General tab** *(page 56)* is not filtering the email you are attempting to convert into tickets.

Note: See **Activation** *(page 51)* to create tickets based on system events, such as alarms.

Note: If a service procedure linked to your service desk *sends* emails, ensure that *outbound* emails are enabled using System > **Outbound Email** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4765.htm)*.

Actions

- **New** - Adds a new reader.
- **Edit** - Edits a selected reader.
- **Delete** - Deletes a selected reader.

Adding/Editing an Email Reader

- **ID** - The name of the email reader.
- **Host Name** - The host name of a POP3 or IMAP email server. For example, `pop.youremailserver.com` for a POP3 email server or `imap.youremailserver.com` for an IMAP email server. The email server may require a host name in the **Host Name** field instead of an IP address if **Use SSL** is also checked.
- **Port Number** - The port number used by the email service. This is normally `110` or `995`.
- **Logon ID** - The email account name. *Do not include the @ suffix.* For example if the entire email address is `acme@yourmail.com`, enter `acme` in this field.
- **Password** - The email account password.
- **Reply Email Address** - Specifies the From address for outbound emails. The From address of an outbound email is specified using the following order of precedence:
  1. If there is a From address in the **sendEmail()** step of a procedure, then that address is used.
  2. Else the sendEmail() step uses the From address provided by a linked Service Desk > **Message Template**, if the link exists and a From address is specified.
  3. Else the sendEmail() step uses the **Reply Email Address** of the Service Desk > **Incoming Email and Alarm Settings** > email reader linked to the service desk. This link between the email reader and the service desk is set using the Service Desk > Desk Definition > Properties > General > **Standard Field Defaults** *(page 35)* > Email field.
  4. Else the **Default Sender Email** address set in System > **Outbound Email** is used.
- **Disable Reader** - If checked, email is no longer retrieved from the email server.
- **Use SSL** - If checked, communication with the email server is encrypted using SSL. Your email server must support SSL to use this feature. Some systems require a SSL connection, and the corresponding port number is 995. The email server may require a host name in the **Host Name** field instead of an IP address if **Use SSL** is also checked.
- **Process HTML content in reply emails** - Checked by default. If blank, ignores HTML formatting in reply email and only examines the plain text reply message. Reply messages may be poorly formatted, confusing the processing of inbound email reply messages. Uncheck this checkbox if reply emails generate errors while attempting to update the original ticket.
- **Receipt Dup** - A **Ticket Request De-Dup** *(page 77)* procedure compares a ticket request with existing tickets to determine if the ticket request is a duplicate. If a ticket request is a duplicate of existing tickets, the ticket request is canceled. See **Source Properties** *(page 73)* for more information.
- **Receipt Map** - A **Ticket Request Mapping** *(page 77)* procedure sets the attributes of a ticket request, just before the ticket is created, including the desk definition used to create the ticket. **Receipt Property Variables** *(page 68)* lists the property variables that are available to test for in a **Ticket Request Mapping** procedure. See **Source Properties** *(page 73)* for more information.

- **Transport** - `POP3` or `IMAP`.

Log

The log lists email reader tasks successfully completed and any errors that may have occurred.

Ticket Associations

A ticket request from an inbound email can be associated with an organization, machine or contact (staff member) in one of three ways, using the following order of precedence:

1. The ticket contains tilda (~) tagged content.
2. A ticket request mapping procedure tests the content of the ticket to determine the associations.
3. The "from" address of the inbound email matches the email of a staff record in the VSA. On the basis of this match, the staff member and the staff member's organization is associated with the ticket.
4. A ticket can be "auto-assigned" to a machine ID if an Service Desk email reader or the Ticketing email reader receives an email from the same email address as the one defined for a machine in the Contact Email field of the Agent > Edit Profile page. Applies when new emails come into a Service Desk email reader or the Ticketing email reader that do not map using any other method of email mapping. if multiple machine IDs have the same Contact Email address, then only one machine ID can have the Auto Assign Ticket checkbox checked.

Inbound Emails Sent in Reply to Service Desk Outbound Emails

Set the **Email Display Name** for emails sent outbound from a desk definition, using the Desk Definition > **General Info tab** *(page 34)*.

Note: The name should be a string of letters unlikely to be found in the subject text or body text of emails. It is used as a delimiter when an email is sent in reply to an outgoing **Service Desk** email.

Email Content

A reader can receive plain or HTML formatted email, with or without attachments, and add the contents to the ticketing system. **RTF formatted emails are not supported.** The following tags can be included in *either the subject or the body* of the email. Tags are case-insensitive.

Existing Tickets Only

- `~ticid='xxx'` - Appends the body of the email to an existing ticket. Optional. When a reply email is received, KSD attempts to use the ticket ID included in the previously sent outbound email to identify the existing ticket to append to.

New and Existing Tickets

- `~hide` - Makes the note a hidden note. The `~hide` tag itself is removed.

New Tickets Only

- `~username='xxx'` - Automatically inserts the value given as `xxx` into the **Submitter Name** field.

  Note: If `~username='xxx'` is *not* included in the either the subject or the body of the email, then the email sender's `From` address is used to populate the **Submitter Name** field.

- `~useremail='xxx'` - Automatically inserts the value given as `xxx` into the **Submitter Email** field.
- `~assignee='xxx'` - Assigns the ticket created to a specific user. The user must exist.
- `~machineid='xxx.xxx'` - Assigns the ticket created to a machine ID. The machine ID must exist.

- ~machinegroup='xxx.xxx' - Assigns the ticket created to a machine ID. The machine ID must exist.
- ~organization='xxx' - Assigns the ticket created to an organization by orgID. The orgID must exist.

> **Note:** The following fields are defined by desk definition.

- ~category='xxx' - Assigns the ticket created to a specific category. The category must exist.
- ~priority='xxx' - Assigns the ticket created to a specific priority. The priority must exist.
- ~status='xxx' - Assigns the ticket created to a specific status. The status must exist.
- ~severity='xxx' - Assigns the ticket created to a specific severity. The severity must exist.
- ~solutiontype ='xxx' - Assigns the ticket created to a specific resolution. The resolution must exist.
- ~customfield='xxx' - Assigns the ticket created to the specified value of a custom field ID. The custom field ID must exist.

Procedure Variables

Service Desk > Configure > Procedure Variables

The **Procedure Variables** page defines variables used in service desk procedures. The default value set for a variable in **Procedure Variables** can be superseded by the default value for a **policy variable** *(page 61)*.

Procedure variables can be *nested* inside of other procedure variables. For example, the procedure variable [=Email_ReviewTeam=] is entered as the value of many pre-defined procedure variables. You need only change the default value of the Email_ReviewTeam procedure variable from enter.review.team.email@yourcompany.com to any valid set of email addresses to update all the other procedure variables that use it.

> **Note:** **Service Desk** installs pre-defined procedure variables. These pre-defined procedure variables are used by one or more pre-configured desk definitions and procedures installed using the **Desk Templates** *(page 42)* page. You can change their contents and your changes will be preserved. But if you delete these pre-defined procedure variables they will be re-created during the next maintenance cycle of the VSA.
>
> **Note:** See **Working with Variables** *(page 67)* for an overview of variables in service desk procedures.

Actions

- **New** - Adds a new procedure variable.
  - **Name** - The variable name. Can't be changed once you save it.
  - **Description** - A brief description of the variable.
  - **Set** - The set determines what kinds of desk definition procedures can use the variable. Use any set type of variable in **Policies** *(page 61)*. You can define two variables with the same name, providing they are assigned different sets. Set options include:

Stage Entry or Exit

Ticket Change

Ticket Request De-Dup

Ticket Request Mapping

Ticket Goals

Ticket Escalations

All - Applies to all types of procedures.

  - **Value** - The default value of the variable.
- **Edit** - Edits a selected procedure variable.
- **Delete** - Deletes a selected procedure variable.

Policies

Service Desk > Configure > Policies

The **Policies** page regulates how a service desk can be used, including when it may be contacted, the method of contact used, the resources available and other requirements. Policies are assigned to a ticket as each ticket is created, by organization, organization type or machine group. See **Automated Processing Using Policies** *(page 4)*.

Policy Configuration

1. Define the hours and days of the year a service desk is open to respond to service requests using **Coverage Schedules** *(page 62)*. This includes hours of coverage to *standard* holidays.
2. Define hours of coverage for specific holidays using **Holidays** *(page 63)*.
3. Define policies regulating how a service desk can be used, using **Policies**.
4. Link a policy to a service desk using using Service Desk > Desk Definition > Processing > **Associated Policies** *(page 40)*.
5. Define variables you associate with service desk policies, using **Procedure Variables** *(page 60)*.
6. Assign procedure variables to policies using the **Variables** tab in **Policies**.
7. Use policy variables in **procedures** *(page 64)* to determine how tickets are processed. An example is automatically setting property values in a ticket based on the value of a policy variable.

Actions

- **New** - Adds a new policy.
- **Edit** - Edits a selected policy.
- **Rename** - Renames a selected policy.
- **Delete** - Deletes a selected policy.

General tab

- **ID** - The name of the policy.
- **Description** - A brief description of this ID.
- **Policy Hours** - Assign a **Coverage Schedules** record to this policy.
- The following settings are for reference purposes only.
  - **Contact by Email** - If checked, the service desk can be contacted by email.
  - **Contact by Phone** - If checked, the service desk can be contacted by phone.
  - **Contact by IM** - If checked, the service desk can be contacted by instant messaging.
  - **Access to Knowledge Base** - If checked, service desk users have access to a knowledge base.
- **Time Measured Against Coverage** - If checked, **goals and escalations** *(page 64)* include **Coverage Schedules** when calculating escalations. For example, if a new ticket is created just before the service desk shuts down for the night, and escalation is scheduled for two hours, then escalation won't occur until two hours after the service desk re-opens the next morning. If unchecked, calendar time alone is used.

Variables tab

Variables can be assigned to policies. Once policies are assigned to tickets, those tickets can be processed by procedures differently, based on the values of their policy variables. Procedures can both read and override the values set by a policy variables.

Select a variable and assign a value. Before you can assign a variable to a policy, the variable must be previously created using Service Desk > **Procedure Variables** *(page 60)*. When a procedure is processed, the default value for a policy variable supersedes the default value set for a variable using **Procedure Variables**.

Actions

- **New** - Assigns a procedure variable to a policy.
  - **ID** - Displays the policy name selected in the middle pane.
  - **Variable** - Select a variable previously defined using Service Desk > **Policies**.
  - **Value** - Enter an alphanumeric value.
- **Edit** - Edits the assignment of a procedure variable with a policy.
- **Delete** - Removes a procedure variable from a policy.

Coverage Schedules

Service Desk > Configure > Coverage Schedules

The **Coverage Schedules** page specifies the hours and days of the week a desk definition is open to respond to service requests. A **Coverage Schedules** record is assigned to a **Policies** *(page 61)* record. **Coverage Schedules** are calculated using the time zone specified for that coverage schedule. If enabled, a **DST** checkbox automatically adjusts for Daylight Savings Time.

Testing for Coverage

An **isWithinCoverage()** *(page 86)* service procedure command can test whether a ticket or KB article is within coverage. You can use this to determine the flow of automation from that point forward in the service procedure.

Time Measured Against Coverage

If the **Time Measured Against Coverage** checkbox in **Policies** *(page 61)* is checked, **goals** *(page 77)* and **escalations** *(page 78)* include **Coverage Schedules** when calculating escalations in service procedures. For example, if a new ticket is created just before the service desk shuts down for the night, and escalation is scheduled for two hours, then escalation will not occur until two hours after the service desk re-opens the next morning. If unchecked, calendar time alone is used.

Actions

- **New** - Adds a coverage schedule.
- **Edit** - Edits a coverage schedule.
- **Delete** - Deletes a coverage schedule.

General tab

- **Name** - The name of the coverage schedule. Can't be changed once you save it.
- **Description** - A brief description of this ID.

Standard Hours tab

The **Standard Hours** tab specifies the hours covered for selected week days and holidays.

Actions

- **New** - Adds an hours covered record.
- **Edit** - Edits a selected hours covered record.
- **Delete** - Deletes a selected hours covered record.

Fields

- **Name** - The name of the hours covered detail record.
- **Days Covered** - Select a week day record or holiday record. Options include individual days in the week, Monday through Friday, weekends, or specific holidays. The holidays you can can select in this drop-down list are defined using the **Holidays** *(page 63)* page. *Holidays older than the current date are not displayed in the drop-down list.*
- **Hours Covered** - Select one of the following:
  - **No Coverage**
  - **24 Hour Covered**
  - **Use these Hours** - If this option is selected, the following fields must be entered:
    - ✓ **Begin Hour** - Enter the hour of the **Days Covered** that service coverage begins. Use 24 hour notation. For example, enter `8` to mean eight o'clock in the morning.
    - ✓ **End Hour** - Enter the hour of the **Days Covered** that service coverage ends. Use 24 hour notation. For example, enter `17` to mean five o'clock in the afternoon.
    - ✓ **Time Zone** - The time zone used to define when a service desk is available to receive calls from customers.

> **Note:** All records are logged using the time zone of the Kaseya Server. The selected time zone only affects how time is entered, displayed and calculated in the user interface.

Default Holidays

You don't have to define hours of coverage for each holiday separately. Instead define hours of coverage for a record called `Default Holidays` in the **Days Covered** drop-down list. *This adds all holidays in the Holidays table to that coverage schedule.* If you don't define hours of coverage for a specific holiday by exception, then each holiday in the `Holidays` table uses the hours of coverage defined for `Default Holidays`.

Holidays

Service Desk > Configure > Holidays

The **Holidays** page is used to define holidays by specific future date. Once defined, all **future dates** are listed in the **Days Covered** drop-down list in the **Standard Hours** tab of the **Coverage Schedules** *(page 62)* page. You can change the date of each holiday at any time—for example, at the beginning of each year—without having to reapply them to your coverage schedules.

Default Holidays

You don't have to define hours of coverage for each holiday separately. Instead define hours of coverage for a record called `Default Holidays` in the **Days Covered** drop-down list. *This adds all holidays in the Holidays table to that coverage schedule.* If you don't define hours of coverage for a specific holiday by exception, then each holiday in the `Holidays` table uses the hours of coverage defined for `Default Holidays`.

Actions

- **New** - Click **New** to display the **Add Holiday** window.
  - **Description** - The name of the holiday. This middle pane identifier can not be changed once you save it.
  - **Date** - Enter the calendar date for the holiday.
- **Edit** - Changes the date of an existing holiday.
- **Delete** - Deletes an existing holiday.

Procedures Definition

Service Desk > Service Desk Procedures

Service Desk procedures automate the processing of tickets and ticket requests. All service desk procedures are created and structured in a similar fashion. All procedures are configured to run by linking them to a service desk definition or another service procedure or by ticket request, a process that requests the creation of a new ticket.

Types of Procedures

- **Stage Entry or Exit** *(page 76)* - Runs when a ticket enters or exits a stage.
- **Ticket Change** *(page 76)* - Runs each time a ticket is changed.
- **Goal** *(page 77)* - Runs a specified time period, usually to record whether or not a goal was met.
- **Escalation** *(page 78)* - Runs a specified time period, usually to escalate a ticket to another level of service.
- **Sub-Procedures** *(page 79)* - Runs when executed by another service desk procedure.

The following procedures apply to *ticket requests* processed using **Incoming Email and Alarm Settings** *(page 56)*.

- **Ticket Request De-Dup** *(page 77)* - Compares a *ticket request* with existing tickets to determine if the ticket request is a duplicate. If a ticket request is a duplicate of existing tickets, the ticket request is canceled.
- **Ticket Request Mapping** *(page 77)* - Sets the attributes of a *ticket request*, just before the ticket is created, including the desk definition used to create the ticket.

Additional Topics

- **Service Procedure Folder Trees** *(page 64)*
- **Procedure Editor** *(page 65)*
- **Working with Variables** *(page 67)*
- **Ticket Properties and Variables** *(page 68)*
- **Service Desk IF-ELSE and STEP Statements** *(page 81)*
- **Service Desk Priority Calculations** *(page 79)*

Service Procedure Folder Trees

Service Desk > Procedure Definition > (Any Type of Procedure)

All service procedures are organized under a single **Shared** cabinet. Use the following options to manage objects in these folder trees.

Always Available

- **(Apply Filter)** - Enter text in the filter edit box, then click the funnel icon to apply filtering to the folder trees. Filtering is case-insensitive. Match occurs if filter text is found anywhere in the folder trees.

When the Shared Cabinet or a Folder is Selected

- **Add Folder** - Creates a new folder underneath the selected cabinet or folder.
- **Import Folder/Procedure** - Imports a folder or procedure as children to the selected folder in the folder tree.

> **Note:** Legacy scripts can be imported into the VSA.

- **Export Folder** - Exports the selected folder and all its procedures as an XML file. The XML file can be re-imported.

  Note: Desk definitions, message templates and tickets can be imported and exported using the System > **Import Center** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6963.htm)*. A desk definition can be exported as a template or as desk. Exporting a desk definition can include all the procedures that support it. Exporting a ticket includes the desk that supports it.

- **Expand All** - Expands the entire folder tree.
- **Collapses All** - Collapses the entire folder tree.

Additional Buttons Displayed When a Folder is Selected

- **Share Folder** - Shares a folder with user roles and individual users. Applies to shared cabinet folders only.
- **Delete Folder** - Deletes a selected folder.
- **Rename Folder** - Renames a selected folder.
- **Take Ownership** - Takes ownership of a folder you do not own. This option only displays for master role users.
- **New Procedure** - Opens the **Procedure Editor** *(page 65)* to create a new procedure in the selected folder of the folder tree.
- **Folder Properties** - Displays the name, description, and owner of a folder, and your access rights to the folder.

Note: See guidelines for share rights to objects within folder trees in the **Folder Rights** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4673.htm)* topic.

When a Procedure is Selected

- **Edit Procedure** - Opens the **Procedure Editor** *(page 65)* to edit the selected procedure.
- **Delete Procedure** - Deletes the selected procedure.
- **Export Procedure** - Exports the selected procedure.
- **Rename Procedure** - Renames the selected procedure.

Viewing Procedure Details

When a service procedure is selected in the middle pane, the following tabs display In the right-hand pane:

- **View Procedure** - Provides a display-only view of the procedure.
- **Used by** - Displays a list of other procedures that execute this procedure. Procedures that are used by other procedures cannot be deleted.

Service Procedure Editor

In any service desk procedure *folder tree* (*http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4673.htm*), click **New Procedure** or **Edit Procedure** to display the **procedure editor**.

- The left hand pane displays the outline of the entire procedure.
- The right hand pane displays the parameters for each statement.

