Genesys Knowledge Center Deployment Guide

Knowledge Center 8.5.3

3/20/2023

Table of Contents

Genesys Knowledge Center Deployment Guide

What is Genesys Knowledge Center

Provides an overview of the Knowledge Center functionality and architecture:

What is Genesys Knowledge Center.

Genesys Knowledge Center Components

High-level architecture

Terminology

Planning your Deployment

Describes major considerations while planning a deployment of your Knowledge Center cluster:

Prerequisites

Sizing Information

Hardware Recommendations

Software Configuration

Installation and Deployment

Step-by-step guide of Knowledge Center service deployment (see main page for full list):

Before You Begin

Installing the Cluster Application

Installing the Server

Installing the GMS

Post Installation and Deployment

Describes tuning options and additional functionalities to tune the Knowledge Center to your business needs:

Access Permissions

Configuration Options

Load Balancing Configuration

Security

GEO Location

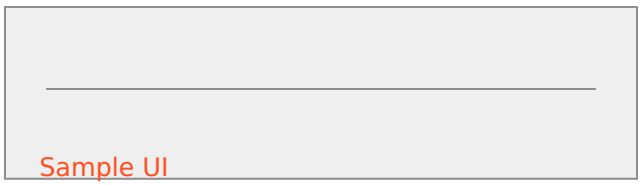
UTF8

Supported Languages

Knowledge Center in Production

Tips & tricks on operating your Knowledge Center cluster in production:

Monitoring Knowledge Center



Sample UI

Importing Data into the Knowledge Center Server

What is Genesys Knowledge Center?

Genesys Knowledge Center allows you to make the best use of your enterprise knowledge by capturing, storing, and distributing it wherever it is needed. Let's take a closer look at the various capabilities of Knowledge Center and some corresponding use cases.

Knowledge-assisted Channels

With Knowledge Center, you can:

- Knowledge-enable channels by providing the right answers to customers in-channel to deflect interactions, leading to cost reduction and better customer service.
  - Knowledge-assisted Email form: Find applicable support articles based on email ticket submission and web form.
- Empower agents with context-appropriate knowledge in a unified desktop for faster resolution when agent-assisted service is needed.

Use Case: Knowledge-assisted Email

Basic Flow	Outcome 1	Outcome 2
1. Tracy clicks on an email web form to find out if GDemoTelecom has service in an area that she is moving to. 2. Tracy types *"Do you have service in Belmont, CA?"* in the subject line. 3. Tracy clicks out of the subject line to type the content in the message body. 4. An FAQ search is invoked.	Tracy is provided with the coverage map for Belmont, CA as a suggested answer. She provides feedback and closes the window.	Tracy ignores the FAQ search and types content in the message body since she has more questions. An email request is logged and placed in queue.

Note: This use case requires customization of Web Form with Knowledge Search API.

Use Case: Knowledge-assisted Social or SMS

Basic Flow	Outcome 1	Outcome 2
1. `@tibwizz` sends a Tweet (or SMS) *"looks like I will miss my connecting flight from LAX to SFO"* to `@blueskyairlines`. 2. Interaction is created and queued. 3. Orchestration script invokes Knowledge API to find answers on what to do when you miss connections.	Answer found. `@Blueskyairlines` auto-responds to `@tibwizz` *"Click here to schedule a call with our travel consultant to rebook"*.	Answer not found. Queue the message for agent.

Note: This use case requires customization of Orchestration logic.

Proactive Knowledge

- Combine Knowledge with Proactive Engagement to proactively provide suggested articles at the right moment.
- Provide knowledge-based assistance for agents if the customer asks for a human-assisted channel escalation.
- Reduce effort, reduce friction and channel escalation.

Use Case: Proactive Knowledge

Basic Flow	Outcome 1	Outcome 2	Outcome 3
1. Jurgen browses www.Gbank.com to research *College Savings Plan*. 2. He navigates to the page. 3. Web Engagement Rules can trigger knowledge article lookup to provide *knowledge nudges*.	Suggested Pages/Info Within the suggested articles section of the page, a few links are populated: • Starting a college savings plan • Transferring an existing college savings plan • College Savings Plan Calculator	Jurgen ignores the suggestions. No action taken.	Jurgen looks at suggestions, but still continues to browse. Proactively offer customers the ability to escalate to assisted service.

Note: This use case requires customization of Rules and Web Page logic.

Knowledge Web Search

Enable dynamic FAQ and channel deflection using natural language search and present knowledge articles to customers via the web.

Use Case: Contact Center Escalation

The following list of outcomes from examples on this page demonstrates how Knowledge Center allows customers to serve themselves if they want to, while providing them with easy ways to contact an agent if they cannot find what they are looking for:

- Outcome 3 in the Web Search and Proactive Knowledge examples
- Outcome 2 in the Knowledge-assisted Email example
- Outcome 3 in the Knowledge Assisted Chat example

Use Case: Web Search

Basic Flow	Outcome 1	Outcome 2	Outcome 3
1. John recently booked an Alaskan vacation for his family on Blue Sky Airlines. 2. John would like to know if he can gate check his baby's stroller and car seat. 3. John goes on www.blueskyairlines.com and in the search box types *"can I gate check my infant car seat and stroller?"*	One Question. One Answer. Knowledge Center finds the right answer in the FAQs and provides the answer to John.	Top 3 Answers. Knowledge Center also provides two other articles that contain information about gate checking guidelines.	John is not satisfied with the answers and says answer was not helpful. John is offered a choice of chat, email, or callback based on agent availability or hours of operation. Agent receiving John's request is presented with all the relevant information about John, his reservations, and the answers viewed by John so that he/she can quickly help John.

Note: This use case requires customization of Rules and web page logic.

Use Case: Fast access to content with auto-complete

Basic Flow	Outcome 1	Outcome 2
1. John goes online to the Blue Sky Airlines website. 2. He navigates the website and finds the page for *Traveling with an infant.* 3. After reviewing the page, John is not clear if he can gate check his stroller. 4. John notices a Search Bar at the top of the page and types *“can I gate check”* 5. Genesys Knowledge Center Auto-complete functionality provides suggestions like *“can I gate check my infant car seat?”*.	John finds the answers to the suggested questions helpful. He provides feedback and closes the window.	John has more questions. Create a chat interaction and place John in queue.

Use Case: Browsing though document categories

Basic Flow	Outcome
1. As John reads the knowledge article about gate checking his infant’s car seat, he also notices a category link called *“Traveling with Infants”*. 2. John clicks on the link and now has access to four other articles: • Travel tips for parents traveling with infants • Baggage allowance for infants • Online check-in for parents traveling with kids	John now has all the information he needs. He answers *“Yes”* to the feedback question from the original article, which now ranks the article higher for subsequent searches. **Note:** Feedback is not available for browsed articles, since all feedback is directly related to a search query.

Knowledge Center Components

Before you start working with Genesys Knowledge Center, you might find it helpful to learn about its components:

- **Knowledge Center Server**—Combines indexing and natural language–based search capabilities to provide effective knowledge article retrieval from one or more knowledge bases.
- **Knowledge Center CMS**—Provides customers who do not have an existing Content Management System (CMS) with the ability to create and update their knowledge bases and push them to the Genesys Knowledge Center Server for indexing and search. This component also allows customers to import and edit knowledge articles from a file.
- **Knowledge Center Plugin for Administrator**—Enables system administrators to use Genesys Administrator to configure their knowledge clusters.
- **Knowledge Center Plugin for Pulse**—Allows contact center managers to view Genesys Knowledge Center reporting at near real-time from the Pulse user interface.
- **Knowledge Center Plugin for Workspace Desktop Edition**—Provides agents with access to knowledge events (searches, article views and feedback) related to the current customer and also allows them to search the knowledge base right from their desktop.
- **Knowledge Center Data Import Tool**—Use this tool to import XML-based QNA data into a Knowledge Center index.
- **Knowledge Center REST API**—Can be used for both client and management functions.
- **Genesys Web Engagement Integration**—Knowledge Center can be used with GWE to provide proactive engagement capabilities.

Knowledge Center Server

The Genesys Knowledge Center Server combines indexing and search capabilities that allow for effective FAQ retrieval over one or more knowledge bases. It is web-based, and can run under the Jetty HTTP Server.

At its core Knowledge Center Server consists of two key parts:

- The Elasticsearch search and analytics engine
- Several Elasticsearch plugins

Elasticsearch is a search server based on Lucene. It provides a distributed, multi-tenant–capable full-text search engine with a RESTful web interface and schema-free JSON documents. ElasticSearch is distributed, which means that indices can be divided into shards and each shard can have zero or more replicas. Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shards.

Other Features of the Knowledge Center Server

- Knowledge Center Server exposes a REST API that can be used for both client and management functions.
- Knowledge Center Server is a cluster application, meaning that several nodes or servers can be grouped within a single cluster.
- Knowledge Center Server requires two application objects in Genesys Administrator:
  - One to describe the server itself (type = *Genesys Knowledge Center Server*)
  - Another for storing high-level options and knowledge base configurations, and for integrating the Knowledge Center server with other applications (type = *Application Cluster*)
- You can use third-party load-balancers above the cluster to organize your servers into a single pool, thereby providing a single point of entry for your users.
- Knowledge Center Server uses Genesys Roles to restrict access, and to authorize and authenticate users.
- The Knowledge Center installation package includes a launcher that can launch both Jetty and all of the applications deployed on Jetty as a standalone Genesys application. To accomplish this goal, the launcher communicates with the Genesys Config Server to fetch the required options.

Knowledge Center CMS

The Knowledge Center Content Management System (CMS) serves several purposes:

- Creates, activates, and deactivates knowledge bases
- Creates, updates, and deletes questions and answers in a knowledge base
- Assigns categories to this content
- Imports historical information from the Knowledge Center Server

The CMS primarily interacts with the Knowledge Center Server when creating or updating index data.

Plugin for Administrator

This plugin lets you manage the structure of the knowledge bases that are controlled by the Knowledge Center Server Cluster application object in Genesys Administrator.

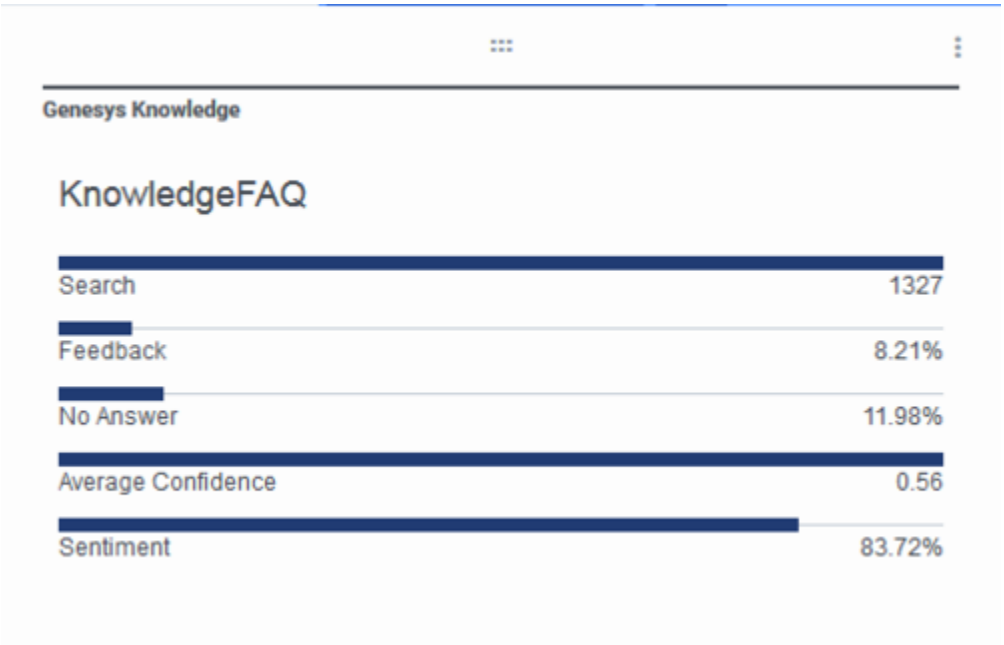
After you install this plugin, you will have access to a separate page in Administrator that displays a user interface for creating new knowledge bases and for editing the descriptions, options, languages, and custom fields in existing knowledge bases.

Plugin for Pulse

The Knowledge Center Plugin for Pulse displays Knowledge Center Server statistics, such as KPIs, user activity, trending topics, like and dislike trends, types of activities, and more.

> **Important**
>
> Pulse plugin is an integral part of Genesys Knowledge Center Server and does not require any additional installation steps.

Here is a sample display of key performance indicators:



Key Performance Indicators

This image shows a sample dashboard containing analytic reports:

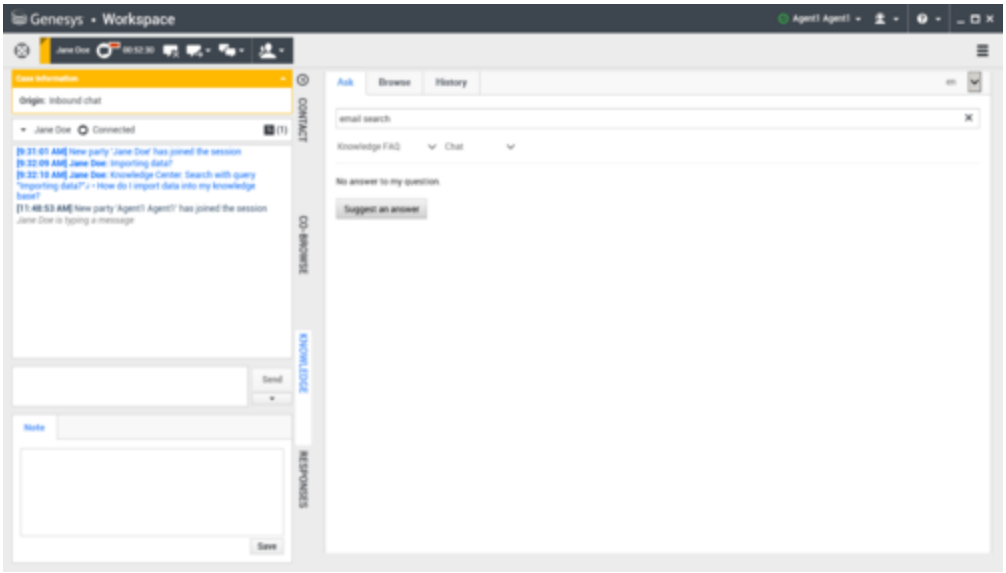
Analytics Report

Plugin for WDE

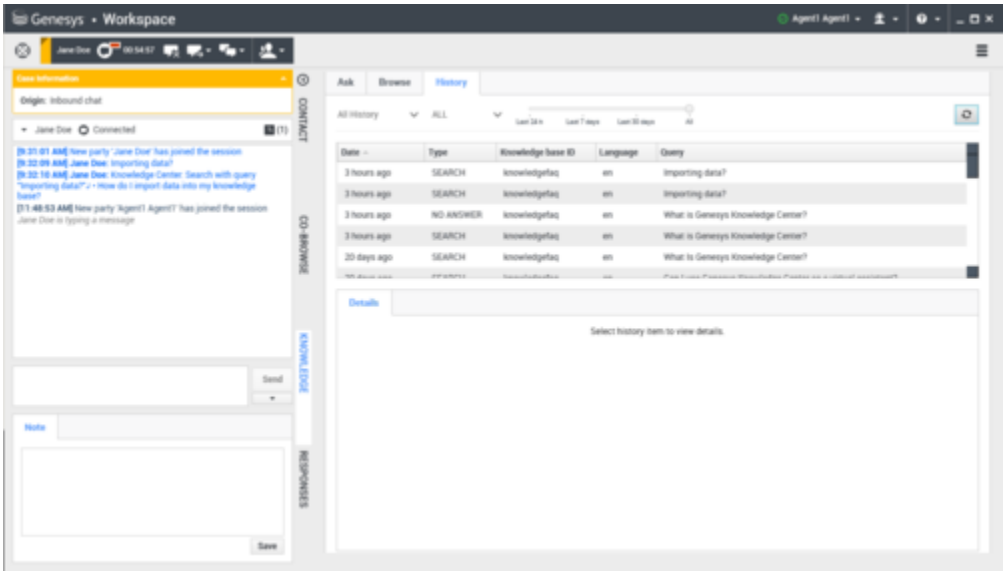
Your agents can use the Knowledge Center Plugin for Workspace Desktop Edition (WDE) to access Knowledge Center data from from their WDE worksession.

For example, if a customer escalates a question using a chat widget and the resulting interaction is routed to an agent, Knowledge Center can pre-populate a search based on the data that is attached to the chat interaction. When the interaction reaches the agent, he or she will see the customer's search history, so the customers needs can be met more quickly. In cases where the customer doesn't authorize automatic search-based access, the agent will also be able to search the customer's session history if the customer allows this during their chat.

The following images show a FAQ search and customer history, respectively.



FAQ Search



Customer History

Data Import Tool

You can use the data import tool to import QNA data from an XML file into a Knowledge Center index . The data in your XML file must be stored in a specific format, as shown in the following simple example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<documents kbsId="gkc" lang="en">
```