Note: See **IF-ELSE-STEP Service Desk Commands** *(page 81)* for a detailed explanation of each statement's parameters.

Action Buttons

These buttons display in the middle pane of the procedure editor.

- **Procedure**
  - **New** - Creates an empty tab for a new procedure.
  - **Open** - Edits an existing procedure.
  - **Save** - Saves the currently selected procedure.
  - **Save As** - Saves the procedure to a different name. A dialog enables you to select the folder used to save the procedure.
- **Edit** - The following buttons are only enabled when one or more statements are selected.
  - **Undo** - Undoes the last edit.
  - **Redo** - Redoes the last edit.
  - **Cut** - Cuts selected lines.
  - **Copy** - Copies selected lines.
  - **Paste** - Pastes copied lines.
  - **Remove** - Removes selected lines.
  - **Goto Line** - Selects the line number you specify.
  - **Search** - Searches for matching text in commands, parameters and values.
  - **Insert Lines** - Inserts a blank line that you can then begin typing into. This displays a drop-down list of commands that you can select a command from and insert into the procedure.
  - **Indent Lines** - Indents selected lines
  - **Outdent Lines** - Outdents selected lines.
- **Help**
  - **Help Tips** - Display tooltips on how to use the procedure editor.
  - **Online Help** - Displays online help.

Drag and Drop

- Drag and drop any statement above or below any other statement.
- Drag and drop any comment above or below any statement.
- A statement is automatically indented when dropped below an IF statement, except for an ELSE statement.
- You can nest steps within multiple IF or ELSE statements. Just drag-and-drop an IF or ELSE statement below an IF statement to insert it as a child statement.

Guidelines

- Click any STEP, IF or ELSE statement in the middle pane to see its settings in the right-hand pane. You can edit these settings in the right hand pane or click any value in a statement directly to edit it.
- Multiple lines can be selected and acted on at one time.
- Right click selected lines to get additional options.
- Enter a value at the top of the left pane to filter the list of statements you can select.
- Hovering the cursor over any statement in the left or middle pane displays a tooltip description of that statement. The same description displays at the top of the third pane.
- Hovering the cursor to the left of selected statements displays  icons. Click these icons to remove, indent or outdent selected statements.
- When entering a value for a variable into a parameter:
  - Enter a < to select from a list of system variables.

  - Enter a # to select from a list of user defined variables.
- Open and work on multiple procedures simultaneously. Each procedure you open displays in a separate tab. Copy and paste selected statements between tabs.
- You can set a STEP to `Continue on Fail`. This allows a procedure to continue running even if that particular STEP fails.
- Click the blank line at the bottom of the procedure to edit the description for the entire procedure.
- You can nest procedures within procedures using the `executeSubProcedure()` command.

Working with Variables

The following guidelines apply to any service desk procedure.

- **Global Procedure Variables** - Global procedure variables are defined using Service Desk > **Procedure Variables** *(page 60)*.
  - The same global variable can be applied to any procedure run by any service desk.
  - The **checkVariable()** *(page 84)* command tests the value of a variable in a procedure.
  - The value of a global variable can be changed within a procedure using the **getVariable()** *(page 93)* command, but that value persists only for the duration of the procedure.
  - Within a procedure, a global variable is referenced *in text* by bracketing its name with the `[=` and `=]`. Example: `[=varname=]`.
  - Global variables are defined by **name** and **set type**. For example, a global variable could be called `varname` and use the set type `All`.The set type `All` means the global variable can be used in any procedure. If a variable is defined using a set type for a specific procedure, it can only be used in procedures matching that set type. For example, a variable defined using the `Goal` set type can only be used in **Ticket Goals** procedures.
  - A global variable of any set type can be associated with a policy. **Policies** *(page 61)* ignore the set type assigned to the global variable.
- **Procedure-Specific Variables** - Within a procedure you can create a *procedure-specific* variable using the **getVariable()** *(page 93)* command and assign it a value.
  - Unlike global variables, the scope of a procedure variable applies only to the procedure it was created in.
  - Like global variables a procedure variable is referenced *in text* by bracketing its name with the `[=` and `=]`. Example: `[=varname=]`.
- **Property Variables** - Property variables reference the values of fields within the VSA.
  - Property variables are distinguished by bracketing their names *in text* with `[$` and `$]`. Example: `[$machine$]`.
  - Types of property variables include`[$TicketId$]` or `[$Organization$]` or `[$Status$]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables.
  - Like global variables and procedure variables, the values of property variables can be tested with IF statements.
  - Setup for the `[$Manager$]` property variable:
    - ✓ Create an Organization.
    - ✓ Create a Department within that Organization.
    - ✓ Create a first Staff member assigned to that Department.
    - ✓ Create a second Staff member assigned to that Department.
    - ✓ There is a field named Supervisor in the Staff record. Assign the *second* Staff member as the *Supervisor* of the first Staff member.
    - ✓ Create a ticket and assign the *first* Staff member to the ticket, the one who has a Supervisor. The Supervisor name does not display in the ticket.

- ✓ Save the ticket.
- ✓ Any associated procedure such as stage entry or ticket change can now test for the ticket's `[$Manager$]` property variable being equal to the name of the *first* staff member.
- ✓ Custom fields property variables are supported.

- **Case Sensitivity** - All three types of variable names are case sensitive.
- **Where Used** - Once variables are created you can include them, in their bracketed format, *in any text entry field* displayed by an IF-ELSE dialog box.
- **Message Templates** - **Property variables** *(page 68)*, global variables and **procedure variables** *(page 60)* can be resolved in message templates.
  - ➢ Any user name included in the **To** field of a message template is automatically converted into that user's corresponding email address. For example, if the user `jsmith` has a corresponding email address of `jsmith@acme.com`, then the same message template—with `jsmith` in the `To` field—can be used by a **sendMessage()** step to send a message to `jsmith` and also used by a **sendEmail()** step to send an email to `jsmith@acme.com`. Email addresses for each user are maintained using System > **Users** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#4576.htm)*.
- **Note Templates** - Property variables, global variables and procedure variables can be resolved in note templates.
- **Procedure Variable Time Zone** - The **Procedure Variable TZ** option on the Desk Definition > **General Info tab** *(page 34)* sets the time zone used by date/time procedure variables such as `[$CreateDateTime$]`, `[$ClosedDateTime$]`, and `[$EscalationDateTime$]`. This time zone also applies to procedure-generated date/time *text* embedded in the **Note** column of the notes history of the ticket and date/time *text* embedded in messages and emails generated by procedures. (The dates/times displayed in the **Time** column of the notes history continue to use the time zone specified for the VSA user in System > **Preferences** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#503.htm)*.)

Ticket Properties and Variables

Ticket Properties

Tickets have many properties. The rules for setting the value of a property are specific to each property. In general, the values of properties are set manually, set by the system and therefore read-only, or read or set using service procedures. Many properties display as columns in the **ticket table** *(page 15)* and in **field permission** *(page 54)* dialogs. The same set of ticket properties are not necessarily available for all combinations of desk definition and user role.

Property Variables

Most of standard fields you see in a ticket have a corresponding property variable. Variable names are case sensitive. **When embedded in the text** of a longer note or a message template, or a longer text statement within a service procedure, a property variable is bracketed using `[$` and `$]` notation to identify it as a property variable that needs to be supplied with a stored value. In a service procedure, if you're prompted to simply enter the name of a property variable **without any surrounding text**, you typically don't have to include the `[$` and `$]` brackets to further identify it as a property variable.

Custom Properties and Property Variables

Custom field property variables are also supported. Use the **assignCustomField()** (page 88) command in service procedures to return or set their values. Uses the format `[$customfieldId$]`. Examples include:

- `[$ReasonforChange$]`
- `[$BusinessAspectsofChange$]`

- [$TechnicalImpact$]

Ticket Request Properties and Property Variables

Ticket request properties are a special set of property variables that apply only when a ticket request is created using an inbound alert or email using **Incoming Email and Alarm Settings** *(page 56)*. They can be:

- Tested using the **testTicketProperty()** *(page 87)* in **Ticket Request Mapping**.
- Embedded in **Message Templates** *(page 49)* when bracketed using [$ and $] notation. The message template must be referenced by a **Ticket Request De-Dup** *(page 77)* and **Ticket Request Mapping** *(page 77)* service procedures.
- Embedded in the edit fields of other **Ticket Request De-Dup** and **Ticket Request Mapping** service procedure commands. *These are the only two service procedures that can use them.* See **Source Properties** *(page 73)* for additional information.

Property/Variable Classifications

- **Read Only Properties/Variables** - Whether displayed in a ticket, note or message template, these properties and their corresponding variables can only be read. They are set by the system or defined in another module, outside of **Service Desk**
- **Can Assign Manually** - This property can be assigned a value manually using the ticket editor.
- **Can Assign by Service Procedure** - This property can be assigned a value by service procedures. Most properties that can be assigned manually can also be assigned by service procedure.
- **Ticket Request Properties/Variables** - These properties and their corresponding property variables are restricted to ticket request service procedures.

Variables	Description	Read Properties/ Variables	Can Assign Manually	Can Assign by Service Procedure	Ticket Request Properties/ Variables
[$ActualTaskHours$]	The total number of hours worked while working on tasks within a ticket.	●			
[$AgentGuid$]	When a ticket is associated with an agent, this is the uniqueID of the agent.	●			
[$AllNotes$]	Displays the entire history of all hidden notes and non-hidden notes.	●			
[$AllPublicNotes$]	Displays the entire history of non-hidden notes.	●			
[$ArchiveFlag$]	If true, this ticket is archived.	●	●	●	
[$Assignee$]	The VSA user currently assigned to the ticket.		●	●	
[$AssigneeEmailAddress$]	The email of the assignee assigned the ticket.	●			
[$Body$]	The body text of the email or alert.				●
[$CCEmailAddr$]	The CC email recipients.				● [2]
[$Category$]	The category assigned to the ticket.		●	●	
[$Changes$]	Displays a summary of changes as plain text, including any newly added note.Applies to public notes only.	●			
[$ChangesAsHtml$]	Displays a summary of changes in HTML, including any newly added note.Applies to public notes only.	●			
[$ClosedDateTime$]	The date/time the ticket was closed.	●			
[$CreateDateTime$]	The creation date/time of the ticket.		●		

[$Description$]	The description of the ticket.		●		
[$EditingUserName$]	The user who last edited the ticket.	●			
[$EmailHeaders$]	The headers of the email.				● [2]
[$EmailPriority$]	The priority of the email, set by the sender.				● [2]
[$EscalationDateTime$]	*(Escalation)* The date/time the ticket will be escalated if it does not exit the current stage.		●		
[$EscalationLevel$]	The current escalation level for this ticket within the current stage.	●			
[$EstimatedTaskHours$]	The estimated number of hours for all tasks within the ticket.	●			
[$ExpectedDueDate$]	The goal date/time for the entire ticket.		●		
[$FullMachineName$]	The fully qualified machine name and group. Equivalent to using [$Machine$].[$MachineGroup$]. Typically used with the scheduleProcedure() step in a procedure to run an agent procedure.	●	●	●	
[$GoalDateTime$]	The goal date/time the ticket will be overdue if it does not exit the current stage.	●			
[$Hidden$]	Always false for ticket requests creating a new ticket. When replying to an outbound ticket, including a ~hide string in the subject or body of the reply email will causes the reply message to be converted into a hidden note.				●
[ID]	The unique ID of the ticket request. Not intended for use in procedures.				●
[$InIncidentUpdate$]	True if an existing ticket is being updated. False if a new ticket is being created.	●			●
[$InMergeTicket$]	True if the procedure is running on a merged ticket command.	●			
[$IsTicketRequest$]	True if created by a system event instead of by an inbound email.				●
[$LastEditDateTime$]	*(Modified On)* The date/time the ticket was last edited.		●		
[$LastPublicUpdate$]	The date/time that a non-hidden note was last added to the ticket.	●			
[$Machine$]	The machineID associated with this ticket.		●	●	
[$MachineGroup$]	The machine group of the machineID associated with the ticket.		●	●	
[$Manager$]	The supervisor of a staff member associated with a ticket. See **Working with Variables** *(page 67)* for instructions on how to use this property variable. Note:This is not the same as the manager of the department the staff member belongs to.	●			
[$Merged$]	True if the ticket was created by a merge command.	●			
[$Organization$]	The organizationID assigned to the ticket.		●	●	●
[$PartitionID$]	The partition the ticket request was directed to. Not intended for use in any service procedure.				●

[$Owner$]	The VSA user ultimately responsible for resolving the ticket.		●	●	
[$OwnerEmailAddress$]	The email address of the owner of the ticket.	●			
[$PolicyName$]	The name of the policy assigned to the ticket.		●	●	
[$Pool$]	The pool of VSA users assigned to the ticket.		●	●	
[$PreviousStage$]	The stage the ticket was previously assigned.	●			
[$Priority$]	The priority assigned to the ticket.		●	●	
[$ProjectedDate$]	The date promised to resolve the ticket.		●	●	
[$RawEml$]	The raw email, without the headers.				● [2]
[$ReadFlag$]	If true, no user has viewed the ticket yet. If false, at least one user has viewed the ticket.		●	●	
[$RelatedTicketsAsBulletList$]	Returns a bulleted list of related tickets.	●			
[$RelatedTicketsAsString$]	Returns a comma separated list of ticketIDs with their summary descriptions.	●			
[$Resolution$]	Displays the resolution assigned to the ticket.		●	●	
[$ResolutionDateTime$]	*(Actual Completed Date)* The date/time the ticket was resolved.	●			
[$ResolutionText$]	Descriptive text entered with a resolution code.		●	●	
[$SDTimeZoneOffset$]	Displays the timezone offset assigned to the ticket.		●	●	
[$ServiceDesk$]	The service desk used to create the ticket.		●		
[$Severity$]	The severity assigned to the ticket.		●	●	
[$SourceType$] [$SourceTag1$] [$SourceTag2$] [$SourceTag3$] [$SourceValue1$] [$SourceValue2$] [$SourceValue3$]	See **testTicketProperty()** *(page 87)* for a description of SourceType, SourceTag and SourceValue property variables.				●
[$Staff$]	Displays as the Contact field in the ticket editor. A contact is a staff member of an organization.		●	●	●
[$StaffEmail$]	*(Contact Email)* Displays the contact's email value.		●		
[$StaffPhoneNumber$]	*(Contact Phone)* The staff phone number the ticket is assigned to.		●	●	
[$Stage$]	The stage assigned to the ticket.		●	●	
[$StageStartDateTime$]	The start date/time of the current stage of the ticket.	●			
[$Status$]	The status assigned to the ticket.		●	●	
[$Subject$]	The subject line of the email or alert.				●
[$SubmitterEmailAddress$]	The email address of the submitter.		●		● [1]
[$SubmitterName$]	The name of the person submitting the ticket. Usually this is the name of the user calling or emailing about a service issue.		●		● [1]

[$SubmitterType$]	The type of person who created the ticket. A PARTICIPANT is a VSA user. A USER is a non-VSA user.	●	●		●
[$Summary$]	A one line summary description of the ticket.		●	●	
[$TargetEmailAddr$]	The email address the inbound email was addressed to.				●
[$TicketId$]	*(ID)* The unique identifier for the ticket.This value cannot be changed.	●			
[$TicketURL$]	Provides a link to the ticket in an email or message.	●			
[$TotalHoursWorked$]	*(Worked Performed Date Time)* The total hours worked on the ticket.		●	●	
[$WorkOrder$]	The work order associated with the ticket.	●			
[$WorkOrderName$]	The name of the work order associated with the ticket.	●			
[$WorkOrderNumber$]	The number of the work order associated with the ticket.	●			
[$WorkPerformedDateTime$]	The date and time work was performed on the ticket.This value is independent of the entries entered for hours worked.	●			

[1] The exceptions are `[$SubmitterName$]` and `[$SubmitterEmailAddress$]` which can be manually changed in the ticket editor.

[2] The `[$CCEmailAddr$]`, `[$EmailHeader$]`, `[$EmailPriority$]`, `[$RawEml$]` property variables apply only to **Ticket Request Mapping** *(page 77)* procedures.

Properties Without Variables					
Activity	The activity type associated with an hours worked entry. Applies only if Service Billing Integration is enabled.		●		
Entry Form	Displays a drop-down list of different forms that can be used to edit a ticket.		●		
Hidden Note	Optionally hides, shows or enables the creation of hidden notes.		●		
Locked By	The user currently editing the ticket. Locking prevents any other user from editing the ticket.		●		
Locked On	The date/time the ticket was locked.		●		
Note	Optionally hides, shows or enables the creation of non-hidden notes.		●		
Organization Address	The address of the organization assigned to the ticket.		●		
Organization Name	The name of the organization assigned to the ticket.		●		
Override Rate	A manually entered rate that overrides the default billing rate for a selected resource type. Applies only if Service Billing Integration is enabled.		●		
ResourceType	The resource type assigned to the ticket. Applies only if Service Billing Integration is enabled.		●		

Source Properties

Creating Tickets from System Events

The VSA can be configured to create tickets when system events occur. The types of system events that create tickets are typically either alerts or inbound email. A system event that creates a ticket also occurs when a VSA user clicks a **Create Ticket** link in various locations throughout the VSA.

Creating a ticket based on a system event starts by adding a row of data to the `dbo.ticRequest` table of the VSA `ksubscribers` database. The row contains all the system event data required to create the ticket.

If **Service Desk** integration is **activated** *(page 51)*, the ticket is created in the **Service Desk** module. If **Service Desk** is not activated, the ticket is created in the **Ticketing** module.

Source Properties

A set of "source" columns are included in each row of data in the `dbo.ticRequest` table. These source columns are only used by **Service Desk**. The set of source columns have the following pattern.

- Source Type
- Source Tag 1
- Source Value 1
- Source Tag 2
- Source Value 2
- Source Tag 3
- Source Value 3
- ...

The entire set of source columns includes one Source Type column and 12 "pairs" of Source Tag and Source Value columns, for a total of 25 columns. The entire set of 25 columns is used to classify the content of the system event used to create the ticket in **Service Desk**. These source columns of information are only available to work with during the time a **Service Desk** ticket is being created from a row of data in the `ticRequest` table.

Creating Tickets from System Events in Service Desk

In **Service Desk** only two types of commands are allowed to access source column data:

- The **isDuplicateRequest()** *(page 85)* IF command inside the **Ticket Request De-Dup** *(page 77)* service procedure.
- The **testTicketProperty()** *(page 87)* IF command inside the **Ticket Request De-Dup** or **Ticket Request Mapping** *(page 77)* service procedures. In this case the list of *source properties* are a subset of many other ticket properties you can test.

Running the Procedures

Both the **Ticket Request De-Dup** and **Ticket Request Mapping** service procedures must be specified by the **Incoming Email and Alarm Settings** *(page 56)* page to run them.

- Use the **General tab** *(page 56)* to create tickets based on system events, such as alerts or users clicking a **Create Ticket** link.
- Use the **Readers tab** *(page 57)* to create tickets based on inbound email.

Sample Data

The following tables show some of the values that might display in selected source columns.

SourceType	SourceTag1	SourceTag2	SourceTag3	SourceTag4
Agent Credential Invalid	agentguid	alertid		

Agent Offline	agentguid	alertid		
Agent Online	agentguid	alertid		
Agent Procedure Failure	agentguid	AlertRegistrationId		
Counter	agentguid	monitorobjectid	monitorvalue	monitorsetname
Disk Drive Change	agentguid	alertid		
Event Log	agentguid	AlertRegistrationId	LogType	LogTypeLabel
KNMi Alert	assetid	monitorid	monitortype	
Low Disk	agentguid	alertid OR AlertRegistrationId		
New Patch	agentguid	alertid		
PCI Card Change	agentguid	alertid		
Process	agentguid	monitorobjectid	monitorvalue	monitorsetname
RAM Change	agentguid	alertid		
Service	agentguid	monitorobjectid	monitorvalue	monitorsetname
SNMP	agentguid	monitorobjectid	monitorvalue	monitorsetname
System Check	agentguid	alertid OR monitorobjectid		
Win AU Change	agentguid	alertid		

SourceType	SourceTag5	SourceTag6	SourceTag7	SourceTag8
Agent Credential Invalid				
Agent Offline				
Agent Online				
Agent Procedure Failure				
Counter	monitorobjectname			
Disk Drive Change				
Event Log	EventSetName	EventId	EventSource	EventMessage
KNMi Alert				
Low Disk				
New Patch				
PCI Card Change				
Process	monitorobjectname			
RAM Change				
Service	monitorobjectname			
SNMP	monitorobjectname			
System Check				
Win AU Change				

Source Value Columns

Not shown in the table above are *source value* columns. A source value column exists for each source

tag column. A source value cell may contain a wide range of values for its corresponding source tag cell. For example, in the table above, the cell for the row `Event Log` and column `Source Tag 4` displays the string `LogTypeLabel`. Some of the different values that can display with in the corresponding `Source Value 4` cells are:

- `Directory Service`
- `Internet Explorer`
- `Application`
- `DNS Server`
- `System`
- `Security`

String Comparisons

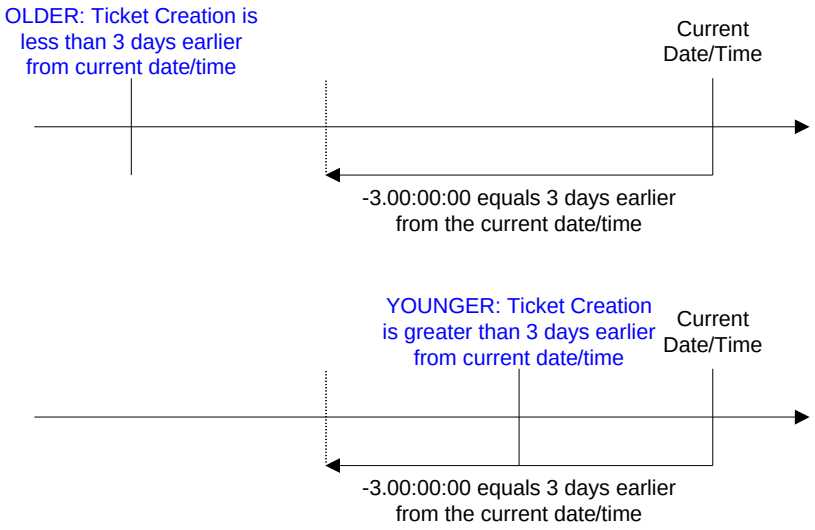
The string comparisons performed in IF commands are not case sensitive. The available tests are:

- `Exists` : true if the variable exists *and* the variable has a non-empty value
- `Does Not Exist` : true if the variable does *not* exist *or* the variable has an empty value
- `Is Equal To` : true if value of the variable equals the test value.
- `Is Not Equal To` : true if value of the variable does *not* equal the test value.
- `Is Greater Than` : true if value of the variable is greater than the test value.
- `Is Greater Than or Equal To` : true if value of the variable is greater than or equal to the test value.
- `Is Less Than` : true if value of the variable is less than the test value.
- `Is Less Than or Equal To` : true if value of the variable is less than or equal to the test value.
- `Contains` : true if the test value is a sub string of the variable (variable must be a string).
- `Does Not Contain` : true if the test value is *not* a sub string of the variable (variable must be a string).
- `Begins With` : true if the variable value begins with the test value.
- `Ends With` : true if the variable value ends with the test value.

For the tests `Is Equal To`, `Is Not Equal To`, `Is Greater Than`, `Is Greater Than or Equal To`, `Is Less Than`, and `Is Less Than or Equal To` the variables compared may be a string, a number, a date in the format of `yyyy/mm/dd` or `yyyy/mm/dd hh:mm` or `yyyy/mm/dd hh:mm:ss`, or a version number containing dots or commas such as `1.2.3` or `4,5,6,7`. Values in variables are stored as strings, so compared numbers must be of equal string length. If a date format is specified, it may be offset using `+ dd:hh:mm:ss` or `- dd:hh:mm:ss`. Only `dd` days are required; `hh` hours, `mm` minutes, and `ss` seconds may be omitted and are assumed to be zero when absent.

Date/Time Comparisons

You can use an IF statement to test whether a time relative to the current time is true or not. Relative time is specified as `[-][d.]hh:mm:ss` where `d` is the number of days, `hh` is the number of hours, `mm` is the number of minutes, and `ss` is the number of seconds . Specifying a – in front causes a time in the past to be calculated. Thus `-1:00:00` means one hour ago, `-3.12:45:08` means three days, 12 hours, 45 minutes, and 8 seconds ago. *Notice that days has a period just after it instead of a colon.* Specifying `d` and `-` are optional. These times do not take into account hours coverage associated with a policy. For example, an IF statement can test to see whether a ticket's create date/time is older than 3 days. The first value is `CreateDateTime`, the operator is `Is Less Than`, and the comparison value is `-3.00:00:00`.



Stage Entry or Exit

Procedures Definition > Stage Entry or Exit

A **Stage Entry or Exit** procedure runs when a ticket transitions to a specified stage. Procedures are linked to stages using the Service Desk > Desk Definition > Processing > Stage > New / Edit > **Procedures** *(page 40)* tab.

Sequencing of Procedures

If you change a stage procedure manually, the stage *exit* procedure runs first, then the stage *entry* procedure for the new stage runs second, then the *ticket change* procedure runs third.

If you set a ticket to a stage *during a procedure*, the procedure waits while the stage exit procedure for the *current* stage runs, then waits while the stage entry procedure for the *new* stage runs, then resumes running the rest of the procedure from that point.

Ticket Change

Procedures Definition > Ticket Change

A **Ticket Change** runs each time a ticket is changed. Typically **Ticket Change** procedures notify the owner, assignee or submitter that a ticket has been updated. **Ticket Change** procedures are linked to the entire desk definition using the Service Desk > Desk Definition > New / Edit > **Procedures** *(page 36)* tab.

Sequencing of Procedures

If you change a stage procedure manually, the stage *exit* procedure runs first, then the stage *entry* procedure for the new stage runs second, then the *ticket change* procedure runs third.

If you set a ticket to a stage *during a procedure*, the procedure waits while the stage exit procedure for the *current* stage runs, then waits while the stage entry procedure for the *new* stage runs, then resumes running the rest of the procedure from that point.

Ticket Request De-Dup

Procedures Definition > Ticket Request De-Dup

A **Ticket Request De-Dup** procedure compares a *ticket request* with existing tickets to determine if the ticket request is a duplicate. If a ticket request is a duplicate of existing tickets, the ticket request is canceled.

Ticket requests are created from either inbound emails or system events, such as alert conditions. **Incoming Email and Alarm Settings** *(page 56)* and linked service desk procedures determine how ticket requests are processed into tickets, what service desks are used to create the tickets, and whether ticket requests are canceled.

Note: See the **isDuplicateRequest()** *(page 85)* command for details on comparing a ticket request with existing tickets. This IF command is only available in a **Ticket Request De-Dup** procedure. See **Source Properties** *(page 73)* for more information.

Note: See **Ticket Properties and Variables** *(page 68)* for a list of property variables that apply to ticket requests.

Ticket Request Mapping

Procedures Definition > Ticket Request Mapping

A **Ticket Request Mapping** procedure sets the attributes of a *ticket request*, just before the ticket is created, including the desk definition used to create the ticket.

Ticket requests are created from either inbound emails or system events, such as alert conditions. **Incoming Email and Alarm Settings** *(page 56)* and linked service desk procedures determine how ticket requests are processed into tickets, what service desks are used to create the tickets, and whether ticket requests are canceled.

Note: See the **testTicketProperty()** *(page 87)* command for details on testing the property values of ticket requests, including the *source* system event of the ticket request. This IF command is only available in a **Ticket Request Mapping** procedure. See **Source Properties** *(page 73)* for more information.

Note: See **Ticket Properties and Variables** *(page 68)* for a list of property variables that apply to ticket requests.

Goal

Procedures Definition > Goal

A **Goal** procedure runs after a specified time period, usually to record whether or not a goal was met.

Goal Procedures for Tickets

A goal procedure is linked to a *desk definition* using the Service Desk > Desk Definition > New / Edit > **Procedures** *(page 36)* tab.

- The ticket goal sets the due date for the entire ticket.
- The ticket editor displays the ticket due date.
- The ticket due date can be set using the **setTicketGoalTime()** *(page 95)* command.
- Ticket goal times can be paused and resumed using the **pauseTicketGoal()** *(page 94)* and **resumeTicketGoal()** *(page 94)* commands.

Goal Procedures for Stages

Goal procedures are linked to *stages* using the Service Desk > Desk Definition > Processing > Stage > New / Edit **Procedures** *(page 40)* tab.

- A stage goal procedure starts the goal clock as soon as the ticket enters the stage. The triggering of a stage goal procedure is canceled as soon as the ticket transitions out of that stage.
- The goal time for a stage can be set using the **setGoalTime()** *(page 95)* command. This overrides the default goal time set in the stage record.
- Color coding by `Due Date` in **Role Preferences** *(page 52)* is determined by a goal procedure linked to a stage.
- Stage goal times can be paused and resumed using the **pauseStageGoal()** *(page 93)* and **resumeStageGoal()** *(page 94)* commands.
- Unlike **escalations** *(page 78)*, goal "levels" do not exist. There is only one for each stage.

Escalation

Procedures Definition > Escalation

An **Escalation** procedure runs after a specified time period, usually to escalate a ticket to another level of service.

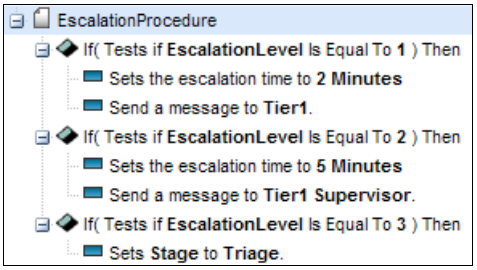
- Escalation procedures are linked to *stages* using the Service Desk > Desk Definition > Processing > Stage > New / Edit > **Procedures** *(page 40)* tab.
- A stage escalation starts the escalation clock as soon as the ticket enters the stage. The escalation procedure, including all levels of the escalation, is canceled when that stage of the ticket is exited.
- The escalation time for a stage can be set using the **setEscalationTime()** *(page 95)* command. This overrides the default escalation time set in the stage record.

Escalation Levels

An escalation procedure can be triggered to run multiple times, based on the escalation level of the ticket. Each time the ticket fails to exit the stage, the escalation level of the ticket increases by one. So a stage starts out with an escalation level value of zero. After the first escalation occurs, the escalation level is equal to one, and so on.

- When a stage is exited all escalation levels for the current stage of that ticket are canceled.
- Procedures can be configured to test the `EscalationLevel` property using the **testIncidentProperty()** *(page 87)* command.
- **Message Templates** *(page 49)* and **Note Templates** *(page 48)* can include the escalation level of a ticket in text using `[$EscalationLevel$]` notation. See **Ticket Properties and Variables** *(page 68)*.

- A ticket escalation procedure can reset the escalation level and time by using a **setEscalationTime()** *(page 95)* command. Each escalation procedure that runs sets up the next escalation event to occur. Example:



Sub-Procedures

Procedures Definition > Sub-Procedures

A **Sub-Procedures** procedure runs when executed by another service desk procedure. This enables you to maintain a single sub-procedure and execute it repeatedly.

- Sub-procedures are executed using the **executeSubProcedure()** *(page 93)* command.
- Any type of procedure can execute a sub-procedure.
- You can nest the execution of procedures to 10 levels.

Note: See **Service Desk Priority Calculations** *(page 79)* for an example of a sub-procedure.

Service Desk Priority Calculations

The `STD Calculate Priority` sub-procedure—used by the ITIL-based **Incident** (page 44) service desk—calculates the priority for a ticket. The calculation considers ticket settings for `Severity` first, then `Urgency` to select:

- The value for the ticket's `Priority` field.
- The **escalation time** *(page 95)* for the stage of the ticket.
- The **goal time** *(page 95)* for the stage of the ticket.

Glossary

- **Severity** – Seriousness of the incident/service request.
- **Urgency** - Requiring or compelling speedy action or attention. The matter is urgent.
- **Priority** - The right of precedence over others. My first priority.

Severity and Urgency

The possible values for both `Severity` and `Urgency` are:

- High
- Medium
- Low

Priorities for Setting Escalation Times

The `STD Calculate Priority` sub-procedure references the following pre-defined **procedure variables** *(page 60)* to set the *escalation time* for the stage of a ticket.

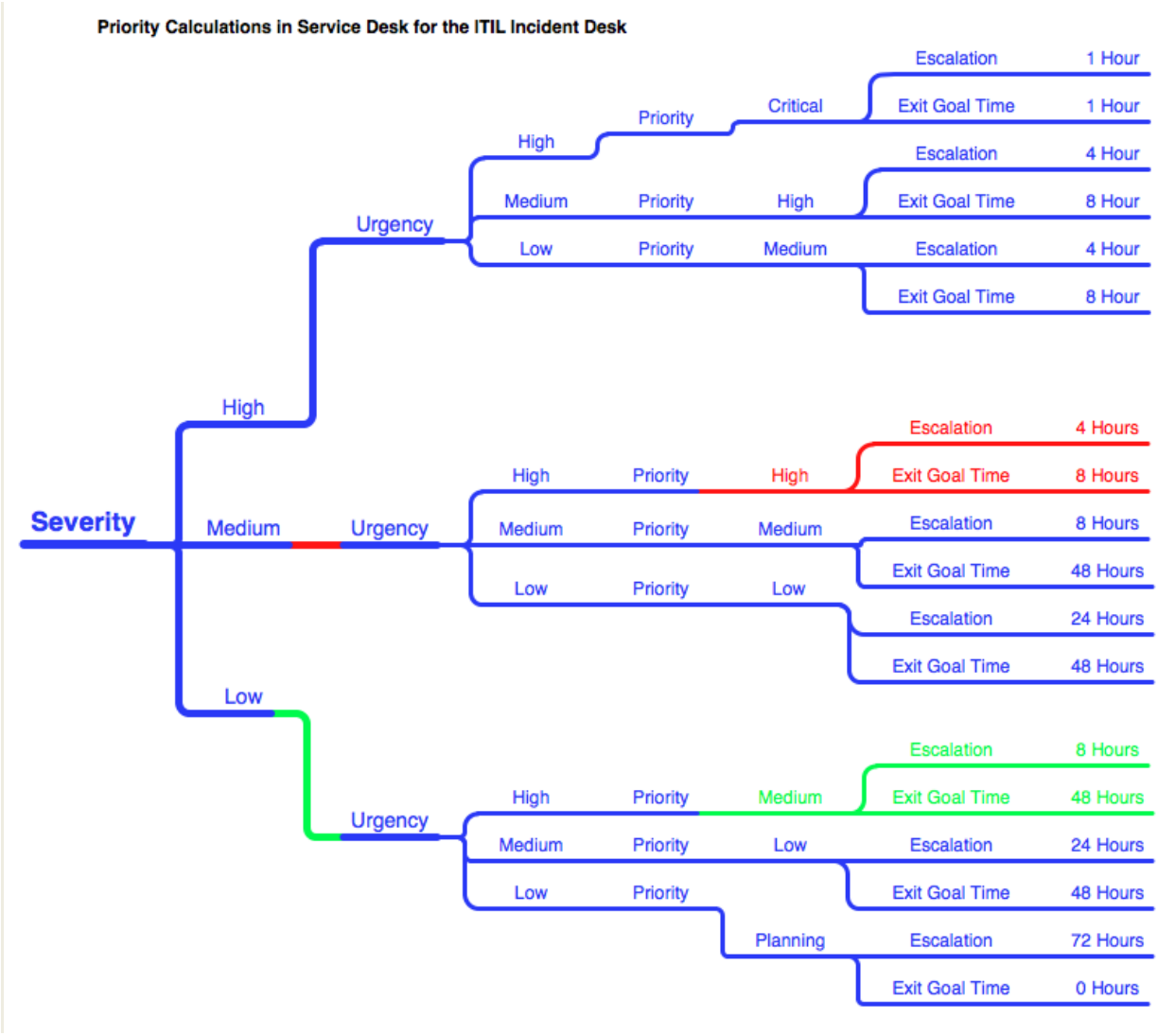
`Escalation_Priority1`	Critical	= 1 hour
`Escalation_Priority2`	High	= 4 hours

`Escalation_Priority3`	Medium	= 8 hours
`Escalation_Priority4`	Low	= 24 Hours
`Escalation_Priority5`	Planned	= 72 Hours

SLA Priorities for Setting Goal Times

The `STD Calculate Priority` sub-procedure references the following pre-defined procedure variables to set the *goal time* for the stage of a ticket.

`SLA_Priority1`	Critical	= 1 hour
`SLA_Priority2`	High	= 8 hours
`SLA_Priority3`	Medium	= 48 hours
`SLA_Priority4`	Low	= 48 Hours
`SLA_Priority5`	Planned	= 0 Hours



IF-ELSE-STEP Service Desk Commands

The following is a summary of **IF-ELSE** and **STEP** statements used in **Service Desk** procedures.