```
    <document>
        <id>gkc_1</id>
        <question>What is Genesys Knowledge Center Server?</question>
        <answer>Genesys Knowledge Center Server combines indexing
and search capabilities that allow for effective FAQ retrieval
over one or more knowledge bases.</answer>
        <categories>
            <category>
                <id>1</id>
                <name>Common article</name>
            </category>
        </categories>
    </document>
</documents>
```

Knowledge Center REST API

Knowledge Center REST API exposes three sets of functionality:

- The Knowledge API can be used by Knowledge Center Server clients who are interested in retrieving FAQ-related information from a knowledge base, including things like the structure of the knowledge base and its feedback data
- The Management API allows service components—such as content management systems, the Knowledge Center Administrator plugin, and data importers—to create, populate, and manage knowledge basess
- The Reporting API provides reporting engines—such as Easy Pulse or third-party products—with data on the various knowledge-related activities carried out by agents and customers

Genesys Web Engagement Integration

While it isn't exactly a component, we thought this would be a good place to mention that you can integrate Knowledge Center with Genesys Web Engagement. GWE helps you monitor, identify, and proactively engage web visitors in conversations that match your business objectives. And Knowledge Center can be used with GWE to provide proactive engagement capabilities.

For more information, see how to integrate Knowledge Center with Genesys Web Engagement.

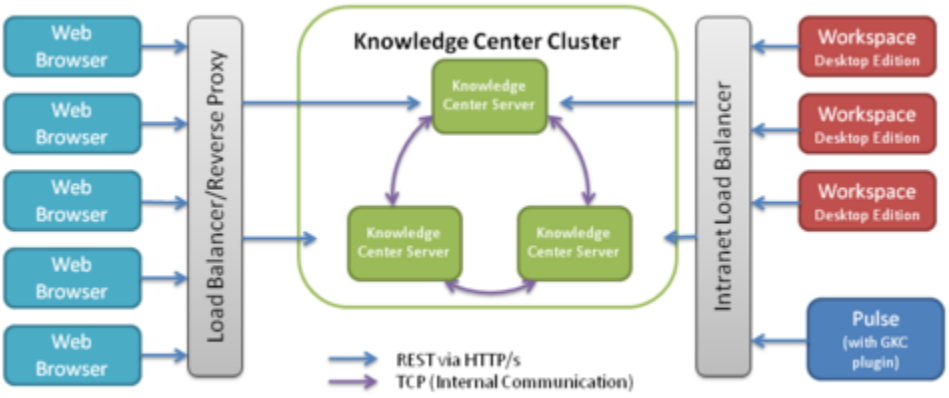
High-Level Architecture

Genesys Knowledge Center consist of several components:

- Genesys Knowledge Center Server
- Genesys Knowledge Center CMS
- Genesys Knowledge Center plugins for the following Genesys products:
  - Workspace Desktop Edition
  - Administrator
  - Pulse

Genesys Knowledge Center Server

Knowledge Center Server is the heart of the Genesys Knowledge Center solution. For purposes of load balancing and reliability, you can logically group your Knowledge Center Servers within a Knowledge Center Cluster. Each server in the cluster owns the same data and can be used to execute any desired queries against this data. These servers must be accessed by means of a properly configured load balancer that distributes the load among the server instances.



Knowledge Center High Level Architecture

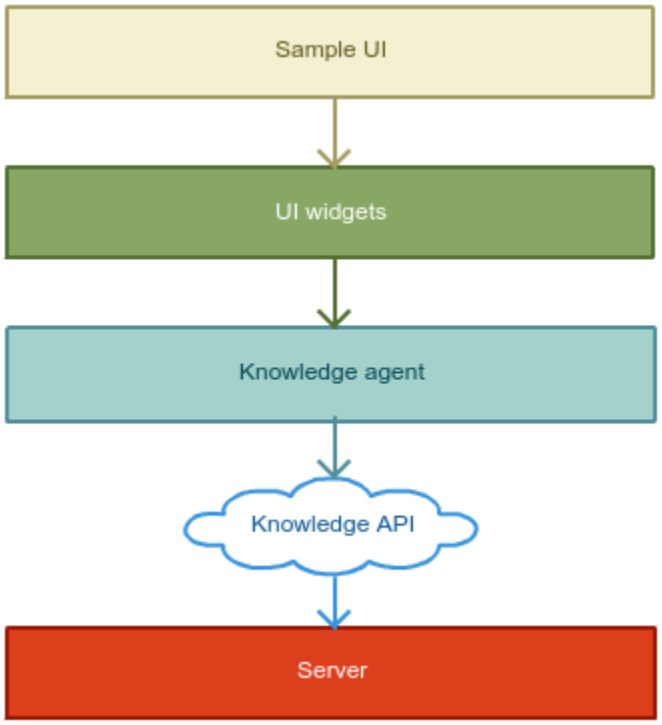
Genesys Knowledge Center Server provides several levels of integration, allowing you to access your knowledge wherever you need it—and in the way that best suits your needs. This includes a set of RESTful APIs that enables you to index data, query the server to find answers, and check usage information.

The Sample UI

The Sample UI is a JavaScript/HTML sample application that you can use as an example of how to integrate Knowledge Center into your corporate site. It runs in the visitors' browser and allows them to find answers to their questions in your corporate knowledge base.

The Sample UI offers all of the available levels of integration, allowing you to chose the one that best suits a particular need, whether it is:

- **The Knowledge API**—The RESTful web service that provides access to the Knowledge Center Server functionality
- **The Knowledge Agent**—A low-level JavaScript mapper that covers the Knowledge API and encapsulates Knowledge session management
- **The UI Widgets**—Basic and atomic UI elements covering different aspects of working with knowledge
- **The Sample UI**—An integrated sample application that implements fully functional access to the knowledge stored in Knowledge Center Server



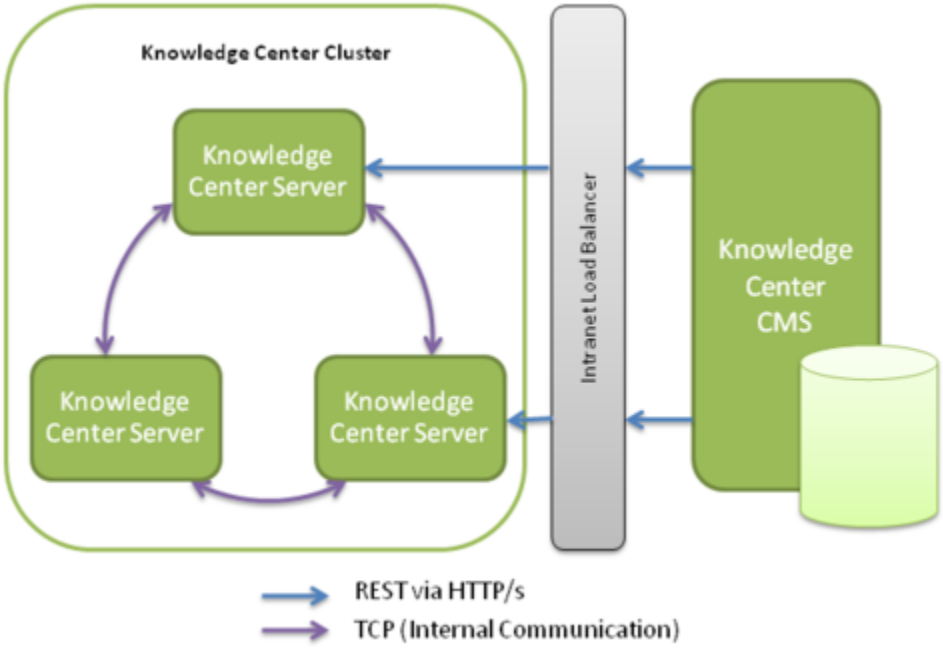
Knowledge Center Sample UI

Genesys Knowledge Center CMS

The Genesys Knowledge Center CMS is an optional component that can be used to store company knowledge and allow role-based access for authoring and improvement. The CMS is seamlessly integrated with Knowledge Center Server using its public REST APIs and allows you to:

- Index authored information into the Knowledge Center Server to expose it for use by agent and customers
- Retrieve usage information from the Knowledge Center Server and use it to better understand customer needs and to improve your knowledge base

For more information, consult the Knowledge Center User's Guide.



Knowledge Center CMS Architecture

Genesys Knowledge Center Plugins

Genesys Knowledge Center comes with three plugins that allow it to be easily integrated into the Genesys environment:

- **The Plugin for Workspace Desktop Edition**—enriching standard agent workplace with the knowledge functionality and history of customer interaction with the knowledge
- **The Plugin for Pulse**—exposing into the Pulse capability to analyze the way how customer and agents interacts with the knowledge
- **The Plugin for Administrator**—simplifies the way you create new knowledge bases via simple step-by-step graphical interface

You can also integrate Knowledge Center with Genesys Web Engagement. This allows you to take actions based on the way your knowledge is used by the customer and agents.

Terminology

Term	Meaning
cluster	Set of the Knowledge Center Servers that are working together using the same knowledge.
knowledge base	Collection of your knowledge sharing same taxonomy (set of categories) and covering same domain/sub-domain.
node	One instance of the Genesys Knowledge Center Server.
replica	backup shard of data used to guarantee data redundancy.
shard	The way to divide your knowledge base on chunks that can be distributed between different servers.

Planning Your Deployment

This chapter helps you to plan the Knowledge Center deployment within your environment. It covers following topics:

> **Important**
>
> The exact deployment architecture and solution size will vary depending on your hardware and your ability to fine-tune the deployed system to get the best performance on your equipment and with your particular user load. However, the estimates in the following topics may give you some basic ideas on how to size your deployment.

Prerequisites

OS Requirements

Knowledge Center Server

- OS Red Hat Enterprise Linux AS 7 (Intel EM64T)
- OS Windows Server 2008 (Intel EM64T)
- OS Windows Server 2012

Knowledge CMS

- OS Red Hat Enterprise Linux AS 7 (Intel EM64T)
- OS Windows Server 2008 (Intel EM64T)
- OS Windows Server 2012 (Intel EM64T)

Genesys Knowledge Center Plugin for Workspace Desktop Edition

- OS Windows Vista (Intel 32-bit)
- OS Windows Server 2008 (Intel 32-bit, Intel EM64T)
- OS Windows 7 (Intel 32-bit, Intel EM64T)
- OS Windows Server 2012 (Intel EM64T)
- OS Windows 8 (Intel EM64T)

Genesys Knowledge Center Plugin for Administrator

- OS Red Hat Enterprise Linux AS 5 (Intel EM64T)
- OS Windows Server 2008 (Intel EM64T)
- OS Windows Server 2012 (Intel EM64T)

Genesys Knowledge Center Plugin for Pulse

- OS Red Hat Enterprise Linux AS 5 (Intel EM64T)
- OS Windows Server 2008 (Intel 32-bit, Intel EM64T)
- OS Windows Server 2012 (Intel EM64T)

Web Browsers

- Google Chrome 34+
- Mozilla Firefox 54+
- Microsoft Internet Explorer 9, 10, 11
- Microsoft Edge
- Apple Safari 10+

Java Requirements

- Java 8 x64 SE Bundle

> **Important**
> Knowledge Center requires Oracle JDK to be used.

Genesys Environment

- Workspace Desktop Edition 8.5
- Pulse 8.5.1 or higher
- Genesys Framework 8.1–8.5
- Configuration Server (8.1.300.21 / 8.5.100.02)
- Genesys Administrator Extension (GAX) 8.5.210.10 or higher

> **Important**
> GAX is needed for Knowledge Center 3.5.302 and earlier only.

Sizing Information

Before deploying the Genesys Knowledge Center solution to your production site, you must estimate the size of the solution that can handle the expected user load. Genesys recommends that you download the Sizing Calculator, an Excel workbook that you can use to help calculate the number of nodes required for your production deployment. **Note:** clicking the link will automatically start the download.

The process of estimation starts from input values, usually given in the terms of business operations (for example, number of knowledge bases, or number of sessions and questions per session). Using some math, and having in mind the workflow that is applied to the input traffic, you can then produce the expected load values in terms of requests per second. Applying these values to the experimentally produced measurements, you can estimate the size of the solution required to be deployed.

> ### Important
> The minimum number of the Knowledge Center Server nodes in Knowledge Center Cluster is 3. When doing the lab testing you can use 1 node configuration.
> A cluster with 2 nodes cannot be used in the lab or in a production deployment. Running a 2 node cluster might lead to data loss in a network outage between these nodes.

Hardware Recommendations

Hardware you can use to deploy the Knowledge Center varies from the particular needs & conditions that you will use it in. Below is the set of recommendations and considerations that will help you to achieve a better understanding how the different components our your environment influence the Knowledge Center performance.

CPUs

Overall application is light on CPU when in comes to finding relevant knowledge to your search requests. This leads to lower dependency on CPU performance. Any modern processor with multiple cores will do the job. These days, two key parameters of the CPU are: speed and number of cores. In this case, you should choose a CPU with more cores than a slightly faster one. This allows the Knowledge Center to service concurrent requests more efficiently.

A modern CPU with 4 or 8 core CPUs is recommended.

Memory

This solution is designed to process tons of data and select only relevant data for each one of your queries. In this case, memory is one of the resources that is intensively used. When planning your host memory you need to ensure that RAM size will be enough to host all your running applications and some extra is left for the system to host the OS filesystem cache.

The absolute minimum of 8Gb RAM must not be crossed. It is recommended to use hosts with 16 Gb. With optimization and running multiple applications on the host, you may end up with 32 or 64 Gb hosts.

> **Important**
>
> In most cases it is recommended to plan your deployment with 50% of your memory allocated to the running application. The remaining 50% is used for the OS filesystem cache allowing for different software (including Genesys Knowledge Center) to work faster and minimizing disk operations.

Discs or Storage

Disks play a key role in the system performance as well. Being data-intensive and doing a lot of the read-write operations, Genesys Knowledge Center Server is highly dependent on the speed of the disk operations.

There are several common ways to improve performance of read-write intensive applications:

- Use the OS cache to minimize the number of read operations required (see recommendation above in the **Memory** section).
- Using faster disks; high-performance (15K) spinning disks are a good choice. Usage of SSD disks will burst the performance of your cluster even mlore.
- Using RAID 0 to improve the speed over any type of disk you are using (no need to go further with redundancy features of RAID, as data replication is the integral part of the solution itself).

The amount of disk space consumed by the Genesys Knowledge Center solution can be estimated using the Sizing Calculator.

> **Important**
>
> Avoid any types of the disk technologies that increase latency and throughput. An example of this solution would be the NAS.

Network

We recommend keeping all your nodes within a cluster in the same network, avoiding cross data center communication. Replication of the data between datacenter is a separate concern and deserves a separate solution.

The key parameters of the network you need to watch for are:

- latency: the Knowledge Center Cluster is the self-managing solution that distributes the load between all available nodes. By sending a request to one node of the cluster, your are employing all nodes (to the extent that it makes sense) to execute your request. Keeping the latency low will guarantee you the maximum speed of distributed execution.
- reliability: minimizing the number of disconnects in the system will guarantee that the Knowledge Center Cluster is fully concentrated on serving the request of your internal and external customers instead of doing house-keeping duties such as replacing dead nodes and relocating the data.
- bandwidth: being quite light in network communication during ordinary work (executing the searches), this solution can be demanding on the network bandwidth while indexing huge amounts of data or recovering from node failure.

Summary

Genesys Knowledge Center Server does not require any enormous hardware configuration to run on. A medium-sized box is the best option to run this application.

Another general recommendation is to keep your hardware set up as solid as possible:

- adding a high performance, huge RAM, SSD RAID 0 enabled server to the cluster of outdated servers will

not provide any noticeable or stable performance improvement overall of the solution. The speed of newly-added servers are completely compromised by the old hardware that executes parts of the same requests making the overall response and performance to be almost unchanged.

- putting one server into a higher latency network segment can improve some access parameters in that particular segment but will result in overall system performance degradation.

The best shot for the deployment of the cluster is using similar hardware for all servers and putting them into similar network conditions. This will ensure the most balanced use of your resources.

> **Important**
>
> Please ensure that the disk's volume that you are using for search indexes has at least of 15% of the total volume of the disk in free space. We are recommending not to use huge volumes as it requires you to have a lot of free space. Genesys Knowledge Center Server is constantly monitoring the remaining free space on the disk and if it finds that there is less then 10% of your disk volume left, it might make a decision to relocate indexes on a different host.

Software Configuration

JVM Settings

We recommended you run the most recent version of Java Virtual Machine. The minimum version supported is Java 1.8 64 bit. The recommended (default) memory settings for the Knowledge Center Server java process (Xms/Xms) is 4 Gb. Extending it to 16 Gb might be a good idea when using large amounts of data. Adding and extra 4 Gb of heap for the process recommended on the nodes that are used to execute history archiving.

> **Important**
> It is strongly advised to not set the Xmx larger than 30Gb.

Co-Locating On Same Host

We do not recommended co-locating two or more nodes of the Genesys Knowledge Center Server on the same host. Loosing the host due to hardware or network failure will result in two or more nodes being lost by the cluster. This will lead to massive network operations, potential loss of data (in the case where the number of replications is configured to 1) and even a complete outage of the cluster (for example, for clusters with 3 nodes).

Genesys Knowledge Center Server can be co-located on the same host with other Genesys or 3rd party solutions.

When co-locating please ensure that:

- The host is equipped with enough resources, especially RAM
- the solutions you are co-locating with are not causing any CPU spike when any other application on the host is blocked

Do not co-locate with other disks' read-write intensive solutions as it could easily overrun capabilities of your disk system.

> **Important**
> In most cases it is recommended to plan your deployment with 50% of your memory allocated to the running application. The remaining 50% is used for the OS filesystem cache allowing for different software (including Genesys Knowledge Center) to work faster and minimizing disk operations.

Genesys Knowledge Center Server/Cluster Configuration

Genesys Knowledge Center Server/Cluster is configured by default to enable smooth and high performance operations. Detailed information on every option exposed, valid values, and considerations behind the solutions are available on the Configurations Options page of the product documentation.

The section below provides additional information and recommendations.

Cluster Setup

The starting point of any cluster configuration is to create one application of the Application Cluster type and as many as needed applications of the Genesys Knowledge Center Server type in Genesys Administrator.

While doing this please ensure that you are creating the application only for the actual nodes of the Server that you will use. Having a spare application added to the cluster will mislead the cluster members while calculating the required number of nodes to be online and to enable cluster functionality.

Generally a cluster expects the N/2+1 node to be online to start servicing the clients (where N is the number of Genesys Knowledge Center Server applications connected to the cluster and is enabled).

For example, if you have configured 3 nodes in the Genesys Administrator and started just one, the started server will refuse any client's requests as it will treat the overall cluster as not started yet. The cluster will become functional only when the second node of The Genesys Knowledge Center Server joins the cluster.

In rare cases when you still need to run such a configuration there are several ways to enforce cluster functionality:

- Disable applications that you are not going to run (by unchecking the **Enable** checkbox in every such application in Genesys Administrator)
- Manually setting the **minimumMasterNodes** value option in the **index** section into a desired value (for example, 1 in the example below).

Important

We do not recommended using manual settings, especially manually defining the **minimumMasterNodes** option as any incorrect values could result in data corruption.

Tip

Running nodes needs to be restarted if you applying one of the listed below recommendations.

Internode Communication

When running a production cluster it is required to disable the multicast node discovery by setting it to *false* **enabled** option in **multicast** section of the cluster application object in the configuration. It is also a good idea to disable multicast in lab deployments.

Multicast may lead to incidental joining of the undesired node to the cluster that will trigger data relocations to rebalance the data between nodes.

Host On-Boarding

Having the majority of the configuration in common between all nodes in the cluster there are few parameters that must be configured to onboard the node on a particular host. An example of these parameters is the folder for log files that is configured in the **log** section of the Genesys Knowledge Center Server application.

The other configuration that you need to set properly is the location of the indexed knowledge that is configured in option **path.data** of **gms.yml** file.

> ### Important
> It is recommended to store your indexed knowledge on the fastest disk you have on the host. By default the data will be placed into the folder where you have installed Genesys Knowledge Center Server.
> If you have two disks attached to the host (for example, one spinning and another SSD) you can reconfigure the application to store the data in the fastest disc (SSD) while using the spinning disc for the application binaries.

Multi-Tenancy

Knowledge Center 8.5.302.xx and earlier is a single tenant solution. This means that if you have a multi-tenant environment and you want to use the Knowledge Center application within several tenants, you need to deploy a separate cluster for every tenant where you plan to use the application.

Starting from 8.5.303.xx release of the product it supports multiple tenants within one cluster deployment. The list of the tenants needs to be explicitly set for the applications in cluster.

> **Important**
>
> Within one Knowledge Center cluster all Knowledge Center Server nodes and all Knowledge Center CMS nodes must be configured to have the same tenant(s) in the **Tenants** section.

Installation and Deployment

Task Summary: Genesys Knowledge Center

The following table outlines the task flow for installing Genesys Knowledge Center.

Objective	Actions
1. Prepare your environment	1. Configure Languages
2. Configure the Knowledge Center Cluster Application	1. Import the Knowledge Center Cluster Application Template 2. Create the Cluster Applications 3. Configure the Cluster Application
3. Install the Knowledge Center Server	1. Import the Knowledge Center Server Application Template 2. Create the Server Applications 3. Configure the Knowledge Center Server Application 4. Install Knowledge Center Server
4. Install the Knowledge Center CMS	1. Install the CMS 2. Configure Data Source (based on the selected provider): • (if using Cassandra as persistent storage), Configure the Cassandra Data Source • (if using Microsoft Server as persistent storage), Configure the CMS to work with Microsoft SQL Server • (if using Oracle as persistent storage), Configure the CMS to work with Oracle 3. Configure the CMS 4. (for 8.5.303.xx and later) Manage the Knowledge Base using CMS

Objective	Actions
5. Install and Use the Administrator Plugin **Important** From the 8.5.303.xx release of the product, Administrator plugin has been discontinued. Please skip this step for these versions.	1. Install the Knowledge Center Plugin for Administrator 2. Manage the Knowledge Bases
6. Configure reporting in Pulse	1. Configure the Genesys Knowledge Center Plugin for Pulse
7. Install the Workspace Desktop Edition Plugin	1. Install the Plugin for Workspace Desktop Edition 2. Configure the WDE Application to work with the WDE Plugin 3. (Optional) Install the WDE Language Pack
8. Configure agent accounts	1. Grant access permission to content authors 2. Provide Knowledge Center Server Access to Agents 3. Provide Knowledge Center Workspace Desktop Edition Plugin Access to Agents

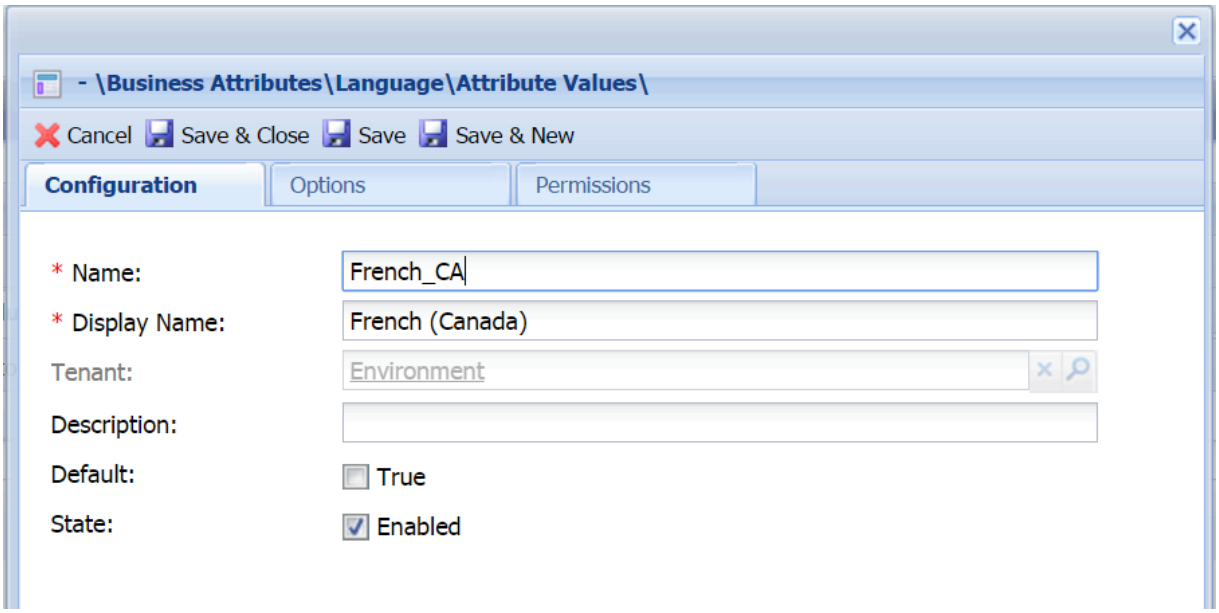
Before you Begin

This chapter describe the step(s) required to prepare your environment for the Knowledge Center installation. Preparation step(s) are:

- Define the languages used in your environment

Configuring Languages

1. Open Genesys Administrator and navigate to **Provisioning** > **Routing/eServices** > **Business Attributes**.
2. Select **Language** business attribute
3. Click **Edit** button
4. Select **Attribute Values** tab
5. Click **New** button or select existing attribute value and press **Edit** button
6. On **Configuration** tab (skip this step if you are editing existing **Attributes Value**. For example, English which is created by default)
   a. Enter **Name**. For instance, French_CA
   b. Enter **Display Name**. For instance French (Canada)
   c. Ensure that **State** check is enabled



7. Select **Options** tab:
   a. Press **New** button to add language code

b. Enter "code" in **Section** field (eg. new section "code" should be created)

c. Enter "language" in **Name** filed (eg. new option "name" should be created)

d. Enter ISO 639-1 alpha-2 code that corresponds to desired language in **Value** field. For instance fr

e. Press **OK** button

8. If you are adding regional language you also need to specify a region code:

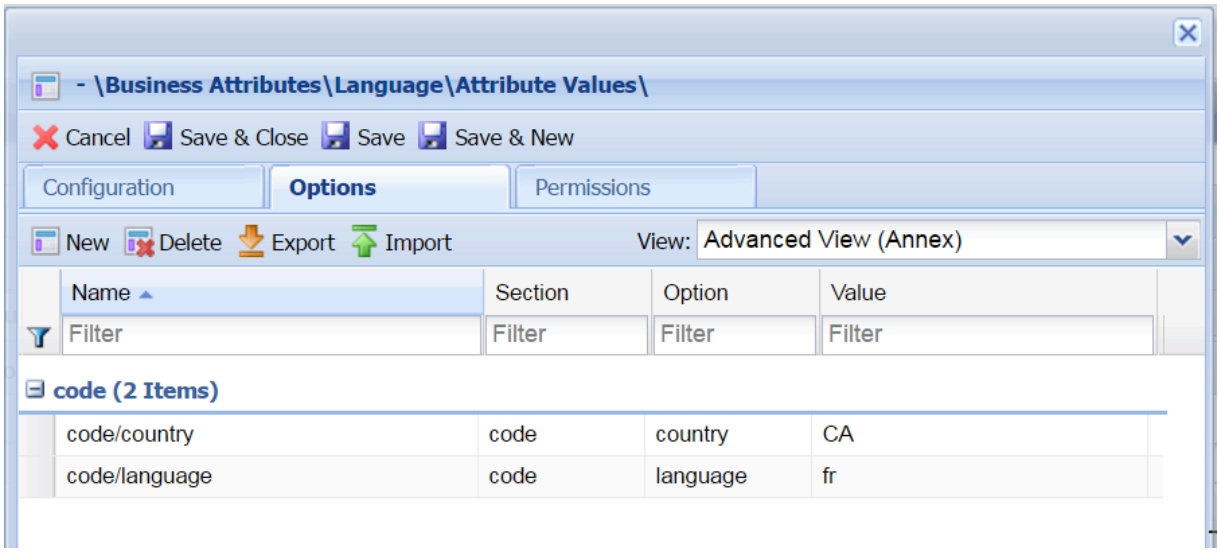
   a. Press **New** button

   b. Enter code in **Section** field

   c. Enter country in **Name** filed

   d. Enter ISO 3166-1 alpha-2 code that corresponds to desired region/country in value filed. For instance CA

   e. Press **OK** button



9. Press **Save & Close** button
   Note: You need to repeat this procedure for every language that you plan to use in your Knowledge Base.

Important

Please ensure that following rules are followed when you are adding Attribute Values to the Language Business Attribute:

- Languages should be create in the same tenant in which Knowlege Server and CMS applications will be configured.
- Every language needs to have language (mandatory) and country (if applicable) codes defined on options tab (languages w/o codes will be ignored)
- Ensure that all language/country combinations are unique (duplicate combinations will be ignored)

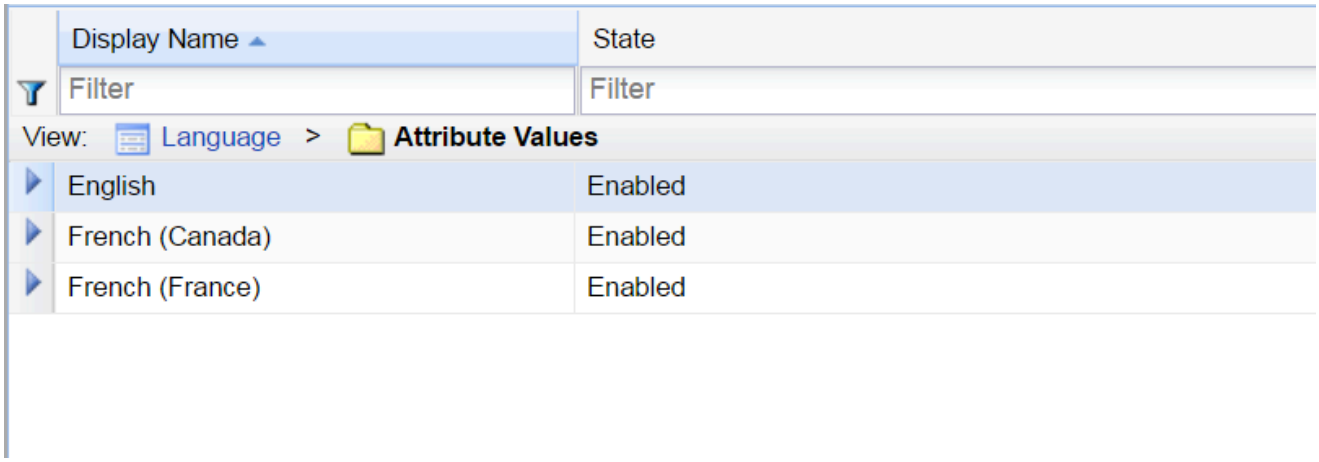
- Do not edit/change defined codes if they are used in the knowledge bases
- Language code needs to correspond to the ISO 639-1 alpha-2 code for the given language (http://www.iso.org/iso/home/standards/language_codes.htm)
- Country code needs to correspond to the ISO 3166-1 alpha-2 code for given language (http://www.iso.org/iso/country_codes)

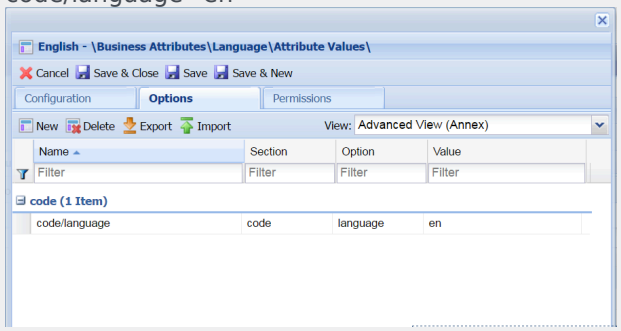
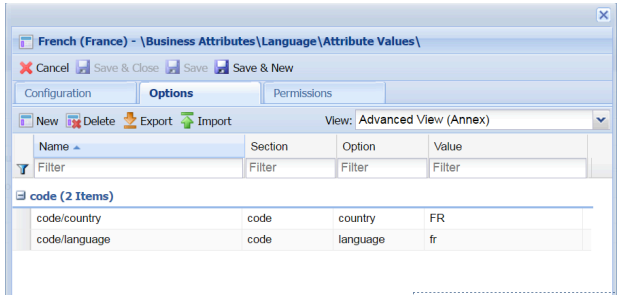
Example

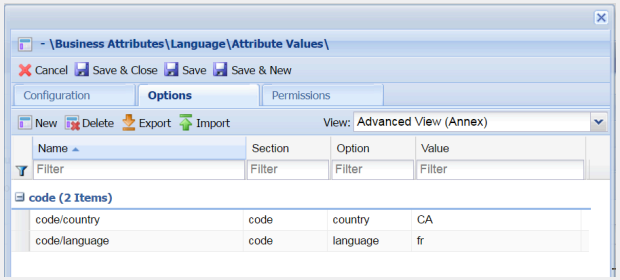
If you wish to have following languages in your environment:

- English
- French (Canadian)
- French (France)

You will need to create 3 attribute values:



Name	Display Name	Options
English	English	code/language=en 
French_FR	French (France)	code/language=fr code/country=FR 
French_CA	French (Canada)	code/language=fr code/country=CA

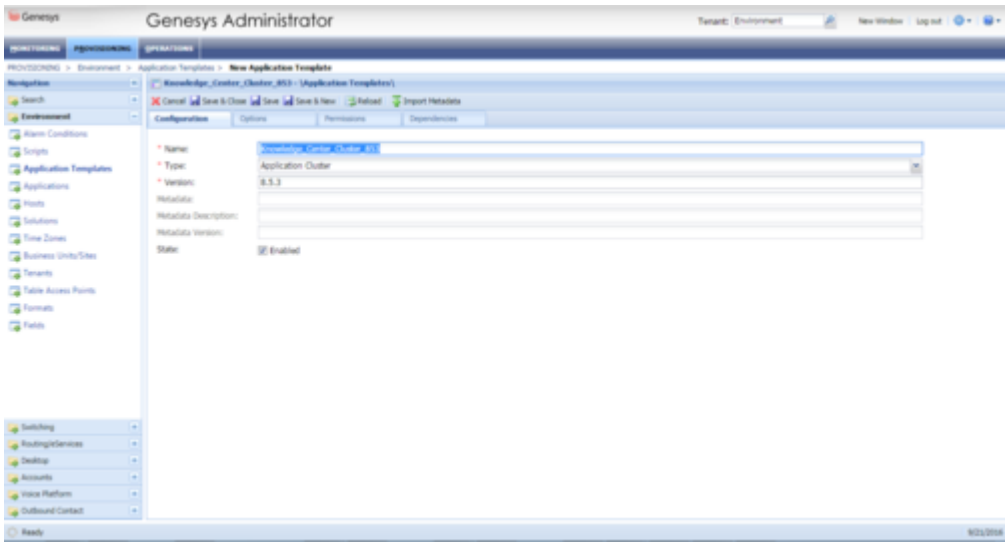
Name	Display Name	Options
		

Installing the Knowledge Center Cluster Application

Carry out the procedures below, in order, to install and configure the Knowledge Center Cluster Application.

Import the Knowledge Center Cluster Application Template

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.
3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Browse to the *Knowledge_Center_Cluster_853.apd* file available in the templates directory of your installation CD.
5. Click open.
6. The **New Application Template** panel opens.

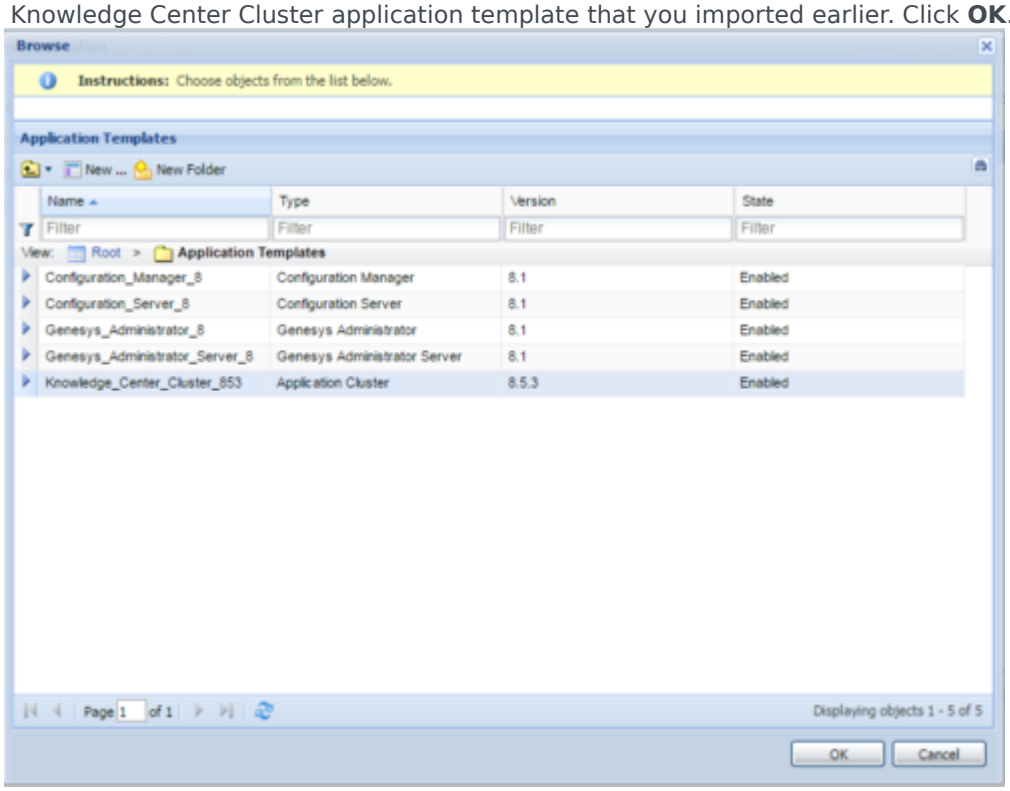


New Application Template Panel

7. Click **Save and Close**.

Create Cluster Applications

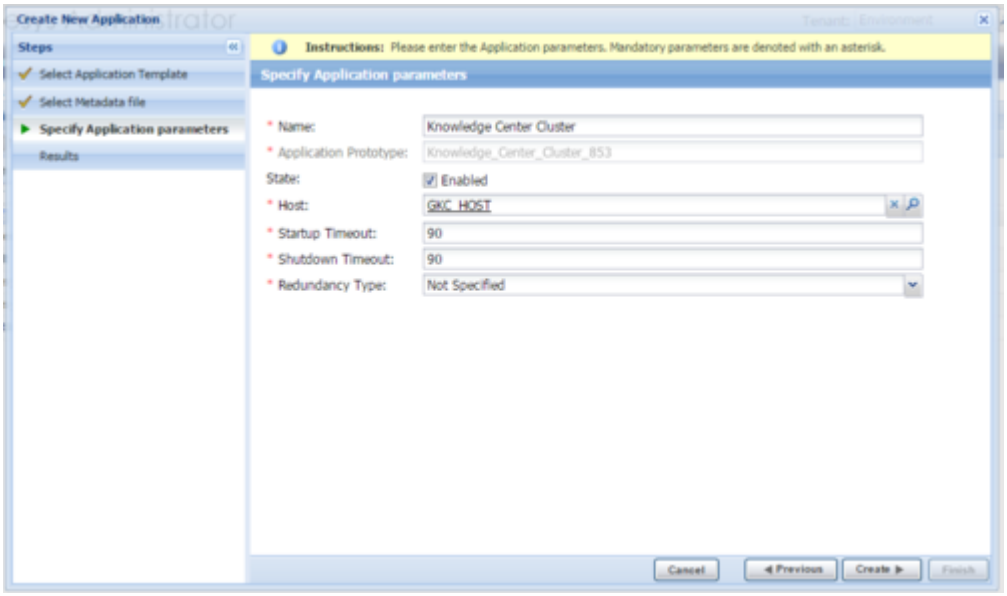
1. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**.
2. In the **Tasks** panel, click **Create New Application**.
3. In the **Select Application Template** panel, click **Browse for Template** and select the Genesys Knowledge Center Cluster application template that you imported earlier. Click **OK**.



Selecting Knowledge Center Cluster Application Template

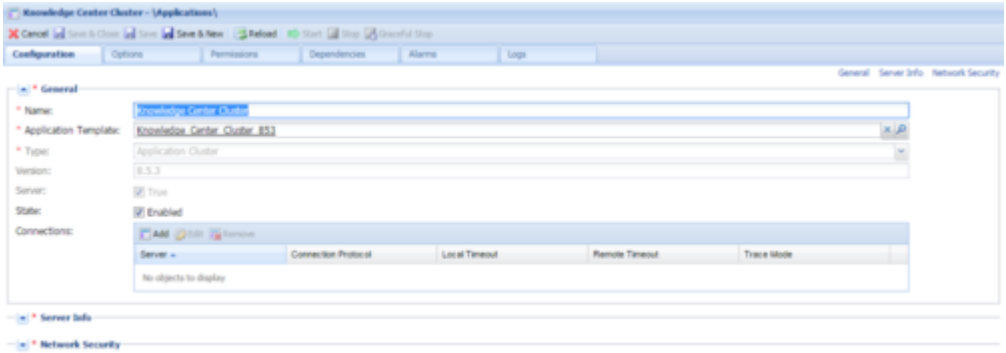
4. The template is added to the **Select Application Template** panel. Click **Next**.
5. In the Select Metadata file panel:
   a. Click **Browse**.
   b. Click **Add**.
   c. Select the *Knowledge_Center_Cluster_853.xml* file available in the templates directory of your installation CD.
   d. Click **Open**.
5. The metadata file is added to the **Select Metadata** file panel. Click **Next**.
6. In **Specify Application parameters**:
   a. Enter a name for your application. For instance, *Knowledge Center Cluster*.
   b. Ensure that **State** is checked.
   c. Select the **Host** on which the Knowledge Center Cluster load-balancer will reside.

d. Click **Create**.



Specifying Knowledge Center Cluster Application Parameters

5. The **Results** panel opens.
6. Enable **Open the Application details form after clicking 'Finish'** and click **Finish**. The Knowledge Center Cluster application form opens and you can start configuring the Cluster application.

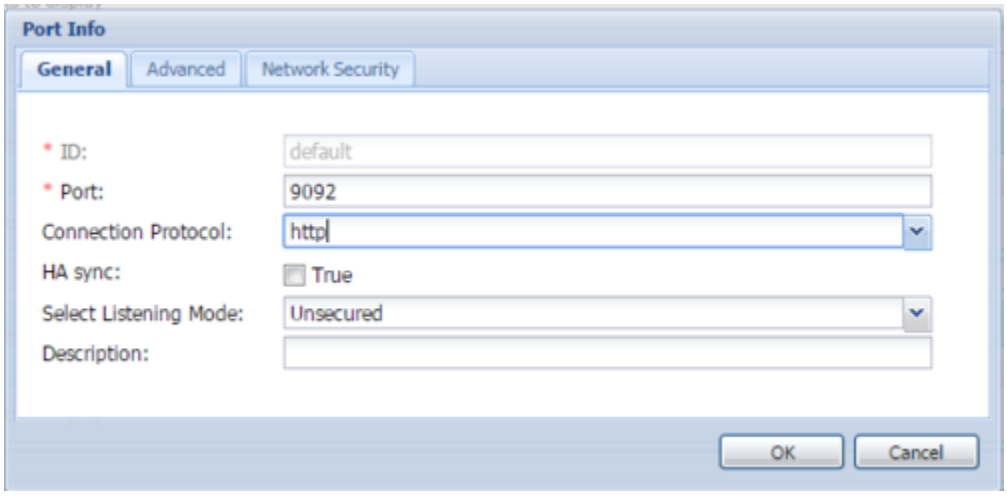


Configuring the Knowledge Center Cluster Application

Configure the Cluster Application

1. If your Knowledge Center Cluster application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the Knowledge Center Cluster and click **Edit...**.
2. Expand the **Server Info** pane.
3. If your **Host** is not defined, click the lookup icon to browse to the host on which the Knowledge Center Cluster load-balancer will reside.

4. In the **Listening Ports** section, create the default port by clicking **Add**. The **Port Info** dialog opens.
   a. Enter the port number for the Knowledge Center Cluster load-balancer, for instance, *9092*.
   b. Choose *http* or "https" for the **Connection Protocol**.
   c. If you will be using a secure connection to the cluster, choose *Secured* for the **Select Listening Mode**.
   d. Click **OK**. The HTTP or HTTPS port with the default identifier appears in the list of **Listening ports**.

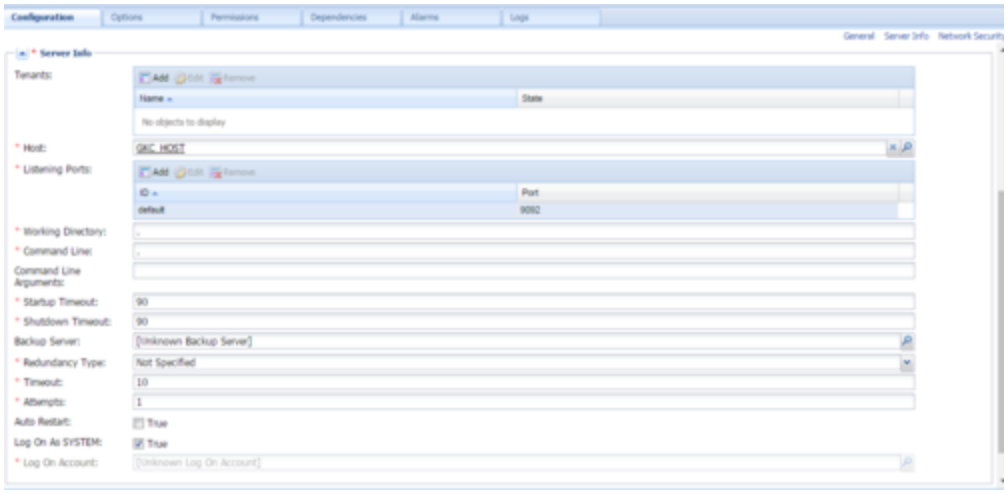


Knowledge Center Cluster Port Information

   e. In the **Tenants** section, add a working tenant by clicking **Add**. Browse and choose the appropriate tenant in the pop-up dialog. Click Ok.

> **Important**
>
> For Knowledge Center 8.5.302.xx and earlier Cluster can only work under a single tenant. Starting from 8.5.303.xx release of the product it supports multiple tenants within one cluster deployment. Application Cluster and GKC server/CMS should be in the same tenant

   f. Ensure the **Working Directory** and **Command Line** fields contain "." (period).



Knowledge Center Cluster Server Information

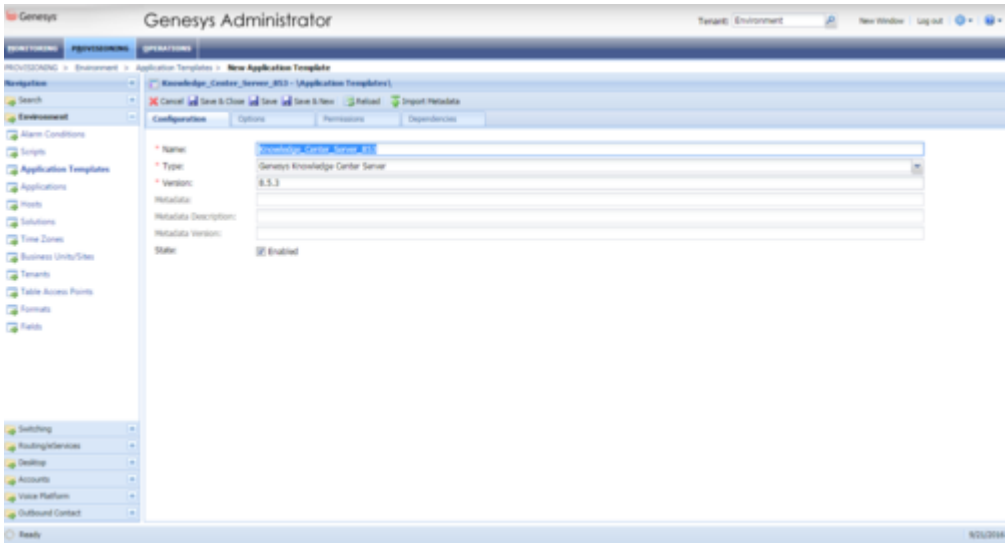
5. Click **Save**.
6. The **Confirmation** dialog for changing the application's port opens. Click **Yes**.

Installing the Knowledge Center Server

Import the Knowledge Center Server Application Template

Start

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.
3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Browse to the *Knowledge_Center_Server_853.apd* file available in the *templates* directory of your installation CD.
5. Click **Open**.
6. The **New Application Template** panel opens.



The Knowledge Center Server Application Template

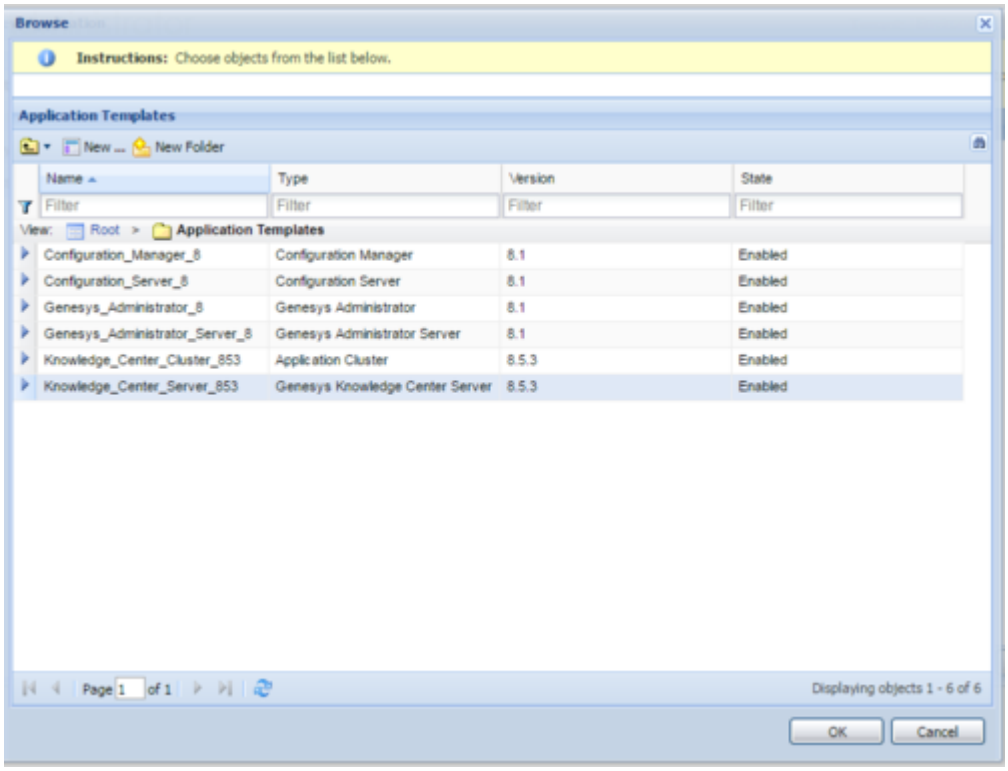
7. Click **Save and Close**.

End

Create Server applications

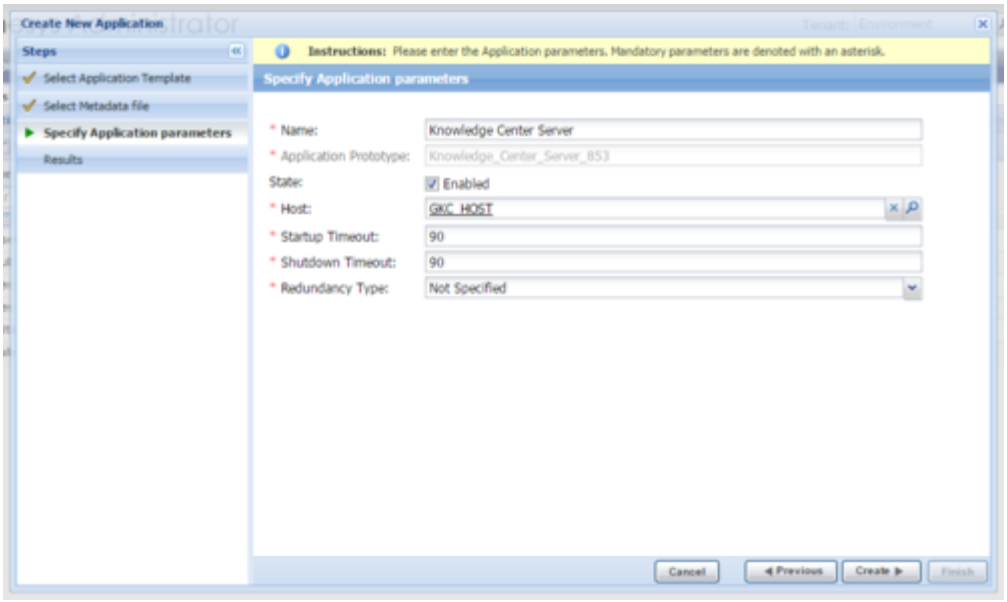
Start

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**.
2. In the **Tasks** panel, click **Create New Application**.
3. In the **Select Application Template** panel, click **Browse for Template** and select the Genesys Knowledge Center Server application template that you imported earlier. Click **OK**.



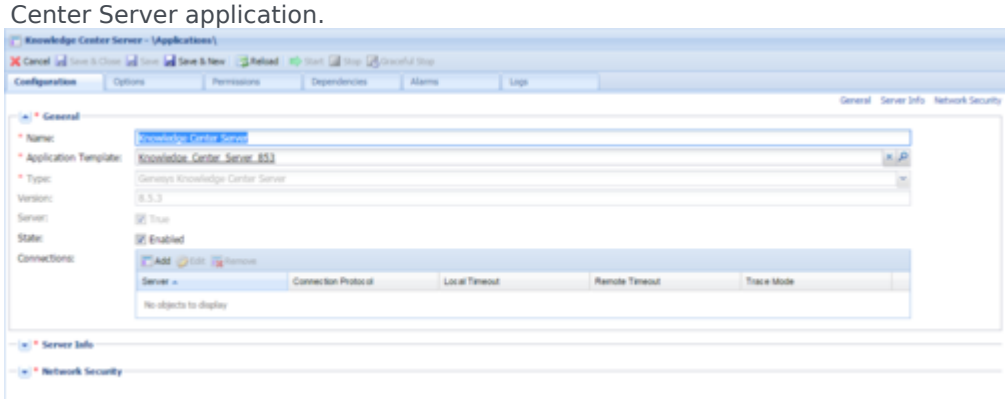
Selecting the Knowledge Center Server Template

4. The template is added to the **Select Application Template** panel. Click **Next**.
5. In the **Select Metadata** file panel:
   a. Click **Browse**
   b. Click **Add**
   c. Select the *Knowledge_Center_Server_853.xml* file available in the templates directory of your installation CD.
   d. Click **Open**
6. The metadata file is added to the **Select Metadata** file panel. Click **Next**.
7. In **Specify Application parameters**:
   a. Enter a name for your application. For instance, *Knowledge Center Server*' .
   b. Enable the **State**
   c. Ensure that **State** checkbox is checked
   d. Select the **Host** on which the Knowledge Center Server will reside
   e. Click **Create**



Creating the Knowledge Center Server Application

6. The **Results** panel opens.
7. Enable **Open the Application details form after clicking Finish** and click **Finish**.
8. The Knowledge Center Server application form opens and you can start configuring the Knowledge Center Server application.



Knowledge Center Server Application Details

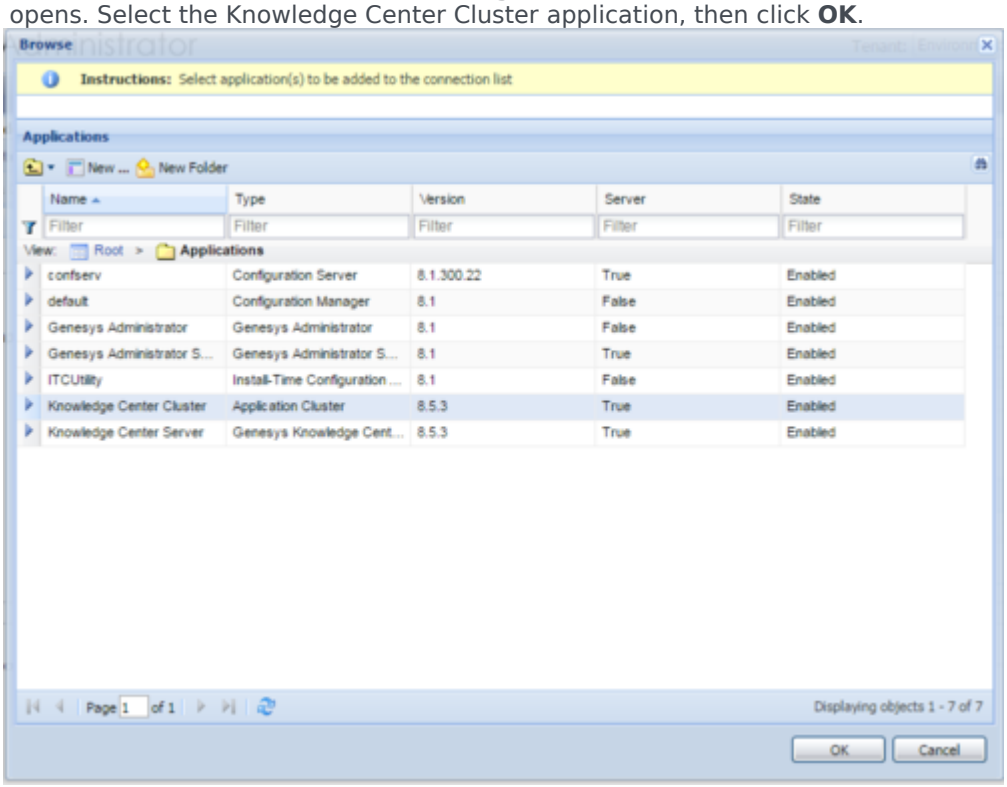
End

Configuring the Knowledge Center Server Application

Start

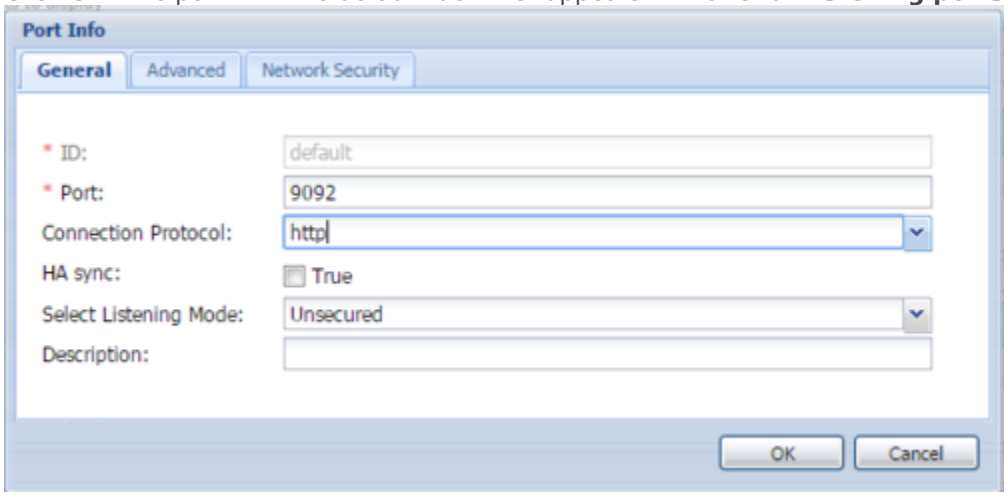
1. If your Knowledge Center Server application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the Knowledge Center Server and click **Edit...**.

2. In the **Connections** section of the **Configuration** tab, click **Add**. The **Browse for applications** panel opens. Select the Knowledge Center Cluster application, then click **OK**.



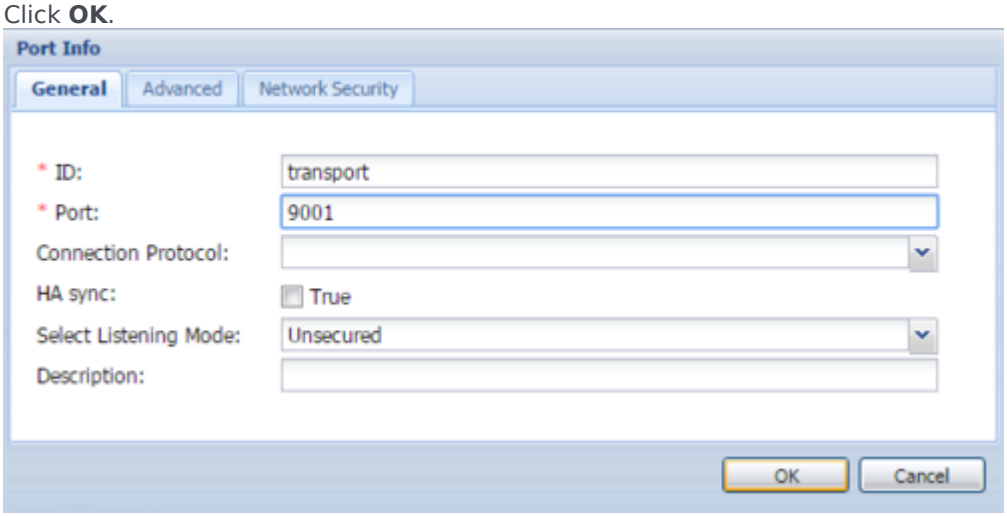
Selecting the Knowledge Center Cluster Application

3. Expand the **Server Info** pane.
4. If your **Host** is not defined, click the lookup icon to browse to the hostname of your application.
5. In the **Listening Ports** section, create the default port by clicking **Add**. The **Port Info** dialog opens.
   a. Enter the **Port**. For instance, *9092*. This should be the port number for the Knowledge Center Server instance.
   b. Click **OK**. The port with the default identifier appears in the list of **Listening ports**.



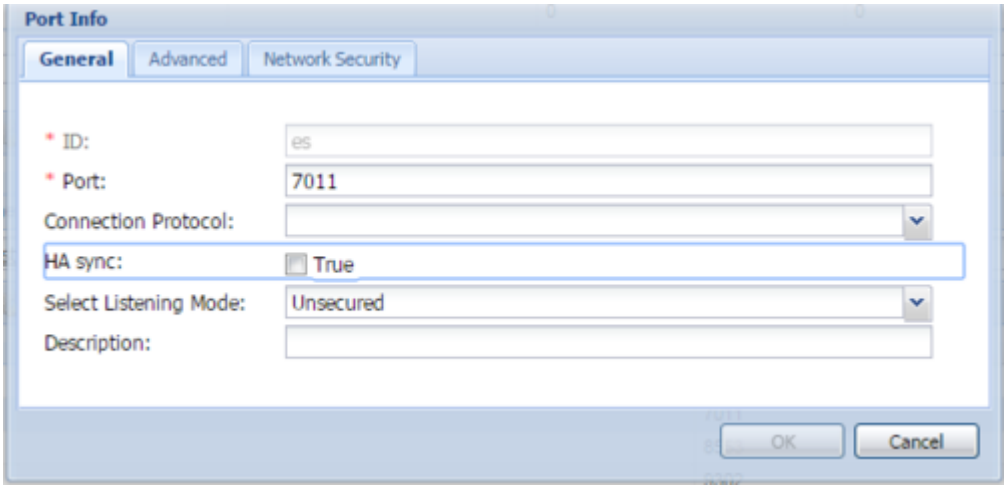
Knowledge Center Server Port Information

6. Optionally, you can explicitly add Transport port for ElasticSearch engine. If you do not define transport port, port 9300 will be used. To specify the port, click the **Add** button. The Port Info dialog opens.
   a. Enter *transport* for the **ID** field.
   b. Enter the **Port**. For instance, 9001.
   c. Click **OK**.



Knowledge Center Server Transport Port Information

4. Optionally, you can explicitly add a port for access to ElasticSearch engine. If you do not define this port, port 9200 will be used. To specify the port, click the **Add** button. The **Port Info** dialog opens.
   a. Enter *es* for the ID field.
   b. Enter the **Port**. For instance, *7011*
   c. Click **OK**.



Knowledge Center Server Elasticsearch REST API Port Information

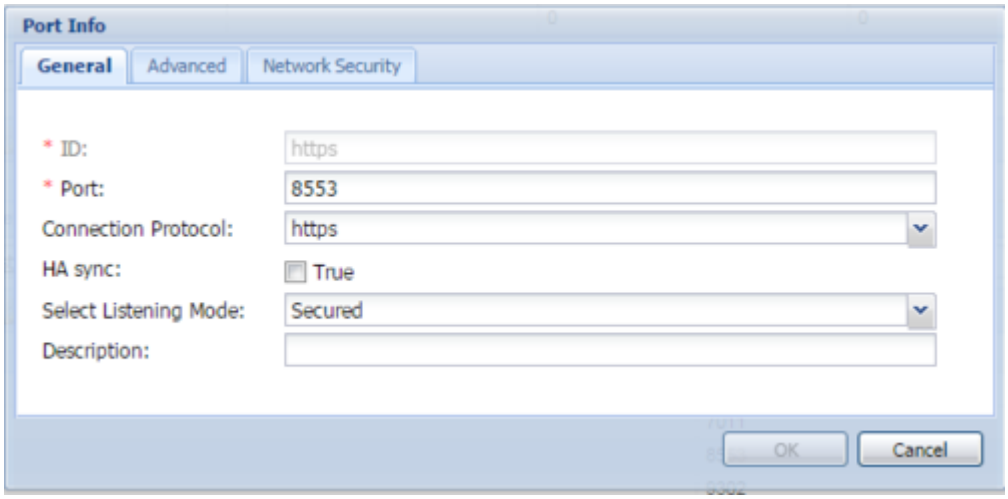
4. Optionally, you can add a secure listening port for authenticated users, secured connections, and secure chat. Click **Add**. The **Port Info** dialog opens.
   a. Enter *https* for the ID field

b. Enter the **port** . For instance, 8553

c. Enter https for the **Connection Protocol**.

d. Choose Secured for the **Select Listening Mode**.

e. Click **OK**.



Knowledge Center Server secure HTTP Port Information

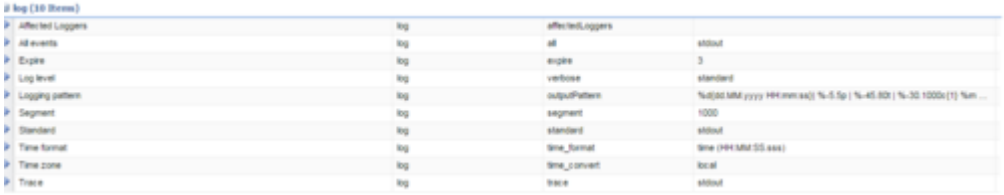
Note: If https port enabled - service will be available only on https port; http connection will be unavailable.

6. Ensure the **Working Directory** and **Command Line** fields contain "." (period).
7. In the **Tenants** section, add a working tenant by clicking **Add**. Browse and choose the appropriate tenant in the pop-up dialog. Click Ok.

> ### Important
>
> For Knowledge Center 8.5.302.xx and earlier Cluster can only work under a single tenant. Starting from 8.5.303.xx release of the product it supports multiple tenants within one cluster deployment. Application Cluster and GKC server/CMS should be in the same tenant

8. If you are using Access Groups to assign privileges to agents:
   - Uncheck **Log On As System**
   - In **Log On Account** specify the user account that has the ability to view access groups (for example, user from the Super Administrators access group).
   - User should have access to the same tenant/tenants in which that Node is configured
   - User should belong to Administrator access group in Environment tenant or be granted "Read and Execute (RX)" and "Read Permissions (E)" permissions for Environment tenant, if the application configured not in the Environment tenant; user should belong to some Administrator Access Group in application's tenant/tenants
9. Click **Save**.
10. The Confirmation dialog for changing the application's port opens. Click **Yes**.

11. (Optional) Select the Options tab. In the [log] section, the all option is set to stdout by default. Enter a filename if you wish to enable logging to a file. For example, you can enter stdout, *C:\Logs\Knowledge\Knowledge_server* to force the system to write logs both to the console and to a file.



Knowledge Center Server Application Logging Options

End

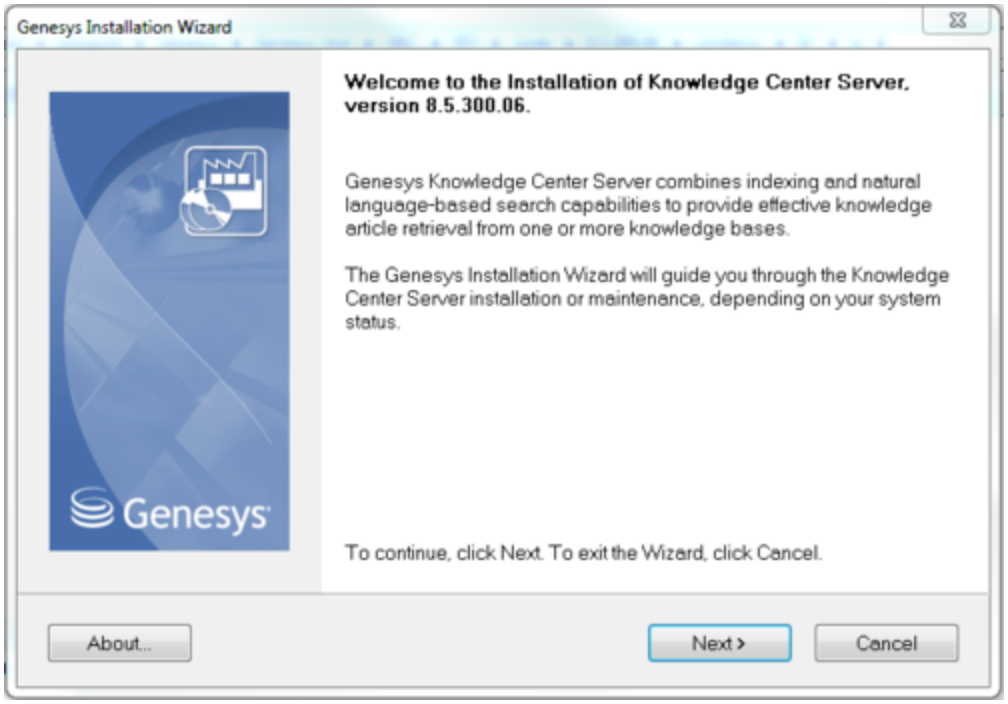
Installing Knowledge Center Server

Windows Installation Procedure

> **Important**
>
> From Knowledger Center Server version 8.5.302.04, you must install the Visual C++ Redistributable Packages run-time components which are required to run C++ applications on Windows.

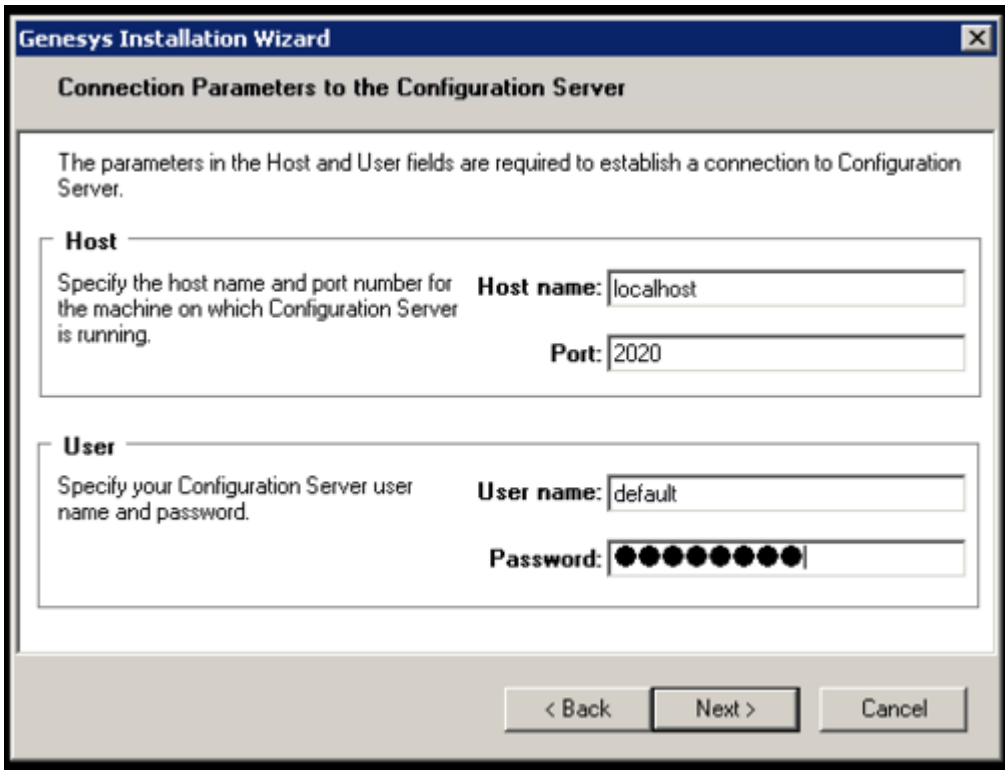
Start

1. In your installation package, locate and double-click the *setup.exe* file. The Install Shield opens the welcome screen.



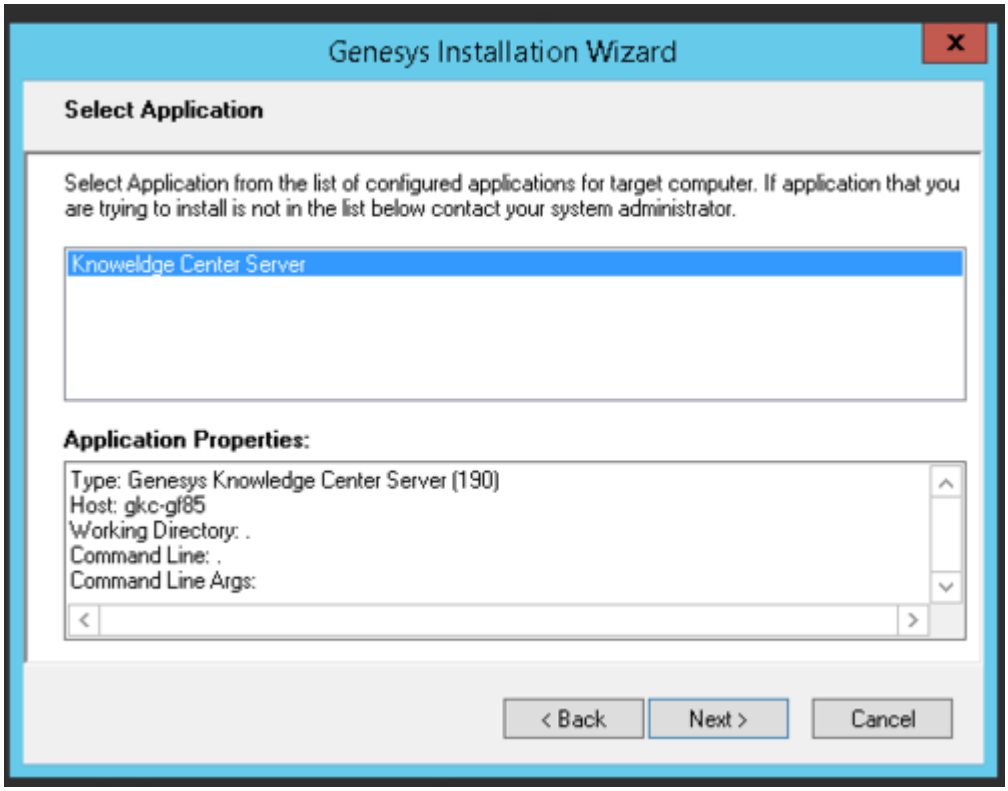
Knowledge Center Server Installation Window

2. Click **Next**. The **Connection Parameters to the Configuration Server** screen appears.



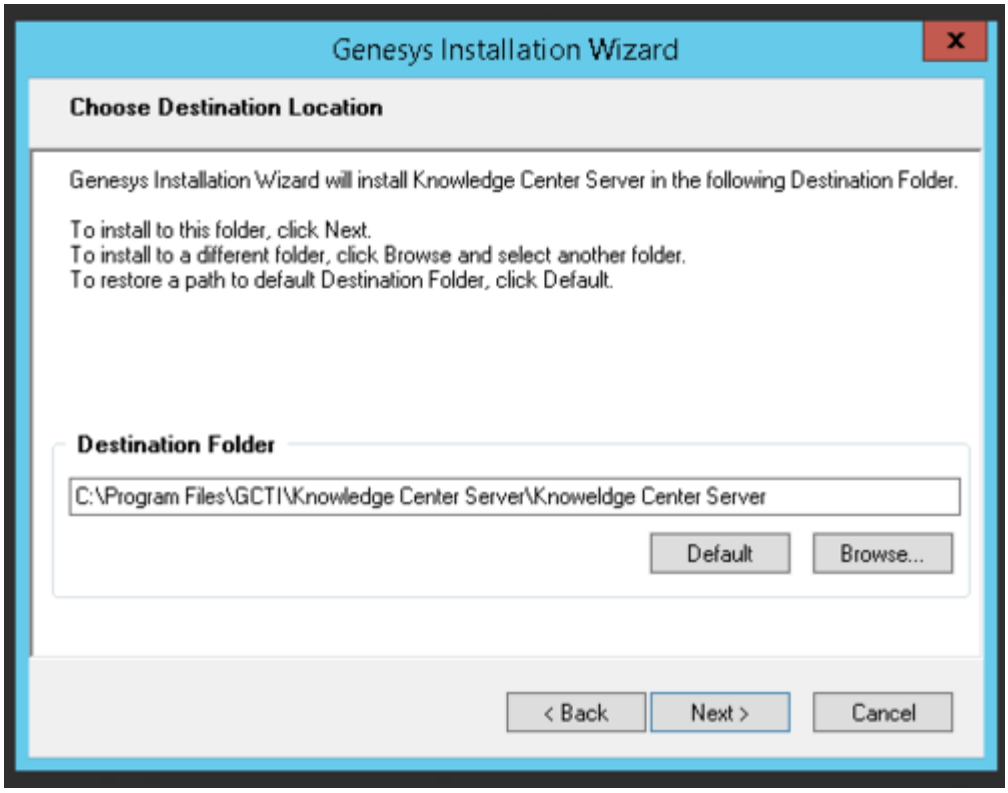
Knowledge Center Server Connection Parameters

3. Under **Host**, specify the host name and port number where Configuration Server is running. (This is the main listening port entered in the **Server Info** tab for Configuration Server.)
4. Under **User**, enter the user name and password for logging into Configuration Server.
5. Click **Next**. The **Select Application** screen appears.



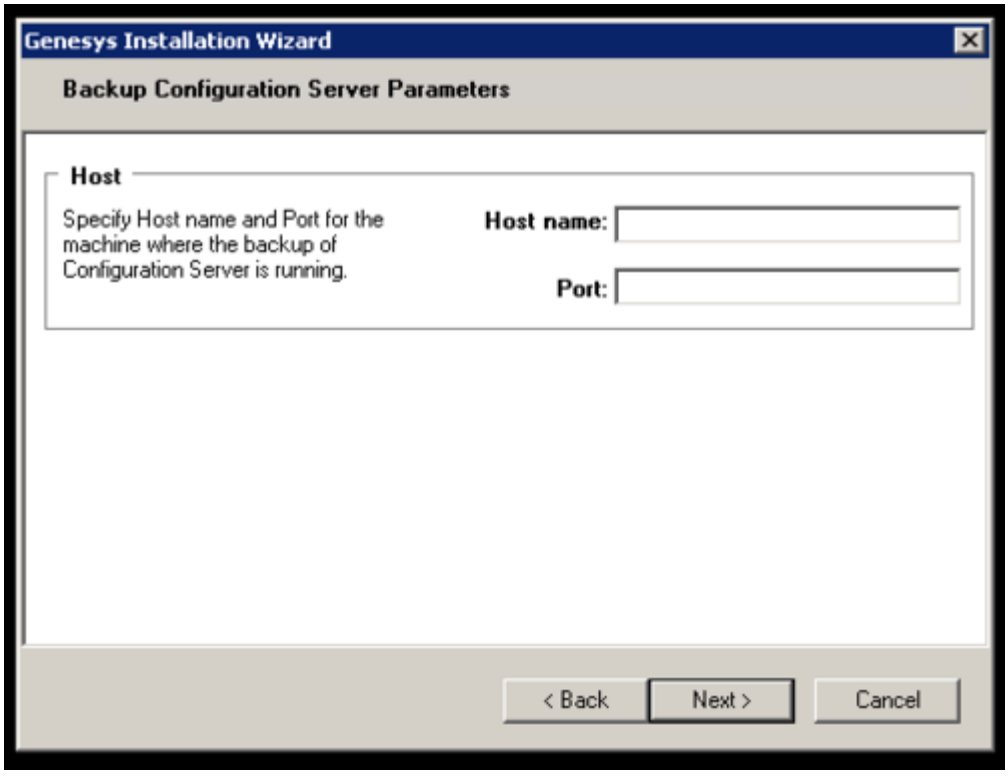
Selecting the Knowledge Center Server Application

6. Select the Knowledge Center Server application that you are installing. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the **Server Info** and **Start Info** tabs of the selected Application object.
   Note: You might see "Reserved Application 6(190)" as the type under the application properties of the selected application. This happens when older versions of Configuration Server are used.
7. Click **Next**. The **Choose Destination Location** screen appears.



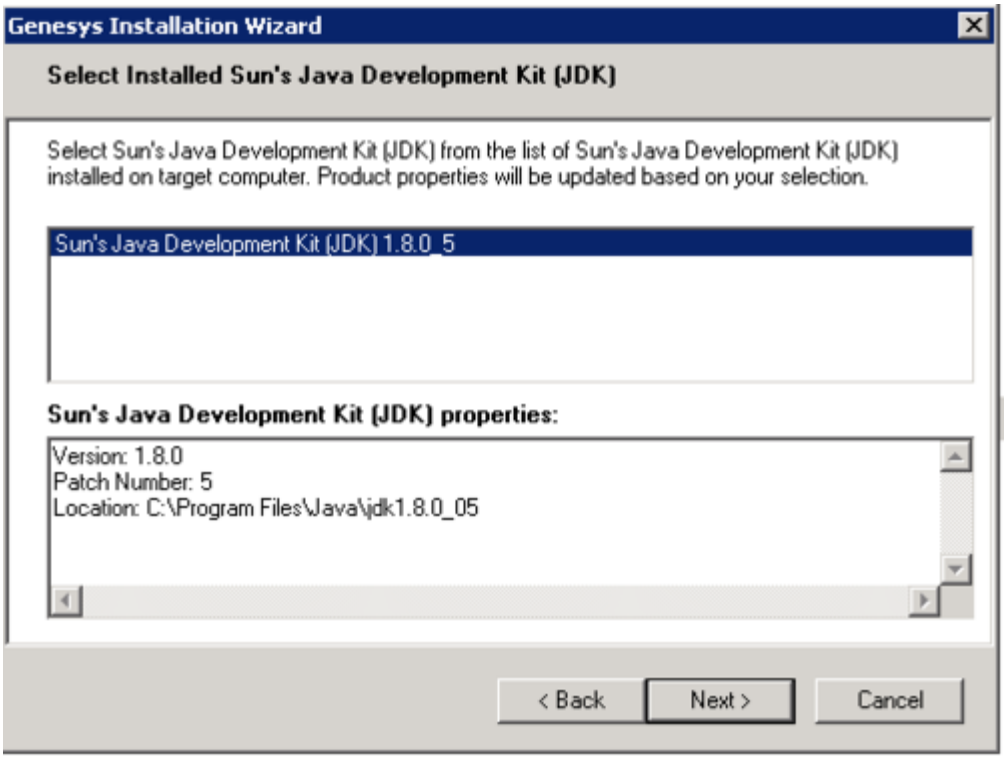
Choosing the Knowledge Center Server Installation Destination

8. Under **Destination Folder**, keep the default value or browse to the desired installation location.
9. Click **Next**. The **Backup Configuration Server Parameters** screen appears.



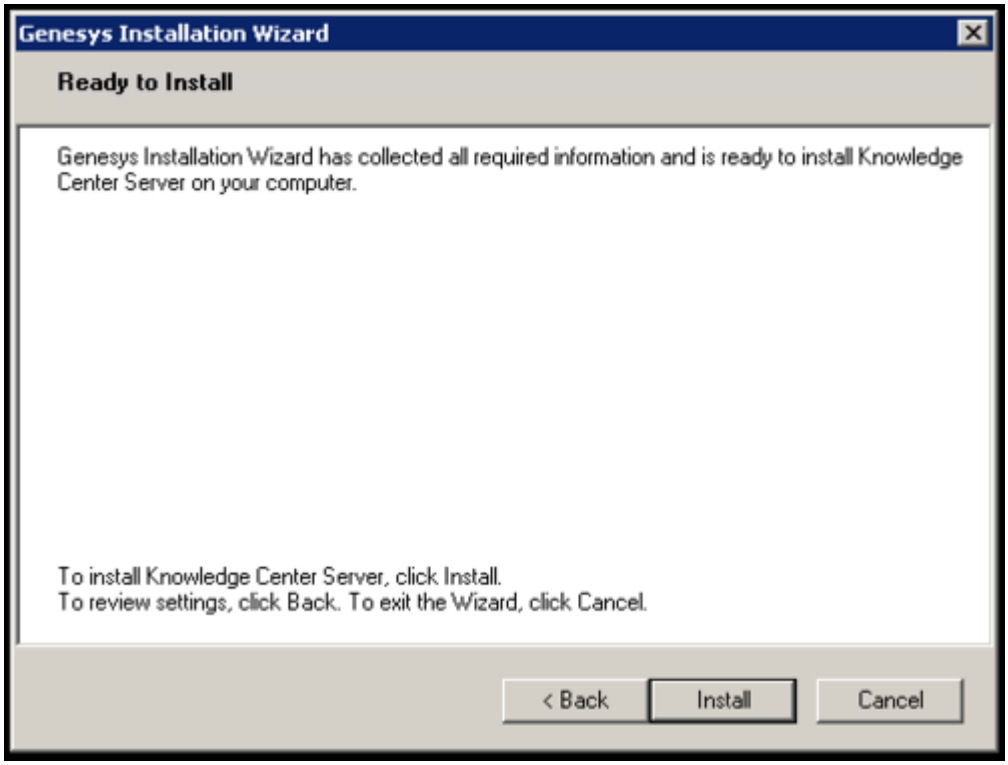
Knowledge Center Backup Config Server Parameters

10. If you have a backup Configuration Server, enter the **Host name** and **Port**.
11. Click **Next**. Choose the appropriate version of the Java JDK.
    Note: Knowledge Center Server requires Java 1.7 or higher.



Selecting the Knowledge Center Server Java Version

12. Click **Next**. The **Ready to Install** screen appears.



Knowledge Center Server is Ready to Install

13. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation. When through, the **Installation Complete** screen appears.
14. Click **Finish** to complete your installation.
15. Inspect the directory tree of your system to make sure that the files have been installed in the location that you intended.

End

Linux Installation Procedure

Start

1. Open a terminal in the Genesys Knowledge Center Server CD/DVD or the Genesys Knowledge Center Server installation package and run the install.sh file. The Genesys installation starts.
2. Enter the hostname of the host on which you are going to install.
3. Enter the connection information required to log in to the Configuration Server:
   a. Hostname—For instance, demosrv.genesyslab.com
   b. Listening port—For instance, 2020
   c. User name—For instance, demo
   d. Password
4. If you have a backup Configuration Server, enter the Host name and Port.
5. If the connection settings are successful, a list of keys and Genesys Knowledge Center Server applications is displayed.
6. Enter the key for the Genesys Knowledge Center Server application that you created previously on Configuration Server.
7. Enter the full path to your installation directory and confirm that it is correct.

If the installation is successful, the console displays the following message: *Installation of Genesys Knowledge Center Server, version 8.5.x has completed successfully.*

End

Installing multiple Server instances

To install multiple server instances you need to repeat following steps for every instance:

1. Create Server applications
2. Configuring the Knowledge Center Server Application
3. Installing Knowledge Center Server

Note: Knowledge Center Cluster Application is created just ones for all server instances working in

the same cluster.

> **Important**
> It is advices to do not co-locate several Knowledge Center Server instances on the same host.

Understanding the Knowledge Center Server Configuration Files

This section describes how to work with the configuration files stored in the Knowledge Center Server.

Indexing Engine Configurationn

1. Go to the *./server* folder and open the *gks.yml* configuration file.
2. Configure the following settings:
   a. index.number_of_shards: # – Number of ElasticSearch shards per each knowledge base (default: 1)
   b. path.data : [PATH] – Path to the folder that contains index data for this node (default: */gks/data*)
   c. path.plugins : [PATH] – path to Pulse Plugins

> **Important**
> - Knowledge Center Server needs to be restarted to apply changed parameters.
> - index.number_of_shards parameter will be applied to the newly created knowledge bases only.

Geo-location database

- Database for Geo-IP Location (the way to translate client IP to it's geographical location)
- The path to */linguatools/geoip/GeoLite2-City.mmdb* can be changed in the *gks.yml* file: *path.geoip*.

Language Resources Configuration

- On Windows:
  - The path to */linguatools/freeling/data/* can be changed in the *gks.yml* file: *path.freeling*.
  - The path to /linguatools/freeling/bin/ can be changed in the gks.yml file: *path.freeling.dll*.

- On Linux:
  - The path to */linguatools/freeling/data/* can be changed in the *gks.yml* file: *path.freeling*.
  - *setenv.sh* exports the following environment variables:
    - FREELINGSHARE – Path to *Path to installation directory/linguatools/freeling/data*
    - LD_LIBRARY_PATH – Path to *Path to installation directory/linguatools/freeling/bin*

Provide Knowledge Center Access to Agents

> **Tip**
>
> Access to a knowledge base may be limited by an agent's assigned skills (see Installing and Using the Administrator Plugin). Please add the appropriate skills so your agent may see the required knowledge bases (see Bulk Assignment of Skills to Agents for more information).

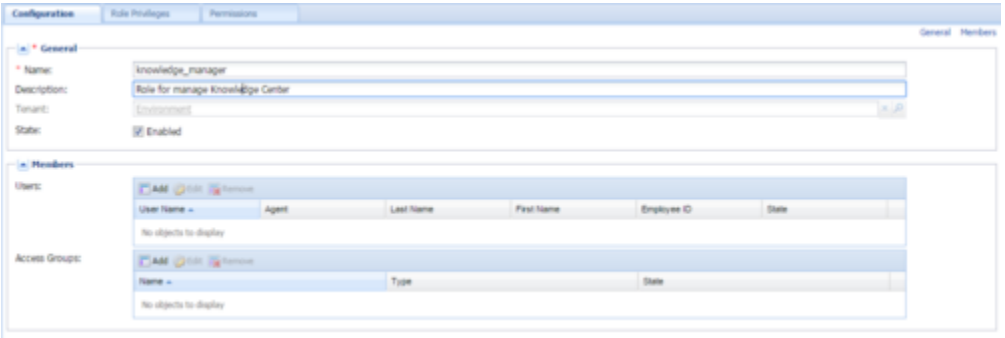
Genesys Knowledge Center supports the following privileges to restrict agent access:

- Allows agent to change data in a knowledge base (suggested for authors)
- Allows agent to manage knowledge bases (suggested for administrators)
- Allows agent to use reporting capabilities (suggested for supervisors)
- Allows to bypass tenants restrictions (suggested for user configured in CMS for "Log On Account" in case of multi-tenant configuration)

To configure the appropriate privileges for an Agent:

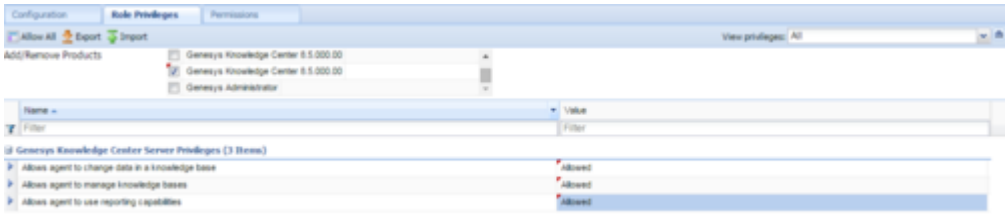
Start

1. Go to **Provisioning > Accounts > Roles**.
2. In the taskbar, click **New** to create a new object.
3. Set the name of the role in the **General** section.



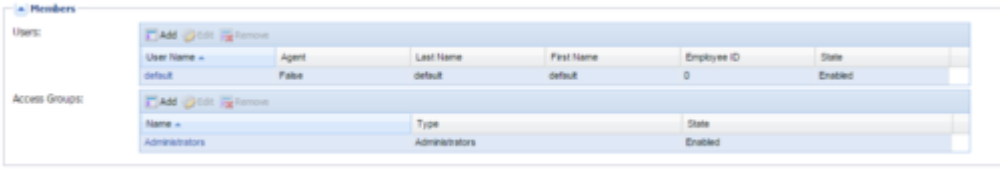
Knowledge Center Server Access Roles

4. Go to the **Role Privileges** tab and select the set of roles for Genesys Knowledge Center.
5. Open the list of privileges for Knowledge Center Server.
6. Set the appropriate privileges to **Allowed**.



Setting Knowledge Center Server Access Privileges

7. Go back to the **Configuration** tab.
8. In the **Members** section, add the appropriate Agent by clicking the **Add**\ button.



Knowledge Center Server Members Section

9. Save and Close.

End

Start and Stop Genesys Knowledge Center Server

Start the Server

Windows:

> **Important**
>
> You can start the Genesys Knowledge Center Server on Windows from:
>
> - Windows Services
> - The server.bat script
> - Genesys Administrator

Start

- You can start the server from Windows Services:
  1. Open Windows Services
  2. Select and start the Genesys Knowledge Center Server [Knowledge Center Server] service.
- You can use the provided server.bat script:
  1. Navigate to the Knowledge Center Server installation server directory and launch the Windows command console (cmd.exe).
  2. Open server directory
  3. Type and execute server.bat, without any parameters.

> **Important**
>
> You can use entry in the **Start > All Programs > Genesys Solutions > Knowledge Center Server [Knowledge Center Server]** menu to start the Server using server.bat

- You can start the server from Genesys Administrator:
  1. Navigate to PROVISIONING > Environment > Applications.
  2. Select the Knowledge Center Server.
  3. Click Start applications in the Runtime panel.

End

The Genesys Knowledge Center Server is shown in Started status in Genesys Administrator.

Linux:

> **Important**
>
> You can start the Genesys Knowledge Center Server on Windows from:
>
> - The server.sh script
> - Genesys Administrator

Start

- You can use the provided server.sh script:
  1. Navigate to the Genesys Knowledge Center Server installation directory in the Unix command console.
  2. Go to server directory

3. Type and execute server.sh, without any parameters.

- You can start the server from Genesys Administrator:
  1. Navigate to PROVISIONING > Environment > Applications.
  2. Select the Knowledge Center Server.
  3. Click Start applications in the Runtime panel.

End

The Genesys Knowledge Center Server is shown in Started status in Genesys Administrator.

After the Server start

After successful Server start you can use following URLs in your browser:

- http://<host>:<default_port>/gks-server - to access the Server REST API
- http://<host>:<default_port>/gks-sample-ui - to access Sample UI application shipped with product (**Note**: you need to load some data to be able to play with this application - reference on Quick Guide.)

Stop the Server

Windows:

> **Important**
>
> You can stop the Genesys Knowledge Center Server on Windows from:
>
> - Windows Services
> - Genesys Administrator
> - A console window

Start

- You can stop the server from Windows Services:
  1. Open Windows Services
  2. Select and stop the Knowledge Center Server service.
- You can stop the server from Genesys Administrator:
  1. Navigate to PROVISIONING > Environment > Applications.
  2. Select the Knowledge Center Server.
  3. Click Stop applications in the Runtime panel.
- If you previously started Genesys Knowledge Center Server in a console window, you can stop the

server by closing the window or navigate to Genesys Knowledge Center Server installation directory in Windows console (cmd.exe), open server directory and execute comand: server.bat stop

End

The Genesys Knowledge Center Server is shown in Stopped status in Genesys Administrator.

Linux:

> **Important**
>
> You can stop the Genesys Knowledge Center Server on Linux from:
>
> - Genesys Administrator
> - A console window

Start

- can stop the server from Genesys Administrator:
  1. Navigate to PROVISIONING > Environment > Applications.
  2. Select the Knowledge Center Server.
  3. Click Stop applications in the Runtime panel.
- Or you can stop the server from the console window where it was started:
  1. Press Ctrl+C while the window is active.
  2. Type Y and press Enter.
- Or you could use provided script server.sh:
  1. Navigate to the Genesys Knowledge Center Server installation directory in the Unix command console.
  2. Go to server directory
  3. Type and execute server.sh with parameter "stop" (for example: server.sh stop)

End

The Genesys Knowledge Center Server is shown in Stopped status in Genesys Administrator.

Installing the Knowledge Center CMS

The 8.5.3 release of the Genesys Knowledge Center incorporates two major changes in the CMS component architecture:

- The cluster mode is the default and the only mode to run CMS (no matter you start with just one node in a cluster or your cluster already have multiple nodes). This puts the restriction on the persistent storages used by the CMS.
- The persistent storage selection is one of the important choices that need to be done when planning a deployment of Genesys Knowledge Center CMS. Knowledge Center support following presistent storages:
  - Microsoft SQL Server 2012
  - Oracle 11g (supported from version 8.5.302.xx)
  - Cassandra 2.2.x (deprecated)

> **Important**
>
> To ensure strong consistency of the data (no matter the configuration of the underlying storage) it was decided to deprecate Cassandra 2.2.x support in 8.5.3 release of the product. Going forward, **support of Cassandra will be discontinued in 9.0 release of the product**.

> **Important**
>
> Starting from 8.5.3 release, usage of the file storage has been discontinued. (It does not provide the desired reliability for storage, particular medium to large data amounts, and it is incompatible with cluster mode.)

- To ensure fast operations with CMS, CMS now comes with indexing engine that helps to execute your requests in a timely manner.

Install the CMS

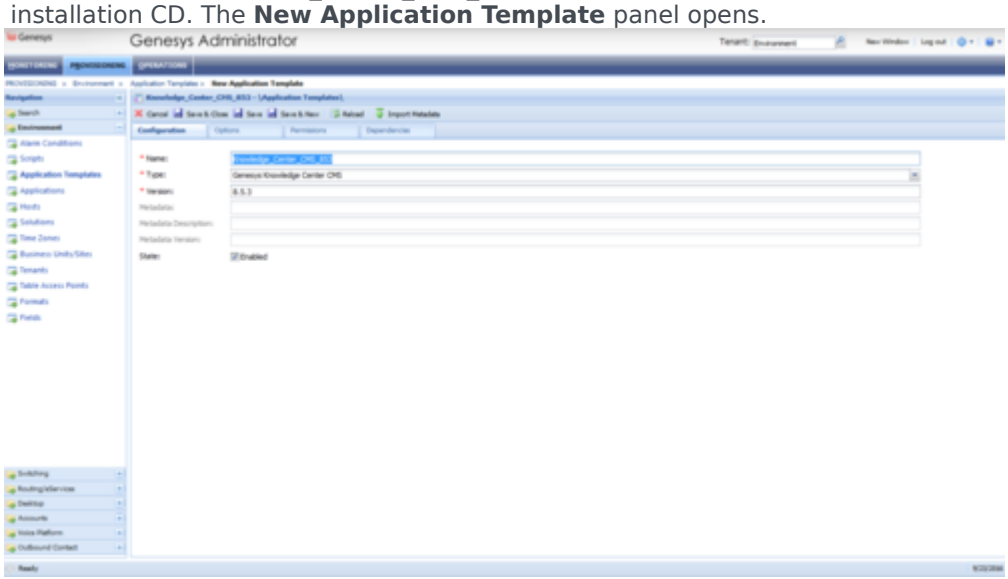
Import the CMS Application Template

Start

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application**

Templates.

2. In the **Tasks** panel, click **Upload Template**.
3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Browse to the *Knowledge_Center_CMS_853.apd* file available in the templates directory of your installation CD. The **New Application Template** panel opens.



The Knowledge Center CMS Application Template

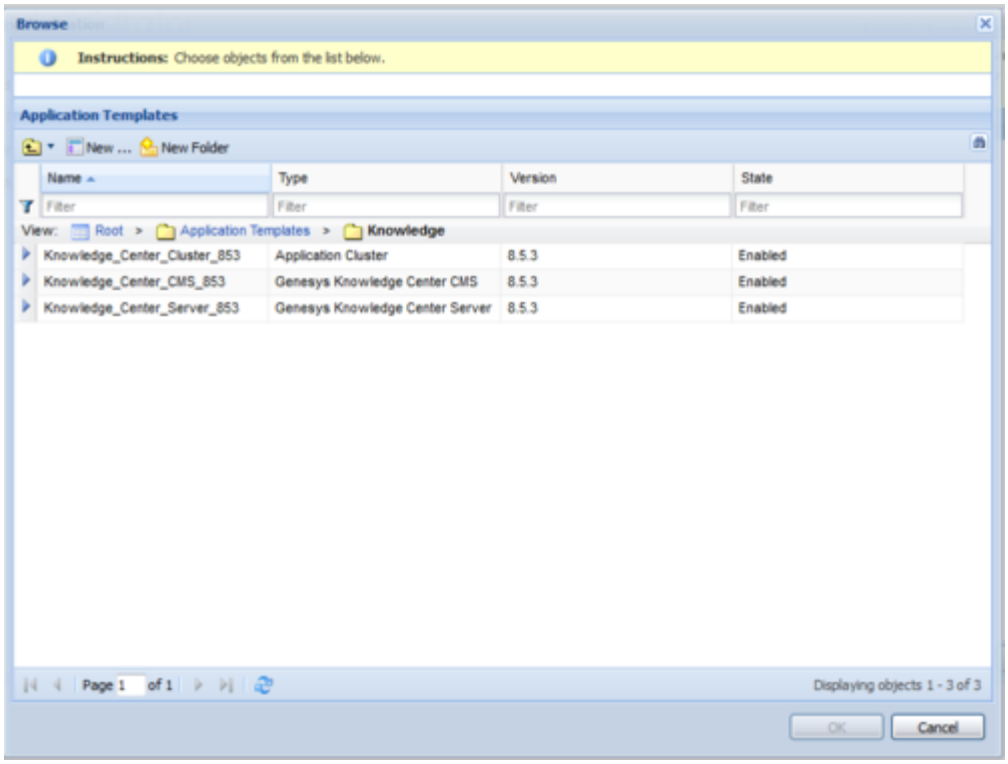
5. Click **Save and Close**.

End

Create CMS Applications

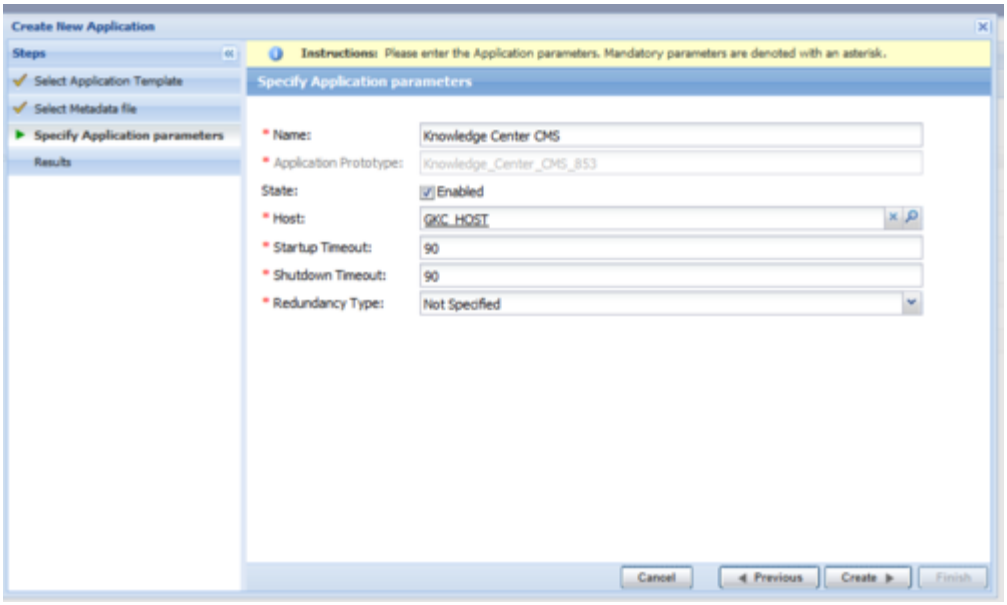
Start

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**.
2. In the **Tasks** panel, click **Create New Application**.
3. In the **Select Application Template** panel, click **Browse for Template** and select the Genesys Knowledge Center CMS application template that you imported earlier. Click **OK**.



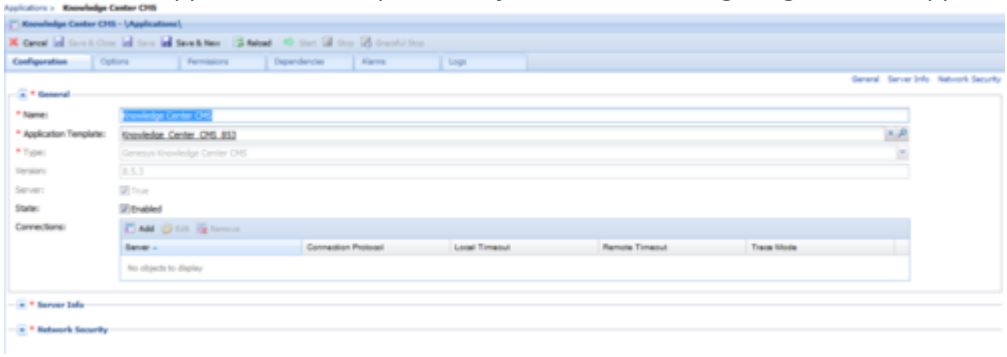
Selecting the Knowledge Center CMS Template

4. The template is added to the **Select Application Template** panel. Click **Next**.
5. In the **Select Metadata** file panel, click **Browse** and select the *Knowledge_Center_CMS_853.xml* file. Click **Open**.
6. The metadata file is added to the **Select Metadata** file panel. Click Next.
7. In **Specify the appropriate application parameters**:
   a. Enter a name for your application. For instance, *Knowledge Center CMS*.
   b. Enable the **State**.
   c. Select the Host on which the CMS will reside.
   d. Click **Create**.



Creating the Knowledge Center CMS Application

5. The **Results** panel opens.
6. Enable **Opens the Application details form after clicking 'Finish'** and click **Finish**. The Knowledge Center CMS application form opens and you can start configuring the CMS application.



Configuring the Knowledge Center CMS

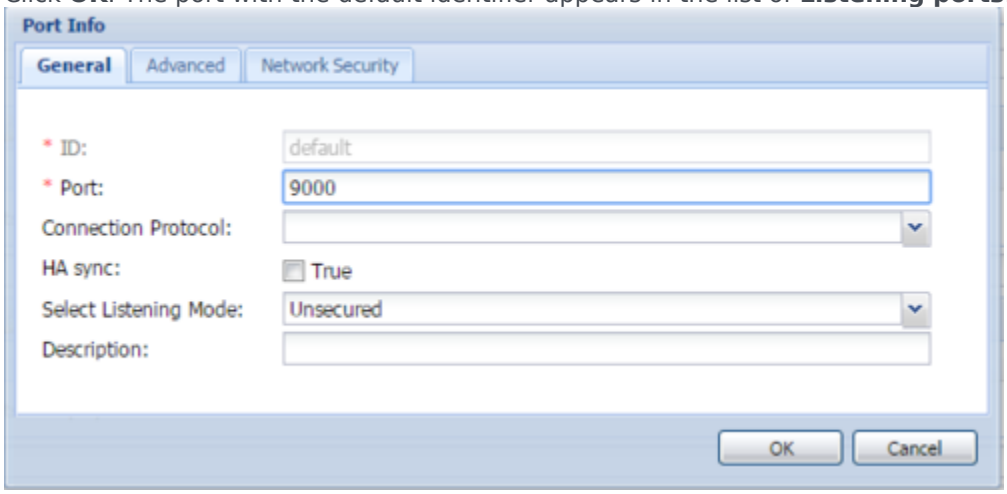
End

Configure the CMS Application

Start

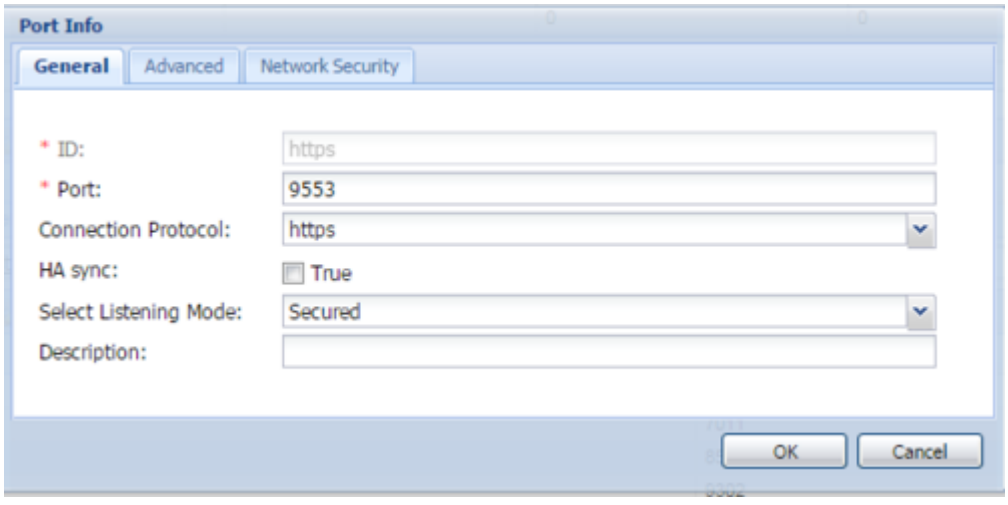
1. If your Knowledge Center CMS application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the Knowledge Center CMS and click **Edit...**.
2. In the **Connections** section of the **Configuration** tab, click **Add**. The **Browse for applications** panel opens.
3. Select the Knowledge Center Cluster application, then click **OK**.
4. Expand the **Server Info** pane.

5. If your Host is not defined, click the lookup icon to browse to the hostname of your application.
   Note: Cassandra Resource access point will need to point on one of the seed nodes you have in your Cassandra deployment.
6. In the **Listening Ports** section, create the default port by clicking **Add**. The **Port Info** dialog opens.
   a. Enter the **Port**. For instance, 9000.
   b. Click **OK**. The port with the default identifier appears in the list of **Listening ports**.



Knowledge Center CMS Port Information

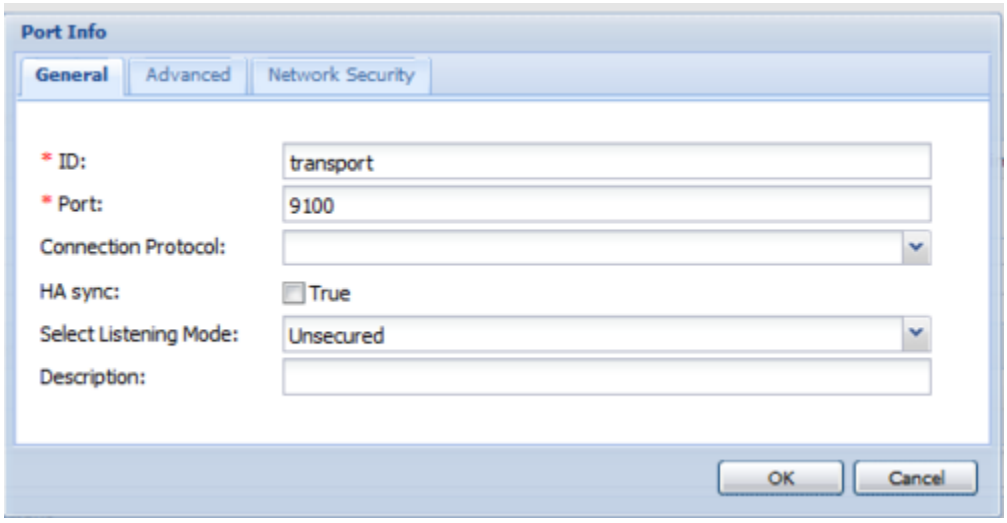
7. Optionally, you can add a secure listening port for authenticated users, secured connections. Click **Add**. The **Port Info** dialog opens.
   a. Enter *https* for the **ID** field.
   b. Enter the port . For instance, 9553.
   c. Enter https for the **Connection Protocol**.
   d. Choose **Secured** for the **Listening Mode**.
   e. Click **OK**.



GKS CMS https port

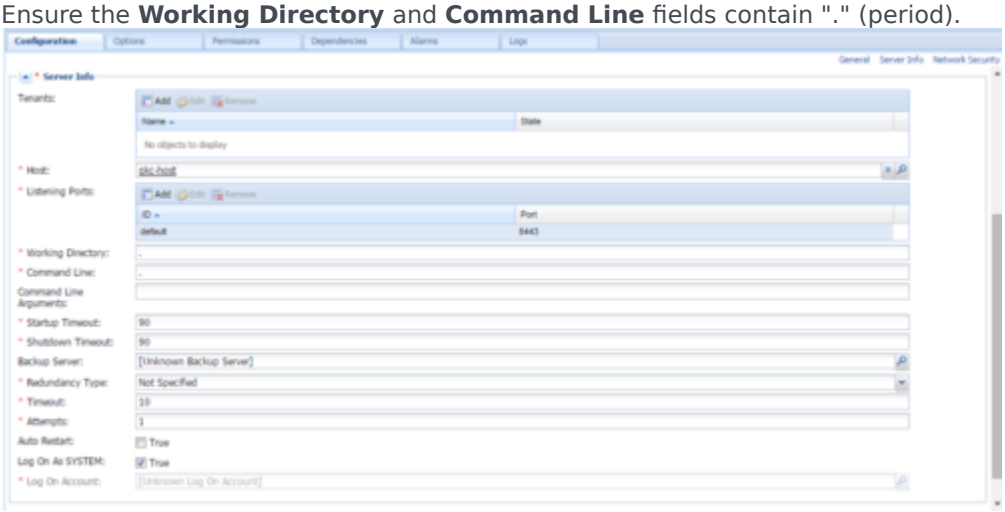
Note: If https port is enabled, service will only be available on https port and http connection will be unavailable.

6. Optionally, you can explicitly add Transport port for ElasticSearch engine. If you do not define transport port, port 9301 will be used. To specify the port, click the **Add** button. The **Port Info** dialog opens.
   a. Enter transport for the **ID** field.
   b. Enter the **Port**. For instance, 9100.
   c. Click **OK**.



7. Optionally, you can explicitly add JGroups for communication between CMS nodes in the cluster. If you do not define this port, default value depending on chosen type of communication will be used . To specify the port, click the **Add** button. The **Port Info** dialog opens.
   a. Enter jgroups for the **ID** field.
   b. Enter the **Port**. For instance, 9110.
   c. Click **OK**.

8. Ensure the **Working Directory** and **Command Line** fields contain "." (period).



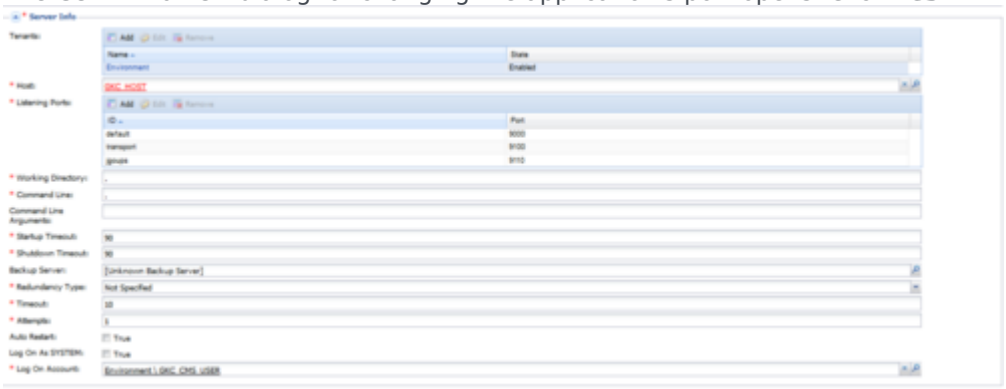
Knowledge Center CMS Information

9. In the **Tenants** section, add a working tenant by clicking **Add**. Browse and choose the appropriate tenant in the pop-up dialog. Click **OK**.

> **Important**
>
> For Knowledge Center 8.5.302.xx and earlier Cluster can only work under a single tenant. Starting from 8.5.303.xx release of the product it supports multiple tenants within one cluster deployment. Application Cluster and GKC server/CMS should be in the same tenant

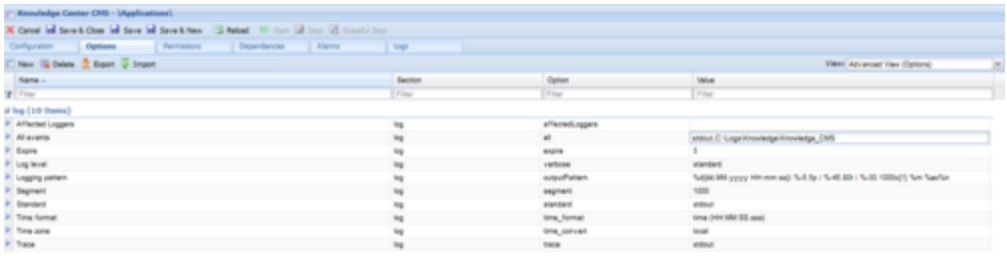
10. Uncheck **Log On As SYSTEM**.
11. In **Log On Account** specify the user account that:
    - has the ability to view access groups (this is required if you use access groups to set privileges for your agents)
    - has **Knowledge.AUTHOR** (Allows agent to change data in a knowledge base) privilege and **Knowledge.MULTITENANT** (Allows to bypass tenants restrictions) in case multi-tenant configuration (required for scheduled synchronization)
    - User should have access to the same tenant/tenants in which that CMS is configured
    - User should be granted "Read and Execute (RX)" and "Read Permissions (E)" permissions for Environment tenant, if the application configured not in the Environment tenant; user should belong to Administrators Access Group in CMS tenants (required for scheduled synchronization)

14. Click **Save**.
15. The **Confirmation** dialog for changing the application's port opens. Click **Yes**.



Knowledge Center CMS Information

16. Go to **Application Cluster** application, open **Options** tab. In section cms.general set valid URL to CMS or CMS cluster load balancer in externalURL option (like http://<cms host>:<CMS default port>/gks-cms).
17. (Optional) Select the **Options** tab. In the [log] section, the all option is set to stdout by default. Enter a filename if you wish to enable logging to a file. For example, you can enter stdout, C:\Logs\Knowledge\Knowledge_CMS to force the system to write logs both to the console and to a file.



End

Configure Data Source

Knowledge Center CMS requires persistent storage to be configured to store all the authored content. Please follow one of the instructions to set up storage of your choice:

> ### Important
> To ensure strong consistency of the data (no matter the configuration of the underlying storage) it was decided to deprecate Cassandra 2.2.x support in 8.5.3 release of the product. Going forward, **support of Cassandra will be discontinued in 9.0 release of the product**.

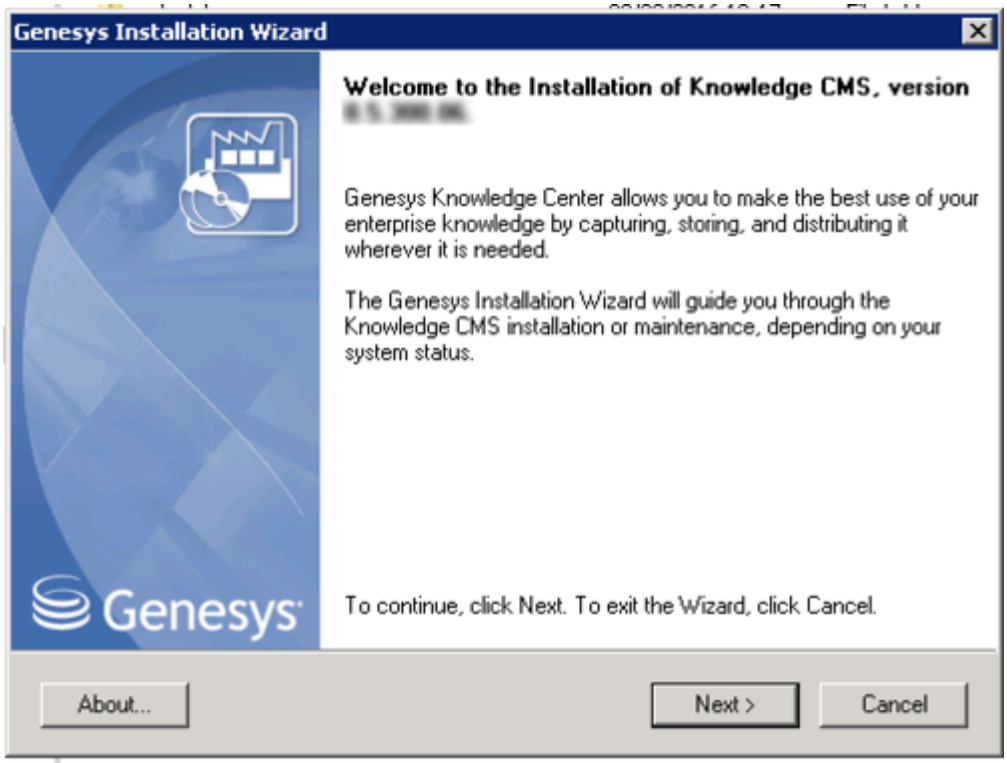
- Microsoft SQL Server - Using CMS with Microsoft SQL Server
- Oracle - Using CMS with Oracle
- Cassandra - Using CMS with Cassandra
- PostgreSQL - Using CMS with PostgreSQL

Installing the CMS

Windows Installation Procedure

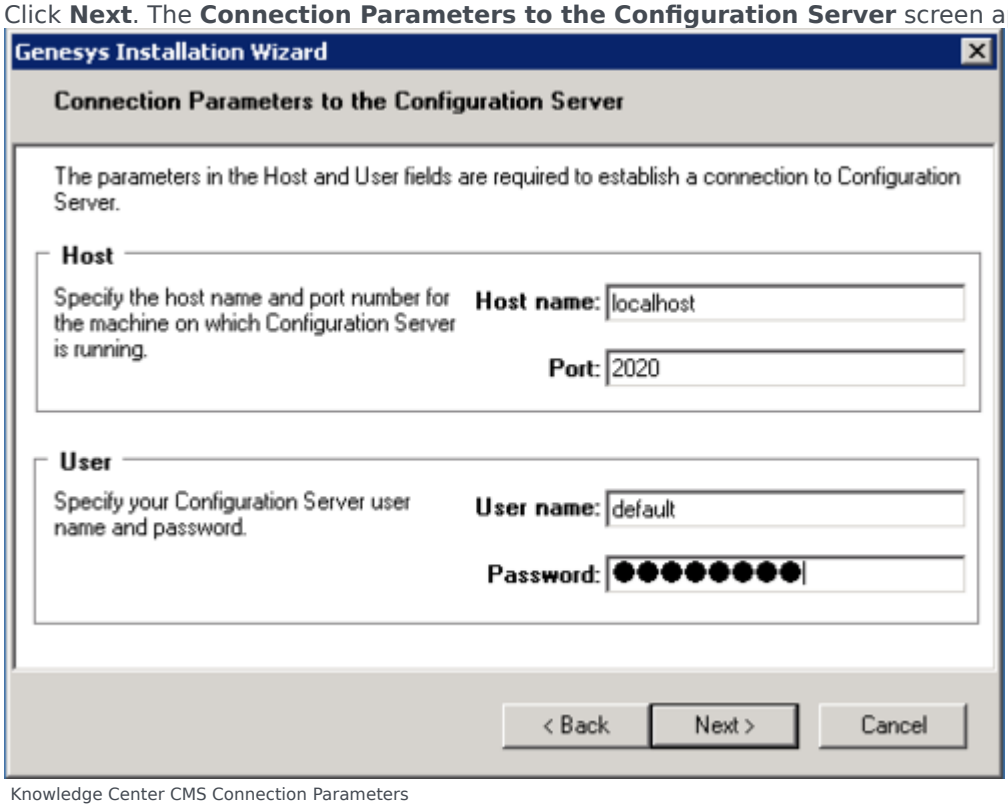
Start

1. In your installation package, locate and double-click the *setup.exe* file. The Install Shield opens the welcome screen.



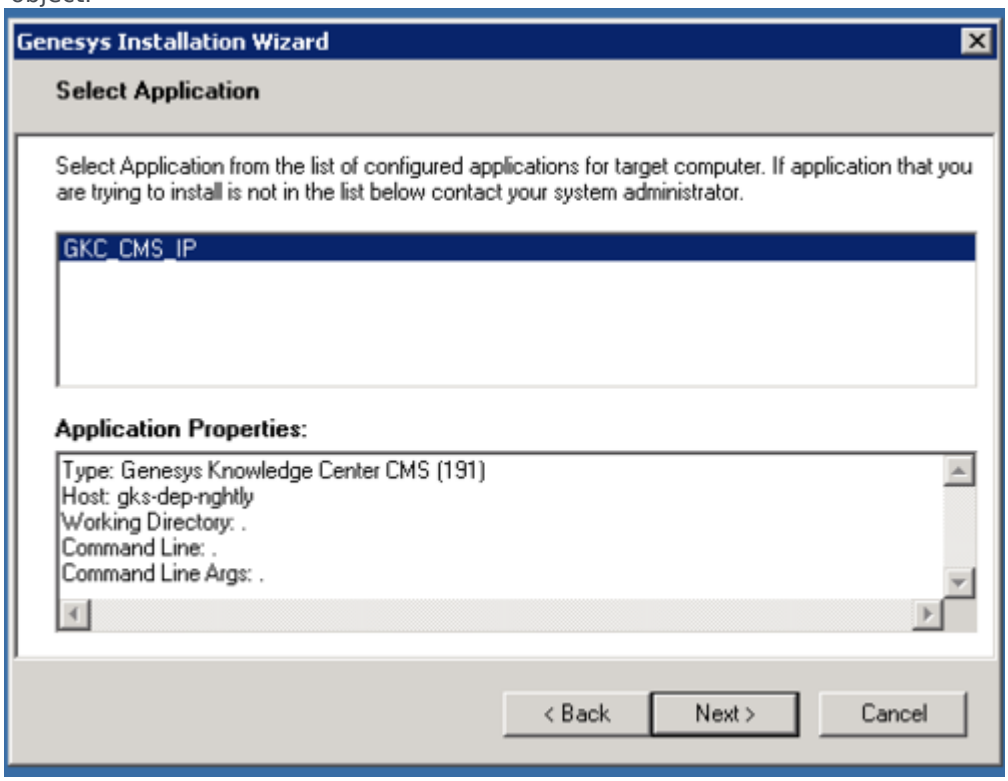
Knowledge Center CMS installation Window

2. Click **Next**. The **Connection Parameters to the Configuration Server** screen appears.



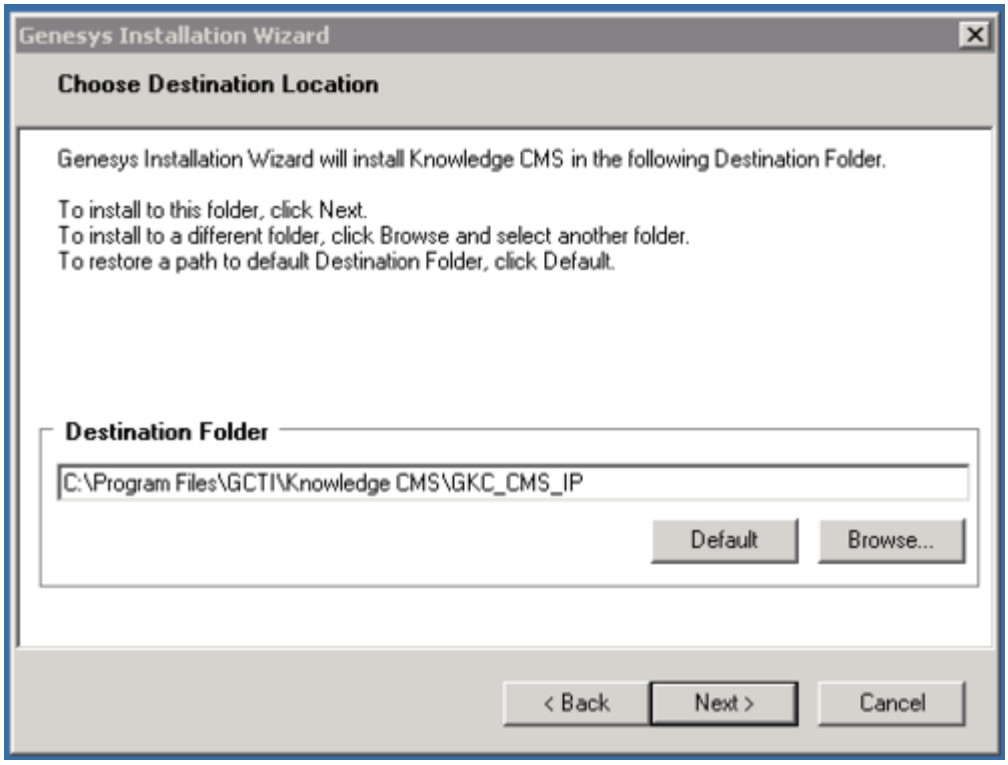
Knowledge Center CMS Connection Parameters

3. Under **Host**, specify the host name and port number where Configuration Server is running. (This is the main listening port entered in the **Server Info** tab for Configuration Server.)
4. Under **User**, enter the user name and password for logging in to Configuration Server.
5. Click **Next**. The **Select Application** screen appears.
6. Select the Knowledge Center CMS that you are installing. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the **Server Info** and **Start Info** tabs of the selected application object.



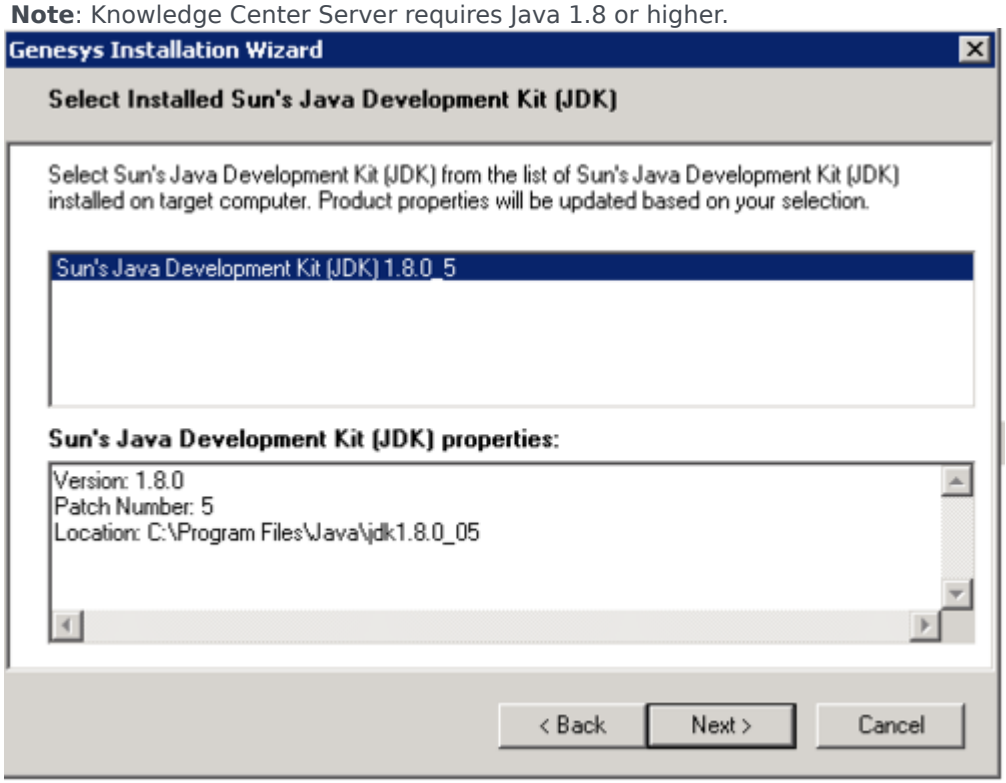
Selecting the Knowledge Center CMS Application

7. Click **Next**. The **Choose Destination Location** screen appears.
8. Under **Destination Folder**, keep the default value or browse for the desired installation location.



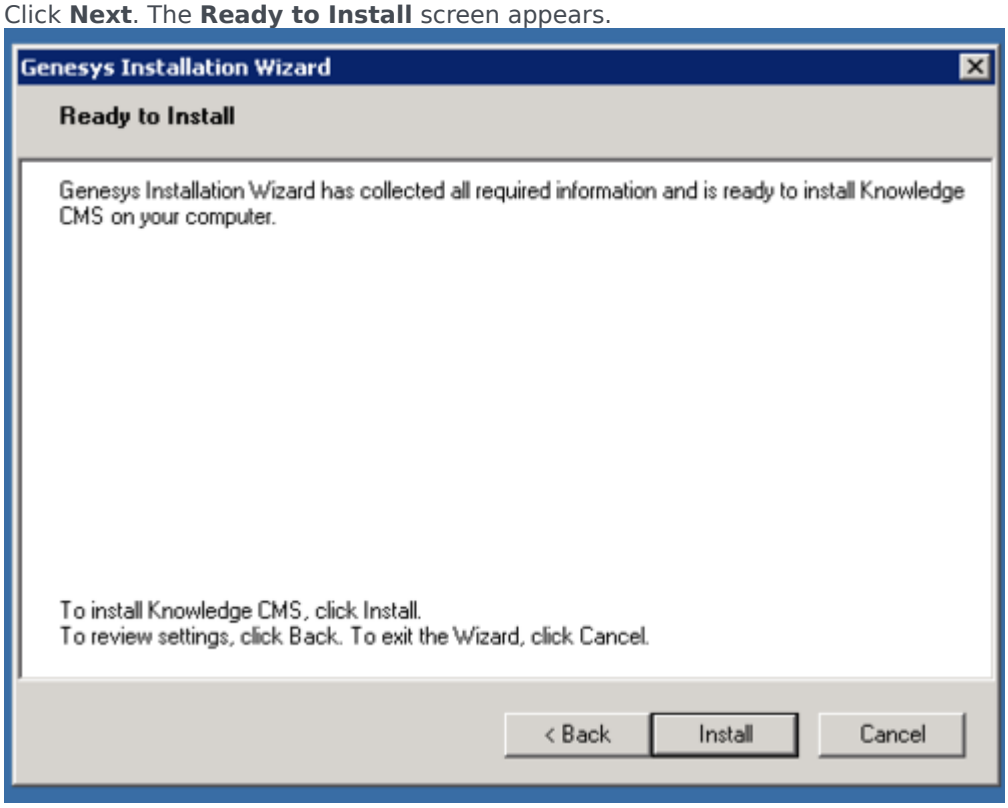
Choosing the Knowledge Center CMS Installation Destination

9. Click **Next**. Choose the appropriate version of the Java JDK.
   Note: Knowledge Center Server requires Java 1.8 or higher.



Selecting the Knowledge Center CMS Java Version

10. Click **Next**. The **Ready to Install** screen appears.



Knowledge Center Knowledge Center is Ready to Install

11. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for the Genesys Knowledge Center CMS. When through, the **Installation Complete** screen appears.
12. Click **Finish** to complete your installation.
13. Inspect the directory tree of your system to make sure that the files have been installed in the location that you intended.

End

Linux Installation Procedure

Start

1. Open a terminal in the CMS installation package, and run the *install.sh* file. The Genesys installation starts.
2. Enter the hostname of the host on which you are going to install.
3. Enter the connection information required to log in to the Configuration Server:
   a. **Hostname**—For instance, *demosrv.genesyslab.com*
   b. **Listening port**—For instance, *2020*
   c. **User name**—For instance, *demo*
   d. **Password**

4. If you have a backup Configuration Server, enter the Host name and Port.
5. If the connection settings are successful, a list of keys and Knowledge Center CMS applications is displayed.
6. Enter the key for the Knowledge Center CMS application that you created previously in Configuration Server.
7. Enter the full path to your installation directory and confirm that it is correct.
8. If the installation is successful, the console displays the following message:
   Installation of Genesys Knowledge CMS, version 8.5.x has completed successfully.

End

Installing multiple CMS instances

To install multiple CMS instances you need to repeat following steps for every instance:

1. Create CMS applications
2. Configuring the Knowledge Center CMS Application
3. Installing Knowledge Center CMS

Note: Knowledge Center Cluster Application is created just ones for all CMS instances working in the same cluster.

Configuring the CMS

The Knowledge Center CMS includes an embedded Jetty server. After installation, you can carry out your initial configuration by creating a *work* directory for temporary Jetty files inside the *./server* folder.
Starting from 8.5.303.xx release of the product "work" folder will be created automatically during installation.

Configure Required CMS Access Options

Genesys Knowledge Center supports the following privileges to restrict agent access:

- Administrator (allow user to carry out Administrators task like create and edit Knowledge bases)
- Approver (allo user to Approve and Publish documents)
- Category Author (allow user to create and update categories)
- Document Author (allow user to create and update documents)
- Multitenant user (Allows user to work with data in all tenants in the CMS)

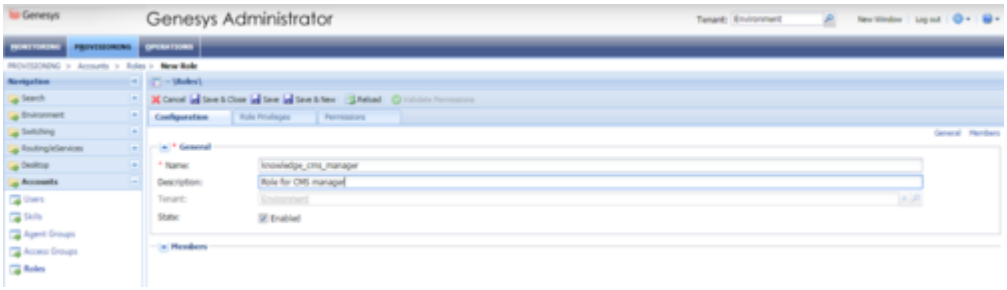
Important

Only agents who have both Document Author and Category Author privileges can successfully import data from XML files into CMS.
To publish document from CMS to Knowledge Server agent also should have "Allows agent to change data in a knowledge base" privilege on Knowledge Server (link to Provide Knowledge Center Access to Agents in Server installation page)

To configure the appropriate privileges for an agent:

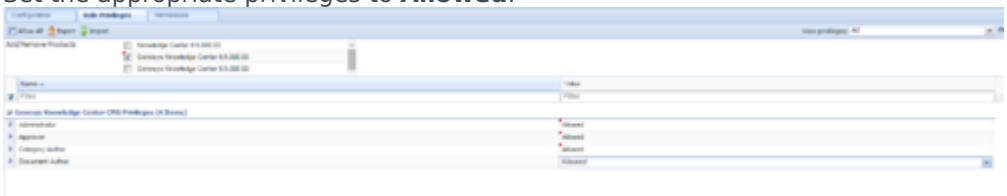
Start

1. Go to **Provisioning > Accounts > Roles.**
2. In the taskbar, click **New** to create a new object.
3. Set the name of the role in the **General** section.



Knowledge Center CMS Access Roles

4. Go to the **Role Privileges** tab and select the set of roles for Genesys Knowledge Center.
5. Open the Genesys Knowledge Center CMS privileges list.
6. Set the appropriate privileges to **Allowed**.



Setting Knowledge Center CMS Access Privileges

7. Go back to the **Configuration** tab.
8. In the **Members Section**, add the appropriate Agent by clicking the **Add** button.

Knowledge Center CMS Members Section

9. Save and Close.

End

Start and Stop Genesys Knowledge Center CMS

Start the CMS

Windows:

> **Important**
>
> You can start the Genesys Knowledge Center CMS on Windows from:
>
> - Windows Services
> - the server.bat script
> - Genesys Administrator

Start

- You can start the server from Windows Services.
  1. Open Windows Services
  2. Select and start the Genesys Knowledge Center CMS [Knowledge Center CMS] service.
- You can use the provided server.bat script.
  1. Navigate to the Knowledge Center CMS installation server directory and launch the Windows command console (cmd.exe).
  2. Open server directory
  3. Type and execute server.bat, without any parameters.

  > **Important**
  >
  > You can use entry in the Start > All Programs > Genesys Solutions > Knowledge Center CMS [Knowledge Center CMS] menu to start the Server using server.bat

- You can start the server from Genesys Administrator.
  1. Navigate to PROVISIONING > Environment > Applications.
  2. Select the Knowledge Center CMS
  3. Click Start applications in the Runtime panel.

End

The Genesys Knowledge Center CMS is shown in Started status in Genesys Administrator.

Linux:

> **Important**
> You can start the Genesys Knowledge Center CMS on Windows from:
>
> - the server.sh script
> - Genesys Administrator

Start

- You can use the provided server.sh script.
  1. Navigate to the Genesys Knowledge Center CMS installation directory in the Unix command console.
  2. Go to server directory
  3. Type and execute server.sh, without any parameters.
- You can start the server from Genesys Administrator
  1. Navigate to **PROVISIONING** > **Environment** > **Applications**.
  2. Select the Knowledge Center CMS.
  3. Click **Start applications** in the **Runtime** panel.

End
The Genesys Knowledge Center CMS is shown in Started status in Genesys Administrator.

After the CMS start

After successful CMS start you can use following URLs in your browser:

- http://<cms host>:<CMS default port>/gks-cms - to access the CMS user interface

Stop the CMS

Windows:

> **Important**
> You can stop the Genesys Knowledge Center CMS on Windows from:

- Windows Services
- Genesys Administrator
- A console window

Start

- You can stop the server from Windows Services.
  1. Open Windows Services
  2. Select and stop the Knowledge Center CMS service.
- You can stop the server from Genesys Administrator.
  1. Navigate to **PROVISIONING** > **Environment** > **Applications**.
  2. Select the Knowledge Center CMS.
  3. Click **Stop applications** in the **Runtime** panel.
- If you previously started Genesys Knowledge Center CMS in a console window, you can stop the server by closing the window or navigate to Genesys Knowledge Center CMS installation directory in Windows console (cmd.exe), open server directory and execute comand: server.bat stop

End
The Genesys Knowledge Center CMS is shown in Stopped status in Genesys Administrator.

Linux:

Important

You can stop the Genesys Knowledge Center CMS on Linux from:

- Genesys Administrator
- A console window

Start

- You can stop the server from Genesys Administrator.
  1. Navigate to **PROVISIONING** > **Environment** > **Applications**.
  2. Select the Knowledge Center CMS.
  3. Click **Stop applications** in the **Runtime** panel.
- Or you can stop the server from the console window where it was started.

1. Press Ctrl+C while the window is active.
2. Type Y and press Enter.

- Or you could use provided script server.sh:
  1. Navigate to the Genesys Knowledge Center CMS installation directory in the Unix command console.
  2. Go to server directory
  3. Type and execute server.sh with parameter "stop" (for example: server.sh stop)

End
The Genesys Knowledge Center CMS is shown in Stopped status in Genesys Administrator.

Deploying Cassandra Cluster

> **Important**
>
> To ensure strong consistency of the data (no matter the configuration of the underlying storage) it was decided to deprecate Cassandra 2.2.x support in 8.5.3 release of the product. Going forward, **support of Cassandra will be discontinued in 9.0 release of the product**.

Genesys recommends using an external Cassandra as the persistent storage for the data stored in Knowledge Center CMS. This chapter describes sample procedure of deploying and configuring Cassandra nodes. For more information please refer to Cassandra documentation.

> **Important**
>
> Genesys recommends that you use Linux when deploying an external Cassandra cluster.
> If you plan to establish secure communications with your Cassandra cluster, Genesys recommends that you carefully evaluate the related security considerations.

Deploy a Cassandra Cluster Node

Linux

Installation

1. Download version 2.2.3 or higher from the Cassandra 2.2 stream.
2. Unpack the archive into the installation directory, for example:

   ```
   cd /genesys tar xzf apache-cassandra-2.2.x-bin.tar.gz
   ```

   > **Important**
   >
   > Do not use paths with spaces when installing Cassandra 2.2

Configuration

1. Go to the directory where you installed your Cassandra node.

2. Edit conf/cassandra.yaml, using the following custom values:
   a. cluster_name: cluster name without spaces, for example GKC_Cassandra_Cluster
   b. seeds: <comma-separated list of fully qualified domain names (FQDN) or IP addresses of one or more Cassandra nodes>
      Note: This value must be the same for all nodes. Here are two examples:
      - 192.168.0.1,192.168.3
      - host1.mydomain.com, host2.mydomain.com
   c. storage_port: 7000 (default value)
   d. ssl_storage_port: 7001 (default value)
   e. listen_address: <current node host name>
      Note: This address is used for inter-node communication, so it must be available for use by other Cassandra nodes in your cluster.
   f. native_transport_port: 9042 (default value)
   g. rpc_address: <current node host name> Note: This address is used by Knowledge Center CMS to connect to
   h. Cassandra, so it must be available to all Knowledge Center CMS hosts.
   i. rpc_port: 9160 (default value)
   j. start_rpc: true
   k. endpoint_snitch: GossipingPropertyFileSnitch
      Note: Make sure that each Cassandra node has access to the ports specified for the other nodes.
3. Edit conf/cassandra-rackdc.properties.
4. Verify that the required communication ports are opened.

Setting Up a Cassandra Service

The sample script described in the following procedure should give you an idea of how to set up Cassandra as a service process.

1. Create the /etc/init.d/cassandra startup script.
2. Edit the contents of the file:

```
#!/bin/sh # # chkconfig: - 80 45 # description:
Starts and stops Cassandra # update daemon path to point to the
cassandra executable DAEMON=<Cassandra_installation_dir>
/bin/cassandra start() { echo -n "Starting Cassandra...
" $DAEMON -p /var/run/cassandra.pid echo "OK"
return 0 } stop() { echo -n "Stopping Cassandra... "
kill $(cat /var/run/cassandra.pid) echo "OK"
return 0 } case "$1" in start) start ;; stop) stop ;;
restart) stop start ;; *) echo $"Usage: $0
{start|stop|restart}" exit 1 esac exit $?
```

3. Make the file executable: sudo chmod +x /etc/init.d/cassandra
4. Add the new service to the list: sudo chkconfig --add cassandra
5. Now you can manage the service from the command line:

- sudo /etc/init.d/cassandra start
- sudo /etc/init.d/cassandra stop

6. Configure the service to be started automatically together with the VM: sudo chkconfig --level 2345 cassandra on

Windows

Installation

1. Download version 2.2.3 or higher from the Cassandra 2.2 stream.
2. Unpack the archive into a path without spaces.

Configuration

1. Go to the directory where you installed your Cassandra node.
2. Edit cassandra.yaml, using the following custom values:
   a. cluster_name: cluster name without spaces, for example GKC_Cassandra_Cluster
   b. seeds: <comma-separated list of fully qualified domain names (FQDN) or IP addresses of one or more Cassandra nodes>
   Note: This value must be the same for all nodes. Here are two examples:
      - 192.168.0.1,192.168.3
      - host1.mydomain.com, host2.mydomain.com
   c. storage_port: 7000 (default value)
   d. ssl_storage_port: 7001 (default value)
   e. listen_address: <current node host name>
   Note: This address is used for inter-node communication, so it must be
   f. available for use by other Cassandra nodes in your cluster.
   g. native_transport_port: 9042 (default value)
   h. rpc_address: <current node host name>
   Note: This address is used by Knowledge Center CMS to connect to
   i. Cassandra, so it must be available to all Knowledge Center CMS hosts.
   j. rpc_port: 9160 (default value)
   k. start_rpc: true
   l. endpoint_snitch: GossipingPropertyFileSnitch
3. Edit conf/cassandra-rackdc.properties.
4. Verify that the required communication ports are opened.
5. Start Cassandra.

Tuning Cassandra Configuration

Configuring cassandra-rackdc.properties

For a single data center, use the following as a guide:

```
dc=<Data Center name>
rack=<RACK ID>
```

Example:

```
dc=OperationalDC
rack=RAC1
```

> **Important**
>
> Genesys recommends that you use the same rack ID if you do not have a clear understanding of your servers' rack usage. For more information about cassandra-rackdc.properties, refer to http://docs.datastax.com/en/cassandra/2.2/cassandra/architecture/archsnitchGossipPF.html

Communication Ports

Cassandra use the following ports for external and internode communication. Note: Either or both of them may not work as expected unless you ensure that these ports are opened for communication between all servers that host Cassandra nodes.

Port	Default	Where to change the value
Cassandra Storage port	7000	storage_port in cassandra.yaml
Cassandra SSL Storage port	7001	ssl_storage_port in cassandra.yaml
Cassandra Thrift port	9160	rpc_port in cassandra.yaml (Knowledge Center CMS uses Thrift protocol to communicate to Cassandra)
Cassandra CQL port	9042	native_transport_port in cassandra.yaml

Working with Cassandra

Starting the Cassandra Cluster Nodes

Your Cassandra nodes must be started in a certain order:

1. Start the seed nodes.
2. Start the other non-seed nodes.

The seed node is one of the nodes specified in the seeds option.

Verifying Your Cassandra Cluster

After you have deployed your Cassandra Cluster, you may want to verify that all of the nodes can communicate with each other. To do this, execute the following command on any Database VM:

Linux

```
cd <Cassandra_installation_dir>/bin ./nodetool -h <hostname> status
```

Windows

```
cd <Cassandra_installation_dir >/bin nodetool -h <hostname> status
```

Sample output

This command should produce output that looks something like this:

```
Datacenter: DC1 ========================== Status=Up/Down |/
State=Normal/Leaving/Joining/Moving -- Address Load Tokens Owns Host ID
Rack UN 10.51.XX.XXX 106,36 KB 256 ? 380d02fb-da6c-4f6a-820e-14538bd24a39
RAC1 UN 10.51.XX.XXX 108,22 KB 256 ? 601f05ac-aa1d-417b-911f-22340ae62c38
RAC1 UN 10.51.XX.XXX 107,61 KB 256 ? 171a15cd-fa4d-410e-431b-51297af13e96
RAC1 Datacenter: DC2 ========================== Status=Up/Down |/
State=Normal/Leaving/Joining/Moving -- Address Load Tokens Owns Host ID
Rack UN 10.51.XX.XXX 104,06 KB 256 ? 48ad4d08-555b-4526-8fab-d7ad021b14af
RAC1 UN 10.51.XX.XXX 109,56 KB 256 ? 8ca0fb45-aef7-4f0a-ac4e-a324ceea90c9
RAC1 UN 10.51.XX.XXX 105,18 KB 256 ? 1c45e1fa-9f82-4bc4-a896-5575bad53808
RAC1
```

Upgrading Cassandra Nodes

You can upgrade your Cassandra version without interrupting service if:

- The version you are upgrading to is in the same stream (for example, from one 2.2.x version to another)
- You are not changing your database schema

Use the following steps for this task:

1. Stop the first Cassandra seed node.
2. Preserve your database storage.
3. Upgrade your Cassandra version, following the instructions in the Release Notes for the new version.
4. Be sure that your database storage is in the preserved state (the same set of files).
5. Start the first Cassandra seed node.
6. Execute steps 1 through 5 for the other seed nodes.
7. Execute steps 1 through 5 for the other non-seed nodes.
8. Verify that the Cassandra cluster is working, as shown above in Verifying Your Cassandra Cluster.

If your upgrade plans include changing your database schema or changing Cassandra versions between streams (for example, from 2.0 to 2.2), then you will have to interrupt service. Use the following steps for this task:

1. Stop all of your Cassandra nodes.
2. If your database schema has been changed since you installed the previous version, update the Cassandra database, following the instructions in the Release Notes for the new version.
3. Configure each node, following the instructions in the Release Notes for the new version.
4. Start the Cassandra seed nodes.
5. Start the other nodes.
6. Verify that the Cassandra cluster is working, as shown above in Verifying Your Cassandra Cluster.

Maintenance

Because Cassandra is a critical component of Knowledge Center CMS, it is essential to keep track of its health. The Datastax documentation provides some really good information about how to do this at http://docs.datastax.com/en/cassandra/2.0/cassandra/tools/toolsNodetool_r.html.
Genesys recommends that you use the nodetool utility that is bundled with your Cassandra installation package and that you make a habit of using the following nodetool commands to monitor the state of your Cassandra cluster.

ring

Displays node status and information about the cluster, as determined by the node being queried. This can give you an idea of the load balance and whether any nodes are down. If your cluster is not properly configured, different nodes may show a different cluster; this is a good way to check that every node views the cluster the same way.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> ring
```

status

Displays cluster information.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> status
```

compactionstats

Displays compaction statistics.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> compactionstats
```

getcompactionthroughput \ setcompactionthhroughput

Displays the compaction throughput on the selected Cassandra instance. By default it is 32 MB/s. You can increase this parameter if you observe permanent growth of database size after the TTL and grace periods are passed. Note that increasing compaction throughput will affect memory and CPU consumption. Because of this, you need make sure to have sufficient hardware to support the rate that you have selected.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> getcompactionthroughput
```

To increase compaction throughput to 64 MB/s, for example, use the following command:

```
nodetool -h <HOST_NAME> -p <JMX_PORT> setcompactionthroughput 64
```

Recovery

Depending on the replication factor and consistency levels of a Cassandra cluster configuration, the Knowledge Center CMS can handle the failure of one or more Cassandra nodes in the data center without any special recovery procedures and without interrupting service or losing functionality. When the failed node is back up, the Knowledge Center CMS automatically reconnects to it. If an eligible number of nodes have failed, you should just restart them.

If too many of the Cassandra nodes in your cluster have failed or stopped, you will lose functionality. To ensure a successful recover from failure of multiple nodes, Genesys recommends that you:

1. Stop every node, one at a time, with at least two minutes between operations.
2. Restart the nodes one at a time, with at least two minutes between operations.

Using CMS with Microsoft SQL Server

Prerequisites

- Create new database in Microsoft SQL Server
- Create user account to access the database

Configuring CMS

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**
2. Select the application defined for the Knowledge Center Cluster and click **Edit**
3. From the **Options** tab in the **cms.cluster** section, set the following options:
   a. set option **type** to value **mssql**

> **Important**
> if you are using version 8.5.300.xx, please type **jdbc** instead of **mssql**

   b. set **dbConnectionUrl** to JDBC connection string for connection to MS SQL Server following the format:
   `jdbc:jtds:sqlserver://<host_of_MSSQL_Server>:<port_of_MSSQL_Server, 1433 by default>;databaseName=<CMS_DB_name>`
   c. **dbUsername** - set to the username that needs to be used to login to MS SQL Server
   d. **dbPassword** - set to the password for db account
   e. **dbDriverClass** - set to **net.sourceforge.jtds.jdbc.Driver**

> **Important**
> Description of options in cms.cluster section can be found in Configuration Options.

If you are installing Knowledge Center CMS 8.5.302.xx or earlier, you need to download and place **Microsoft SQL Server JDBC driver** on every CMS host:

1. Open in your browser https://sourceforge.net/projects/jtds/files/jtds/1.2.7/

2. Download **jtds-1.2.7-dist.zip** (Oracle account is needed)
3. Unpack downloaded archive
4. Place **jtds-1.2.7.jar** into **<CMS_installation_folder>/lib/ext**

> **Important**
>
> The driver must be added to the installation folder of every CMS node in your deployment.
> Starting from the 8.5.303.xx release of the product, drivers are embedded into the CMS's IP.

Using CMS with Oracle

> **Important**
> Oracle 11g supported from version 8.5.302.xx of the product.

Prerequisites

- Create new database in Oracle 11g
- Create user account to access the database

Configuring CMS

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**
2. Select the application defined for the Knowledge Center Cluster and click **Edit**
3. From the **Options** tab in the **cms.cluster** section, set the following options:
   a. set option **type** to value **oracle**
   b. set **dbConnectionUrl** to JDBC connection string for connection to Oracle following the format:
   `jdbc:oracle:thin:@<host_with_Oracle>:<port_of_Oracle, 1521 by default>:<CMS_DB_SID>`
   c. **dbUsername** - set to the username that needs to be used to login to Oracle
   d. **dbPassword** - set to the password for db account
   e. dbDriverClass - set to **oracle.jdbc.driver.OracleDriver**

> **Important**
> Description of options in cms.cluster section can be found in Configuration Options.

If you are installing Knowledge Center CMS 8.5.302.xx, you need to download and place **Oracle JDBC driver** on every CMS host:

1. Open in your browser http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html
2. Accept the license

3. Download **ojdbc6.jar** (Oracle account is needed)
4. Place **ojdbc6.jar** into **<CMS_installation_folder>/lib/ext**

> **Important**
>
> The driver must be added to the installation folder of every CMS node in your deployment.
> Starting from the 8.5.303.xx release of the product, drivers are embedded into the CMS's IP.

Using CMS with Cassandra

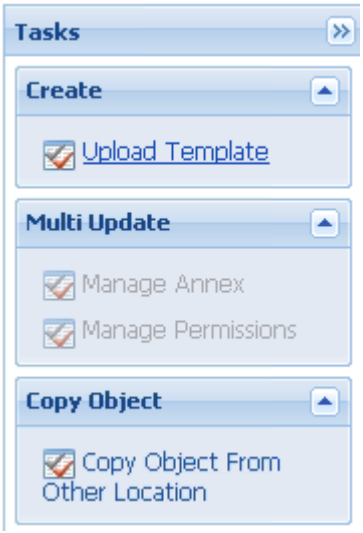
> **Important**
>
> To ensure strong consistency of the data (no matter the configuration of the underlying storage) it was decided to deprecate Cassandra 2.2.x support in 8.5.3 release of the product. Going forward, **support of Cassandra will be discontinued in 9.0 release of the product**.

> **Important**
>
> This step is required in case Cassandra 2.2.x is planned to be used as the storage. If you need guidance deploying Cassandra cluster, please refer to Deploying Cassandra Cluster. Detailed Cassandra documentation can be found on the Cassandra website. If you are using another data source as the CMS Repository, go directly to Install the CMS.

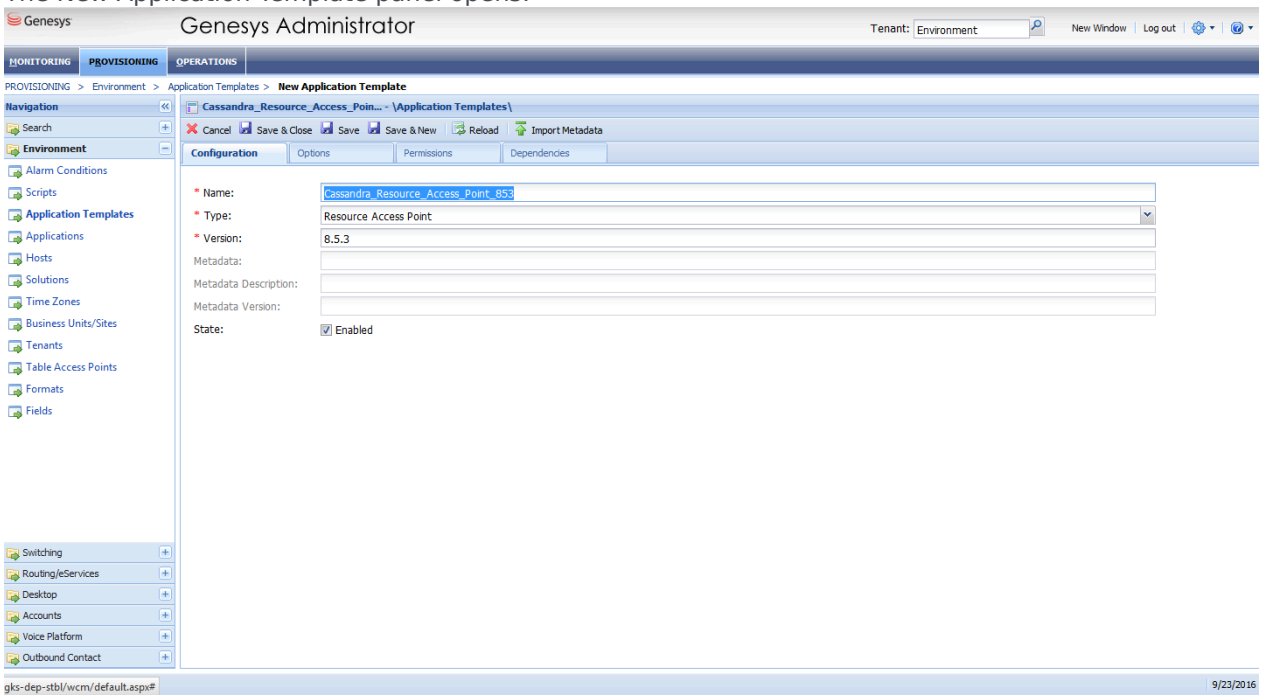
Importing the Cassandra Resource Access Point Template

Start

1. Open Genesys Administrator and navigate to **Provisioning** > **Environment** > **Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.



3. Click **Add** and choose application template (APD) file to import window, click **Add**.
4. Browse to the Cassandra_Resource_Access_Point_853.apd file. Click **Open**.
   The New Application Template panel opens:



5. Click **Save & Close**.

End

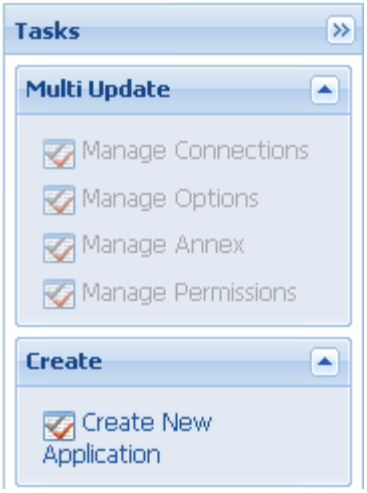
Creating the Cassandra Resource Access Point Application

Prerequisites

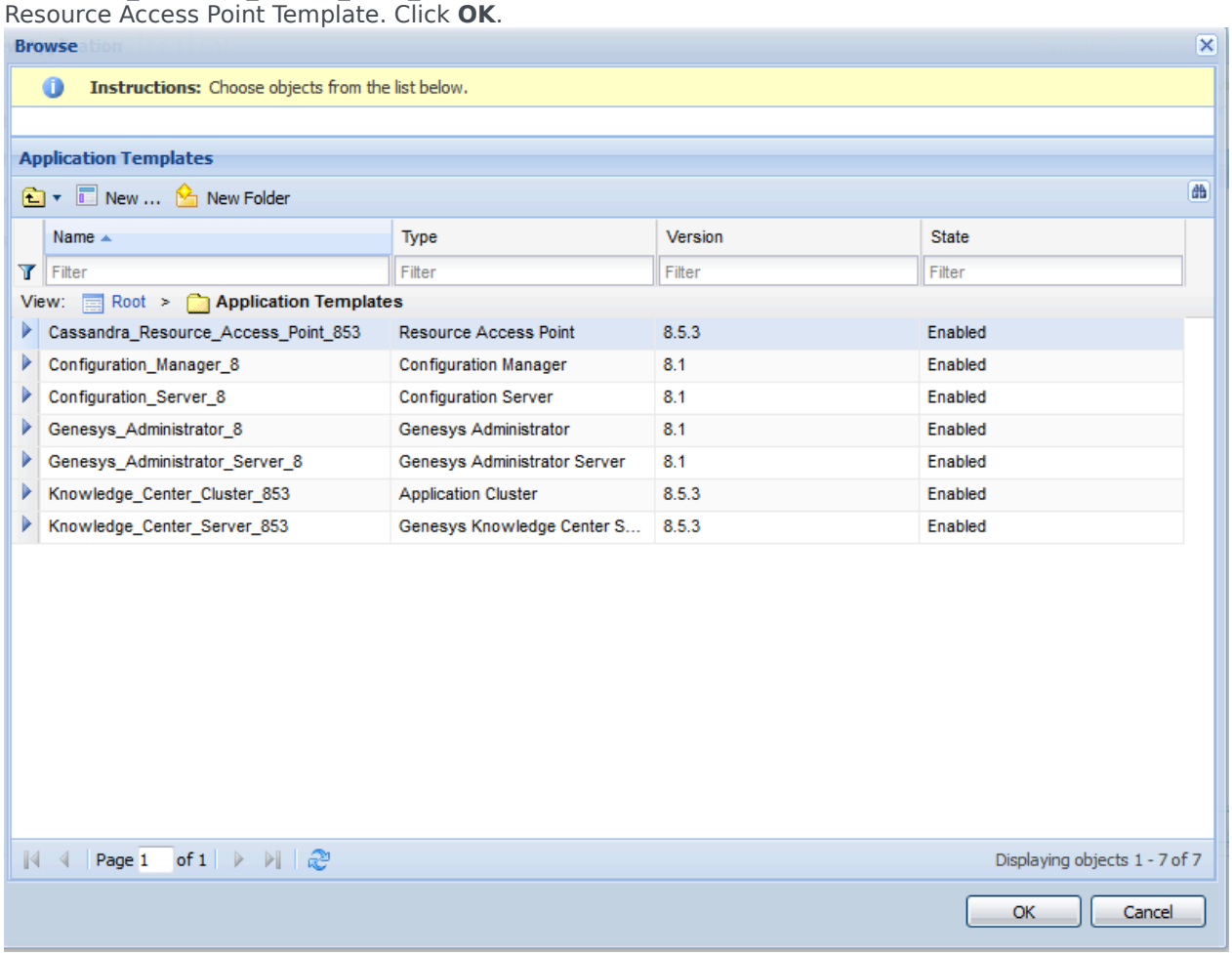
- You completed Importing the Cassandra Resource Access Point Template.

Start

1. Open Genesys Administrator and navigate to **Provisioning** > **Environment** > **Applications**.
2. In the **Tasks** panel, click **Create New Application**.

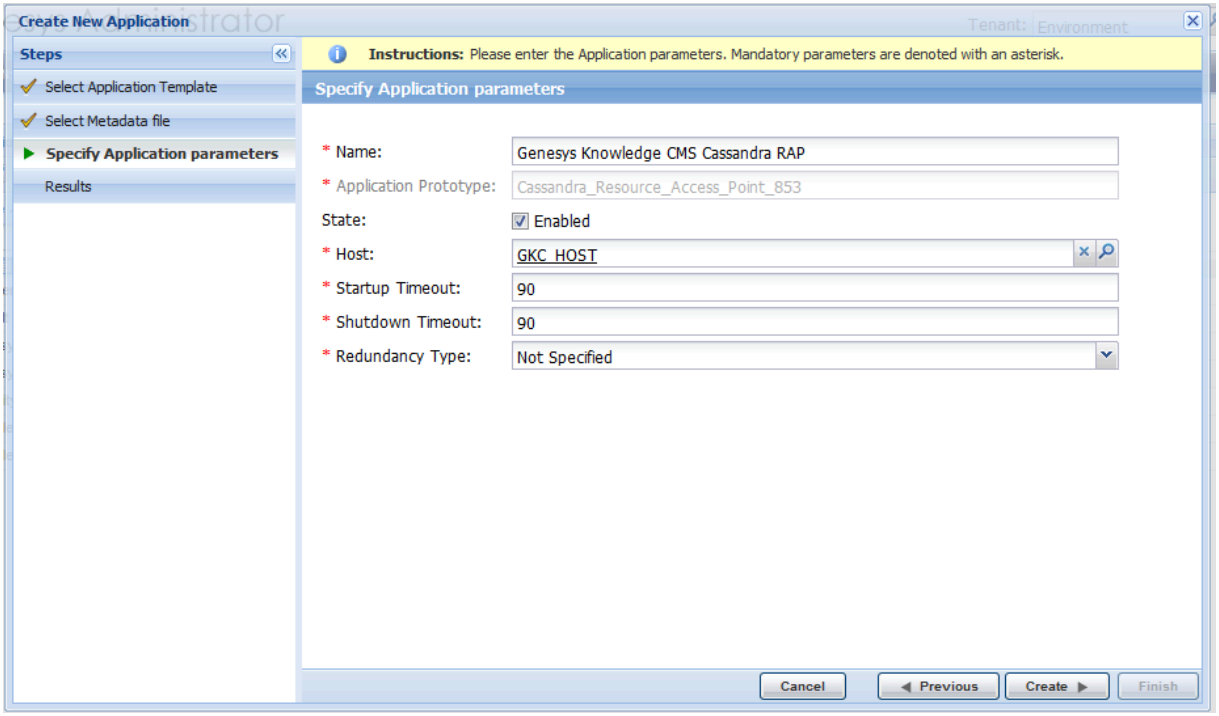


3. In the **Select Application Template** panel, click **Browse for Template** and select the Cassandra_Resource_Access_Point_853 template that you imported in Importing the Cassandra Resource Access Point Template. Click **OK**.

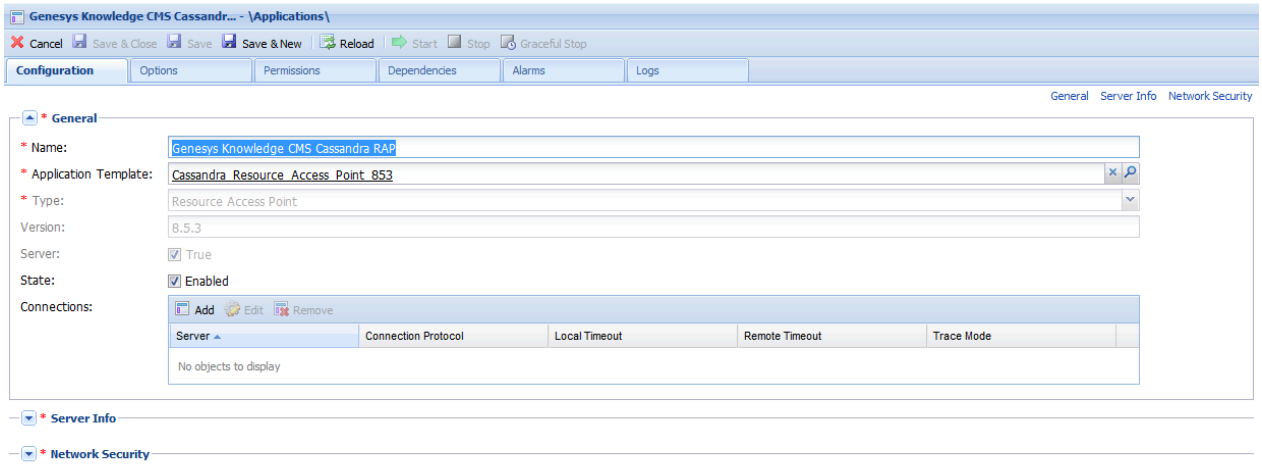


4. The template is added to the **Select Application Template** panel. Click **Next**.

5. In the **Select Metadata** file panel,
   a. click **Browse**
   b. click **Add**
   c. select the Cassandra_Resource_Access_Point_853.xml file
   d. Click **Open**
6. The metadata file is added to the Select Metadata file panel. Click **Next**.
7. In **Specify Application** parameters:
   a. Enter a name for your application. For instance, "Genesys Knowledge CMS Cassandra RAP"
   b. Make sure that **State** is enabled
   c. Select the **Host** on which the Access Point will reside
   d. Click **Create**



8. The **Results** panel opens.
9. Enable Opens the Application details form after clicking 'Finish' and click **Finish**. The Cassandra Resource Access Point application form opens and you can start configuring its properties.



End

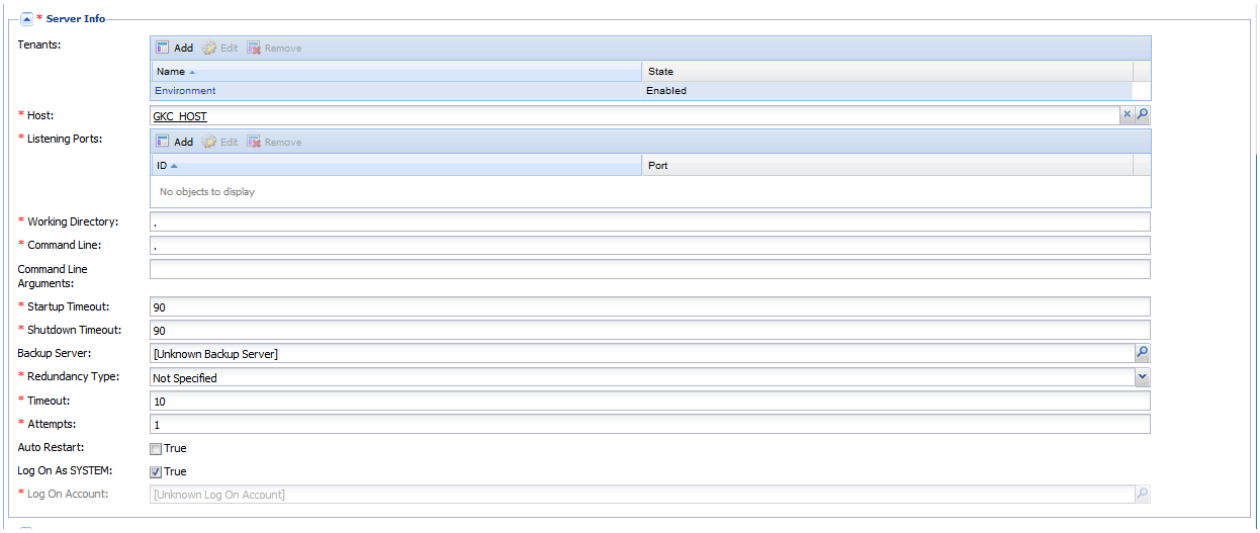
Configuring the Cassandra Resource Access Point Application

Prerequisites

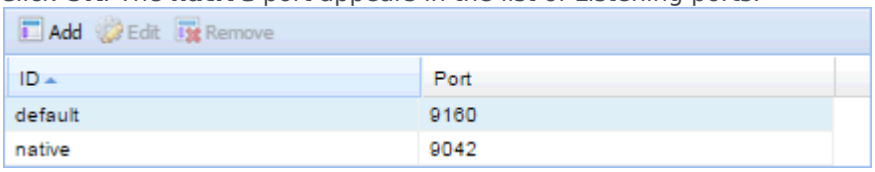
- You completed Creating the Cassandra Resource Access Point Application.

Start

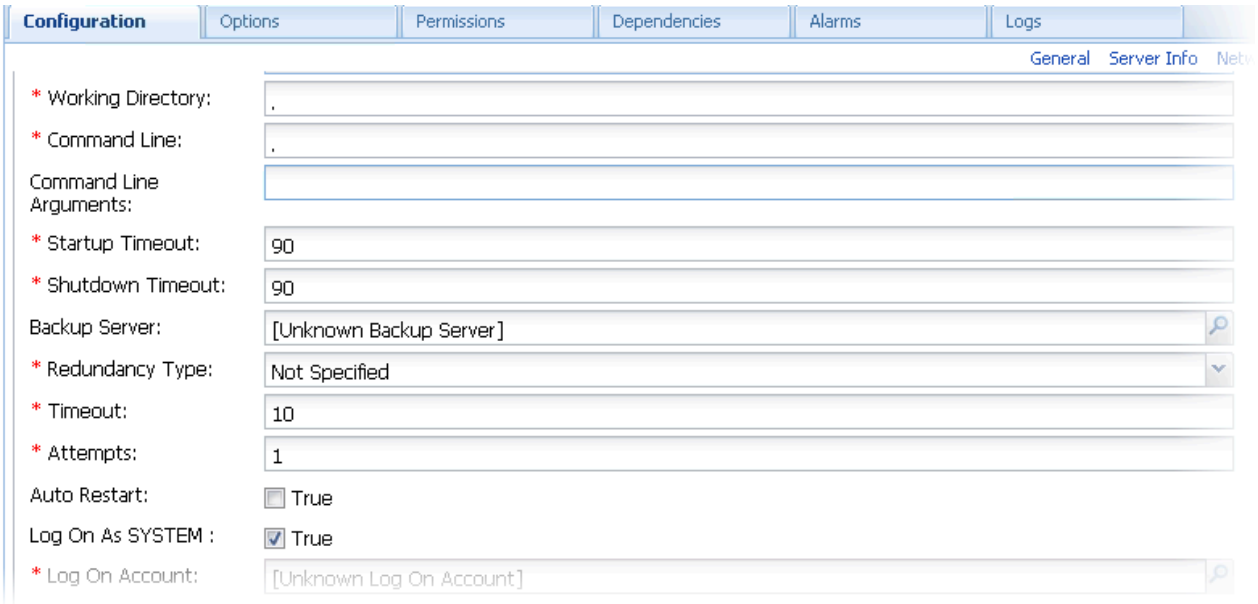
1. If your Cassandra Resource Access Point application form is not open in Genesys Administrator, navigate to **Provisioning** > **Environment** > **Applications**. Select the application defined for the Cassandra Resource Access Point and click **Edit**.
2. Expand the **Server Info** pane.
3. In the **Tenant** section, click **Add** and select your tenant. For instance, Environment. Click **OK**. (Tenant should be same as for previously created Genesys Application cluster <link to chapter about installing Cluster>)
4. If your **Host** is not defined, click the lookup icon to browse to the hostname of your application, which should point to the host where you plan to locate your Cassandra node.



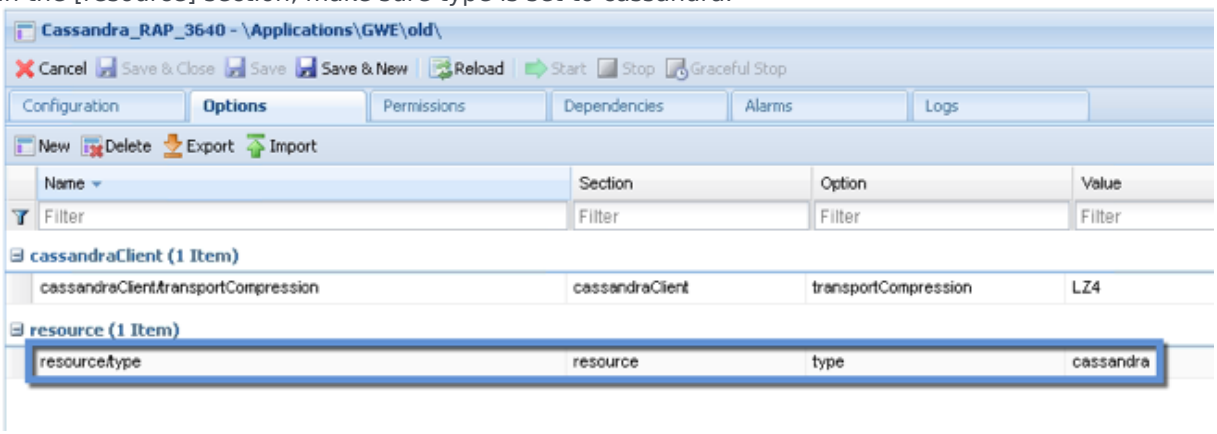
5. In the **Listening Ports** section, create the default port by clicking **Add**. The Port Info dialog opens.
   a. Enter the **Port**. For instance, 9160 (use value of rpc_port of your Cassandra).
   b. Click **OK**. The **default** port appears in the list of **Listening** ports.
6. Click **Add** again. The **Port Info** dialog opens.
   a. In the **ID** field, enter **native**.
   b. Enter the **Port**. For instance, 9042 (use value of native_transport_port of your Cassandra).
   c. Click **OK**. The **native** port appears in the list of Listening ports.



7. Ensure the **Working Directory** and **Command Line** fields contain "." (period).



8. Click **Save**.
9. The **Confirmation** dialog for changing the application's port opens. Click **Yes**.
10. Select the **Options** tab.
    - In the [resource] section, make sure type is set to cassandra.



11. Click **Save & Close**. If the **Confirmation** dialog opens, click **Yes**.

End

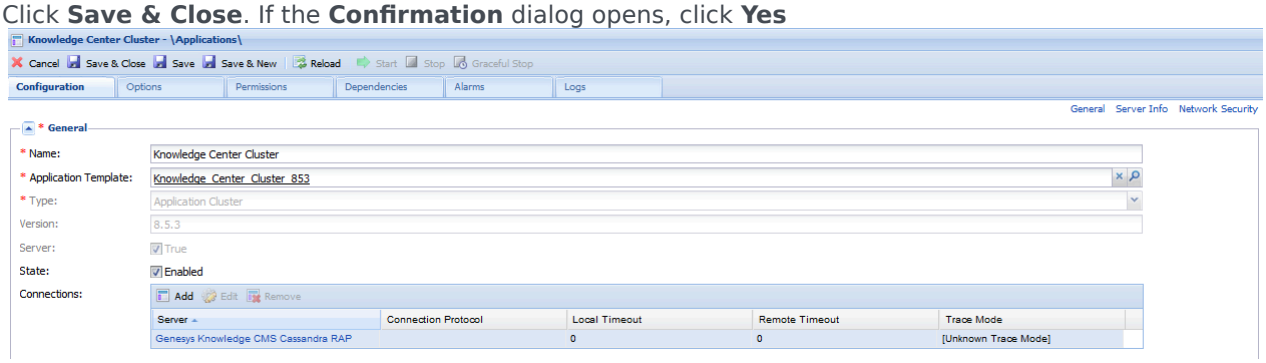
Configuring the Knowledge Center Cluster for Use with Cassandra

Prerequisites

- You completed Configuring the Cassandra Resource Access Point Application.

Start

1. Navigate to **Provisioning** > **Environment** > **Applications**. Select the application defined for the Knowledge Center Cluster and click **Edit**.
2. In the **Connections** section of the **Configuration** tab, click **Add**. The Browse for applications panel opens. Select a Genesys application defined as a Cassandra Resource Access Point, then click **OK**.
3. Select added connection to application, click **Edit** and ensure that the default connection port selected as ID
4. (Optional) Open Options tab. In section cassandra-keyspace set valid values to use your Cassandra node.
5. Click **Save & Close**. If the **Confirmation** dialog opens, click **Yes**



End

Configuring CMS to Work with PostgreSQL

> **Important**
> PostgreSQL is supported starting from 8.5.304.xx release of the Genesys Knowledge CMS Please use the latest stable version of PostgreSQL.

Prerequisites

- Create new database in PostgreSQL
- Create user account to access the database

Configuring CMS

1. Configure database properties in the Genesys Knowledge Center Cluster application.
2. Open Genesys Administrator and navigate to **Provisioning > Environment > Applications**.
3. Select the application defined for the Knowledge Center Cluster and click **Edit**.
4. From the **Options** tab in the **`cms.cluster`** section, set the following options:
   - for option **type** set **postgre** value.
   - **dbConnectionUrl** - to JDBC connection string for connection to PostreSQL Server following the format:

     ```
     jdbc:postgresql://<host_of_PostgreSQL>:<port_of_PostgreSQL>/<CMS_DB_name>
     ```

   - **dbUsername** - set to the username that needs to be used to login to PostgreSQL.
   - **dbPassword** - set to the password for db account.

   Note: Description of options in `cms.cluster` section can be found in Configuration Options.

Installing and Using the Administrator Plugin

> **Important**
>
> Genesys Knowledge Center Plugin for Genesys Administrator has been discontinued as of release 8.5.303.xx of the product. All functionality of the plugin has been migrated into the Knowledge Center CMS which now permits content authoring as well as knowledge base management. For more information, please see Installing the Knowledge Center CMS.
> Please skip this chapter if you are using 8.5.303.xx or higher installation version.

Installing the Knowledge Center Plugin for Administrator

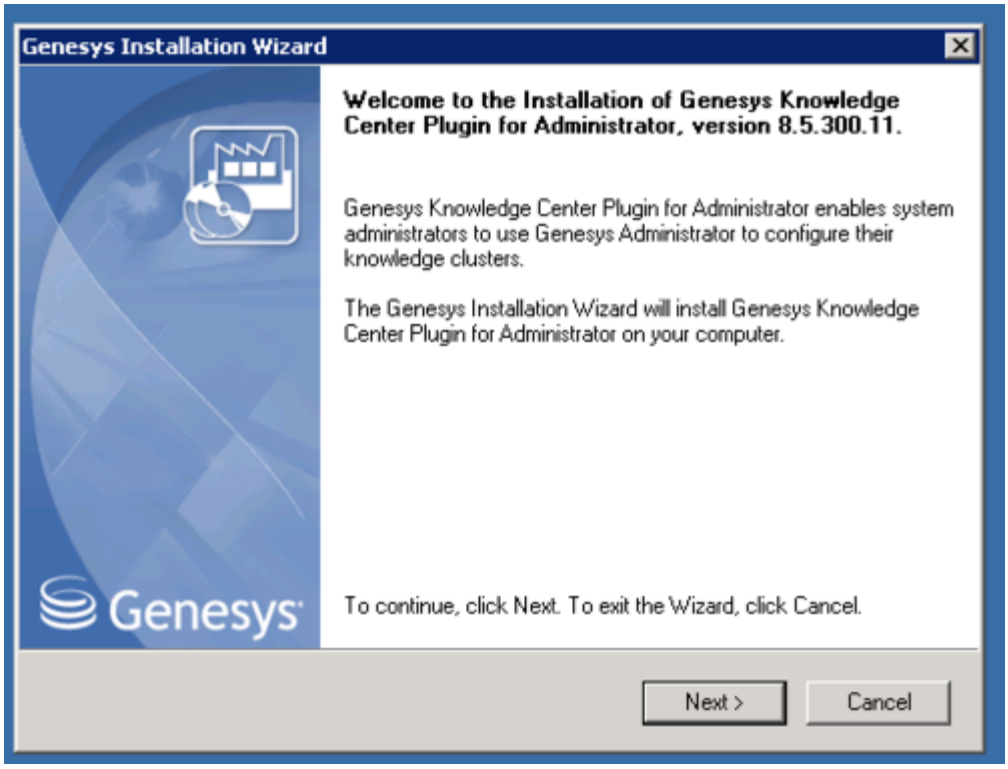
Prerequisites

- Genesys Administrator must have been installed, but should be stopped before installing the plugin
- If the Administrator Plugin was previously installed on the current host, manually remove the previous version from the */plug-ins* folder in the Genesys Administrator installation directory

Windows Installation Procedure

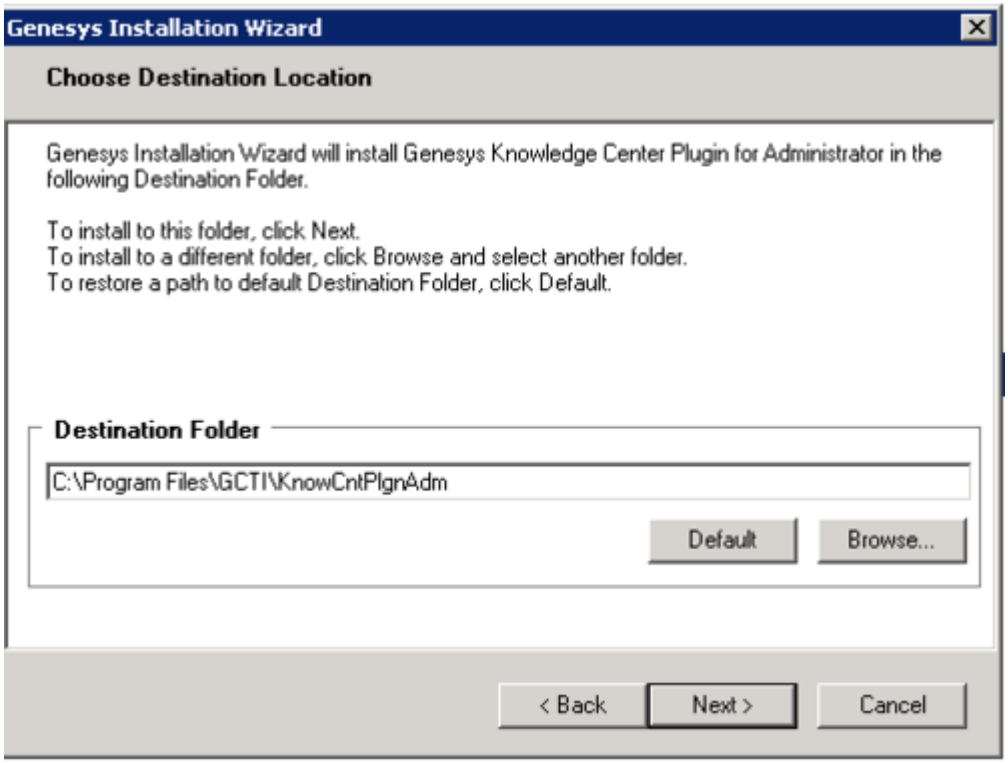
Start

1. In your installation package, locate and double-click the **setup.exe** file. Install Shield opens its welcome screen.



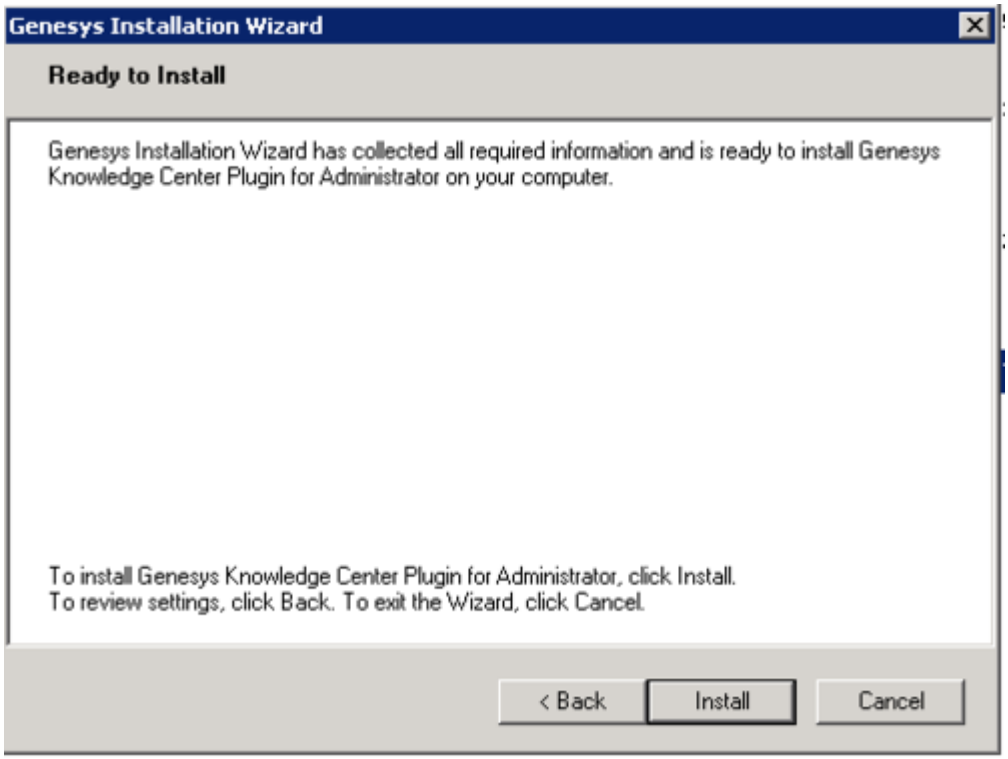
Knowledge Center Administrator Plugin Install Shield Window

2. Click **Next**. The **Choose Destination Location** screen appears.



Knowledge Center Administrator Plugin Destination Window

3. Under **Destination Folder**, keep the default value or browse to the desired installation location. Click **Next**.
4. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation. When it has finished, the **Installation Complete** screen appears.



Knowledge Center Administrator Plugin Installation Complete

5. Click **Finish** to complete your installation.
6. Inspect the directory tree of your system to make sure that the files have been installed in the location that you intended.
7. *gax-plugin-knowledge.jar* should be added as a Genesys Administrator plugin.
8. Restart Genesys Administrator.

End

Linux Installation Procedure

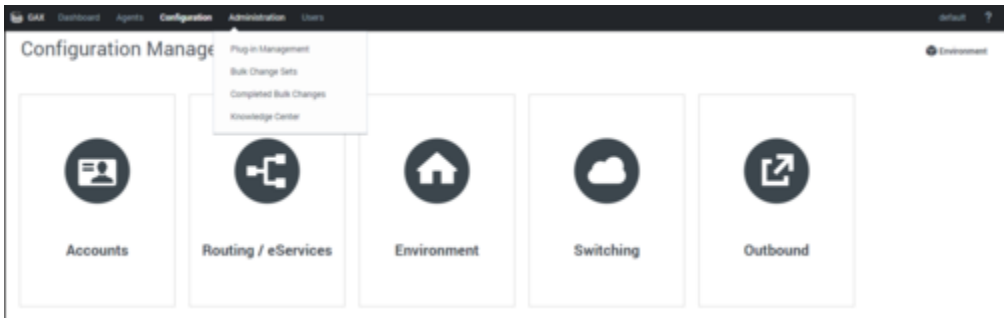
Start

1. Open a terminal in the Genesys Knowledge Center Plugin for Administrator IP, and run the *install.sh* file. The Genesys Installation starts.
2. Enter the full path to the GAX installation directory.
3. Enter the full path to your installation directory for the plugin and confirm it.

4. If the installation is successful, the console displays the following message: Installation of Genesys Knowledge Center Plugin for Administrator, version 8.5.x has completed successfully.
5. *gax-plugin-knowledge.jar* should be added as a Genesys Administrator plugin.
6. Restart Genesys Administrator.

End

A **Knowledge Center** item should appear under the Administration menu.



Knowledge Center in Administrator Menu

Providing access to Knowledge Center Plugin for Administrator

> ### Important
> Users must have the next privilege in order to use the Administrator plugin.
>
> - Allows agent to manage knowledge bases (Knowledge. ADMINISTER) — Enables access to the Knowledge Center Plugin for Administrator tab in Genesys Administrator (in "Genesys Administrator Extensions - Genesys Knowledge Center Plug-in" privileges)
> - To save a created configuration the user should at least belong to the Administrators Access Group

To configure the appropriate role for an agent:

Start

1. Go to **Provisioning** > **Environment** > **Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.
3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Choose the application template (APD) file from the import window and click **Add**.
5. Browse to the Knowledge_Center_GAX_Plugin_853.apd file available in the templates directory of your installation CD. The **New Application Template** panel opens.

6. Click **Import Metadata**.
7. Click **Add** and select the Knowledge_Center_GAX_Plugin_853.xml file.
8. Click **Open**.
9. Information from the metadata file will be added to the template and the appropriate privilege will be added into the framework.
10. Save and Close.
11. Go to **Provisioning** > **Accounts** > **Roles**.
12. In the taskbar click **New** to create a new object.
13. Set the name of the role in the **General** section.
14. Go to the **Role Privileges** tab, and select the set of roles for Genesys Knowledge Center.
15. Open the **Genesys Administrator privileges** list and select the Genesys Knowledge Center Plug-In Privileges section.
16. Set the appropriate privileges as allowed.
17. Go back to the **Configuration** tab.
18. Add the appropriate **Agent** to the **Members** section by clicking the **Add** button.
19. Save and Close.

End

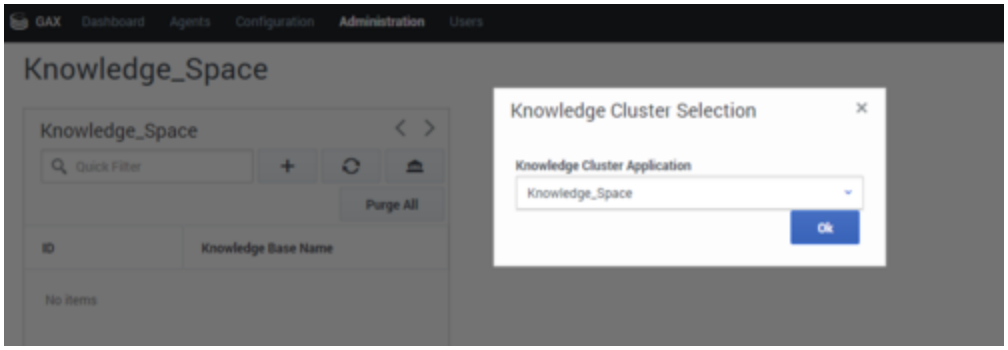
Managing Knowledge Bases

In order to use Knowledge Center Server you need to create at least one knowledge base in the Knowledge Center Cluster application, using the Knowledge Center Plugin for Administrator. This section describes the structure and specific options you need in order to create an index for this knowledge base in Knowledge Center Server.

Selecting the Knowledge Center Cluster Application

Start

1. Log in to Genesys Administrator and navigate to the **Administration > Knowledge Center** menu item.
2. Using the  button, open the menu for **Select Knowledge Cluster**. Select the appropriate cluster from the drop-down and click the **Ok** button. A list of the knowledge bases that have been defined for this cluster will be displayed.



Selecting a Knowledge Cluster

End

Creating a Knowledge Base

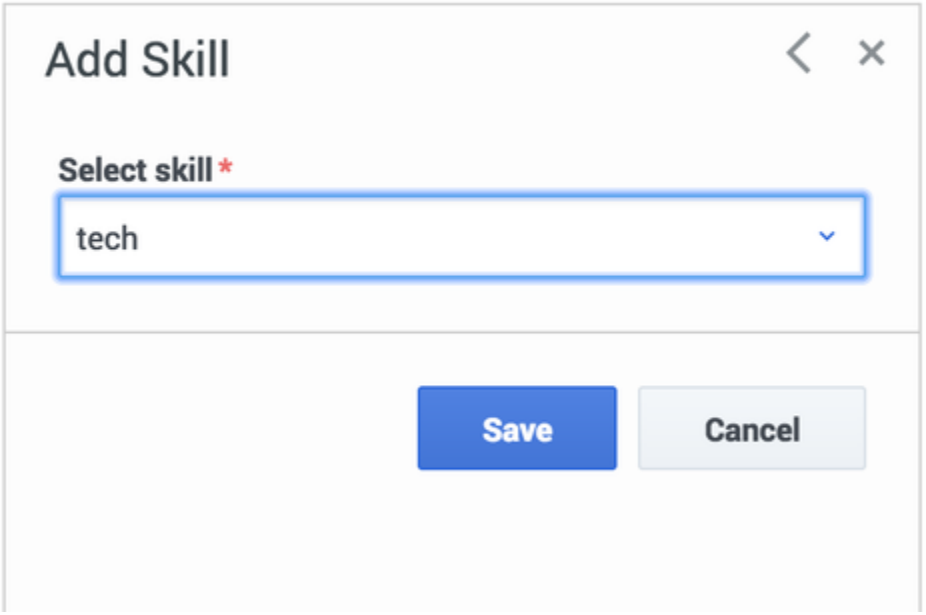
Start

1. Click the **+** button. A panel with the main knowledge base parameters will be displayed. Fill in the following fields:
   - **ID**—The ID should only contain numbers, lower-case Latin letters, and underscores, with a maximum length of 50 characters. The limitation to lower-case letters is because ElasticSearch is case-insensitive and will therefore render all names as lower-case.
   - **Name**—Maximum length is 50 characters
   - **Description**—optional
   - Select the type of the content to be stored in knowledge base - Content type:
     - FAQ - Frequently asked questions
     - Articles - rich-text documents
   - Select the allowed feedback type for the knowledge base - Feedback types support:
     - Voting (relevance feedback for search query) and Rating (5-star rating of the document content)
     - only voting
     - only raiting
     - none (all feedback capabilities are disabled)
   - Select the first knowledge base language.
   - Make the knowledge base public or private. (If the knowledge base is made public, it will be visible to all users, whether or not they are authorized.)

> **Important**
>
> Later for private knowledge bases you can specify whether the knowledge base should be available to all of your agents or only to the agents that have one of the specified skills. In the case where you have specified several skills for the knowledge base, the agent needs to have at least one of them to access

> the knowledge base. Skill level does not influence ability to access the knowledge base.

- Make the knowledge base active or inactive. If you un-check **Knowledge Base is active** neither your customers nor your agents will be able to search for information in that knowledge base. Authors, agents with the privilege **Knowledge.Author** and administrators with the privilege **Knowledge.Admin**, can still use the base to prepare content stored in it.
- You can make your private knowledge bases (when **Knowledge base is public** is not check-marked) available to only a subset of your agents by selecting **Skill-based access to knowledge base**. If you choose to make the knowledge base accessible to an agent with specific skills, you will need to select the skills that will grant an agent access to the knowledge base.



Making a Knowledge Base Accessible by Skill

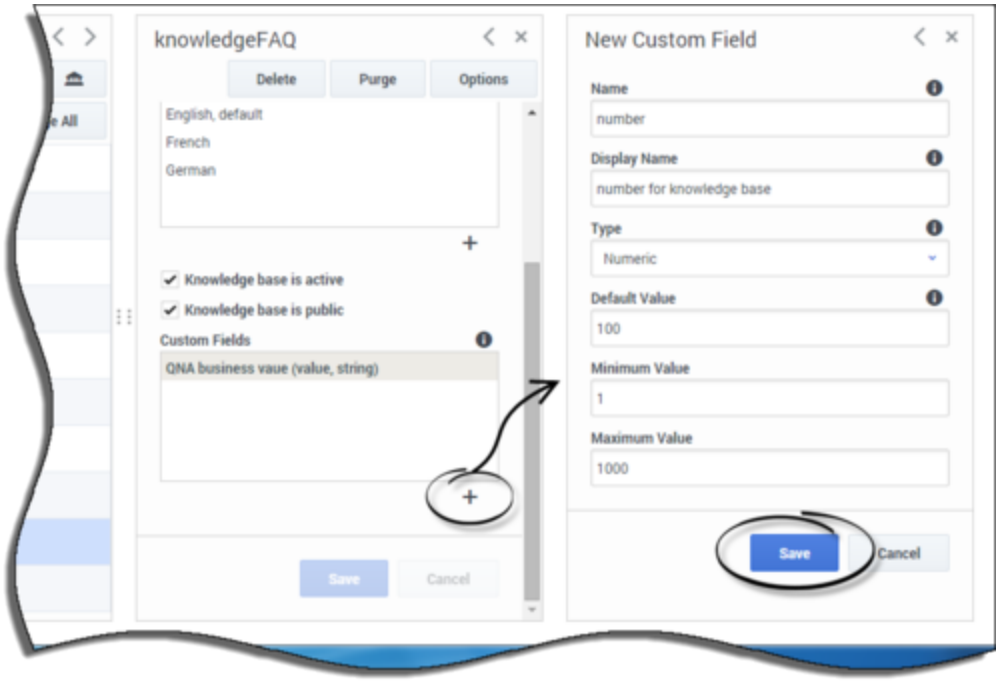
2. Click **Save**. The knowledge base will be created.

End

Creating Custom Fields

Start

1. Click the **+** sign under the **Custom Fields** section. The **New Custom Field** panel will be displayed.



2. To define a custom field, fill in the following information:

- **Name**—Should consist only of numbers, Latin letters and underscores, with a maximum length of 50 characters.
- **Display name**
- Select the type of field
- For **String** fields define:
  - Default value (optional)
  - If the field can be left empty, set the check box to **Allow empty**
- For **Numeric** fields define:
  - Default value (optional)
  - Minimum value (optional)
  - Maximum value (optional)
- For **DateTime** fields define:
  - Default value (optional; format should be yyyy-MM-dd HH:mm:ss)
- For Boolean
  - Default value (optional)
- For List of values
  - List of the allowed values

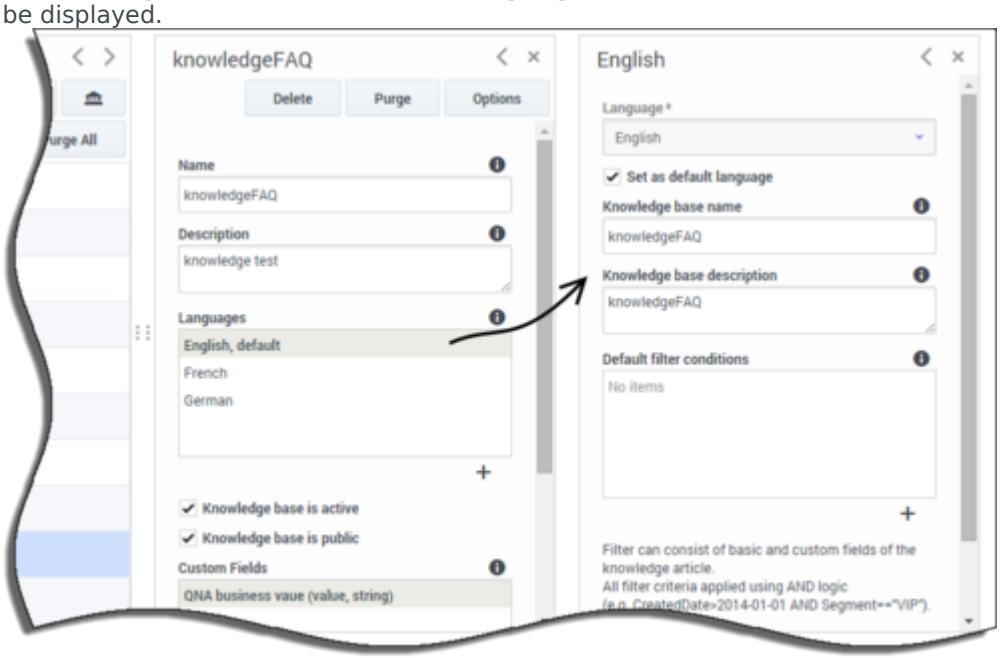
    - Default value (one from the list)
  - Visibility of the custom field: whether it visible for agent and customer or just agents.
- Click **Save** to save your changes.

End

Adding Language-specific Information

Start

1. Click the **English, default** row in the **Languages** section. A panel with language-specific settings will be displayed.



2. You can define the following parameters in this section:

   - A localized knowledge base name
   - A localized knowledge base description
   - Whether or not the selected language is the default

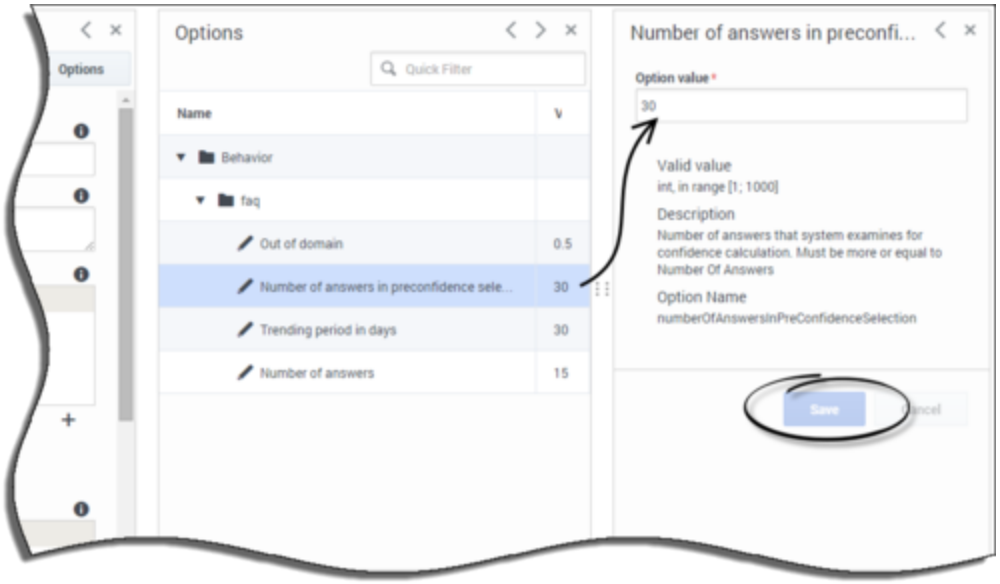
- Click the **Save** button

End

Editing Knowledge Base Options

Start

1. To edit the options for a particular knowledge base, click the **Options** button and then click the appropriate option to edit its value. The options are initialized with their default values.



2. Enter the new option value and click the **Save** button.

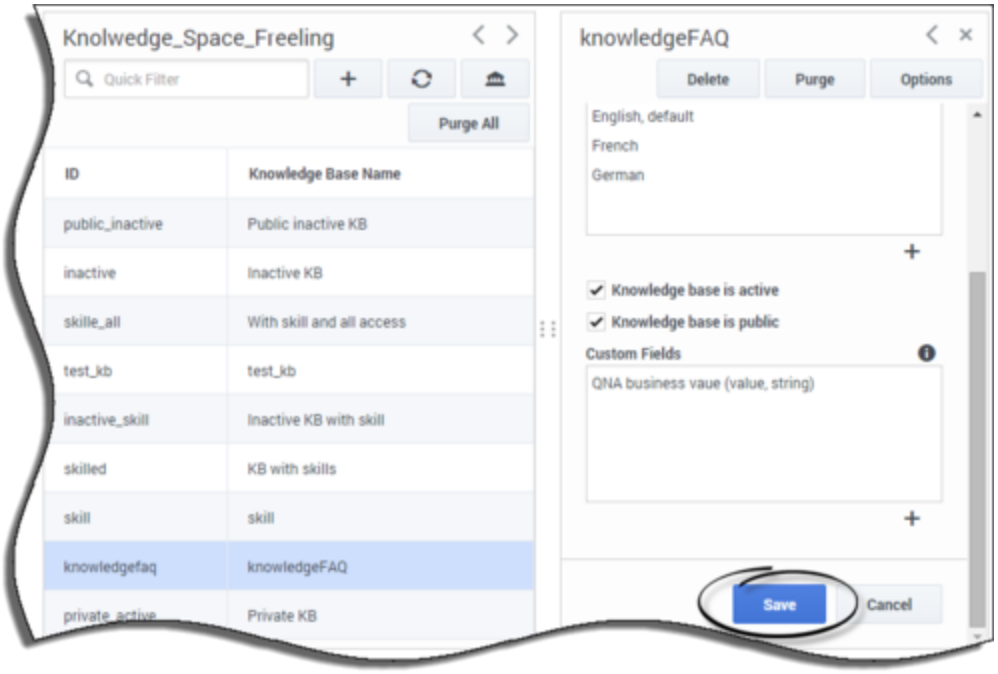
End

> ### Important
> It is not recommended to set the out-of-domain value higher than 0.75 as it represents an exact match of the question with no feedback accumulated for the query. The optional setting is 0.5 (default value).

Editing a Knowledge Base Definition

Start

1. Select a knowledge base from the list.



2. Edit the knowledge base definition and click the **Save** button.

End

Deleting a Knowledge Base Definition

Start

1. Select a knowledge base from the list.
2. Press the **Delete** button and confirm the action.

End

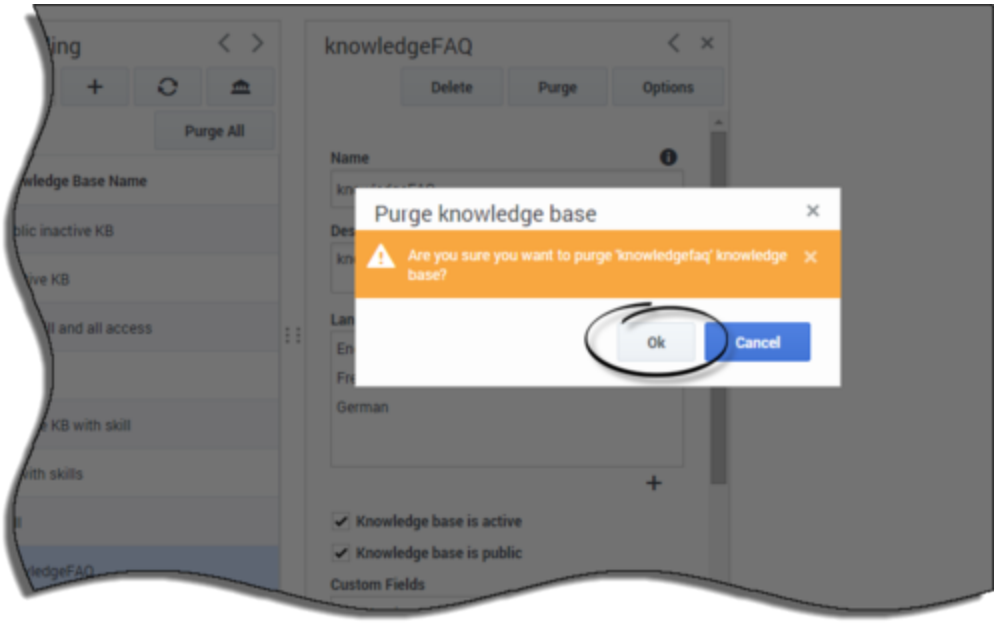
Purging Knowledge Bases

Prerequisites

- The Administrator user must have **Knowledge.ADMINISTER** privileges
- You must create and select a Knowledge Center Cluster application

Start

1. To purge a particular knowledge base, select it from the list, press the **Purge** button, and confirm the action.



2. To purge all knowledge bases, use the **Purge All** button.

End

Installing the Pulse Plugin

The Genesys Knowledge Center Plugin for Pulse provides access to Knowledge Center Server statistics such as KPI, user activity, trending topics, like and dislike trends, and activity types.

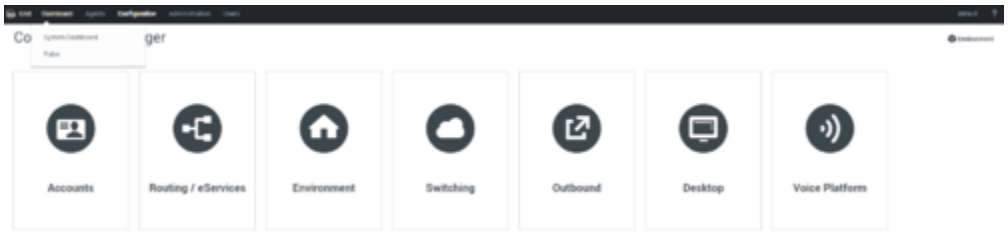
Install Genesys Knowledge Center Plugin for Pulse

Components required for Pulse plugin come pre-integrated into every deployment of Genesys Knowledge Center Server. So you do not need any additional steps to install them, please proceed directly to the configuration.

Configure Genesys Knowledge Center Plugin for Pulse

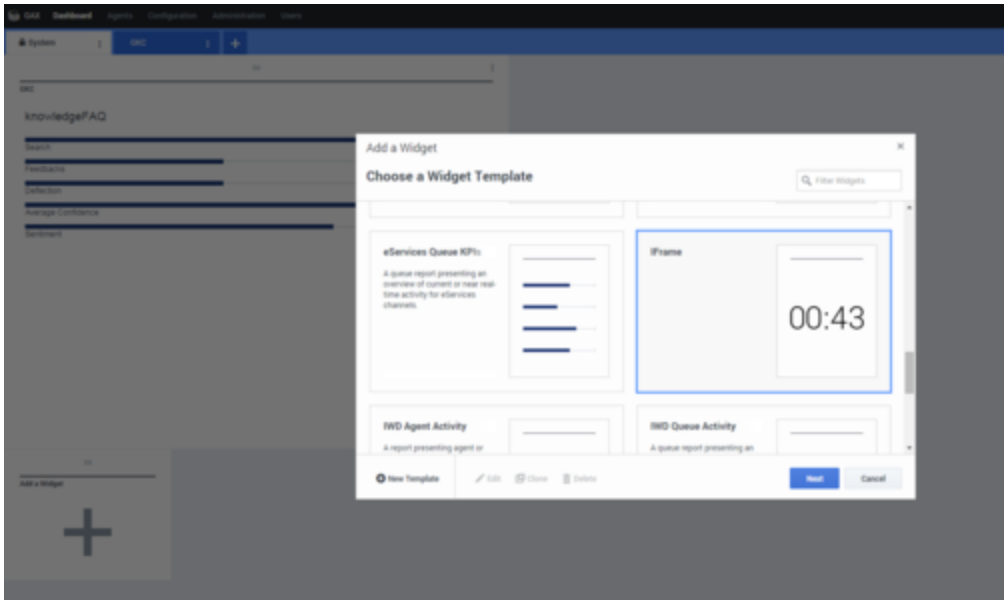
Start

1. Log into Genesys Administrator.
2. Go to **Dashboard > Pulse**.



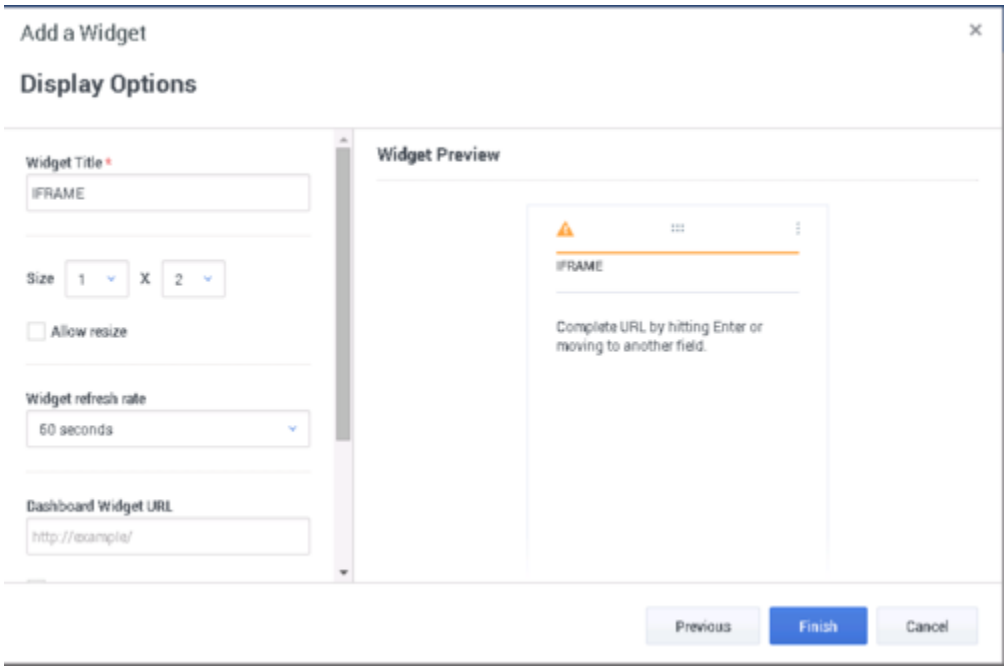
Selecting the Pulse Dashboard options in Genesys Administrator

3. Click **Add a Widget**.
4. Select the **IFrame** widget type.



Adding a Pulse iFrame widget

5. Set the name of the widget.



Setting the Pulse widget options

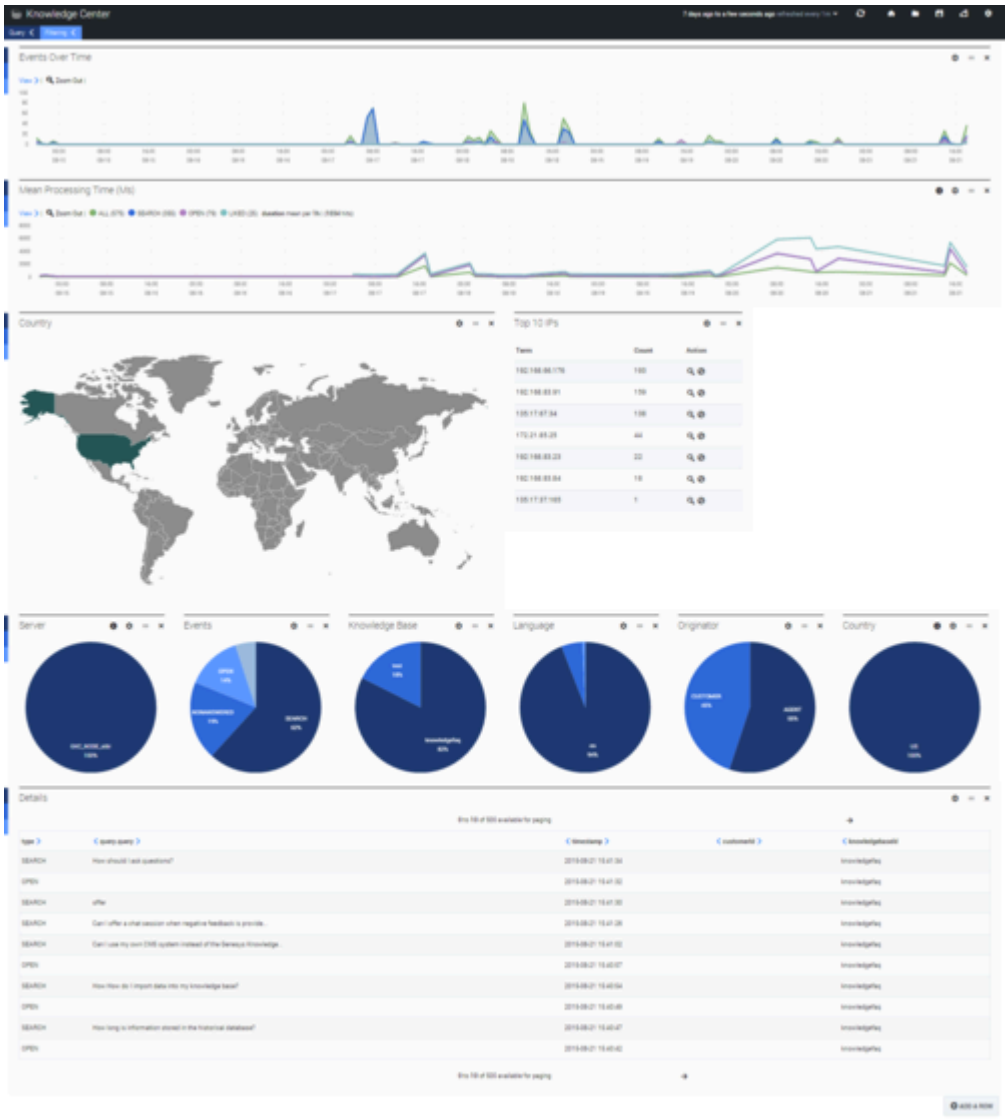
6. Set the widget URL to: http://<*host*>:<*es_port*>/_plugin/gkc-kpi/?kbId=<knowledge_base_id>(=<chosen language>&tenantId=<tenatId>&timeframe=<timeframe> (see Knowledge Center Pulse Plugin Configuration Options for more information about parameters).
7. Set the Maximized widget URL. You can set it to the Default Dashboard (http://<*host*>:<*es_port*>/_plugin/gkc-dashboard/#/dashboard/file/default.json) or the Performance

Dashboard (http://<*host*>:<*es_port*>/_plugin/gkc-dashboard/#/dashboard/file/performance.json).

8. Click **Finish**.


Knowledge Center
Daily Trend
0.37% (ALL)
0.7% (SEARCH)
0.45% (OPEN)
0% (USE)
Events Over Time
Hourly Trend
0.37% (ALL)
0.7% (SEARCH)
0.45% (OPEN)
0% (USE)
Mean Confidence
Activity Heatmap
SEARCH
143
OPEN
221
NO ANSWER
18
FEEDBACK
54
RATING
94
USED
40
Events
Produced By
Channels
Clients
Country
Knowledge Base
Language
Feedback
Rating
Top 10 Viewed
Top 10 Categories
Top Keywords
Top 10 Used
Top Rated
Worst Rated
Top 10 Liked
Top 10 Disliked

Pulse Dashboard Widget



Pulse Performance Dashboard Widget

You have successfully added a widget for accessing Knowledge Center statistics.

End

Knowledge Center Pulse Plugin Configuration Options

You can customize the KPI widget by defining parameters in the URL:

http://<*host*>:<*es_port*>/_plugin/gkc-kpi/?kbId=<knowledge_base_id>(=<chosen language>&tenatId=<tenantId>&timeframe=<timeframe>

- kbId=<knowledge_base_id>— Set which knowledge base id to generate metrics for. If not defined, the metrics will be calculated for all accessible knowledge bases (within defined tenant, if provided).
- lang=<chosen language>— Set the language metrics will be generated for. If not defined, the metrics will be generated in all available languages within the knowledge base and/or tenant.
- tenantId=<tenantId> — Set which tenant to generate metrics for. If not defined, the metrics will be generated for all available tenants (not recommended for multi-tenant environments). **Note**: this option was added in the 8.5.303 release of the product.
- timeframe=<timeframe>— Timeframe to generate metrics (for example now-1M). If not defined, the metrics will be generated for the last hour (now-1h).

Important

Timeframe expression must start with an “anchor” date **- now** and follow by a math expression starting from **- and / (rounding)**. The units supported are **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), and **s** (second). For example, now-1h, now-1h-1m, now-1h/d.

Installing the Workspace Desktop Edition Plugin

Installing the Plugin for Workspace Desktop Edition

Agents can use the Knowledge Center Plugin for Workspace Desktop Edition (WDE) to access knowledge-related information right from their desktop. For example, if a customer asks a question using a chat widget and the corresponding interaction is routed to an agent, Knowledge Center can execute a pre-populated search based on data attached to the new interaction, as well as displaying the customer's search history and providing the agent with full access to the knowledge base access. And if the customer has not authorized during their search, the agent can link their session history to that customer's ID to access their full history while working with the interaction. To use this plugin complete the procedures below, in order.

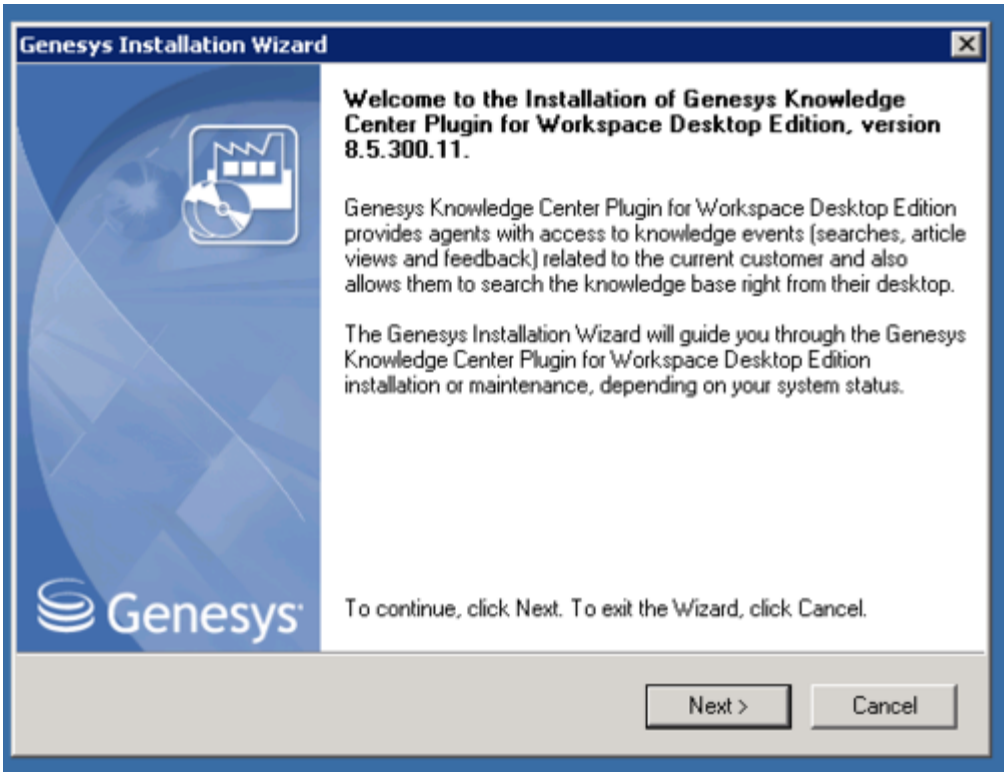
Installing the Plugin for Workspace Desktop Edition

Prerequisites

Workspace Desktop Edition must be installed and configured to work with voice or media interactions.

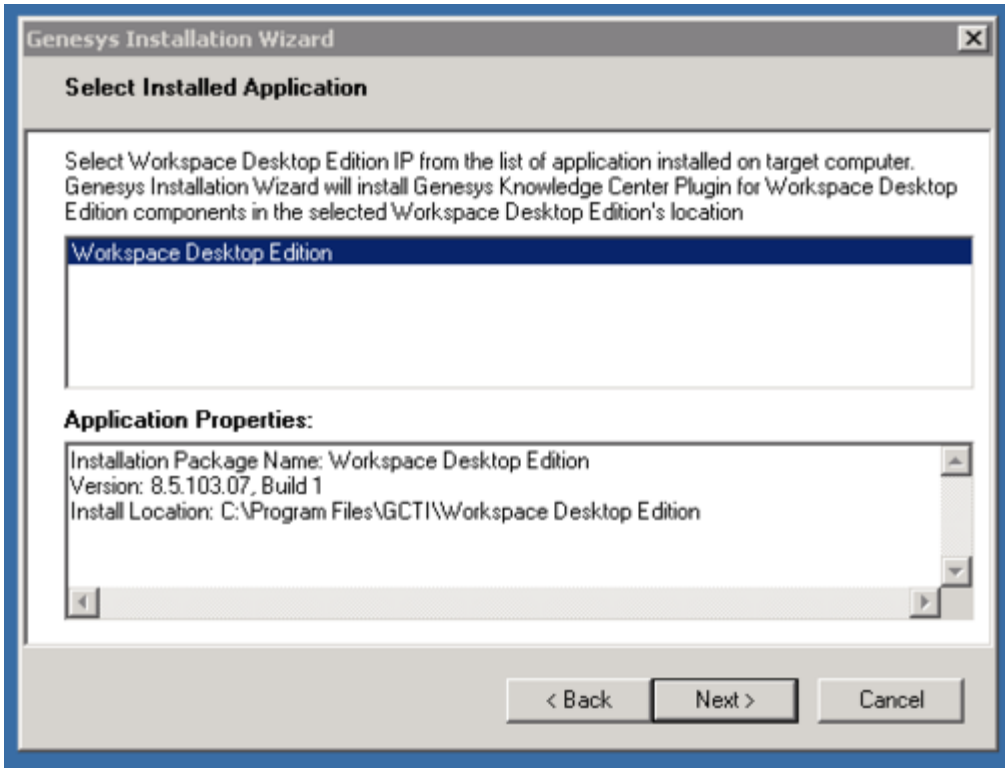
Start

1. In your installation package, locate and double-click the **setup.exe** file. The Install Shield opens the welcome screen.



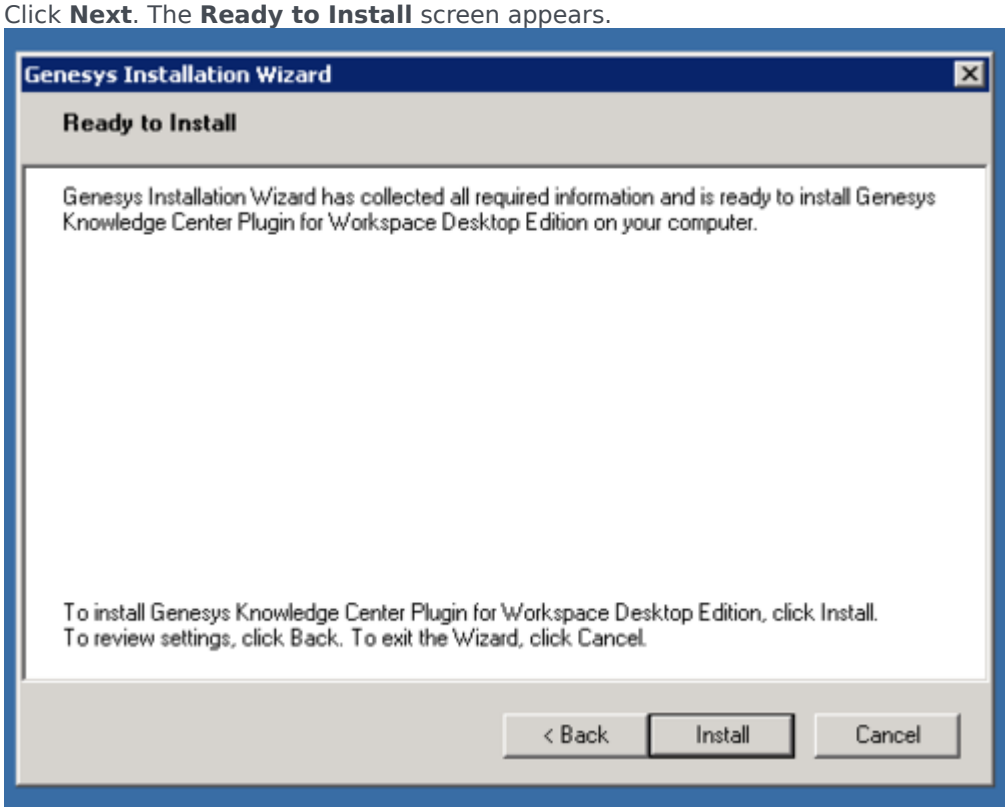
Knowledge Center WDE Plugin—Install Shield Screen

2. Click **Next**. The **Select Installed Application** screen appears.
3. Select the installed Workspace Desktop Edition Application for which you want to install the plugin. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the Server Info and Start Info tabs of the selected Application object.



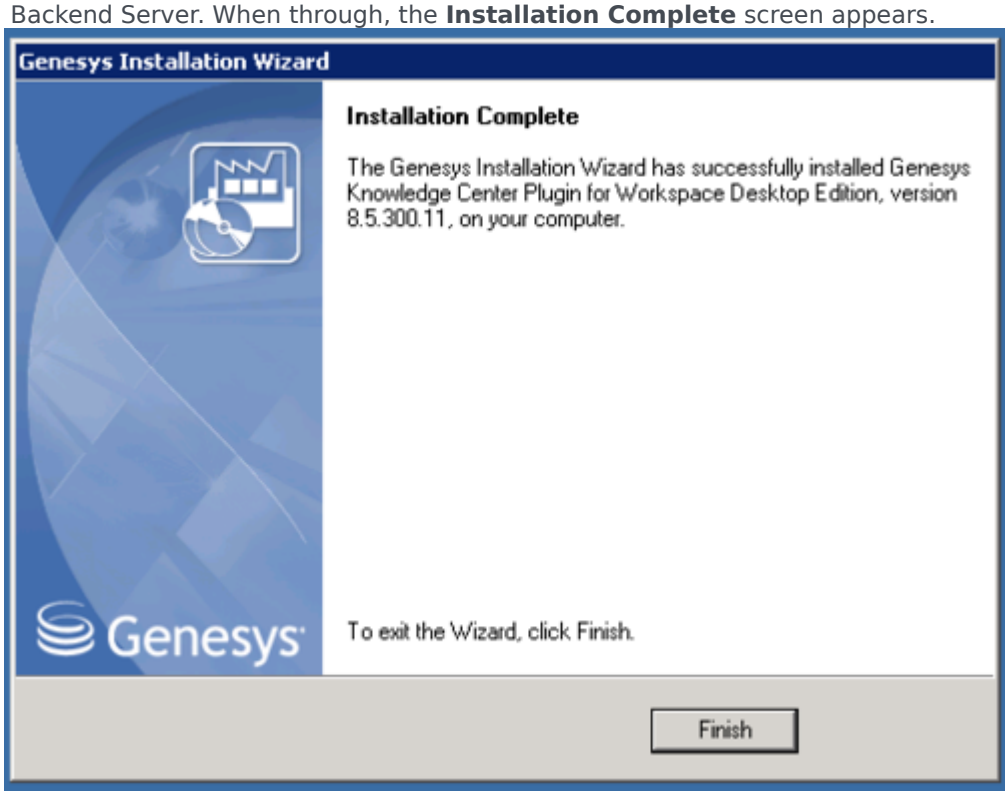
Select Installed Workspace Desktop Edition Application

4. Click **Next**. The **Ready to Install** screen appears.



Knowledge Center WDE Plugin—Ready to Install

5. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for Backend Server. When through, the **Installation Complete** screen appears.



Knowledge Center WDE Plugin—Installation Complete

6. Click **Finish** to complete your installation.
7. Inspect the directory tree of your system to make sure that the following files have been installed in the location that you intended:

- *GWEInstallationFolder\Genesyslab.Desktop.Modules.Knowledge.dll*
- *GWEInstallationFolder\Genesyslab.Desktop.Modules.Knowledge.module-config*
- *GWEInstallationFolder\Genesyslab.Desktop.Modules.Knowledge.pdb*
- *GWEInstallationFolder\Newtonsoft.Json.dll*
- *GWEInstallationFolder\RestSharp.dll*
- *GWEInstallationFolder\System.Net.Http.Formatting.dll*
- *GWEInstallationFolder\Language\Genesyslab.Desktop.Modules.Knowledge.en-US.xml*

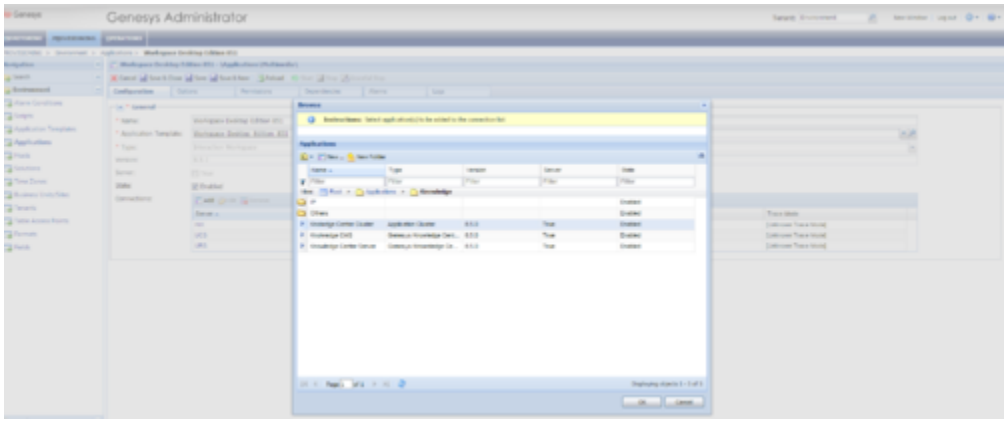
End

Configuring the WDE Application to work with the WDE Plugin

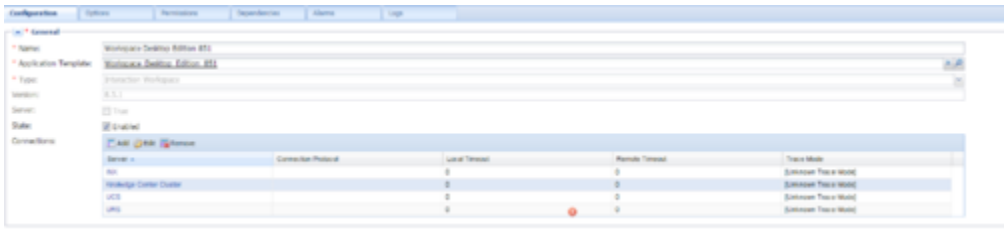
> **Important**
>
> To run the WDE plugin correctly, the local storage must be enabled in Internet Explorer on the host with the WDE client application. To verify this, open **Settings** > **Internet Options** then click the **Advanced** tab > **Security**. Confirm that "Enable DOM-Storage" is checked. If it is not, click the check box and then save your updated settings.

Add the Knowledge Center Cluster to Your WDE Connections

1. If your Workspace Desktop Edition application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the Workspace Desktop Edition and click **Edit...**.
2. In the **Connections** section of the **Configuration** tab, click **Add**. The **Browse for applications** panel opens. Select the **Knowledge Center Cluster application**, then click **OK**.



Knowledge Center WDE Plugin—Browse for applications 1



Knowledge Center WDE Plugin—Browse for applications 2

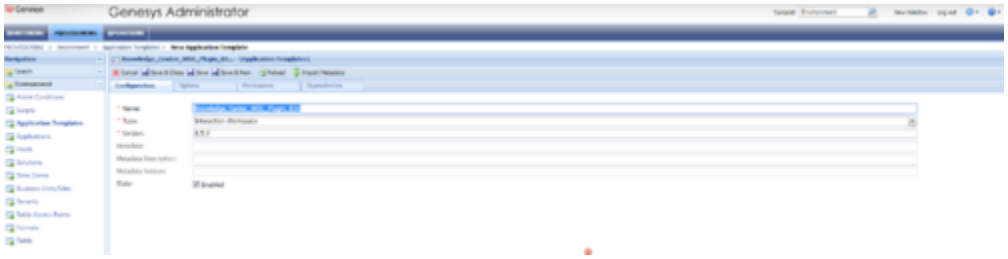
Add Knowledge Center Options to Your WDE Application

To use the Knowledge Center Plugin for WDE, you need to add some options to your WDE application so that it can gather knowledge-related information from incoming interactions. You can add these

options to the the **interaction-workspace** section of the WDE application.

Start

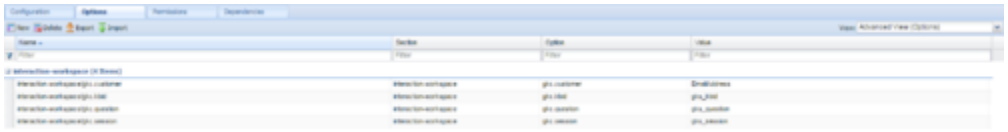
1. Import the template with the additional options:
    1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application Templates**.
    2. In the **Tasks** panel, click **Upload Template**.
    3. In the *Click 'Add' and choose application template (APD) file to import* window, click **Add**.
    4. Choose the application template (APD) file from the import window and click **Add**.
    5. Browse to the *Knowledge_Center_WDE_Plugin_852.apd* file available in the templates directory of your installation CD. The **New Application Template** panel opens.



Knowledge Center WDE Plugin—New Application Template panel

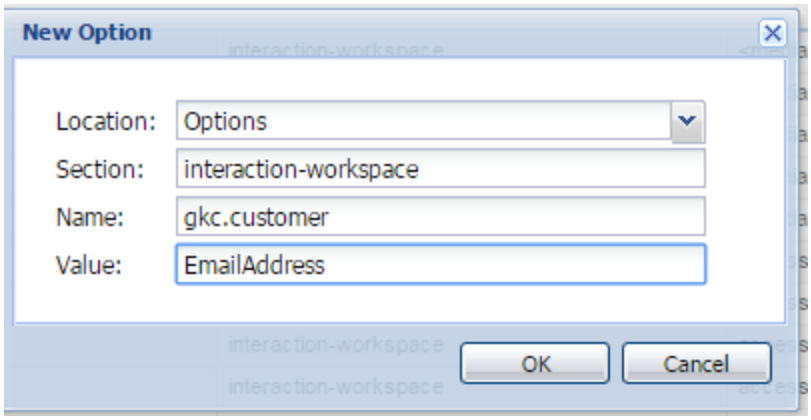
    6. Click **Save and Close**

2. Open the **Options** tab of the uploaded application and review the new options.



Knowledge Center WDE Plugin—Options tab of uploaded application

3. Navigate to **Provisioning > Environment > Applications**. Select the application defined for Workspace Desktop Edition and click **Edit...**.
4. Open the **Options** tab.
5. Add the plugin options to the **interaction-workspace** section using the **New** button.



Knowledge Center WDE Plugin—Add plugin options

End

The Knowledge Center Plugin for WDE uses the following additional options:

Section	Option	Default value	Allowed values	Description	Takes effect
interaction-workspace	gkc.proactive-search	true	true false	Enables or disables Proactive Knowledge in a Chat feature. For additional details, see the Advanced features page.	Next agent session
interaction-workspace	gkc.question	gks_question	any valid user data key	Interaction user data key that contains search query that will be pre-populated in Desktop	Next agent session
interaction-workspace	gkc.kbid	gks_kbid	any valid user data key	Interaction user data key containing the knowledge base Id to search knowledge in	Next agent session
interaction-workspace	gkc.customer	EmailAddress	any valid user data key	Interaction user data key that contains customer identification (for example email address of the customer)	Next agent session

Section	Option	Default value	Allowed values	Description	Takes effect
interaction-workspace	gkc.session	gks_session	any valid user data key	User data key that contains knowledge session Id associated with the interaction	Next agent session
interaction-workspace	gkc.language	**before 8.5.303:** Language **since 8.5.303:** gks_lang, Language	**before 8.5.303:** any valid user data key **after 8.5.303:** comma separated list of valid user data keys	Interaction user data key that contains language of interaction **Note**: since 8.5.303 this option can contain comma-separated ordered list of keys. for example "gks_lang, Language"; in case of several keys in attached data - first key in list from option will be used	Next agent session
interaction-workspace	gkc.country	Country	any valid user data key	Interaction user data key that contains region of interaction (used for multi-regional languages, for example en_US, en_UK)	Next agent session
interaction-workspace	gkc.spellcheck	false	true false	Enables or disables spell check correction of the searched query	Next agent session
interaction-workspace	gkc.can-upvote-on-copy-content	`true`	`true`, `false`	Enables or disables to vote automatic positive feedback after clicking the copy content button. Important Available from 8.5.304.33	Next agent session

Providing Knowledge Center Access to Agents

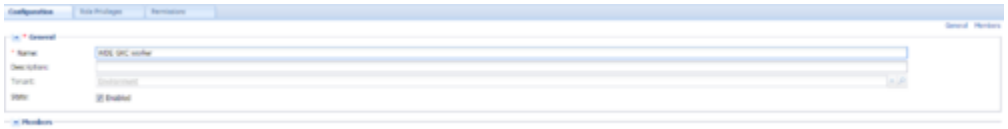
Genesys Knowledge Center supports the following privilege in order to restrict Agent access:

- **Knowledge.WORKER** — Enables access to the Genesys Knowledge Center tab in WDE
- **Knowledge.AUTHOR** — Enables ability to suggest new knowledge to knowledge bases.

To configure the appropriate role for an agent:

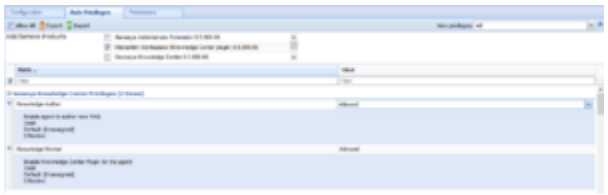
Start

1. Go to **Provisioning > Environment > Application Templates**.
2. Select the application template defined for Workspace Desktop Edition and click **Edit...**.
3. Click **Import Metadata**.
4. Click **Add** and select the *Knowledge_Center_WDE_Plugin_852.xml* file.
5. Click **Open**.
6. Information from the metadata file will be added to the template and the appropriate privilege will be added into the framework.
7. Save and Close.
8. Go to **Provisioning > Accounts > Roles**.
9. In the taskbar click **New** to create a new object.
10. Set the name of the role in the **General** section.



Knowledge Center WDE Plugin—Set Role Names

11. Go to the **Role Privileges** tab, and select the set of roles for Genesys Knowledge Center.
12. Open the WDE Knowledge Center Plugin privileges list and select the **Genesys Knowledge Center Privileges** section.
13. Create the appropriate privileges as allowed.



Knowledge Center WDE Plugin—Create Privileges

14. Go back to the **Configuration** tab.

15. Add the appropriate Agent to the **Members** section by clicking the **Add** button.

Knowledge Center WDE Plugin—Members Section

16. Save and Close.

End

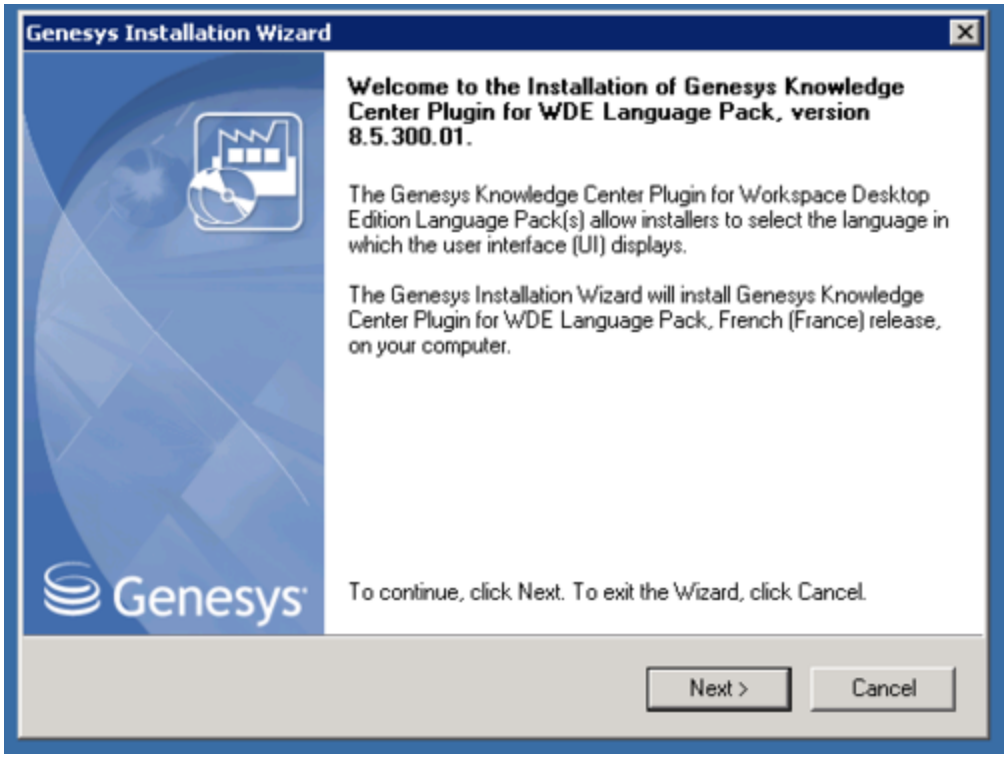
Installing the WDE Language Pack

Prerequisites:

- Must have Genesys Knowledge Center 8.5.3 or higher installed
- Must have Workspace Desktop Edition 8.5.1 or higher installed
- Must have Knowledge Center Workspace Desktop Edition Plugin 8.5.3 or higher installed

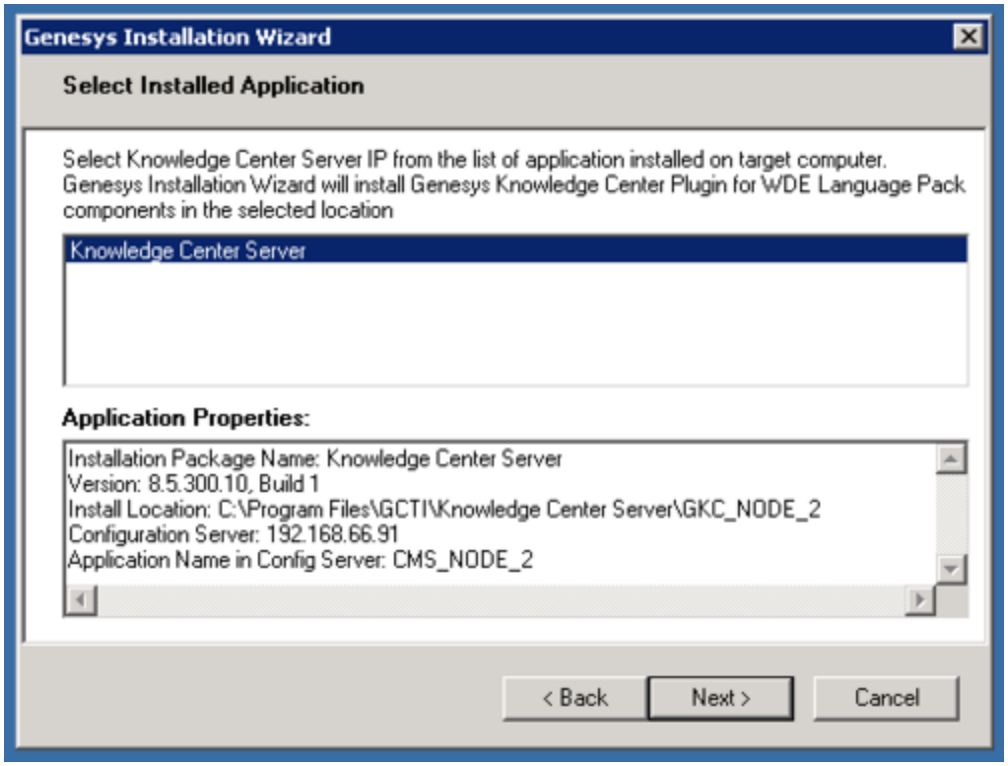
Start

1. In your Language Pack installation package, locate and double-click the **setup.exe** file. The Install Shield opens the welcome screen.



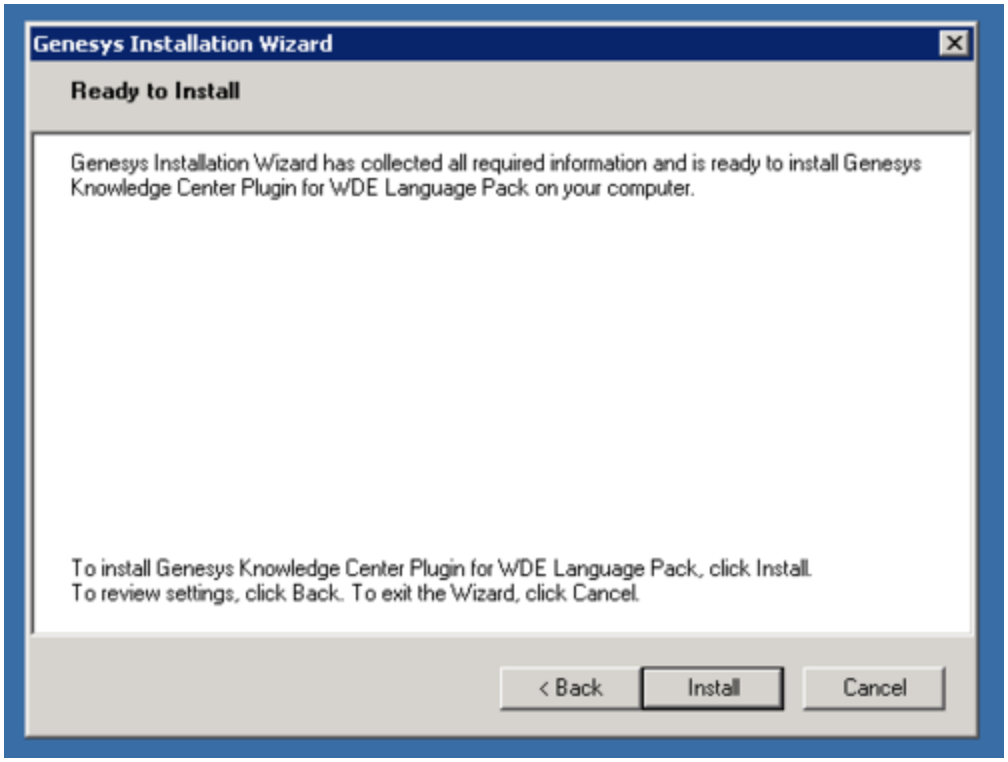
WDE Language Pack Installation Welcome Screen

2. Click **Next**. The **Select Installed Application** screen appears.
3. Select the installed Knowledge Center Server Application for which you want to install the plugin. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the Server Info and Start Info tabs of the selected Application object.



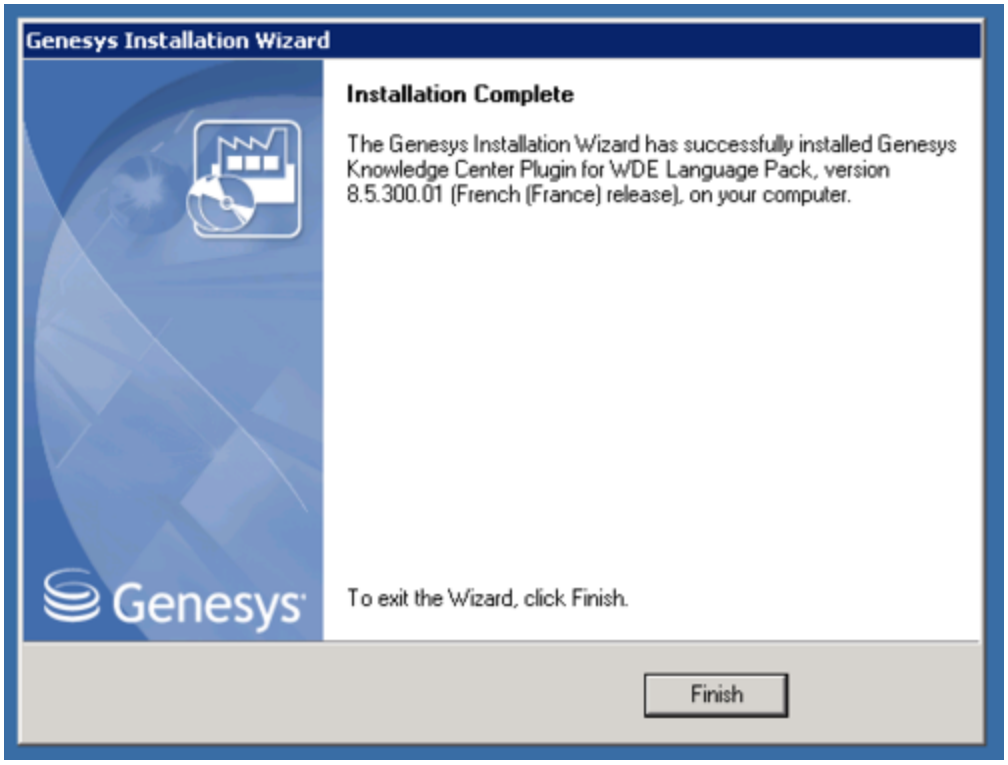
Select Installed Knowledge Center Server Application

4. Click **Next**. The **Ready to Install** screen appears.



WDE Language Pack—Ready to Install

5. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for Backend Server. When through, the **Installation Complete** screen appears.



WDE Language Pack—Installation Complete

6. Click **Finish** to complete your installation.
7. Inspect the directory tree of your system to make sure that the following files, based on the language of your language pack, have been installed in the location that you intended:
    - *<KnowledgeCenterServer_InstallationFolder>\server\resources\wde_de.properties*
    - *<KnowledgeCenterServer_InstallationFolder>\server\resources\wde_fr.properties*
    - *<KnowledgeCenterServer_InstallationFolder>\server\resources\wde_es.properties*
    - *<KnowledgeCenterServer_InstallationFolder>\server\resources\wde_pt.properties*

End

Post Installation and Deployment

This chapter provides you with information on what to do after you've deployed the Knowledge Center within your environment. It covers following topics:

Access Permissions

Overview

Before starting up with Knowledge Center you need to:

- Define access rules for every knowledge base you have created
- Set up permissions for your knowledge team
- Set up access to knowledge for your agents

Knowledge Center leverages privileges and skills to define desired access level:

- Privileges used to grant access to functional capabilities such as authoring ability, approval rights, ability to work with knowledge in Workspace or ability to suggest content. Privileges assigned to roles that you can assign to your personnel.