IF-ELSE Statements

checkVariable() *(page 84)*	Evaluate a given variable.
else *(page 85)*	Adds an else command underneath a corresponding if command.
isDuplicateRequest() *(page 85)*	Compares inbound email to determine if it is a duplicate of an existing ticket.
inReopenTicket() *(page 86)*	Tests whether the ticket is a reopened ticket.
isTicketClosed() *(page 86)*	Tests whether a ticket is closed.
isWithinCoverage() *(page 86)*	Tests whether a ticket or KB article is currently within the coverage schedule.
testIncidentCustomField() *(page 86)*	Compares a custom field value in a ticket to a specified value or variable.

testIncidentProperty() *(page 87)*	Compares the value of a field in a ticket to a specified value or variable.
testTicketProperty() *(page 87)*	Compares the value of a field in an inbound email to a specified value or variable.

STEP Statements

addNote() *(page 87)*	Adds a note to the ticket being processed by the procedure.
addNoteWithTime() *(page 87)*	Adds a note and time to the ticket being processed by the procedure.
assignCustomField() *(page 88)*	Assigns a value to a custom field in a ticket.
assignCustomObjectField() *(page 88)*	Assigns a value to the custom field of a *new ticket* created using the **createTicket()** *(page 89)* command.
assignObjectProperty() *(page 88)*	Assigns a value to the property of a *new ticket* created using the **createTicket()** *(page 89)* command.
assignTicketProperty() *(page 89)*	Assigns a value to the property of the *current* ticket.
cancelTicketCreation() *(page 89)*	Cancels the creation of the ticket.
comment() *(page 89)*	Adds a one line comment to the procedure.
createTicket() *(page 89)*	Creates a new ticket in a specified service desk.
deleteTicket() *(page 89)*	Deletes the ticket being processed by the procedure.
executeExternalScript() *(page 90)*	Executes a script written in VB or C#.
executeShellCommand() *(page 91)*	Executes a shell command on the Kaseya Server.
executeSqlNonQuery() *(page 92)*	Executes an SQL non-query command on the Kaseya Server SQL Server database.
executeSqlQuery() *(page 92)*	Executes an SQL select query command on the Kaseya Server SQL Server database.
executeSubProcedure() *(page 93)*	Starts another procedure.
exitProcedure() *(page 93)*	Exits the current procedure.
failCurrentTicketTransaction() *(page 93)*	Changes made to a ticket are not saved to the database.
getAgentInfo() *(page 93)*	Returns a selected property value from the last audit of the agent machine associated with the ticket.
getVariable() *(page 93)*	Gets a entered value or a value from a file stored on the Kaseya Server.
linkTickets() *(page 93)*	Associate another ticket with the ticket being processed.
pauseStageGoal() *(page 93)*	Pause counting elapsed time against the stage goal.
pauseTicketGoal() *(page 94)*	Pause counting elapsed time against the ticket goal.
reOpenTicket() *(page 94)*	Reopens a closed ticket.
resumeStageGoal() *(page 94)*	Resume counting elapsed time against the stage goal.
resumeTicketGoal() *(page 94)*	Resume counting elapsed time against the ticket goal.
scheduleProcedure() *(page 94)*	Runs a selected *agent procedure* for a specified machine.
sendEmail() *(page 94)*	Sends an email to one or more recipients.
sendMessage() *(page 94)*	Sends a message to the inbox of a user.

sendMessageToPool() *(page 95)*	Sends a message to the inbox of each user included in a pool.
sendPost() *(page 95)*	Sends a POST request to a specified URL.
setEscalationTime() *(page 95)*	Sets the time a ticket is escalated.
setGoalTime() *(page 95)*	Sets the time a ticket or stage goal is scheduled to be completed.
setTicketGoalTime() *(page 94)*	Set the due date for the entire ticket.

Where Used

These **IF-ELSE** and **STEP** statements are used by the following types of **Service Desk** procedures.

IF-ELSE Statements	Stage Entry or Exit	Ticket Change	Ticket Request De-Dup	Ticket Request Mapping	Ticket Goals	Ticket Escalations	Sub-Procedures
checkVariable() *(page 84)*	●	●	●	●	●	●	●
else *(page 85)*	●	●	●	●	●	●	●
isDuplicateRequest() *(page 85)*			●				
inReopenTicket() *(page 86)*	●	●					
isTicketClosed() *(page 86)*				●			
isWithinCoverage() *(page 86)*	●	●			●	●	●
testIncidentCustomField() *(page 86)*	●	●			●	●	●
testIncidentProperty() *(page 87)*	●	●			●	●	●
testTicketProperty() *(page 87)*				●			

STEP Statements	Stage Entry or Exit	Ticket Change	Ticket Request De-Dup	Ticket Request Mapping	Ticket Goals	Ticket Escalations	Sub-Procedures
addNote() *(page 87)*	●	●	●	●	●	●	●
addNoteWithTime() *(page 87)*	●	●	●	●	●	●	●
assignCustomField() *(page 88)*	●	●	●	●	●	●	●
assignObjectProperty() *(page 88)*	●	●			●	●	●
assignTicketProperty() *(page 89)*	●	●	●	●	●	●	●
cancelTicketCreation() *(page 89)*				●			

comment() *(page 89)*	●	●	●	●	●	●	●
createTicket() *(page 89)*	●	●			●	●	●
deleteTicket() *(page 89)*	●	●			●	●	●
executeExecuteScript() *(page 90)*	●	●	●	●	●	●	●
executeShellCommand() *(page 91)*	●	●	●	●	●	●	●
executeSqlNonQuery() *(page 92)*	●	●	●	●	●	●	●
executeSqlQuery() *(page 92)*	●	●	●	●	●	●	●
executeSubProcedure() *(page 93)*	●	●			●	●	●
exitProcedure() *(page 93)*	●	●	●	●	●	●	●
failCurrentTicketTransaction() *(page 93)*	●	●			●	●	●
getAgentInfo() *(page 93)*	●	●	●	●	●	●	●
getVariable() *(page 93)*	●	●	●	●	●	●	●
linkTickets() *(page 93)*	●	●			●	●	●
pauseStageGoal() *(page 93)*	●	●			●	●	●
pauseTicketGoal() *(page 94)*	●	●			●	●	●
reOpenTicket() *(page 94)*				●			
resumeStageGoal() *(page 94)*	●	●			●	●	●
resumeTicketGoal() *(page 94)*	●	●			●	●	●
scheduleProcedure() *(page 94)*	●	●	●	●	●	●	●
sendEmail() *(page 94)*	●	●	●	●	●	●	●
sendMessage() *(page 94)*	●	●	●	●	●	●	●
sendMessageToPool() *(page 95)*	●	●	●	●	●	●	●
sendPost() *(page 95)*	●	●	●	●	●	●	●
setEscalationTime() *(page 95)*	●	●			●	●	●
setGoalTime() *(page 95)*	●	●			●	●	●
setTicketGoalTime() *(page 95)*	●	●			●	●	●

IF-ELSE in Detail

checkVariable()

Enter the name of the variable in the format `#var_name#`. It can be a global variable or a procedure variable that was defined earlier in the current procedure or a parent procedure. If a global variable, then the global variable's **Set** value must be compatible with the procedure being run. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

Enter a string comparison operator. The same string comparison options are available for all IF commands. See **String Comparisons** *(page 75)* and **Date/Time Comparisons** *(page 76)*.

Then enter a second value to compare against the first value. Instead of a value you can also enter a property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables.

If the test is true, **IF** steps are executed. If the test is false, **ELSE** steps are executed.

else

Adds an **Else** command underneath a corresponding **If** command. Any steps listed under the **Else** command are executed when the corresponding **If** command returns a `False` result.

Example - Sample Procedures.Managed Services.Disk Mgmt.Clean.Windows Disk Cleanup (wdc)

```
If
hasRegistryKey("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches\Compress old files") Does Not Exist
  executeProcedure(WDC Step 1", " ", "Immediate", "All Operating Systems", "Halt on Fail")
Else
  executeProcedure(WDC Step 2", " ", "Immediate", "All Operating Systems", "Halt on Fail")
```

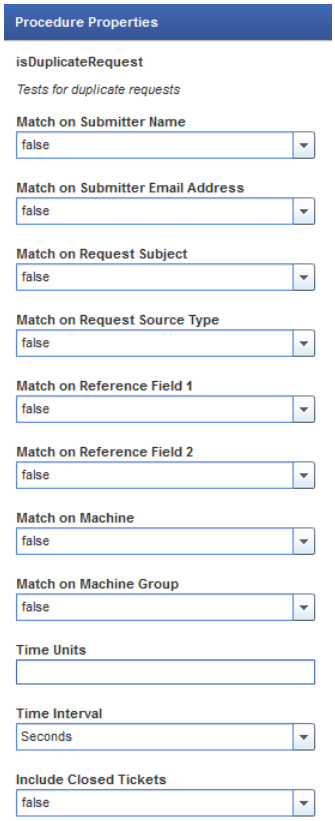
isDuplicateRequest()

This IF command is only available in a **Ticket Request De-Dup** *(page 77)* procedure.

Tests a *ticket request* to see if it is a duplicate of any existing ticket. Prevents multiple tickets from being created for the ticket request, within a specified time limit.

Ticket requests are created from either inbound emails or system events, such as alert conditions. **Incoming Email and Alarm Settings** *(page 56)* and linked service desk procedures determine how ticket requests are processed into tickets, what service desks are used to create the tickets, and whether ticket requests are canceled.

The following dialog displays in the right hand pane of the ticket editor when this command is selected in the procedure editor. The test tests true if *all fields that have an entered value* match their corresponding values in a ticket request record. See **Source Properties** *(page 73)* for more information.



- **Match the submitter name**
- **Match the submitter email address**
- **Match the request subject**
- **Match the source type** - The **Source Type column** *(page 73)*. Example values include:
  - Counter
  - Service
  - Process
  - SNMP
  - Alert
  - System Check
  - KES
  - Log Parser
- **Match the first reference field** - The **Source Value 1 column** *(page 73)*.
- **Match the second reference field** - The **Source Value 2 column** *(page 73)*. Example values include:
  - web server check
  - dns check
  - port check
  - ping
  - custom
- **Match the machine**
- **Match the machine group**
- **Enter Recurrence Time (Integer) / The Unit of Time** - Limits matching to the most recent <N> number of periods, based on the ticket creation date.
- **Include Closed Tickets** - If checked, open and closed tickets are searched for duplicates. If unchecked, only open tickets are searched for duplicates.

isTicketClosed()

Tests whether the ticket being mapped is closed. Typically used prior to a **reOpenTicket()** *(page 94)* step. Applies only to a **Ticket Request Mapping** *(page 77)* procedure.

inReopenTicket()

Tests whether the ticket was previously reopened.

isWithinCoverage()

This IF command is not available for **Ticket Request De-Dup** *(page 77)* and **Ticket Request Mapping** *(page 77)* procedures.

Tests whether a ticket or KB article is currently within the **coverage schedule** *(page 62)*.

testIncidentCustomField()

Enter the name of the custom field in the format `custom_field`, without bracket characters of any kind.

Enter a string comparison operator. The same string comparison options are available for all IF commands. See **String Comparisons** *(page 75)* and **Date/Time Comparisons** *(page 76)*.

Then enter a second value to compare against the first value. Instead of a value you can also enter a property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about

variables.

If the test is true, **IF** steps are executed. If the test is false, **ELSE** steps are executed.

testIncidentProperty()

Select a ticket property from the drop-down list.

Enter a string comparison operator. The same string comparison options are available for all IF commands. See **String Comparisons** *(page 75)* and **Date/Time Comparisons** *(page 76)*.

Then enter a second value to compare against the first value. Instead of a value you can also enter a property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Property and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

For testing the value of a **Category** *(page 37)* property, use two bar characters `||` to delimit levels of categories. For example: `Hardware||CPU||Overheating`

If the check is true, **IF** steps are executed. If the check is false, **ELSE** steps are executed.

testTicketProperty()

This IF command is only available in a **Ticket Request Mapping** *(page 77)* and **Ticket Request De-Dup** *(page 77)* procedures.

Select a ticket property from the drop-down list, in the format `objectname`, without bracket characters of any kind.

Enter a string comparison operator. The same string comparison options are available for all IF commands. See **String Comparisons** *(page 75)* and **Date/Time Comparisons** *(page 76)*.

Then enter a second value to compare against the first value. Instead of a value you can also enter a property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

For testing the value of a **Category** *(page 37)* property, use two bar characters `||` to delimit levels of categories. For example: `Hardware||CPU||Overheating`

You can also test a **source property** *(page 73)*.

If the test is true, **IF** steps are executed. If the test is false, **ELSE** steps are executed.

STEP in Detail

addNote()

Adds a note to the ticket being processed by the procedure. Additional parameters include:

- Selecting a note template instead of entering a note.
- Making the note hidden or not.

Note text can include an property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

addNoteWithTime()

Adds a note and time to the ticket being processed by the procedure. Additional parameters include:

- Selecting a note template instead of entering a note.
- Making the note hidden or not.
- The time in minutes added with the note.
- Marking time added as billable or not.
- Selecting a Default User to associate the time entry with.
- Overriding the Default User with the user manually editing the ticket.

> **Note:** Overriding the Default User is unavailable for **Goal** *(page 77)*, **Escalation** *(page 78)*, **Ticket Request De-Dup** *(page 77)*, or **Ticket Request Mapping** *(page 77)*, since no user is manually editing the ticket just before these procedures are run.

Note text can include an property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

assignCustomField()

Assigns a value to a custom field in a ticket. Enter the name of the custom field in the format `custom_field`, without bracket characters of any kind. Then enter a value that matches the range of possible values allowed for this custom field. The range of possible values is specified using the Service Desk > Desk Definition > Properties > **Custom Fields** *(page 38)* tab.

You can also enter a property variable in the format `[$objectname$]` or a global variable or procedure variable in the format `[=var_name=]`. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

assignCustomObjectField()

Assigns a value to the custom field of a *new ticket created in a previous service procedure step* using the **createTicket()** *(page 89)* command.

- **The name of the global variable used to reference the New Ticket** - Enter the name of the global variable without [= and =] brackets. If multiple **createTicket()** steps are included in the same procedure, each should specify a different global variable so that each new ticket can be uniquely identified in subsequent steps.
- **The name of the property to assign** - The name of the custom field to assign.
- **The value for the property** - The value to assign the custom field. The range of possible values is specified using the Service Desk > Desk Definition > Properties > **Custom Fields** *(page 38)* tab.

assignObjectProperty()

Assigns a value to the property of a *new ticket created in a previous service procedure step* using the **createTicket()** *(page 89)* command.

- **The name of the global variable used to reference the New Ticket** - Enter the name of the global variable without [= and =] brackets. If multiple **createTicket()** steps are included in the same procedure, each should specify a different global variable so that each new ticket can be uniquely identified in subsequent steps.
- **The name of the property to assign** - The name of the ticket property to assign.
- **The value for the property** - The value to assign the ticket property.

The set of properties that can be assigned are displayed in a drop-down list. See **Working with Variables** *(page 67)* for more information about variables.

For assigning the value of a **Category** *(page 37)* property, use two bar characters `||` to delimit levels of categories. For example: `Hardware||CPU||Overheating`

> **Note:** Use **Assign Ticket Property** *(page 89)* to assign a value to the property of the *current* ticket.

assignTicketProperty()

Assigns a value to the property of the *current* ticket. The set of properties that can be assigned are displayed in a drop-down list. The supplied value may be a global variable name or procedure variable name in the format `[=var_name=]`. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

For assigning the value of a **Category** *(page 37)* property, use two bar characters `||` to delimit levels of categories. For example: `Hardware||CPU||Overheating`

> **Note:** Use **assignObjectProperty()** *(page 88)* to assign a value to the property of a *new ticket* created using the **createTicket()** *(page 89)* command.

cancelTicketCreation()

Cancels the creation of the ticket. There are no parameters to set. Applies to **Ticket Request Mapping** *(page 77)* procedures only.

comment()

Adds a one line comment to the procedure.

Example

```
// The IRPStackSize setting for this machine is #IRPStackSize#
```

createTicket()

Creates a new ticket in a specified service desk.

- **The name of the service** - The name of the service desk.
- **Summary for the new ticket** - The summary of the ticket.
- **(Description)** - The description of the ticket.
- **The global variable where the reference to the new ticket** - The name of a global variable used to reference this newly created ticket in subsequent **assignObjectProperty()** *(page 88)* steps. If multiple **createTicket()** steps are included in the same procedure, each should specify a different global variable so that each new ticket can be uniquely identified in subsequent steps. Enter the name of the global variable without [= and =] brackets.

Events occur in the following order:

1. **assignObjectProperty()** steps in subsequent steps assign values to the properties of the newly create ticket.
2. After the new tickets are created, if there are properties of the ticket that have not been set using **assignObjectProperty()** steps, but have defaults in the service desk, those default values are applied to the ticket.
3. If there is a stage entry procedure for the service desk associated with the new ticket's beginning stage, that is run last.

deleteTicket()

Deletes the ticket being processed by the procedure.

executeExternalScript()

> **Note:** A `No Approved External Scripts` message displays in the **Choose the external script to execute** drop-down list if no external scripts have been manually added to the Kaseya Server, as described below.

Executes a script written in VB.Net (.vb) or CSharp (.cs) on the Kaseya Server from the following file locations.

- `<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\0` - Files stored in the `0` directory are available in every partition. (Default)
- `<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\1` - The `1` directory in this example matches partition ID 1. Files stored in a partition-specific directory are available only in that partition.

executeExternalScript() XML files can have any filename. You can have multiple scripts in one file. You can also have multiple files. The scripts referenced by an XML file should be stored in the same location as the XML file under a `\Script` folder.

Example of VB.Net Script

This script writes to a file if the file exists.

```
Dim FILE_NAME As String = "c:\temp\test.txt"
If System.IO.File.Exists(FILE_NAME) = True Then
     Dim objWriter As New System.IO.StreamWriter( FILE_NAME )
     objWriter.Write( "Hello" )
     objWriter.Close()
End If
```

Place the script file in this location:

```
<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\0\script\CreateFile.vb
```

> **Note:** This example requires the creation of `c:\temp\test.txt` prior to execution. Also, the group `IIS_IUSRS` must have permissions to `C:\temp`.

Example of XML File

The XML should be structured like the following example, which includes 2 external scripts.

```
<?xml version="1.0" encoding="utf-8" ?>
<externalScriptList>
   <externalScriptDef label="First Script" fileName="createFile.vb">
   </externalScriptDef>
   <externalScriptDef label="Second Script" fileName="createFile2.vb">
   </externalScriptDef>
</externalScriptList>
```

Place the script file in this location:

```
<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\0\VB.xml
```

Example of C#.Net Script

This script writes to a file if the file exists.

```
string FILE_NAME = "c:\\temp\\test.txt";
if (System.IO.File.Exists(FILE_NAME) == true) {
 System.IO.StreamWriter objWriter = new System.IO.StreamWriter(FILE_NAME);
 objWriter.Write("Hello");
 objWriter.Close();
}
```

Place the script file in this location:

```
<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\0\script\CreateFile.cs
```

Note: This example requires the creation of `c:\temp\test.txt` prior to execution. Also, the group `IIS_IUSRS` must have permissions to `C:\temp`.

Example of XML File

The XML should be structured like the following example, which includes 1 external script.

```
<?xml version="1.0" encoding="utf-8" ?>
<externalScriptList>
   <externalScriptDef label="Write to a file" fileName="CreateFile.cs">
   </externalScriptDef>
</externalScriptList>
```

Place the script file in this location:

```
<Kaseya_Installation_Directory>\xml\SDProcExecuteScript\0\CS.xml
```

Definitions

- `label` - The name that displays in the drop-down list of available scripts to execute. Should be informative of what the script does. Required.
- `filename` - The filename of the script. The script must be located in the same folder as the XML statement under a `\Script` folder. Required.
- `name` - The name of an `additionalAssembly`. Each `additionalAssembly` should have its own node in the XML statement. Optional.

executeShellCommand()

Note: A `No Approved Shell Command` message displays in the **Choose the command to execute in a command shell** drop-down list if no SQL commands have been manually added to the Kaseya Server, as described below.

Executes a shell command on the Kaseya Server from the following file locations.

- `<Kaseya_Installation_Directory>\xml\SDProcShellCommand\0` - Files stored in the `0` directory are available in every partition.
- `<Kaseya_Installation_Directory>\xml\SDProcShellCommand\1` - The `1` directory in this example matches partition ID 1. Files stored in a partition-specific directory are available only in that partition.

Shell commands stored in these locations are formatted as XML files and can have any filename. You can have multiple commands in one XML file. You can also have multiple XML files.

Example

The XML should be structured like the following example, which includes 2 shell commands.

```
<?xml version="1.0" encoding="utf-8" ?>
<shellCommandList>
  <shellCommandDef label="Shell Command 1" commandToExecute="echo hello"
waitForCompletion="true" sendOutputToLog="true" outputFile="c:\temp\fromcmd.log"
outputVariable="" />
  <shellCommandDef label="Shell Command 2" commandToExecute="echo hello"
waitForCompletion="true" sendOutputToLog="true" outputFile="c:\temp\fromcmd2.log"
outputVariable="" />
</shellCommandList>
```

Definitions

- `label` - The name that displays in the drop-down list of available commands to execute. Should be informative of what the shell command does. Required.
- `commandToExecute` - The shell command. Required.
- `waitforCompletion` - True to wait. False to not wait. Required.
- `sendOutputToLog` - True to send output. False to not send output. Required.

At least one of the following is required. If both are filled in `outputFile` has precedence.

- `outputFile` - Full file path and name where the output is saved if sendOutputToLog is true. Optional.
- `outputVariable` - Name of the variable the output goes to. Optional.

executeSqlNonQuery()

> **Note:** A `No Approved SQL` message displays in the **SQL Command to Execute** drop-down list if no SQL commands have been manually added to the Kaseya Server, as described below.

Updates the database—such as updating the value in a column or inserting a row—by running a selected SQL "nonquery" statement. Global "nonquery" statements are specified in the following location: `<C:\Kaseya\xml\SDProcSQL\0\SQLNonQuery\<filename.xml>`.

> **Warning:** Direct manipulation of the Kaseya Server database can result in a massive loss of data.

Filenames can be any name with an `.xml` extension so long as they are formatted correctly internally. Multiple statements specified using one or more XML files display as a single combined combo box list in the user interface. Each SQL statement in the XML file has a unique label, and only the labels are shown in the combo box. If no SQL statements are defined, then `*No Approved SQL*` displays in the combo box.

Partition-Specific Statements

Partition-specific folders can contain partition-specific SQL statements. For example: `<C:\Kaseya\xml\SDProcSQL\123456789\SQLNonQuery\<filename.xml>`. Users can select and run all `0` folder SQL "nonquery" statements and all SQL "nonquery" statements located in the partition path that matches the partition they are using.

Example Format

```
<?xml version="1.0" encoding="utf-8" ?>
<queryList>
   <queryDef label="Update Table" sql="UPDATE table1 SET column2 = value2 WHERE column1 = value1" />
</queryList>
```

executeSqlQuery()

> **Note:** A `No Approved SQL` message displays in the **SQL Command to Execute** drop-down list if no SQL commands have been manually added to the Kaseya Server, as described below.

Returns a value from the database and stores it to a named variable by running a selected SQL "query" statement. Global "query" statements are specified in the following location: `<C:\Kaseya\xml\SDProcSQL\0\SQLQuery\<filename.xml>`

Filenames can be any name with an `.xml` extension so long as they are formatted correctly internally. Multiple statements specified using one or more XML files display as a single combined combo box list in the user interface. Each SQL statement in the XML file has a unique label, and only the labels are shown in the combo box. If no SQL statements are defined, then `*No Approved SQL*` displays in the combo box.

Partition-Specific Statements

Partition-specific folders can contain partition-specific SQL statements. For example: `<C:\Kaseya\xml\SDProcSQL\123456789\SQLQuery\<filename.xml>.` Users can select and run all `0` folder SQL "query" statements and all SQL "query" statements located in the partition path that matches the partition they are using.

Example Format

```
<?xml version="1.0" encoding="utf-8" ?>
<queryList>
    <queryDef label="Agent Guid" sql="SELECT machName FROM dbo.machNameTab WHERE agentGuid =
12345678910111213141516171" />
</queryList>
```

executeSubProcedure()

Causes a **sub-procedure** *(page 79)* to execute. Use this capability to string multiple **IF-ELSE** clauses together. You can nest the execution of procedures to 10 levels.

exitProcedure()

Exits the current procedure. There are no parameters to set.

failCurrentTicketTransaction()

If included in a procedure, cancels all changes to the ticket. Cancels both ticket changes made manually before saving the ticket and ticket changes made within the procedure being processed prior to this step being run. Does not cancel **sendEmail()** or **sendMessage()** steps processed before this step is run. *Cancels all steps in the procedure after this step is run*.

getAgentInfo()

Returns a selected property value from the last audit of the agent machine associated with the ticket.

- **The vMachine property to retrieve from the agent** - See Audit > **Audit Summary** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6779.htm)* for a description of each property.
- **The variable to hold the value of the agent property** - Specify the variable name without brackets. Example: `var_name`.
- **Perform Step On** - Specify the type of operating system to perform this step on.

getVariable()

Defines a new procedure variable and assigns it a value. Specify the variable name without brackets. Example: `var_name`.

You can refer to this variable in text fields in any subsequent steps by adding `[=` and `=]` around the variable name. Example: `[=var_name=]`. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

- **Constant Value** - Enter a value for the variable.

linkTickets()

Links the ticket being processed to a specified ticket ID or knowledge base article ID.

pauseStageGoal()

Pauses counting elapsed time against the current stage **goal** *(page 77)*. Goal procedures are linked to *stages* using the Service Desk > Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab.

pauseTicketGoal()

Pauses counting elapsed time against the ticket **goal** *(page 77)*. The ticket goal determines the due date for the entire ticket. Goals are linked to a service desk using the Desk Definition > Properties > General > **Procedures** *(page 36)* tab.

reOpenTicket()

Changes an End-stage (for example, `Closed`) ticket to a Begin-stage (for example, `New`) ticket. Applies only to a **Ticket Request Mapping** *(page 77)* procedure.

resumeStageGoal()

Resumes counting elapsed time against the current stage **goal** *(page 77)*. Goal procedures are linked to *stages* using the Service Desk > Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab.

resumeTicketGoal()

Resumes counting elapsed time against the ticket **goal** *(page 77)*. The ticket goal determines the due date for the entire ticket. Goals are linked to a service desk using the Desk Definition > Properties > General > **Procedures** *(page 36)* tab.

scheduleProcedure()

Runs a selected *agent procedure* for a specified machine. Use `[$FullMachineName$]` to specify the machine name and machine group, assuming a machine is associated with the ticket. Otherwise you can enter a procedure variable for the machine name or specify a fixed machine name value.

sendEmail()

Sends an email to one or more recipients. Specifies the subject and body text of the email.

Optionally check the **Send the email if the body contains the [$ChangesAsHtml$] or [$Changes$] and those properties are empty due to hidden notes.** This prevents emails with "empty" notes from being sent.

You can include property variables, global variables and procedure variables in the **To**, **From**, **Subject**, and **Body** fields of a **sendEmail()** command. You can reference property variables, global variables and procedure variables in the message templates you link to in a **sendEmail()** command. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

You can also enter the name of a pool of users. Names are delimited using commas or semi-colons. The following order of precedence determines the email addresses of recipients. The name in the To field of a **sendEmail()** step is checked to see if it:

- Exists as a user name. If that name is found, and an email address exists for that user, then that email address is used.
- Conforms to an SMTP email address. For example, `user@domain.com`. If the name matches that pattern, that name is used for the email address. If a service desk pool name conforms to an email address, then email will be sent to that email address, not to the pool of users in that pool.
- The name is looked up as a service desk pool. The email is sent to the email addresses associated with each user in the pool.

sendMessage()

Sends a message to the **Inbox** of a user.

You can only enter one user name in the **To** field of the **sendMessage()** dialog box. Use **sendMessage() to User Pool** *(page 95)* to send a message to multiple users.

You can include property variables, global variables and procedure variables in the **To**, **From**, **Subject**, and **Body** fields of a **sendMessage()** command. You can reference property variables, global variables and procedure variables in the message templates you link to in a **sendMessage()** command. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

sendMessageToPool()

Sends a message to the **Inbox** of all users in a selected pool.

You can include property variables, global variables and procedure variables in the **Subject**, and **Body** fields of a **sendMessageToPool()** command. You can reference property variables, global variables and procedure variables in the message templates you link to in a **Send Message to User Pool** command. See **Ticket Properties and Variables** *(page 68)* for a list of property variables. See **Procedure Variables** *(page 60)* and **Working with Variables** *(page 67)* for more information about variables.

sendPost()

Sends a POST request to a specified URL. The request always sends the content you enter in json format.

- If the command specifies a series of fields and values in json format, then running the command sends a POST request formatted as follows:

```
 {"PartitionId":"<paritionId>",
"ServerName":"<server name","ProcedureName":"<procedure name>",
"MyField1":"<myField1>",
"MyField2":"<myField2>",
... }
```

- If the command specifies just text, then running the command sends a POST request formatted as follows:

```
 {"PartitionId":"<paritionId>",
"ServerName":"<server name","ProcedureName":"<procedure name>",
"PostData":"<Text to be sent>"}
```

The target URL receives this string and processes it.

setEscalationTime()

Sets the time a ticket is escalated. Overrides the default escalation time set for a stage using the Service Desk > Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab. The escalation time can be calculated using the coverage schedule associated with the ticket or using calendar time. For example, if a coverage schedule does not include weekend coverage, and the ticket escalation procedure runs during the weekend, escalation of the ticket will not start counting until coverage begins on Monday morning. If escalation time is calculated using:

- Calendar time, and a coverage schedule is associated with the ticket, calendar time is used.
- A coverage schedule, and no coverage schedule is associated with the ticket, calendar time is used.

setGoalTime()

Sets the time the goal for a stage is scheduled to be completed. Overrides the default goal time set for a stage using the Service Desk > Desk Definition > Processing > Stage > **Procedures** *(page 40)* tab.

setTicketGoalTime()

Sets the time the goal for a ticket is scheduled to be completed. The ticket goal determines the due date for the entire ticket. Overrides the default goal time set for a ticket using the Desk Definition > New

or Edit > **Procedures** *(page 36)* tab.

Service Desk Reports

Data sets are available to support the creation of custom **Service Desk** report definitions and report templates. They are located in the Info Center > Configure & Design > **Report Parts**.

In addition, the following legacy "fixed format" report definitions are provided.

- **Service Desk - Custom Ticket** *(page 96)*
- **Service Desk - Service Goals** *(page 97)*
- **Service Desk - Service Hours** *(page 97)*
- **Service Desk - Service Times** *(page 98)*.
- **Service Desk - Service Volumes** *(page 98)*
- **Service Desk - Tickets** *(page 98)*

Service Desk - Custom Tickets

Info Center > Reporting > Reports > Service Desk - Custom Tickets

- Displays only if the Service Desk add-on module is installed.

The **Custom Tickets** report definition generates a report displaying **Service Desk** ticket summary information and ticket details.

Configure your report definition using the following parameters.

General

- **Service Desk**
- **Notes / Summary / Submitter Filter** - List only tickets or ticket counts containing this string in any note, summary line or submitter information line. Use * for wildcard.
- **Display all Tickets** - If checked, list all tickets individually.
- **Display Notes with each ticket** - If checked, display notes with each ticket.
- **Hide Hidden Notes** - If checked, hide hidden notes.
- **Display Ticket Status Chart for each Admin** - Displays a separate ticket status bar chart for each user plus for unassigned.
- **Display pie chart for each selected Ticket Category Column of Data** - `Assignee`, `Status`, `Priority`, `Category`, `Sub Category`.

Time Range

- **Select the Time Range Type** - Filters by a fixed type of date range.
- **Display all open tickets plus tickets closed within the last N days** - Applies only if `Last N Days` is selected time range type.
- **Custom Start DateTime** - Applies only if `Fixed Range` is select time range type.
- **Custom End DateTime** - Applies only if `Fixed Range` is select time range type.

Columns

Values for all desk definitions are displayed in the drop-down lists. Select multiple items using `Ctrl+Click` and `Shift+Click`, unless otherwise noted.

- **Sort Column** - Select the column to sort tickets on.
- **Sort Direction** - `Ascending`, `Descending`.

Filters

- **Assignee Filter** - Only one item can be selected.
- **Status Filter**
- **Priority Filter**
- **Category Filter**
- **SubCategory Filter** - Only displays subcategories for selected categories in the **Category Filter**.

Service Desk - Service Goals

`Info Center > Reporting > Reports > Service Desk - Service Goals`

- Displays only if the Service Desk add-on module is installed.

The **Service Goals** report definition generates a report displaying summary information and ticket details related to meeting **Service Desk** goals.

Configure your report definition using the following parameters:

Time Selection

- **Select the Time Range Type** - Filters by a fixed type of date range.
- **Number Of Days** - Applies only if `Last N Days` is selected time range type.
- **Custom Start DateTime** - Applies only if `Fixed Range` is select time range type.
- **Custom End DateTime** - Applies only if `Fixed Range` is select time range type.

Parameters

- **Include Only Tickets with Goals** - If checked, only tickets with goals are displayed.
- **Select Report-By Type** - `Service Goals by Ticket`, `Ticket Number`.
- **Sort Column** - Select the column to sort tickets on.
- **Sort Direction** - `Ascending`, `Descending`.

Service Desk - Service Hours

`Info Center > Reporting > Reports > Service Desk - Service Hours`

- Displays only if the Service Desk add-on module is installed.

The **Service Hours** report definition generates a report displaying summary information and ticket details related to **Service Desk** hours worked.

Configure your report definition using the following parameters:

Time Selection

- **Select the Time Range Type** - Filters by a fixed type of date range.
- **Number Of Days** - Applies only if `Last N Days` is selected time range type.
- **Custom Start DateTime** - Applies only if `Fixed Range` is select time range type.
- **Custom End DateTime** - Applies only if `Fixed Range` is select time range type.

Parameters

- **Include Only Tickets with Goals** - If checked, only tickets with goals are displayed.
- **Select Report-By Type** - `Service Hours by Ticket`, `Service Hours by Contributor`, `Service Hours by Organization`.
- **Sort Column** - Select the column to sort tickets on.
- **Sort Direction** - `Ascending`, `Descending`.

Service Desk - Service Times

Info Center > Reporting > Reports > Service Desk - Service Times

- Displays only if the Service Desk add-on module is installed.

The **Service Times** report definition generates a 12-month report, starting with a specified month and year, showing how many tickets have been created, closed, resolved, past due within fixed time buckets.

Configure your report definition using the following parameters:

Parameters

- **Month** - Select a month.
- **Year** - Select a year.
- **Display Tickets Created** - If checked, display tickets created.
- **Display Tickets Closed** - If checked, display tickets closed.
- **Display Tickets Resolved** - If checked, display tickets resolved.
- **Display Tickets Past Due** - If checked, display tickets past due.
- **Display Ticket Service Time Details Tables** - If checked, display tickets detail tables.

Service Desk - Service Volumes

Info Center > Reporting > Reports > Service Desk - Service Volumes

- Displays only if the Service Desk add-on module is installed.

The **Service Volumes** report definition generates a 12-month report, starting with a specified month and year, showing the number of tickets in each month that belong to each possible value in a specified ticket column.

Configure your report definition using the following parameters:

Parameters

- **Group by** - Select the column to group by.
- **Sort Column Direction** - `Ascending`, `Descending`.
- **Month** - Select a month.
- **Year** - Select a year.
- **Display Ticket Volumes Chart** - If checked, display a tickets volumes chart.

Service Desk - Tickets

Info Center > Reporting > Reports > Service Desk - Tickets

- Displays only if the Service Desk add-on module is installed.

The **Tickets** report definition generates a report displaying **Service Desk** ticket summary information and ticket details.

Configure your report definition using the following parameters:

Time Selection

- **Select the Time Range Type** - Filters by a fixed type of date range.

- **Display all open tickets plus tickets closed within the last N days** - Applies only if `Last N Days` is selected time range type.
- **Custom Start DateTime** - Applies only if `Fixed Range` is select time range type.
- **Custom End DateTime** - Applies only if `Fixed Range` is select time range type.

Parameters

- **Notes / Summary / Submitter Filter** - List only tickets or ticket counts containing this string in any note, summary line or submitter information line. Use * for wildcard.
- **Display all Tickets** - If checked, list all tickets individually.
- **Display Notes with each ticket** - If checked, display notes with each ticket.
- **Hide Hidden Notes** - If checked, hide hidden notes.
- **Sort Column** - Select the column to sort tickets on.
- **Sort Direction** - `Ascending`, `Descending`.
- **Display Ticket Status Chart for each Admin** - Displays a separate ticket status bar chart for each user plus for unassigned.
- **Display pie chart for each selected Ticket Category Column of Data** - `Assignee`, `Status`, `Priority`, `Category`, `Sub Category`.

Column Filters

Values for all desk definitions are displayed in the drop-down lists. Select multiple items using `Ctrl+Click` and `Shift+Click`, unless otherwise noted.

- **Assignee Filter** - Only one item can be selected.
- **Status Filter**
- **Priority Filter**
- **Category Filter**
- **SubCategory Filter** - Only displays subcategories for selected categories in the **Category Filter**.

Application Logging

Service Desk > Administration > Application Logging

The **Application Logging** page displays a log of **Service Desk** module activity by:

- **Event ID**
- **Event Name**
- **Message**
- **Admin**
- **Event Date**

This table supports **selectable columns, column sorting, column filtering and flexible columns widths** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#6875.htm)*.

Resend Time Entries

Service Desk > Administration > Resend Time Entries

- This page only displays for master role users.

Master role users can use this page to resend time entries that failed to to be processed from **Service Desk** to the **Time Tracking** module.

Failed time entries will be sent dating back to one week from today unless you change the date below.

If you blank out the date or input an invalid date format, it will default to one week from today as well.

- **Since When** - Enter a date.
- **Run Now** - Resends time entries to the **Time Tracking** module.

KSD API Web Service

Enabling KSD SOAP API

See the **VSA SOAP API** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#3433.htm)* user guide for a general introduction to the Kaseya API.

To enable the KSD API Web Service:

- Display the System > **Configure** *(http://help.kaseya.com/webhelp/EN/VSA/9030000/index.asp#248.htm)* page in the VSA.
- Check the **Enable VSA API Web Service** checkbox.
- Access the KSD API web service using `http://<your-KServer>/vsaWS/vsaServiceDeskWS.asmx`

KSD SOAP API Data Types

The following are the major data types used in the **KSD SOAP API**. These data types are defined in the XML schema document in the `XML\Schemas\ServiceDesk\ServiceDeskDefinition.xsd` file located in the directory where the Kaseya software is installed.

Note: In the descriptions that follow, where the name says `(content)` that means the value is the content of the element.

Legend

- A - AddIncident
- G - GetIncident
- L - ListIncidents
- U - UpdateIncident

RefItem

The **RefItem** describes an item that is a reference item in the service desk. These have an internal database ID value, an internal name, an optional description and the display value.

G	ref	string	The internal name of the item. This is usually prefixed by the service desk name and `||`, such as `Standard||Open`.
G	id	string	The internal database key for the item.
G	description	string	The optional description for the item.
G	(content)	string	The user readable form of the item.

CustomField

The **CustomField** describes the value of a custom field in an incident.

AGU	fieldName	string	The name of the field within the service desk.
AGU	(content)	string	The value of the custom field.

Note

The **Note** describes a single note attached to a ticket.

G	User	string	The name of the user that created the note.
G	Timestamp	dateTime	The time the note was created.
AG	Text	string	The content of the note. This may be HTML formatted, and may include references to attachments.
AG	Hidden	boolean	True if the note should be hidden.
AG	HoursWorked	Decimal	The number of hours worked in this update of the ticket.
AG	SuppressNotify	Boolean	True if notifications for this update should be suppressed.

Attachment

The **Attachment** describes a single attachment for the ticket.

A	Name	string	A unique identifying string for this attachment.
A	DisplayName	string	The name of the attachment as seen by the user.
A	FileName	string	The original name of the file or URL.
A	DocumentType	string	The MIME format of the attachment.
A	Content	Base64Binary	The base 64 encoded content for the attachment.

RelatedIncident

The **RelatedIncident** is another incident that has been related to this current incident

AGU	IncidentNumber	string	The unique identifier for the incident.
G	Summary	string	The summary of the related incident.
G	Status	string	The user readable status of the related incident.
G	Description	string	The description field of the incident.

ServiceDeskDefinition

The following **ServiceDeskDefinition** elements returned describe the desk definition used to edit the ticket. This provides each of the possible values for each field in the ticket.

A single record of the following elements returned.

ServiceDeskDefinition	id="decimal"	A unique identifier.
Name	string	The name of the desk definition.

Description	string	A brief description of the desk definition.
RequireTime	boolean	If true, entering hours worked is required.
DisplayMachineInfo	boolean	If true, machine lookup field is displayed.
RequireMachineInfo	boolean	If true, machine lookup association is required.
DisplayOrgInfo	boolean	If true, organization lookup field is displayed.
RequireOrgInfo	boolean	If true, organization lookup association is required.
DisplayCI	boolean	obsolete
RequireCI	boolean	obsolete
AllAdmins	boolean	obsolete
AutoStartClock	boolean	If true, a clock is automatically started when the user starts to edit the ticket.
AutoSaveClock	boolean	If true, when the ticket is saved, the difference between the current time and the start time is entered as the Hours Worked.
AutoInsertNote	boolean	If true, notes are automatically added to each ticket for the changes made to the ticket.
AutoInsertNoteHidden	boolean	If true, automatically generated notes are made hidden.
NeedStatusNote	boolean	obsolete
SDPrefix	string	The prefix code added to the beginning of the ticket ID.
DefaultStatus	decimal	Default status value. Refers to one of the elements with the matching id attribute in the Status section.
DefaultStage	decimal	Default stage value. Refers to one of the elements with the matching id attribute in the Stage section.
DefaultPriority	decimal	Default priority value. Refers to one of the elements with the matching id attribute in the Priority section.
DefaultSeverity	decimal	Default severity value. Refers to one of the elements with the matching id attribute in the Severity section.
DefaultResolution	decimal	Default resolution value. Refers to one of the elements with the matching id attribute in the Resolution section.
DefaultCategory	decimal	Default category value. Refers to one of the elements with the matching id attribute in the Category section.
DefaultSubCategory	decimal	Obsolete
DefaultServiceDesk	boolean	If true, this is the default service desk, the first one selected when creating new tickets.
TemplateName	string	The template file used to initially create the service desk. Not used otherwise.
TemplateType	int	The type of service desk: 1=ticket, 3=knowledge base.
SequenceName	string	For internal development use only.
EditingTemplate	string	The name of the form used to edit tickets for the service desk.
ShowNotesPane	boolean	If true, notes pane displays in lower pane of Tickets table.
ShowWorkOrders	boolean	If true, display work order and work order line in ticket editor.
ShowSessionTimers	boolean	If true, display session timers in ticket editor.
ShowTasks	boolean	If true, display tasks tab and task related fields.
EstimatedHours	double	Total number of hours worked estimated to resolve this ticket.
ActualHours	double	Total number of hours entered to resolve this ticket.
EmailReader	string	The email reader associated with the service desk.
Administrator	string	The user that is the "desk administrator" of the service desk. The desk

		administrator is notified of certain errors within the service desk.
DefaultPolicy	string	The default policy assigned to the desk.
Status	RefItem	Returns a list of child elements of each Status value in the service desk.
Priority	RefItem	Returns a list of child elements of each Priority value in the service desk.
Severity	RefItem	Returns a list of child elements of each Severity value in the service desk.
Resolution	RefItem	Returns a list of child elements of each Resolution value in the service desk.
TaskStatus	RefItem	Returns a list of child elements of each TaskStatus value in the service desk.
Categories	RefItem	Returns a list of child elements of each Category value in the service desk.
Stages		Returns a list of child elements of each Stage value in the service desk. Each Stage is identified by a Begin, Middle, or End stagetype attribute. Each stage has the following child elements: • Item - The name of the stage. • Initialization - The Stage Entry procedure linked to the stage. • Escalation - The Escalation procedure linked to the stage. Time and Units are specified as attributes. • Goal - The Goal linked to the stage. The Goal procedure linked to the stage. Time and Units are specified as attributes. • NextStage – One of the next stages that this stage may transition to.
Participants	RefItem	The list of users as pools that may be assignees or owners for the service desk.
CurrentContact		Contact information about the user logged on during this transaction. If the user is associated with a staff record, then the `CurrentContact` information is culled from the staff record. If the currently logged on user is a machine user using **Portal Access**, then `CurrentContact` information is culled from the Home > Change Profile tab of **Portal Access**. • ContactName • PhoneNumber • Organization • EmailAddress
SubmitterTypes	string	Type of person submitting the ticket: • UNKNOWN • PARTICIPANT - A participant is a VSA user. • USER - Someone not known to VSA.
CustomFields		Returns zero or more Field elements, each with the following hierarchy: • Caption - Screen caption. • Title - Report title. • Fieldname - Name of the field. • FieldFormat - Data type. • DefaultValue - Default value, if a List data type. • Values - collection element, if a List data type. ○ Item - List item value.

AccessRights		Returns a hierarchy of child elements:
		• ViewHiddenNotes - true or false
		• ChangeHiddenNotes - true or false
		• Field Rights>Field Right - collection elements
		  ○ FieldName - Name of the ticket field
		  ○ AccessType - Required, Edit, View Only, Hidden

NoteTemplates		Returns a list of note templates, each representing standard text that can be added to ticket notes.
ChangeProcedure	string	The Change Ticket procedure associated with the service desk.
GoalProcedure	decimal	The Goal procedure associated with the service desk. • time – the amount of time for goal • unit – The units of time • (content) – the name of the goal procedure.
ResourceTypes		The list of resource types that can be assigned to a ticket.
TaskDefinitions		The list of task values that can be assigned to a task status.
AssocPolicies		The list of policies that can be associated with a ticket.

Incident Summary

The **IncidentSummary** contains the basic description of a ticket.

AGLU	ServiceDeskName	string	The name of the desk definition.
GLU	IncidentNumber	string	The ticket identifier.
AGLU	Summary	string	The ticket summary text.
AGLU	Description	string	The ticket description. text.
AGLU	Status	string	The ref status of the ticket.
AGLU	Priority	string	The ref priority of the ticket.
AGLU	Resolution	string	The ref resolution type of the ticket.
AGLU	Stage	string	The ref stage of the ticket.
AGLU	Severity	string	The ref severity of the ticket.
AGLU	Category	string	The ref category of the ticket.
AGLU	SubCategory	string	The ref subcategory of the ticket.
GL	Policy	string	The policy of the ticket.
GL	CreateDateTime	dateTime	The date time the ticket was created.
GL	LastEditDateTime	dateTime	The date time the ticket was last edited.
GL	CloseDateTime	dateTime	The date time the ticket was closed.
AGLU	OrgID	decimal	Unique identifier of the organization associated with the ticket.
AGLU	OrganizationName	string	The organization name associated withe ticket.
AGLU	Organization	string	The organization ID associated with the ticket.
AGLU	OrganizationStaffName	string	The organization staff member name associated with the ticket.
AGLU	OrganizationStaff	string	The organization staff member unique ID associated with the ticket.
AGLU	OrganizationStaffEmail	string	The email of the organization staff member associated with the ticket.
AGLU	Machine	string	The machine associated with the ticket.
AGLU	MachineGuid	decimal	The GUID of the machine associated with the ticket.
AGLU	MachineGroup	string	The machine group of the machine associated with the ticket.
AGLU	MachineGroupGuid	decimal	The GUID of the machine group associated with the the ticket.
AGLU	Submitter	string	The name of the submitter who submitted the ticket.
AGLU	SubmitterEmail	string	The email of the ticket submitter.
AGLU	SubmitterPhone	string	The phone of the ticket submitter.

AGLU	SubmitterType	string	Type of person submitting the ticket: • UNKNOWN • PARTICIPANT - A participant is a VSA user. • USER - Someone not known to VSA.
GL	IsUnread	boolean	If true, the ticket has not been viewed by the currently logged on user.

Incident

The **Incident** is derived from the **IncidentSummary** and contains all of the fields of the **IncidentSummary** in addition to these fields.

G	IsParticipant	boolean	obsolete
G	IsClosed	boolean	True if closed.
G	CurrentStageEscalation DateTime	dateTime	Stage escalation date and time.
G	CurrentGoalDateTime	dateTime	Stage goal date and time.
AGU	Owner	string	Owner of the ticket.
	Participant	string	obsolete
AGU	AssigneeType	string	Type of assignee: • UNKNOWN • PARTICIPANT - individual assignee • POOL - a pool of users
AGU	Assignee	string	Assignee name.
AGU	AssigneeEmail	string	Assignee email.
G	ActualCompletionDate	dateTime	obsolete
G	ExpectedCompletion Date	dateTime	Date time the ticket is or was expected to be closed, (the ticket goal due date).
G	ActualResolutionDate	dateTime	Date time a resolution type was set for the ticket.
AGU	PromisedDate	dateTime	Date time promise date entered by the customer representative to resolve the ticket.
G	IsArchived	boolean	True if ticket is archived.
G	IsError	boolean	obsolete
G	IsPoolAssignee	boolean	obsolete
	ErrorMessage	string	obsolete
	Notify	boolean	obsolete
G	CurrentStage	string	The current stage.
AGU	ResolutionNote	string	Descriptive text entered with the resolution type.
G	LockTime	dateTime	Date time the ticket was locked by opening the ticket for editing.
G	LockUser	string	User locking the ticket by opening the ticket for editing.
G	StageGoalTime Remaining	int	The time remaining before the stage goal timer executes the goal procedure. Rrelevant when the stage goal has been paused.

AGU	SourceType	string	The source type, either a system event or email, that generated a ticket request. • Email • Backup • KES • Patch • Monitor • Alarm • Portal • ServiceDesk • Other
	OrgAddress/Address	string	Org address 1
	OrgAddress/Address	string	Org address 2
	OrgAddress/City	string	Org city
	OrgAddress/State	string	Org state
	OrgAddress/Zip	string	Org zip
	OrgAddress/Country	string	Org address
AGLU	Field	CustomField	Zero or more custom fields values
AGU	Notes	Note	Zero or more notes.
AGU	Attachments	Attachment	Zero or more attachments
AGU	RelatedIncidents	Related Incident	Zero or more related incidents
	StartDate	datetime	start date/time of the task
	EndDate	datetime	end date/time of the task
	UpdateTime	datetime	last date/time this task was updated
	FollowupDate	datetime	date/time to followup on this task
	CompletionDate	datetime	completion date/time of this task
	ApprovalDate	datetime	approval date/time of this task
	PromiseDate	datetime	promise date/time for this task
	PercentCompletion	int	percent completion of this task
	TaskStatus	string	status of this task
	ActualHours	double	total hours worked for this task
	Resource	Resource	Zero or more resources
	Assignee	string	assignee assigned to this task
	EstimatedHours	decimal	Estimated total hours worked for this ticket.
	TotalHours	decimal	Actual hours worked for this ticket.
	PreviousStage	string	PreviousStage of this ticket.
	WorkPerformedDateTime	datetime	Datetime work was performed on this ticket.
	EditingTemplate	string	Editing template used to edit this ticket.
GU	ServiceDeskDefinition	ServiceDesk Definition	

KSD SOAP API - Operations

The following operations can be performed using the **KSD SOAP API**.

AddIncident

The request is:

AddSDIncident	Incident	The content of the new incident to create. Only fields marked with an A in the first column can be set.
SessionId	Decimal	The session ID.

A single record of the following fields is returned.

IncidentNumber	string	The unique identifier of the ticket.
IncidentID	decimal	The identifier of the ticket.
Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

AddServDeskToScope

The request is:

servDeskName	string	The name of the service desk.
scopeName	string	The name of the scope.
SessionId	decimal	The session ID.

A single record of the following fields is returned.

Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

GetIncident

Retrieves a single incident from the database. The request is:

IncidentRequest		The incident to retrieve. This has the following fields: • `IncidentNumber` – The ticket ID as seen by the user, such as STD000001 • `IncidentId` – The database ID of the ticket to retrieve. • `IncludeNotes` – true to include notes in the retrieved ticket • `IncludeDefinition` – true to include the desk definition in the response • `IncludeAttachment` - true to include attachments in the retrieved ticket

SessionId	Decimal	The SOAP API session ID.

A single record of the following fields is returned.

IncidentResponse	Incident	The retrieved incident.
Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

GetIncidentList

Retrieves a list of incidents matching the request criteria.

The request is:

IncidentListRequest		The list of incidents to retrieve. The list can be filtered using the following elements: • `ServiceDeskName` – The name of the service desk to query. • `Status` – One or more status values to match. If no status values are supplied, then tickets are retrieved regardless of status. • `Priority` – One or more priority values to match. If no priority values are supplied, then tickets are retrieved regardless of priority. • `Stage` – One or more stage values to match. If no stage values are supplied, then tickets are retrieved regardless offstage. • `SummarySearch` – a string or expression to search the summary of tickets. • `Organization` – The name or partial name of organizations to match. If not supplied, then tickets are retrieved for all organizations within the scope. • `OrganizationStaff` – The name of an organizational staff member. associated with tickets. If not supplied, then tickets are retrieved for all organizations within the scope. • `Machine` – The name of a machine to match. If not supplied, then tickets are retrieved for all machines within the scope. • `MachineGroup` – The name of a machine group to match. If not supplied, then tickets are retrieved for all machine groups within the scope. • `Assignee` – The name or partial name of assignees to match. If not supplied, then tickers are retrieved for all assignees within scope. • `StartingIncident` – When paging, this is the next incident number to retrieve. This value comes from the nextStartingIncident value of a previous GetIncidentList request. • `IncidentCount` – When present, specifies the number of incidents to retrieve. • `SortField` – When present, sorts the results on the field name.
SessionId	decimal	The SOAP API session ID.
ReturnDescription	boolean	If true, the response includes a `Description` element for each incident in the list. If false, the `Description` element is not included in the response.

The response is the following:

IncidentList		The list of matching incidents. The following attributes are returned. • `totalIncidents` – The total number of incidents that match the

		request. • `nextStartingIncident` – the Id of the next incident to retrieve. Each incident returns the following elements: • `ServiceDeskName` • `IncidentNumber` • `Summary` - The summary line of the ticket. • `Description` - The description of the incident. Provided only if the request includes the `ReturnDescription` element set to true. • `Status` • `Stage` • `CreateDateTime` • `LastEditDateTime` • `OrgID` • `OrganizationName` • `Machine` • `MachineGroup` • `IsUnread` - The ticket has not yet been opened.
Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

GetIncidentList2

Returns all values in **GetIncidentList** *(page 109)* plus the custom fields and values per incident.

Multiple records of the following field is returned, if applicable.

CustomFields	string or null	The value of the custom field that was specified in the request.

A single record of the following fields is returned.

Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

GetServiceDesk

Retrieves the definition of a service desk. This should be called prior to creating a user interface to allow the user to enter a ticket. The request is:

ServiceDeskDefinitionRequest		The service desk to retrieve. This has the following elements: • ServiceDeskName – The name of the service desk to retrieve. • ServiceDeskID – the database of the service desk to retrieve. Should not be used.
SessionId	Decimal	The SOAP API session ID.

A single record of the following elements returned.

ServiceDeskDefinitionResponse	ServiceDeskDefinition	The retrieved desk definition.
Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

GetServiceDesks

Multiple records of the following fields are returned, if applicable. The request is:

IsDefault	boolean	If true, the service desk is the default service desk.
ServiceDeskID	decimal	A unique identifier.
ServiceDeskName	string	The name of the service desk.

A single record of the following fields is returned.

Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

Primitives

The following primitive Datatype operations are also provided. Each primitive operation uses the same xml contract as their corresponding multiple-columns operation. Each primitive returns a string value that requires subsequent processing. You are strongly discouraged from using these methods.

Primitive	Result	Datatype
PrimitiveAddIncident	PrimitiveAddIncidentResult	string
PrimitiveAddServDeskToScope	PrimitiveAddServDeskToScopeResult	string
PrimitiveGetIncident	PrimitiveGetIncidentResult	string
PrimitiveGetIncidentList	PrimitiveGetIncidentListResult	string
PrimitiveGetServiceDesk	PrimitiveGetServiceDeskResult	string
PrimitiveGetServiceDesks	PrimitiveGetServiceDesksResult	string
PrimitiveUpdateIncident	PrimitiveUpdateIncidentResult	string

QueueAddIncident

Queues an **AddIncident** *(page 108)* request. This is normally used in high volume situations where many tickets are being created via the API in a small amount of time, so the system does not time out. The add incident request gets added to a table and an ongoing event will pull from this table to create tickets, so the request does not need to wait for the ticket to be created.

The request is:

AddSDIncident	Incident	The incident to add.
SessionId	Decimal	The SOAP API session ID.

A single record of the following fields are returned.

Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

UpdateIncident

Updates a single incident in the database. The request is:

UpdateSDIncident	Incident	The incident to update. See the first column of the Incident data type for the fields that are valid on update.
SessionId	Decimal	The SOAP API session ID.

A single record of the following fields is returned.

Method	string	The operation that requested this response.
TransactionID	decimal	The unique message ID for this message.
ErrorMessage	string	If blank, no error was returned.
ErrorLocation	string	If blank, no error was returned.

Sample Messages

Sample data is included in the following XMLs.

GetServiceDesks Request

```
<GetServiceDesks xmlns="vsaServiceDeskWS">
  <req>
    <SessionID>626484243835763212925457 55</SessionID>
  </req>
</GetServiceDesks>
```

GetServiceDesks Response

```
<GetServiceDesksResponse xmlns="vsaServiceDeskWS">
  <GetServiceDesksResult>
    <ServiceDesks>
      <ServiceDesk>
        <IsDefault>false</IsDefault>
        <ServiceDeskID>291273277175176</ServiceDeskID>
        <ServiceDeskName>KnowledgeBase</ServiceDeskName>
      </ServiceDesk>
      <ServiceDesk>
        <IsDefault>false</IsDefault>
        <ServiceDeskID>696191121914314</ServiceDeskID>
        <ServiceDeskName>Standard</ServiceDeskName>
      </ServiceDesk>
    </ServiceDesks>
    <Method>GetServiceDesks</Method>
    <TransactionID>144</TransactionID>
    <ErrorMessage/>
```

```
    <ErrorLocation/>
  </GetServiceDesksResult>
</GetServiceDesksResponse>
```

GetServiceDesk Request

```
<GetServiceDesk xmlns="vsaServiceDeskWS">
  <req>
    <ServiceDeskDefinitionRequest>
      <ServiceDeskName
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard</ServiceDeskName>
      <ServiceDeskID
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">696191121914314</ServiceDeskID>
    </ServiceDeskDefinitionRequest>
    <SessionID>626484243835763212925457 55</SessionID>
  </req>
</GetServiceDesk>
```

GetServiceDesk Response

```
<GetServiceDeskResponse xmlns="vsaServiceDeskWS">
  <GetServiceDeskResult>
    <ServiceDeskDefinitionResponse id="696191121914314">
      <Name xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard</Name>
      <Description xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard
SD</Description>
      <RequireTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</RequireTime>
      <DisplayMachineInfo
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</DisplayMachineInfo>
      <RequireMachineInfo
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</RequireMachineInfo>
      <DisplayOrgInfo
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</DisplayOrgInfo>
      <RequireOrgInfo
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</RequireOrgInfo>
      <DisplayCI xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</DisplayCI>
      <RequireCI xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</RequireCI>
      <AllAdmins xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</AllAdmins>
      <AutoStartClock
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</AutoStartClock>
      <AutoSaveClock
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</AutoSaveClock>
      <AutoInsertNote
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</AutoInsertNote>
      <AutoInsertNoteHidden
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</AutoInsertNoteHidden>
      <NeedStatusNote
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</NeedStatusNote>
      <SDPrefix xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">STD</SDPrefix>
      <DefaultStatus
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">218924116119912</DefaultStatus>
      <DefaultStage
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">831768438118427</DefaultStage>
      <DefaultPriority
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">693719171716599</DefaultPriority>
      <DefaultSeverity
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">0</DefaultSeverity>
      <DefaultResolution
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">0</DefaultResolution>
      <DefaultCategory
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">0</DefaultCategory>
      <DefaultServiceDesk
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</DefaultServiceDesk>
      <TemplateType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">1</TemplateType>
      <SequenceName
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">SEQ129</SequenceName>
```

```
      <EditingTemplate
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Fixed_Width_Tabbed.xml</EditingTe
mplate>
      <Status xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Item ref="Standard||AwaitingHardware" id="541491145218711">Awaiting Hardware</Item>
        <Item ref="Standard||AwaitingUserFeedback" id="281767467828324">Awaiting User Feedback</Item>
        <Item ref="Standard||Closed" id="989295147216226">Closed</Item>
        <Item ref="Standard||Escalated" id="551271771474242">Escalated</Item>
        <Item ref="Standard||Hold" id="172151822788151">Hold</Item>
        <Item ref="Standard||InProgress" id="111313126312233">In Progress</Item>
        <Item ref="Standard||New" id="218924116119912">New</Item>
      </Status>
      <Priority xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Item ref="Standard||CriticalHigh" id="744512181719881">Critical High</Item>
        <Item ref="Standard||High" id="982525519923522">High</Item>
        <Item ref="Standard||Low" id="291721863176342">Low</Item>
        <Item ref="Standard||Medium" id="693719171716599">Medium</Item>
        <Item ref="Standard||Planning" id="176222131631332">Planning</Item>
      </Priority>
      <Severity xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Item ref="Standard||CompanyWide(High)" id="315477225242249">Whole Company (High)</Item>
        <Item ref="Standard||MultipleUsers(Medium)" id="262164368749722">Multiple users (Medium)</Item>
        <Item ref="Standard||OneUser(Low)" id="917688316816914">Single User (Low)</Item>
      </Severity>
      <Resolution xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Item ref="Standard||AdviceGiven" id="498162732192611">Advice Given</Item>
        <Item ref="Standard||CannotDuplicate" id="262514419248621">Cannot Duplicate</Item>
        <Item ref="Standard||ClosedbyCustomerRequest" id="525192125718333">Closed by Customer
Request</Item>
        <Item ref="Standard||HardwareReplaced" id="432262321578326">Hardware Replaced</Item>
        <Item ref="Standard||HotFixReleased" id="189239616133249">Hot Fix Released</Item>
        <Item ref="Standard||InstallationCompleted" id="139764799836252">Installation Completed</Item>
        <Item ref="Standard||NewSoftwareInstalled" id="521637923418319">New Software Installed</Item>
        <Item ref="Standard||Noresponsefromuser" id="115424612244857">No response from user</Item>
        <Item ref="Standard||OSReinstalled" id="531617444692623">OS Reinstalled</Item>
        <Item ref="Standard||Other" id="711261961631328">Other</Item>
        <Item ref="Standard||PassedtoSales" id="191482475814123">Passed to Sales</Item>
        <Item ref="Standard||Pendingscriptcleared" id="762515513181192">Pending script cleared</Item>
        <Item ref="Standard||ReapplySchema" id="525317525441497">Reapply Schema</Item>
        <Item ref="Standard||Reboot" id="832182442825238">Reboot</Item>
        <Item ref="Standard||ResolvedbyCustomer" id="243623591961272">Resolved by Customer</Item>
        <Item ref="Standard||ResolvedbyTechnition" id="423939164212169">Resolved</Item>
        <Item ref="Standard||SolvedwithKBarticle" id="272199179212412">Solved with KB article</Item>
        <Item ref="Standard||TrainingGiven" id="622224812237126">Training Given</Item>
      </Resolution>
      <Categories xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Category>
          <Item ref="Standard||Advice&amp;amp;Guidance" id="161211171768212">Advice &amp;amp; Guidance</Item>
          <SubCategory ref="Standard||Advice&amp;amp;Guidance||General"
id="561699795215782">General</SubCategory>
        </Category>
        <Category>
          <Item ref="Standard||Kaseya" id="641881726251641">Kaseya</Item>
          <SubCategory ref="Standard||Kaseya||AgentIcon" id="821781865922435">Agent Icon</SubCategory>
          <SubCategory ref="Standard||Kaseya||Alarm" id="481422361723261">Alarm</SubCategory>
          <SubCategory ref="Standard||Kaseya||ApplicationChanges" id="525187874623717">Application
Changes</SubCategory>
          <SubCategory ref="Standard||Kaseya||Disk" id="919621482151882">Disk</SubCategory>
          <SubCategory ref="Standard||Kaseya||Eventlog" id="814714713317798">Eventlog</SubCategory>
          <SubCategory ref="Standard||Kaseya||GetFile" id="322618792314914">Get File</SubCategory>
          <SubCategory ref="Standard||Kaseya||Hardware" id="176166136238942">Hardware</SubCategory>
          <SubCategory ref="Standard||Kaseya||Lanwatch" id="214791394922624">Lanwatch</SubCategory>
          <SubCategory ref="Standard||Kaseya||Logon_Admin"
id="943315515116292">Logon_Admin</SubCategory>
          <SubCategory ref="Standard||Kaseya||Logon_User"
id="636613429245187">Logon_User</SubCategory>
          <SubCategory ref="Standard||Kaseya||NewAgent" id="557214511134217">New Agent</SubCategory>
          <SubCategory ref="Standard||Kaseya||Other" id="631281678197153">Other</SubCategory>
          <SubCategory ref="Standard||Kaseya||PatchManagement" id="462824113621914">Patch
```

```
Management</SubCategory>
          <SubCategory ref="Standard||Kaseya||Procedure" id="274262311559714">Procedure</SubCategory>
          <SubCategory ref="Standard||Kaseya||RCDisabled" id="641624812335116">RC
Disabled</SubCategory>
          <SubCategory ref="Standard||Kaseya||Script" id="471482131991414">Script</SubCategory>
          <SubCategory ref="Standard||Kaseya||SystemOffline" id="113411182222324">System
Offline</SubCategory>
          <SubCategory ref="Standard||Kaseya||SystemOnline" id="251814418923368">System
Online</SubCategory>
          <SubCategory ref="Standard||Kaseya||Unidentified"
id="617313577253122">Unidentified</SubCategory>
        </Category>
        <Category>
          <Item ref="Standard||Network" id="414766231875111">Network</Item>
          <SubCategory ref="Standard||Network||Connectivity"
id="122145211361321">Connectivity</SubCategory>
          <SubCategory ref="Standard||Network||Design" id="495611529142242">Design</SubCategory>
          <SubCategory ref="Standard||Network||Firewall" id="812515316323522">Firewall</SubCategory>
          <SubCategory ref="Standard||Network||Other" id="946227769167531">Other</SubCategory>
          <SubCategory ref="Standard||Network||Performance"
id="941891772111717">Performance</SubCategory>
        </Category>
        <Category>
          <Item ref="Standard||Printer" id="155243642251342">Printer</Item>
          <SubCategory ref="Standard||Printer||Other" id="341431321188813">Other</SubCategory>
          <SubCategory ref="Standard||Printer||PrinterProblem" id="851831547314111">Printer
Problem</SubCategory>
          <SubCategory ref="Standard||Printer||PrinterSetup" id="619395216749723">Printer
Setup</SubCategory>
          <SubCategory ref="Standard||Printer||Toner" id="161984536861723">Toner</SubCategory>
        </Category>
        <Category>
          <Item ref="Standard||ServiceRequest" id="541124124415221">Service Request</Item>
          <SubCategory ref="Standard||ServiceRequest||EquipmentMove" id="862712311517672">Equipment
Move</SubCategory>
          <SubCategory ref="Standard||ServiceRequest||NewLaptop" id="266812518245792">New
Laptop</SubCategory>
          <SubCategory ref="Standard||ServiceRequest||NewServer" id="322872913227349">New
Server</SubCategory>
          <SubCategory ref="Standard||ServiceRequest||NewWorkstation" id="224115236352441">New
Workstation</SubCategory>
        </Category>
      </Categories>
      <Stages xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Stage stageType="End">
          <Item ref="Standard||Closed" id="213813735111171" description="Auto Generated">Closed</Item>
          <Initialization>Standard Enters Closed</Initialization>
        </Stage>
        <Stage stageType="Begin">
          <Item ref="Standard||Identified" id="831768438118427" description="New ticket is
received">Identified</Item>
          <Initialization>Standard Enters Identified</Initialization>
          <Escalation time="15" unit="MINUTE">Incident is Escalated</Escalation>
          <Goal time="1" unit="HOUR">Identified Goal</Goal>
          <NextStage ref="Standard||Tier1" id="546812745461511" description="Tier 1
Support">Tier1</NextStage>
        </Stage>
        <Stage stageType="Middle">
          <Item ref="Standard||Tier1" id="546812745461511" description="Tier 1 Support">Tier1</Item>
          <Initialization>Standard Enters Tier1</Initialization>
          <Escalation time="3" unit="HOUR">Incident is Escalated</Escalation>
          <Goal time="2" unit="HOUR">Tier1 Goal</Goal>
          <NextStage ref="Standard||Closed" id="213813735111171" description="Auto
Generated">Closed</NextStage>
          <NextStage ref="Standard||Tier2" id="318527191192719" description="Tier 2 Specialist
Support">Tier2</NextStage>
        </Stage>
        <Stage stageType="Middle">
          <Item ref="Standard||Tier2" id="318527191192719" description="Tier 2 Specialist
```

```
Support">Tier2</Item>
          <Initialization>Standard Enters Tier2</Initialization>
          <Escalation time="3" unit="HOUR">Incident is Escalated</Escalation>
          <Goal time="4" unit="HOUR">Tier2 Goal</Goal>
          <NextStage ref="Standard||Closed" id="213813735111171" description="Auto
Generated">Closed</NextStage>
        </Stage>
      </Stages>
      <Participants xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Participant ref="garyw" id="675118836391351128914163l3" isPool="false">garyw</Participant>
        <Participant ref="jschenck" id="72381729521421633172123416"
isPool="false">jschenck</Participant>
        <Participant ref="NickT" id="961719213153499239246342491" isPool="false">NickT</Participant>
        <Participant ref="Standard||SupportManager" id="654222596258293" isPool="true">SupportManager
(Pool)</Participant>
        <Participant ref="Standard||Tier1Support" id="352161952139188" isPool="true">Tier1Support
(Pool)</Participant>
        <Participant ref="Standard||Tier2Support" id="921522231318131" isPool="true">Tier2Support
(Pool)</Participant>
      </Participants>
      <CustomFields xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Field id="221552971661261">
          <Caption>Source</Caption>
          <Title>Source</Title>
          <FieldName>Source</FieldName>
          <FieldFormat>List</FieldFormat>
          <DefaultValue>Call</DefaultValue>
          <Values>
            <Item ref="Call" id="0">Call</Item>
            <Item ref="EMail" id="0">EMail</Item>
            <Item ref="Text" id="0">Text</Item>
          </Values>
        </Field>
        <Field id="818831117157241">
          <Caption>Urgency</Caption>
          <Title>Urgency</Title>
          <FieldName>Urgency</FieldName>
          <FieldFormat>List</FieldFormat>
          <DefaultValue>Medium</DefaultValue>
          <Values>
            <Item ref="High" id="0">High</Item>
            <Item ref="Low" id="0">Low</Item>
            <Item ref="Medium" id="0">Medium</Item>
          </Values>
        </Field>
        <Field id="513119818455188">
          <Caption>KB Article created</Caption>
          <Title>KB Article Created</Title>
          <FieldName>KB_Article</FieldName>
          <FieldFormat>List</FieldFormat>
          <DefaultValue>No</DefaultValue>
          <Values>
            <Item ref="No" id="0">No</Item>
            <Item ref="Yes" id="0">Yes</Item>
          </Values>
        </Field>
        <Field id="291214644251233">
          <Caption>Dept</Caption>
          <Title>Department</Title>
          <FieldName>Dept</FieldName>
          <FieldFormat>List</FieldFormat>
          <DefaultValue>IT</DefaultValue>
          <Values>
            <Item ref="Accounting" id="0">Accounting</Item>
            <Item ref="Accounts Payable" id="0">Accounts Payable</Item>
            <Item ref="Facilities" id="0">Facilities</Item>
            <Item ref="HR" id="0">HR</Item>
            <Item ref="IT" id="0">IT</Item>
            <Item ref="Other" id="0">Other</Item>
```

```
          <Item ref="Payroll" id="0">Payroll</Item>
          <Item ref="Sales" id="0">Sales</Item>
          <Item ref="Telecom" id="0">Telecom</Item>
        </Values>
      </Field>
    </CustomFields>
    <AccessRights xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
      <ViewHiddenNotes>true</ViewHiddenNotes>
      <ChangeHiddenNotes>true</ChangeHiddenNotes>
      <FieldRights>
        <FieldRight>
          <FieldName>ID</FieldName>
          <AccessType>Required</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Summary</FieldName>
          <AccessType>Required</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Description</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>CreationDtTm</FieldName>
          <AccessType>ViewOnly</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>SubmitterName</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>SubmitterEmailAddr</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>ContactPhone</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>OrgName</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>OrgID</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>StaffID</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>ContactEmail</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>MachineID</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Note</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>ClosedDtTm</FieldName>
          <AccessType>ViewOnly</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>PromiseDtTm</FieldName>
          <AccessType>Edit</AccessType>
```

```
        </FieldRight>
        <FieldRight>
          <FieldName>DueDtTm</FieldName>
          <AccessType>ViewOnly</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>ActualCompletedDate</FieldName>
          <AccessType>ViewOnly</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>HiddenNote</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Owner</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>LockUser</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>EditDtTm</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>current_esc_datetime</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>current_goal_datetime</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>lockTime</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>sourceType</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Status</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Priority</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Severity</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Category</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>SubCategory</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Stage</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
        <FieldRight>
          <FieldName>Resolution</FieldName>
          <AccessType>Edit</AccessType>
        </FieldRight>
```

```
          <FieldRight>
            <FieldName>Assignee</FieldName>
            <AccessType>Edit</AccessType>
          </FieldRight>
          <FieldRight>
            <FieldName>Source</FieldName>
            <AccessType>Edit</AccessType>
          </FieldRight>
          <FieldRight>
            <FieldName>Urgency</FieldName>
            <AccessType>Edit</AccessType>
          </FieldRight>
          <FieldRight>
            <FieldName>KB_Article</FieldName>
            <AccessType>Edit</AccessType>
          </FieldRight>
          <FieldRight>
            <FieldName>Dept</FieldName>
            <AccessType>Edit</AccessType>
          </FieldRight>
        </FieldRights>
      </AccessRights>
      <NoteTemplates xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Item ref="My Note" id="196429316815241">My Note</Item>
        <Item ref="Note 2" id="167218821431219">Second note</Item>
      </NoteTemplates>
      <ChangeProcedure xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard is
Changed</ChangeProcedure>
      <GoalProcedure time="1" unit="DAY"
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard Goal - All
Stages</GoalProcedure>
    </ServiceDeskDefinitionResponse>
    <Method>GetServiceDesk</Method>
    <TransactionID>146</TransactionID>
    <ErrorMessage/>
    <ErrorLocation/>
  </GetServiceDeskResult>
</GetServiceDeskResponse>
```

GetIncidentList Request

```
<GetIncidentList xmlns="vsaServiceDeskWS">
  <req>
    <IncidentListRequest>
      <ServiceDeskName
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard</ServiceDeskName>
      <IncidentCount
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">30</IncidentCount>
    </IncidentListRequest>
    <SessionID>626484243835763212925457 55</SessionID>
    <ReturnDescription>true</ReturnDescription>
  </req>
</GetIncidentList>
```

GetIncidentList Response

```
<GetIncidentListResponse xmlns="vsaServiceDeskWS">
  <GetIncidentListResult>
    <IncidentList>
      <Incident xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <ServiceDeskName>Standard</ServiceDeskName>
        <IncidentNumber>STD000001</IncidentNumber>
        <Summary>Monitoring generated Counter ALARM at 7:03:04 am 05-Feb-10 on
ag-yellow-w732.root.myOrg</Summary>
   <Description>Monitoring generated Counter ALARM at 7:03:04 am 05-Feb-10 on
ag-yellow-w732.root.myOrg<br/><br/>SNMP Device: N/A<br/>Monitor Set: _ZC-CM2-Disk Drive C Free
Space_Copy<br/>Type: Counter<br/>Log Object Name: DiskSpace...</Description>
        <Status>Closed</Status>
```

```
        <Priority>Low</Priority>
        <Stage>Closed</Stage>
        <CreateDateTime>2010-02-05T17:07:21.55-08:00</CreateDateTime>
        <LastEditDateTime>2010-02-05T22:59:22.64-08:00</LastEditDateTime>
        <Submitter>Kaseya Support</Submitter>
        <SubmitterEmail>noreply@kaseya.com</SubmitterEmail>
      </Incident>
    </IncidentList>
    <Method>GetIncidentList</Method>
    <TransactionID>147</TransactionID>
    <ErrorMessage/>
    <ErrorLocation/>
  </GetIncidentListResult>
</GetIncidentListResponse>
```

GetIncident Request

```
<GetIncident xmlns="vsaServiceDeskWS">
  <req>
    <IncidentRequest>
      <IncidentNumber
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">STD000001</IncidentNumber>
      <IncludeNotes
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</IncludeNotes>
      <IncludeDefinition
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</IncludeDefinition>
    </IncidentRequest>
    <SessionID>672232251143169126734902 69</SessionID>
  </req>
</GetIncident>
```

GetIncident Response

```
<GetIncidentResponse xmlns="vsaServiceDeskWS">
  <GetIncidentResult>
    <IncidentResponse id="611922114996841">
      <IncidentNumber
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">STD000001</IncidentNumber>
      <Summary xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Getting Started with
Service Desk - PLEASE READ!</Summary>
      <Description xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
    &lt;p&gt;&lt;strong&gt;&lt;span
style='font-size:11.0pt;font-family:"Calibri","sans-serif";color:blue'&gt;WELCOME TO SERVICE
DESK&lt;/span&gt;&lt;/strong&gt;&lt;br/&gt;
    Your Service Desk module has been pre-configured with a template-driven Standard service desk, and
a Knowledge Base desk. Only a few short customization steps are required to use these desks immediately.
See &lt;a href="http://help.kaseya.com/WebHelp/EN/KSD/1000000/index.htm?toc.htm?5982.htm"&gt;Getting
Started&lt;/a&gt; to quickstart your implementation of Service Desk.
    &lt;/p&gt;
  </Description>
      <Status
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Closed</Status>
      <Priority
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Low</Priority>
      <Stage
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Closed</Stage>
      <Category
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Advice&amp;Guidance</Ca
tegory>
      <CreateDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-02-05T17:07:21.55-08:00</Cre
ateDateTime>
      <LastEditDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-02-05T22:59:22.64-08:00</Las
tEditDateTime>
      <Submitter xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Kaseya
Support</Submitter>
      <SubmitterEmail
```

```
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">noreply@kaseya.com</SubmitterEmai
l>
      <SubmitterType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">UNKNOWN</SubmitterType>
      <IsUnread xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</IsUnread>
      <IsParticipant
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsParticipant>
      <Owner xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">garyw</Owner>
      <AssigneeType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">POOL</AssigneeType>
      <Assignee
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Tier1Support</Assignee>
      <ActualCompletionDate
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-02-05T22:59:29.28-08:00</Act
ualCompletionDate>
      <ExpectedCompletionDate
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-02-06T17:07:22.283-08:00</Ex
pectedCompletionDate>
      <IsArchived
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsArchived>
      <IsError xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsError>
      <Notify xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</Notify>
      <SourceType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">ServiceDesk</SourceType>
      <CustomFields xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Field fieldName="Source">Text</Field>
        <Field fieldName="Urgency">Low</Field>
        <Field fieldName="KB_Article">No</Field>
        <Field fieldName="Dept">Sales</Field>
      </CustomFields>
      <Notes xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Note id="213494962391116">
          <Timestamp>2010-02-05T22:59:25.127-08:00</Timestamp>
          <Text>Auto Generated Note:&lt;br/&gt;
Ticket Changed&lt;br/&gt;    'currentStageGoalDateTime' cleared&lt;br/&gt;</Text>
          <Hidden>true</Hidden>
        </Note>
        <Note id="356934215185622">
          <User>garyw</User>
          <Timestamp>2010-02-05T17:07:21.55-08:00</Timestamp>
          <Text>Auto Generated Note:&lt;br/&gt;
Ticket Added&lt;br/&gt;</Text>
          <Hidden>true</Hidden>
        </Note>
      </Notes>
    </IncidentResponse>
    <Method>GetIncident</Method>
    <TransactionID>200</TransactionID>
    <ErrorMessage/>
    <ErrorLocation/>
  </GetIncidentResult>
</GetIncidentResponse>
```

AddIncident Request

```
<AddIncident xmlns="vsaServiceDeskWS">
  <req>
    <AddSDIncident>
      <ServiceDeskName
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard</ServiceDeskName>
      <Summary xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Test Ticket From Web
Service</Summary>
      <Description xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">This ticket was
created with the web service.</Description>
      <Status
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||New</Status>
      <Priority
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Medium</Priority>
```

```
      <Category
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Network</Category>
    </AddSDIncident>
    <SessionID>672232251143169126734902 69</SessionID>
  </req>
</AddIncident>
```

AddIncident Response

```
<AddIncidentResponse xmlns="vsaServiceDeskWS">
  <AddIncidentResult>
    <AddSDIncidentResponse>
      <IncidentNumber
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">STD000002</IncidentNumber>
      <IncidentID
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">249259141859248</IncidentID>
    </AddSDIncidentResponse>
    <Method>AddIncident</Method>
    <TransactionID>203</TransactionID>
    <ErrorMessage/>
    <ErrorLocation/>
  </AddIncidentResult>
</AddIncidentResponse>
```

Update Incident Request

```
<UpdateIncident xmlns="vsaServiceDeskWS">
  <req>
    <UpdateSDIncident id="894212819800719301574914 35">
      <ServiceDeskName
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Customer_SD_Basic</ServiceDeskNam
e>
      <IncidentNumber
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">CSN000380</IncidentNumber>
      <Summary xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Test Ticket From Web
Service</Summary>
      <Description xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">This ticket was
created with the web service.</Description>
      <Status
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||InProgress</Status>
      <Priority
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Low</Priority>
      <Stage
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Standard||Identified</Stage>
      <CreateDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-03-10T21:07:31.923-08:00</Cr
eateDateTime>
      <LastEditDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-03-10T21:07:31.923-08:00</La
stEditDateTime>
      <Submitter xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">garyw</Submitter>
      <SubmitterType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">UNKNOWN</SubmitterType>
      <IsUnread xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">true</IsUnread>
      <IsParticipant
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsParticipant>
      <CurrentStageEscalationDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-03-10T21:22:43.063-08:00</Cu
rrentStageEscalationDateTime>
      <CurrentGoalDateTime
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-03-10T22:07:43.077-08:00</Cu
rrentGoalDateTime>
      <Owner xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">garyw</Owner>
      <AssigneeType
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">POOL</AssigneeType>
      <Assignee
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">Tier1Support</Assignee>
      <ExpectedCompletionDate
```

```
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">2010-03-11T21:07:43.077-08:00</Ex
pectedCompletionDate>
      <IsArchived
xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsArchived>
      <IsError xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</IsError>
      <Notify xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">false</Notify>
      <Notes xmlns="http://www.kaseya.com/vsa/2007/12/ServiceDeskDefinition.xsd">
        <Note id="281273717819319">
          <User>garyw</User>
          <Timestamp>2010-03-10T21:07:31.923-08:00</Timestamp>
          <Text>Auto Generated Note:&lt;br/&gt; Ticket Added&lt;br/&gt;</Text>
          <Hidden>true</Hidden>
        </Note>
      </Notes>
    </UpdateSDIncident>
    <SessionID xmlns="">98782788528483188965186776</SessionID>
  </req>
</UpdateIncident>
```

UpdateIncident Response

```
<UpdateIncidentResponse xmlns="vsaServiceDeskWS">
  <UpdateIncidentResult>
    <Method>UpdateIncident</Method>
    <TransactionID>205</TransactionID>
    <ErrorMessage/>
    <ErrorLocation/>
  </UpdateIncidentResult>
</UpdateIncidentResponse>
```

Glossary

Default Field Permissions

Portal Access users (machine users) *(page 9)* use ticket field level permissions defined for the `Default` machine role. The `Default` machine role also applies to VSA users using a user role that does not include either the `Service Desk Administrator` or `Service Desk Technician` roletype. When a VSA user is using the `Default` machine role to view or edit a ticket, a `Default Permissions Apply` message displays at the top of a service desk ticket. If even the `Default` machine role does not provide access to a ticket, then an error message tells the user their role does not permit access to the ticket.

Desk Definition Templates

To help you get started quickly, new desk definitions are created in Service Desk by copying them from one of several **desk definition templates**, each configured to support a typical business process. The template you select determines whether *tickets* or *KB articles* are created, based on the desk definition.

- `Blank KnowledgeBase` - Creates a blank knowledge base desk definition that contains no pre-defined values for any standard fields.
- `Blank Ticket` - Creates a blank desk definition that contains no pre-defined values for any standard fields.
- `ITIL ChangeRequests` - A change request desk, based on ITIL guidelines, used to track detailed requirements for changes or enhancements to a product or service, usually in response to a problem. The `ChangeRequest` (page 47) service desk is created from this template.
- `ITIL Incident` - An advanced service desk, based on ITIL guidelines. The `Incidents` (page 44) service desk is created from this template. Integrates with pre-defined sample user roles, user pools, organization types, procedure variables, message templates, policies, coverage schedules, and holidays. Additional configuration is required to use this service desk.
- `ITIL KnowledgeBase` - A knowledge base desk based on ITIL guidelines. The `Knowledge Base` (page 47) desk is created from this template. Serves as a repository of known errors and recommended solutions.
- `ITIL Problem` - A service desk, based on ITIL guidelines, used to track broadly-based product or service problems separately from immediate customer support issues. The `Problem` (page 46) service desk is created from this template.
- `Service_Desk_Standard` - Provides basic automation with a simplified workflow. The `Customer_SD_Automation (page 43)` service desk is created from this template. Minimal configuration is required to use this service desk.

Editing Template

An **editing template** serves three purposes:

1. The editing template defines the layout of the dialog used to edit a ticket or KB article.
2. An editing template may *mask* selected fields, even though the fields are defined by the desk definition. *Using an editing template to mask a field overrides whatever field permission is set for that field.*
3. The editing templates also sets default **field permissions** *(page 126)* for editing a ticket or KB article. Whether assigned by role or by user, you can override the default field permissions set by the editing template to suit your business requirements.

An editing template is applied to a combination of desk definition and user role (or machine role) using **Role Preferences** *(page 52)* or the Desk Definition > Access > **Roles** *(page 41)* tab. An editing template can also be applied to a combination of desk definition and user using **User Preferences** *(page 55)*. User Preferences has precedence over Role Preferences. The default editing template for all roles and

all users working with a service desk is specified in the Service Desk > Desk Definition > New or Edit > **General Info** *(page 34)* tab.

Field Permissions

Field permissions are set by role. For VSA users using roles linked to the `Service Desk Technicians` role type, **field permissions** *determine what fields a user can view or edit within the ticket editor or KB article editor.* Typical field permissions include: `Editable`, `View Only`, `Hidden`, or `Required`. Default field permissions are set by **editing template** *(page 125)*.

> **Note:** VSA users using a role linked to the `Service Desk Administrators` role type can see and work with any field in any ticket editor or KB article editor. `Master` role users also always have complete field permission access, regardless of roletype assignment.

Machine Group

Machines are always defined by **machine group** and machine groups are always defined by organization. You can define multi-level hierarchies of machine groups by identifying a parent machine group for a machine group. You can also move a machine group and all of its associated machines to a different parent machine group within the same organization.

Managed Machine

A monitored machine with an installed agent and active machine ID / group ID account on the Kaseya Server. Each managed machine uses up one agent license.

myOrg

`myOrg` is the **organization** *(page 126)* of the service provider using the VSA. All other organizations in the VSA are second party organizations doing business with `myOrg`. The default name of `myOrg`, called `My Organization`, should be renamed to match the service provider's company or organization name. *This name displays at the top of various reports to brand the report.* Agents installed to internally managed machines can be assigned to this organization. *VSA user logons are typically associated with staff records in the* `myOrg` *organization.* `myOrg` cannot be assigned a parent organization.

On Premises

An **on premises** hardware/software installation of the VSA is a maintained by a service provider and typically used only by the service provider. See **Software as a Service (SaaS)** *(page 127)*.

Org

The VSA supports three different kinds of business relationships:

- **Organizations** - Supports machine groups and manages machines using agents.
- **Customers** - Supports the billing of customers using **Service Billing**.
- **Vendors** - Supports the procurement of materials using **Service Billing**.

The `Org` table is a support table shared by *organizations*, *customers* and *vendors*. Each record in the `Org` table is identified by a unique `orgID`. The `Org` table contains basic information you'd generally need to maintain about any kind of business relationship: mailing address, primary phone number, duns number, yearly revenue, etc. Because the `Org` table is shared, you can easily convert:

- A customer into an organization or vendor.
- A vendor into an organization or customer.
- An organization into a customer or vendor.

> **Note:** `myOrg` (page 126) is the organization of the service provider using the VSA.

Pools

Users can be assigned to **pools** of users, regardless of the roles they belong to. Tickets can be

assigned to pools using the **Assigned To** field in the editor, or by procedure using the `assignTicketProperty() > Pool` command. When a user in that pool opens the ticket, the ticket is automatically *reassigned* to the individual user.

Resource Type

A **resource type** sets a default rate for a billable labor item or entry. Typically a resource type represents a skill performed by a staff member. A billing rate and standard cost is defined for each resource type required to perform the service. The rate can be overridden when selected. The classification of resource types typically reflects the production requirements of a company. Labor entries are classified by both resource type and by activity type.

Software as a Service (SaaS)

Kaseya provides "software as a service" (SaaS) deployment of **Virtual System Administrator™**. Service providers contract with Kaseya to access a VSA hosted and maintained by Kaseya and can install a specified number of their customer agents. Service providers are allocated a unique *tenant partition* of a shared Kaseya Server and database. Within their assigned partition, service providers can only see their own organizations, machine groups, procedures, reports and tickets. Service providers in a tenant partition have full access to all the functions of the VSA except system maintenance, which is the responsibility of Kaseya. See **On Premises** *(page 126)*.

Stop Notification and Email Send Delay

Pending email notification of selected tickets can be stopped using **Stop Notification** in **Tickets** *(page 15)*. This option is used in conjunction with **Email Send Delay** in **Role Preferences** *(page 52)* and **User Preferences** *(page 55)*. For example, if **Email Send Delay** is set to `5` minutes for the role you are using, then you have `5` minutes to cancel any emails that are pending for a ticket using **Stop Notification**. Applies to ticket rows displaying a  icon in the icon column of the ticket table. **Stop Notification** only stops currently pending email.

Task

Tasks organize a **Service Desk** ticket into a series of steps. Each task can be assigned a different assignee and tracked individually for completion. You can add notes and hours worked by individual task. All tasks for all tickets can be listed, sorted, filtered, searched and viewed using the **Tasks Associated with Tickets** *(page 29)* page.

Ticket

A **ticket** tracks the detection, reporting, and resolution of an issue. Every communication and action taken on behalf of the issue can be recorded in the ticket record.

Ticket Requests

Ticket requests are created from either inbound emails or system events, such as alert conditions. **Incoming Email and Alarm Settings** *(page 56)* and linked service desk procedures determine how ticket requests are processed into tickets, what service desks are used to create the tickets, and whether ticket requests are canceled.

Work Order

A **work order** specifies the delivery of a non-recurring service (billable item). A work order lets you estimate and review costs *before you create billable entries*. Work orders can be converted from a sales order, but a sales order is not required. The four types of billable items you can include on a work order are *labor*, *parts*, *expenses* or *general*.

Work Types

Work types determine how time entries are integrated with other functions in the VSA. The work type options displayed in your VSA depend on the modules installed.

- **Admin Tasks** - A recurring operational activity not associated with any project.

- **Work Orders** - Only displays if the **Service Billing** is installed.
- **Service Desk Tickets** - Only displays if **Service Desk** 1.3 or later is installed.

Index

N

O

P

Q

R

S

T

U

V

W