- Skills allow you define knowledge areas that an agent or author can access. Skills are highly dynamic and allow you to provide additional knowledge while you assign areas of responsibilities to your agents.

Let's review these tasks a bit closer in the following sections.

Access permissions

Restricting access to Knowledge

To restrict access to the knowledge you need to define these restrictions for a knowledge base. All documents within knowledge base will follow the defined restrictions.

A knowledge base can be:

- Public - its content is accessible to all agents and customers
- Private - its content is accessible to agents only
- Private with skill restriction - content of the knowledge base is accessible to an agent with specific skills

Skill-based access allows you dynamically manage access to the knowledge along with managing the distribution of the customer's interactions. That ensures that there are no additional actions required when you assign an agent to a new area.

For example, say you have a knowledge base with restricted access for agents who have the skill "technical support", and the same skill is used for routing interactions to your group of agents. Adding the agent to the group by assigning him the "technical support" skill will automatically give him access to the proper knowledge bases.

Setting up Knowledge Team

The next task is to grant access to the CMS and knowledge bases to the members of your authoring team. CMS allows the following privileges to be granted:

- Administrator - able to create, modify and delete knowledge bases' properties
- Author: Categories - gives content author ability to create new categories, and modify and delete existing ones
- Author: Documents - allows content author ability to create new documents, and modify and delete existing ones
- Approver - designated for content managers who validate and approve created documents and categories

At least one of the above privileges are required to be able to work with CMS.

Also, content authors and managers need to be assigned proper skills to get access to the private knowledge bases with skill restrictions. **Note:** Administrators have access to any knowledge base, no matter the skill restrictions applied.

Granting access to agents

Agents follow the same concept. They require *privileges* to get access to functionality and *skills* for getting access to private knowledge bases with skill restrictions.

An agent can be assigned following privileges:

- Knowledge Worker - allows access to Knowledge Center functionality in the Workspace
- Knowledge Author - allows agent to suggest knowledge content from the Workspace

Also, agents need to be assigned proper skills to get access to the private knowledge bases with skill restrictions.

Configuration Procedures

Knowledge Center Privileges

Knowledge Center supports following privileges

Privilege	Product	Description	Since
Knowledge.AUTHOR	Knowledge Center Server	Allows changing data in a knowledge base. This privilege is required for agents that are running data synchronization from Genesys Knowledge Center CMS or third-party sources.	8.5.0

Privilege	Product	Description	Since
Knowledge.ADMINISTER	Knowledge Center Server	Allows to manage knowledge bases	8.5.0
Knowledge.REPORTING	Knowledge Center Server	Allows using reporting capabilities	8.5.0
Knowledge.MULTITENANT	Knowledge Center Server	Allows to bypass tenants restrictions while importing data into knowledge bases. **Important** Required only for the multi-tenant deployments. Will not affect other privileges or access rights except authoring.	8.5.303.14
Knowledge.CMS.Document.Author	Knowledge Center CMS	Allows to create and update knowledge documents	8.5.0
Knowledge.CMS.Category.Author	Knowledge Center CMS	Allows to create and update categories	8.5.0
Knowledge.CMS.Approver	Knowledge Center CMS	Allows approving and publishing edited categories and document for use by agent and customers	8.5.0
Knowledge.CMS.Administrator	Knowledge Center CMS	Allows to manage knowledge bases (create new, modify and delete)	8.5.0
Knowledge.CMS.Multitenant	Knowledge Center CMS	Allows bypassing tenant restrictions while working through API	8.5.303.14
Knowledge.Worker	Workspace Desktop Edition	Enable the Knowledge Center Plugin for the agent	8.5.0
Knowledge.Author	Workspace Desktop Edition	Allows agent to propose new knowledge documents from Workspace	8.5.0
Knowledge.ADMINISTER	Genesys Administrator	Allows to manage knowledge bases **Important** Discontinued from 8.5.303.14 release of the product	8.5.0

Privileges for typical Roles

The table below shows examples of typical roles and privileges required for them:

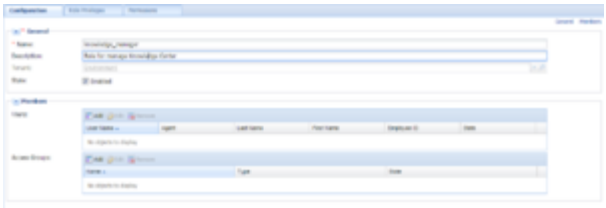
Role	Description	Privileges
CMS Administrator	• Manages Knowledge Bases • Sets up publishing schedules • Doing maintenance procedures with knowledge	• Knowledge.CMS.Administrator (CMS) • Knowledge.ADMINISTER (Server) • Knowledge.ADMINISTER (Administrator) - 8.5.302.xx or earlier release
Knowledge Manager	• Approves content produced by authors • Publish knowledge documents to be used by agents and customer	• Knowledge.CMS.Approver (CMS) • Knowledge.AUTHOR (Server) • Knowledge.REPORTING (Server)
Knowledge Author	• Creates knowledge documents • Creates knowledge categories • Reviews usage feedback and update knowledge content	• Knowledge.CMS.Document.Author (CMS) • Knowledge.CMS.Category.Author (CMS)
Agent	• Handles customers' interactions	• Knowledge.Worker (WDE) • Knowledge.Author (WDE)

Assigning a Privilege

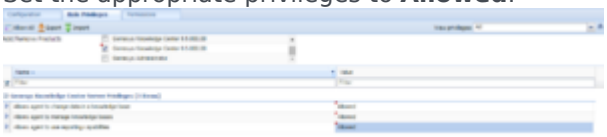
To configure the appropriate privileges for an Agent:

Start

1. Go to **Provisioning** > **Accounts** > Roles.
2. In the taskbar, click **New** to create a new object.
3. Set the name of the role in the **General** section.



4. Go to the **Role Privileges** tab and select the set of roles for Genesys Knowledge Center.
5. Open the list of privileges for Knowledge Center Server.
6. Set the appropriate privileges to **Allowed**.



7. Go back to the **Configuration** tab.
8. In the **Members** section, add the appropriate agent by clicking the **Add** button.

9. Save and Close.

End

Assignining Skills

To configure the appropriate skills for an Agent:

Start

1. Go to **Provisioning** > **Accounts** > **Users**.
2. Select **Agent** from the table
3. Click **Edit...** button
4. Expand Agent Info panel
5. Click **Add...** in the **Skills Level** section

6. In the **Skills Level** dialog:

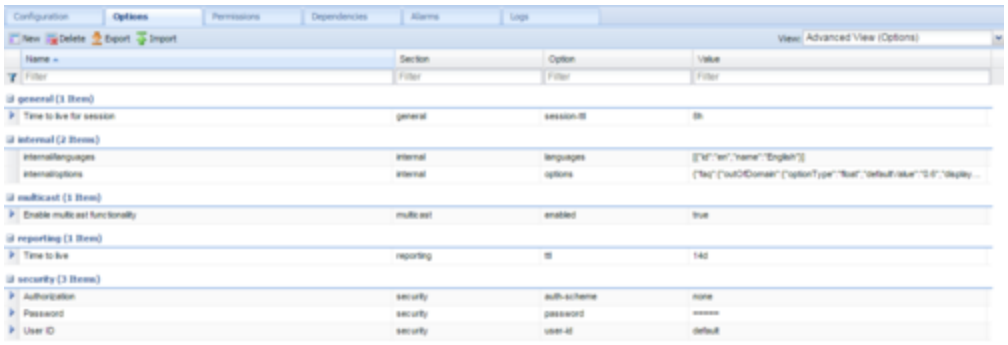
- select the skill
- enter the skill level
- click OK

7. Save and Close.

End

Configuration Options

Knowledge Center Cluster Application Options



Knowledge Center Cluster Application Configuration Options

Name	Description	Value
Section: cms.cluster		
type	Type of storage used for repository Applies to: Genesys Knowledge Center CMS	**Default:** cassandra **Valid Values:** jdbc*, cassandra, mssql, oracle, postegre** **Changes Take Effect:** After restart **Note**: *Since 8.5.302.xx 'jdbc' value is deprecated, 'mssql' value must be used instead* Important • mssql and oracle values were added beginning with version 8.5.302.xx of the product • PostgreSQL support added in version 8.5.304

dbConnectionUrl	Correct connection string for connection to MS SQL/Oracle/PostreSQL,
	
	this options should be configured when **type** option in the same section is set to mssql or oracle
	Applies to: Genesys Knowledge Center CMS
		Default: n/a
		
		Valid Values:
		MSSQL example:
		
		`jdbc:jtds:sqlserver://<host of MS SQL server>:<port of MS SQL server; 1433 by default>;databaseName=<CMS DB name>`
		
		Oracle example:
		
		`jdbc:oracle:thin:<userName>/<password>@<host of Oracle DB>:<port of Oracle DB; 1521 by default>:<SID. e.g. ORCL>`
		
		PostreSQL example:

Name	Description	Value
		`jdbc:postgresql://<host of PostreSQL server>:<port of PostreSQL server; 5432 by default>/<CMS DB name>` **Changes Take Effect:** After restart
dbDriverClass	Driver class name for MS SQL/Oracle connector; when **type** option in the same section is set to mssql or oracle Applies to: Genesys Knowledge Center CMS Important As of version 8.5.304 you can ignore this option when `type` option is set to either mssql, oracle, or postgre. CMS will use the proper driver based on the database selected.	**Default:** net.sourceforge.jtds.jdbc.Driver **Valid Values:** example for jTDS 1.2.7 for MS SQL - net.sourceforge.jtds.jdbc.Driver example for ojdbc6 for Oracle - oracle.jdbc.driver.OracleDriver example for PostreSQL jdbc driver - org.postgresql.Driver **Changes Take Effect:** After restart
dbUsername	Name of user for access JDBC database Applies to: Genesys Knowledge Center CMS Important Does not have have an effect when `type` is set to cassandra. For Cassandra, use `userName` option in cassandra-keyspace section	**Default:** n/a **Valid Values:** correct username **Changes Take Effect:** After restart
dbPassword	Password for user for access JDBC database Applies to: Genesys Knowledge Center CMS Important Does not have have an effect when `type` is set to cassandra. For Cassandra, use `userName` option in cassandra-keyspace section	**Default:** n/a **Valid Values:** correct password **Changes Take Effect:** After restart

Name	Description	Value
jgroupsConfiguration	Determine the communication approach used to interact between a servers. Applies to: Genesys Knowledge Center CMS	**Default:** TCP **Valid Values:** JGROUPS_UPD, JGROUPS_TCP, JGROUPS_EC2, TCP, TCP_NIO, TCP_GOSSIP, TUNNEL, UDP_LARGECLUSTER **Changes Take Effect:** After restart
cacheName	Cache name, table(Oracle, MSSQL) or column family(Cassandra) name for main infinispan cache Applies to: Genesys Knowledge Center CMS	**Default:** gkc **Valid Values:** any string < 32 characters **Changes Take Effect:** after restart
cacheBinaryName	Binary cache name, table(Oracle, MSSQL) or column family(Cassandra) name for binary infinispan cache Applies to: Genesys Knowledge Center CMS	**Default:** gkc-binary **Valid Values:** any string < 32 characters **Changes Take Effect:** after restart
cacheMetadataName	Metadata cache name, table(Oracle, MSSQL) or column family(Cassandra) name for metadata infinispan cache Applies to: Genesys Knowledge Center CMS	**Default:** gkc-metadata **Valid Values:** any string < 32 characters **Changes Take Effect:** after restart
Section: cms.general		
externalURL	Public URL that is used to access the CMS (like http://<cms host>:<CMS default port>/gks-cms). This URL will be used to build the link on the attachments in knowledge documents. Applies to: Genesys Knowledge Center CMS	**Default:** none **Valid Values:** Valid URL **Changes Take Effect:** Immediately
Section: cms.index		
minNodes	Defines a minimal number of CMS nodes required to form a functioning CMS Cluster. If a value is 0 - cluster will work properly in case if (N/2)+1 CMS nodes started, where N is a count of all configured CMS nodes in the cluster.	**Default:** 0 **Valid Values:** int, in range [0; 999] **Changes Take Effect:** After restart

Name	Description	Value
	Applies to: Genesys Knowledge Center CMS	
Section: general		
session-ttl	Specify time that server will store session information while no activities are taking place. Applies to: Genesys Knowledge Center Server	**Default:** 8h **Valid Values:** number + unit, for example 1d or 3m. Supported units: d (days), m (minutes), h (hours), or w(weeks) **Changes Take Effect:** After restart.
esReadOnly	Allow only read operation over ES port. Important Enabling write access through the native ElasticSearch REST API may result in data loss and/or corruption. Please ensure that only designated users/ hosts will have access to this API. Applies to: Genesys Knowledge Center Server	**Default:** true **Valid Values:** true, false **Changes Take Effect:** Immediately
knowledgebaseFolder	Name of the folder that will store information about the knowledge bases definitions in Configuration Server. The folder will be placed in Script objects within Tenant that knowledge bases belong to. **Note:** Since 8.5.303.14 Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** knowledge **Valid values:** any string **Changes Take Effect:** After restart.
Section: search		
numberOfAnswers	Number of documents returned in the result. Applies to: Genesys Knowledge Center Server	**Default:** 6 **Valid Values:**: int, in range [1; 65535] **Changes Take Effect:** Immediately
Section: index		

Name	Description	Value
minimumMasterNodes	Defines a minimal number of server nodes required to form a functioning Server Cluster. If value is 0 - Cluster will work properly in case if (N/2)+1 nodes started, where N is count of all configured GKS nodes in GKS cluster. Applies to: Genesys Knowledge Center Server	**Default:** 0 **Valid Values:** int, in range [1; 100] **Changes Take Effect:** After restart
docstatNumberOfShards	Number of shards for "docstat" index of each knowledge base Applies to: Genesys Knowledge Center Server	**Default:** 1 **Valid Values:** int, in range [1; 10] **Changes Take Effect:** Immediately Changes takes no effect after creation of index 'docstat'.
docstatNumberOfReplicas	Number of replicas for "docstat" index of each knowledge base. Applies to: Genesys Knowledge Center Server	**Default:** 1 **Valid Values:** int, in range [1; 10] **Changes Take Effect:** Immediately
historyNumberOfShards	Number of shards for index of "history". Applies to: Genesys Knowledge Center Server	**Default:** 1 **Valid Values:** int, in range [1; 10] **Changes Take Effect:** Immediately Changes takes effect at the moment of new segment of index 'history' creation.
historyNumberOfReplicas	Number of replicas for index of "history". Applies to: Genesys Knowledge Center Server	**Default:** 1 **Valid Values:** int, in range [1; 10] **Changes Take Effect:** Immediately
Section: reporting		
geo	Determine the precision of the IP geo-location algorithm. Applies to: Genesys Knowledge Center Server	**Default:** CITY **Valid Values:** OFF - Disabled, IP - Customer's IP Address, COUNTRY - Customer's country, CITY - Customer's city **Changes Take Effect:** Immediately
ttl	Specify time that records will be stored in the	**Default:** 365d

Name	Description	Value
	history. Applies to: Genesys Knowledge Center Server	**Valid Values:**number + unit, for example 1d or 3m. Supported units: d (days), m (minutes), h (hours), or w(weeks) **Changes Take Effect:** After restart.
Section: log		
all	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** stdout **Valid Values:** (log output types) **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the all log level option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's

Name	Description	Value
		working directory. **Changes Take Effect:** After start or restart.
standard	Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example: standard = stderr, network. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Name	Description	Value
		Changes Take Effect: Immediately
trace	Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: trace = stderr, network. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** Immediately
verbose	Determines whether a log output is created. If it is, specifies the minimum level of log events	**Default:** standard

Name	Description	Value
	generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Valid Values:** **[+] all** All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated. **[+] debug** The same as all. **[+] trace** Log events of the Trace level and higher (that is, log eventsof the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated. **[+] interaction** Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated. **[+] standard** Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated. **[+] none** No output is produced. **Changes Take Effect:** Immediately

Name	Description	Value
segment	Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** 1000 **Valid Values:** **[+] false** No segmentation is allowed. **[+] <number> KB or <number>** Sets the maximum segment size, in kilobytes. The minimum segment size is 100 KB. **[+] <number> MB** Sets the maximum segment size, in megabytes. **[+] <number> hr** Sets the number of hours for the segment to stay open. The minimum number is 1 hour. **Changes Take Effect:** After restart.
expire	Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** 3 **Valid Values:** **[+] false** No expiration; all generated segments are stored. **[+] <number> file or <number>** Sets the maximum number of log files to store. Specify a number from 1—1000.

Name	Description	Value
		[+] <number> day Sets the maximum number of days before log files are deleted. Specify a number from 1—100. **Changes Take Effect:** After restart. Important If an option's value is not set within the range of valid values, it will automatically be reset to 10.
affectedLoggers	Verbosity settings are explicitly applied for the following loggers: • Loggers that are not declared explicitly in the *log4j2.xml* configuration file. • Loggers that are specified explicitly in the *log4j2.xml* and are specified in the value for this affectedLoggers option. For other loggers specified in *log4j2.xml*, but not mentioned in the value for this option, the verbosity level is not re-applied. Here is a use case for when you might need to set this option: • Cassandra needs to write error messages to a log file, and at the same time, Genesys components also need to write debug messages to the log file. To resolve this use case, you would: 1. Specify the following logger in *log4j2.xml*: *<logger name="org.apache.cassandra" level="error" additivity="false">*	**Default:** None **Valid Values:** The names of loggers, separated by a semicolon (;), specified in the LOG4J2.xml. For example: *com.genesyslab.wmcbcore, com.genesyslab.qna.api.sdk,* org.elasticsearch, com.genesyslab.platform, com.genesys.knowledge.api.processors, com.genesys.knowledge.server.configuration, com.genesys.elasticsearch.index.analysis.filters, com.genesys.elasticsearch.index.analysis. tokenizers, com.genesys.knowledge.security.proxy, com.genesys.knowledge.aspects. LoggingRestAspect, com.genesys.knowledge.web.filters. RequestLoggingFilter **Changes Take Effect:** Immediately

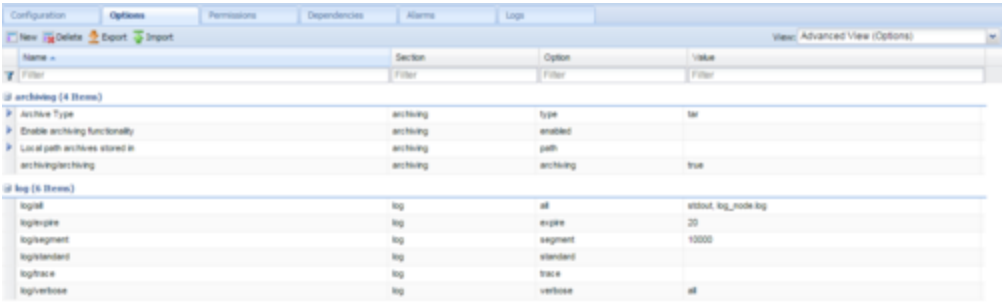
Name	Description	Value
	2. **Do not** include *org.apache.cassandra* in the value for the **affectedLoggers** option. 3. The default *log4j2.xml* file contains the following logger: *<logger name= "com.genesyslab.platform" level="info"* additivity="false"> 1. Include *com.genesyslab.platform* in the value for the **affectedLoggers** option. 2. Set the **verbose** option to *debug*. In the sample above, the value of **affectedLoggers** should be *com.genesyslab.platform*. Error (but not debug or info) messages from Cassandra will be available in logs, and debug messages from *com.genesyslab.platform* will be available in logs. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	
time_format	Specifies how to represent, in a log file, the time when an application generates log records. A log record's time field in the ISO 8601 format looks like this: 2001-07-24T04:58:10.123. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** time **Valid Values:** **[+] time** The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. **[+] locale** The time string is formatted according to the system's locale. **[+] ISO8601** The date in the time string is formatted according to the ISO

Name	Description	Value
		8601 format. Fractional seconds are given in milliseconds. **Changes Take Effect:** Immediately
time_convert	Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since 00:00:00 UTC, January 1, 1970. Applies to: Genesys Knowledge Center Server, Genesys Knowledge Center CMS	**Default:** local **Valid Values:** **[+] local** The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. **[+] utc** The time of log record generation is expressed as Coordinated Universal Time (UTC). **Changes Take Effect:** Immediately
Section: security		
auth-scheme	Specifies the HTTP authentication scheme used to secure REST API requests to the Knowledge Server. With the Basic scheme, clients must be authenticated with a user ID and password. Applies to: Genesys Knowledge Center Server	**Default:** none **Valid Values:** none, basic **Changes Take Effect:** After restart.
user-id	The user identifier (login) used in authentication for the REST API. Applies to: Genesys Knowledge Center Server	**Default:** n/a **Valid Values:** string **Changes Take Effect:** After restart.

Name	Description	Value
password	The user password used in authentication for the REST API. Applies to: Genesys Knowledge Center Server	**Default:** n/a **Valid Values:**string **Changes Take Effect:** After restart.
Section: cassandra-keyspace **Important:** Cassandra support is deprecated.		
name	name of the cassandra keyspace. Applies to: Genesys Knowledge Center CMS	**Default:** gkccms **Valid Values:** string, any valid cassandra keyspace name **Changes Take Effect:** After restart
dataCompression	The compression algorithm to use. Applies to: Genesys Knowledge Center CMS	**Default:** LZ4Compressor **Valid Values:** None, LZ4Compressor, SnappyCompressor, and DeflateCompressor **Changes Take Effect:** After restart.
userName	Cassandra user name. Applies to: Genesys Knowledge Center CMS	**Default:**n/a **Changes Take Effect:** After restart.
password	Cassandra password. Applies to: Genesys Knowledge Center CMS	**Default:**n/a **Changes Take Effect:** After restart.
replicationStrategy	Cassandra replication strategy. Applies to: Genesys Knowledge Center CMS	**Default:** SimpleStrategy **Valid Values:** valid replication strategy name **Changes Take Effect:** After restart.
replicationStrategyParams	Cassandra replication strategy params. Applies to: Genesys Knowledge Center CMS	**Default:** 'replication_factor':3 **Changes Take Effect:** After restart.
readConsistencyLevel	Cassandra consistency level for reading. Applies to: Genesys Knowledge Center CMS	**Default:** ONE **Valid Values:** 1-ONE,

Name	Description	Value
		2-QUORUM,3-LOCAL_QUORUM,4-EACH_QUORUM,5-ALL,6-ANY,7-TWO,8-THREE,9-S **Changes Take Effect:** After restart.
writeConsistencyLevel	Cassandra consistency level for writing. Applies to: Genesys Knowledge Center CMS	**Default:** ONE **Valid Values:** 1-ONE, 2-QUORUM,3-LOCAL_QUORUM,4-EACH_QUORUM,5-ALL,6-ANY,7-TWO,8-THREE,9-S **Changes Take Effect:** After restart.
Section: cassandra-security **Important:** Cassandra support is deprecated		
enable-ssl	Enables or disables SSL for connection to Cassandra cluster. Applies to: Genesys Knowledge Center CMS	**Default:** false **Valid Values:** true, false **Changes Take Effect:** After restart.
truststore-path	Path to truststore. Applies to: Genesys Knowledge Center CMS	**Default:** n/a **Changes Take Effect:** After restart.
truststore-password	Truststore password. Applies to: Genesys Knowledge Center CMS	**Default:** n/a **Changes Take Effect:** After restart.
Section: internal		
Important Knowledge Center Server uses this section to store internal initialization parameters. Do not attempt to change these options.		

Knowledge Center Server Application Options



Knowledge Center Server Application Configuration Options

Name	Description	Value
Section: archiving		
enabled	Specifies whether a node will allow to execute archiving using its API. Enabling archiving on the node does not affect other nodes of the cluster. Archiving is resource consuming functionality - use it wisely.	**Default:** true **Valid Values:** true, false **Changes Take Effect:** After restart.
type	Defines format of resulted archive will be stored in.	**Default:** tar **Valid Values:** tar, zip, cpio **Changes Take Effect:** After restart.
path	Path to the stored archive. The archive will be stored as <path>/history_<requested_date_range>.<archive>	**Default:** none **Valid Values:** string **Changes Take Effect:** After restart.
Section: log		
all	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile	**Default:** stdout **Valid Values:** (log output types) **[+] stddout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the all log level option to the network output enables an application to send log events

Name	Description	Value
		of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** After start or restart.
standard	Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example: standard = stderr, network	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.

Name	Description	Value
		[+] memory Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** Immediately
trace	Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: trace = stderr, network.	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk.

Name	Description	Value
		This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** Immediately
verbose	Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.	**Default:** standard **Valid Values:** **[+] all** All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated. **[+] debug** The same as all. **[+] trace** Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated. **[+] interaction** Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.

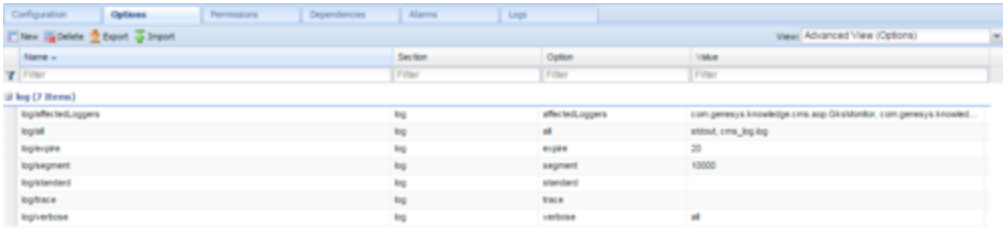
Name	Description	Value
		[+] standard Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated. **[+] none** No output is produced. **Changes Take Effect:** Immediately
segment	Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.	**Default:** 1000 **Valid Values:** **[+] false** No segmentation is allowed. **[+] <number> KB or <number>** Sets the maximum segment size, in kilobytes. The minimum segment size is 100 KB. **[+] <number> MB** Sets the maximum segment size, in megabytes. **[+] <number> hr** Sets the number of hours for the segment to stay open. The minimum number is 1 hour.

Name	Description	Value
		Changes Take Effect: After restart.
expire	Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.	**Default:** 3 **Valid Values:** **[+] false** No expiration; all generated segments are stored. **[+] <number> file or <number>** Sets the maximum number of log files to store. Specify a number from 1—1000. **[+] <number> day** Sets the maximum number of days before log files are deleted. Specify a number from 1—100. **Changes Take Effect:** After restart. Important If an option's value is not set within the range of valid values, it will automatically be reset to 10.
affectedLoggers	Verbosity settings are explicitly applied for the following loggers: • Loggers that are not declared explicitly in the *log4j2.xml* configuration file. • Loggers that are specified explicitly in the *log4j2.xml* and are specified in the value for this affectedLoggers option.	**Default:** com.genesys.knowledge.server.configuration, com.genesyslab.wmcbcore, com.genesys.knowledge.manager **Valid Values:** The names of loggers, separated by a semicolon (;), specified in the LOG4J2.xml. For example: com.genesyslab.wmcbcore, com.genesyslab.qna.api.sdk, org.elasticsearch, com.genesyslab.platform, com.genesys.knowledge.api.processors,

Name	Description	Value
	For other loggers specified in *log4j2.xml*, but not mentioned in the value for this option, the verbosity level is not re-applied. Here is a use case for when you might need to set this option: • Cassandra needs to write error messages to a log file, and at the same time, Genesys components also need to write debug messages to the log file. To resolve this use case, you would: 1. Specify the following logger in *log4j2.xml*: *<logger name="org.apache.cassandra" level="error" additivity="false">* 2. **Do not** include *org.apache.cassandra* in the value for the **affectedLoggers** option. 3. The default *log4j2.xml* file contains the following logger: *<logger name="com.genesyslab.platform" level="info" additivity="false">* 4. Include *com.genesyslab.platform* in the value for the **affectedLoggers** option. 5. Set the **verbose** option to *debug*. In the sample above, the value of **affectedLoggers** should be *com.genesyslab.platform*. Error (but not debug or info) messages from Cassandra will be available in logs, and debug messages from *com.genesyslab.platform* will be available in logs.	com.genesys.knowledge.server.configuration, com.genesys.elasticsearch.index.analysis.filters, com.genesys.elasticsearch.index.analysis.tokenizers, com.genesys.knowledge.security.proxy, com.genesys.knowledge.aspects.LoggingRestAspect, com.genesys.knowledge.web.filters.RequestLoggingFilter **Changes Take Effect:** Immediately
time_format	Specifies how to represent, in a log file, the time when an application generates log records. A log record's time field in the ISO 8601 format looks like this: 2001-07-24T04:58:10.123	**Default:** time **Valid Values:** **[+] time**

Name	Description	Value
		The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. **[+] locale** The time string is formatted according to the system's locale. **[+] ISO8601** The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds. **Changes Take Effect:** Immediately
time_convert	Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since 00:00:00 UTC, January 1, 1970.	**Default:** local **Valid Values:** **[+] local** The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. **[+] utc** The time of log record generation is expressed as Coordinated Universal Time (UTC). **Changes Take Effect:** Immediately

Knowledge Center CMS Application Options



Knowledge Center CMS Application Options

Name	Description	Value
Section: cassandra-security **Important:** Cassandra support is deprecated.		
truststore-path	Path to truststore	**Default:** /trustStore.jks **Valid Values:** valid path to trust store **Changes Take Effect:** After start or restart.
truststore-password	Truststore password	**Default:** n/a **Valid Values:** valid path to trust store **Changes Take Effect:** After start or restart.
Section: gkc-security		
enable-ssl	Enables/disables secure connection from CMS to the Genesys Knowledge Center Server	**Default:** false **Valid Values:** true, false **Changes Take Effect:** After start or restart.
truststore-path	Path to truststore	**Default:** ./trustStore.jks **Changes Take Effect:** After start or restart.
truststore-password	Truststore password	**Default:** n/a **Changes Take Effect:** After start or restart.
Section: log		
all	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile	**Default:** stdout **Valid Values:** (log output types) **[+] stddout** Log events are sent to the Standard output (stdout). **[+] stderr**

Name	Description	Value
		Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the all log level option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** After start or restart.
standard	Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example: standard = stderr, network	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr**

Name	Description	Value
		Log events are sent to the Standard error output (stderr). **[+] network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** Immediately
trace	Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: trace = stderr, network.	**Default:** stdout **Valid Values:** **[+] stdout** Log events are sent to the Standard output (stdout). **[+] stderr** Log events are sent to the Standard error output (stderr). **[+] network**

Name	Description	Value
		Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. **[+] memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. **[+] [filename]** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. **Changes Take Effect:** Immediately
verbose	Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.	**Default:** standard **Valid Values:** **[+] all** All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated. **[+] debug** The same as all. **[+] trace** Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.

Name	Description	Value
		[+] interaction Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated. **[+] standard** Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated. **[+] none** No output is produced. **Changes Take Effect:** Immediately
segment	Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.	**Default:** 1000 **Valid Values:** **[+] false** No segmentation is allowed. **[+] <number> KB or <number>** Sets the maximum segment size, in kilobytes. The minimum segment size is 100 KB. **[+] <number> MB** Sets the maximum segment size, in megabytes.

Name	Description	Value
		[+] <number> hr Sets the number of hours for the segment to stay open. The minimum number is 1 hour. **Changes Take Effect:** After restart.
expire	Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.	**Default:** 3 **Valid Values:** **[+] false** No expiration; all generated segments are stored. **[+] <number> file or <number>** Sets the maximum number of log files to store. Specify a number from 1—1000. **[+] <number> day** Sets the maximum number of days before log files are deleted. Specify a number from 1—100. **Changes Take Effect:** After restart. Important If an option's value is not set within the range of valid values, it will automatically be reset to 10.
affectedLoggers	Verbosity settings are explicitly applied for the following loggers:	**Default:** com.genesys.knowledge.cms.service, com.genesys.jcr, org.modeshape.jcr.spi.index.provider, com.genesys.knowledge.server.configuration,

Name	Description	Value
	• Loggers that are not declared explicitly in the *log4j2.xml* configuration file. • Loggers that are specified explicitly in the *log4j2.xml* and are specified in the value for this affectedLoggers option. For other loggers specified in *log4j2.xml*, but not mentioned in the value for this option, the verbosity level is not re-applied. Here is a use case for when you might need to set this option: • Cassandra needs to write error messages to a log file, and at the same time, Genesys components also need to write debug messages to the log file. To resolve this use case, you would: 1. Specify the following logger in *log4j2.xml*: *<logger name="org.apache.cassandra" level="error" additivity="false">* 2. **Do not** include *org.apache.cassandra* in the value for the **affectedLoggers** option. 3. The default *log4j2.xml* file contains the following logger: *<logger name="com.genesyslab.platform" level="info" additivity="false">* 4. Include *com.genesyslab.platform* in the value for the **affectedLoggers** option. 5. Set the **verbose** option to *debug*. In the sample above, the value of **affectedLoggers** should be *com.genesyslab.platform*. Error (but not debug or info) messages from Cassandra will be available in logs, and debug messages from *com.genesyslab.platform* will be available in	com.genesyslab.wmcbcore, com.genesys.knowledge.manager, com.genesys.knowledge.manager.aop.ManagerRespositoryMonito com.genesys.knowledge.cms.gks, com.genesys.knowledge.cms.aop.GksMonitor, com.genesys.knowledge.cms.scheduling, com.genesys.knowledge.cms.rest **Valid Values:** The names of loggers, separated by a semicolon (;), specified in the LOG4J2.xml. Troubles with modeShape start, indexing, cassandra connection etc: es_details, org.elasticsearch.gateway, org.elasticsearch.action, org.elasticsearch.transport, org.elasticsearch.discovery.zen.ping.unicast, org.modeshape, org.infinispan, net.dataforte.cassandra, com.genesys.modeshape, com.genesys.modeshape.jcr.index.elasticsearch, org.elasticsearch, org.modeshape.jcr.spi.index.provider Troubles with config-server connection, configuration etc: com.genesyslab.wmcbcore, PSDK, com.genesyslab.platform, com.genesys.knowledge.server.configuration Low-level http logging: com.genesys.knowledge.cms.web.filters.RequestLoggingFilter Troubles with documents, categories or another content workflow: com.genesys.knowledge.cms.repository, com.genesys.knowledge.cms.service, com.genesys.jcr Troubles with GKS communication, synchronization, scheduling etc: com.genesys.knowledge.cms.gks, com.genesys.knowledge.cms.aop.GksMonitor, com.genesys.knowledge.cms.scheduling Troubles with UCS-data workflow: com.genesys.knowledge.manager, com.genesys.knowledge.manager.aop.ManagerRespositoryMonitor

Name	Description	Value
	logs.	
time_format	Specifies how to represent, in a log file, the time when an application generates log records. A log record's time field in the ISO 8601 format looks like this: 2001-07-24T04:58:10.123	**Default:** time **Valid Values:** **[+] time** The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. **[+] locale** The time string is formatted according to the system's locale. **[+] ISO8601** The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds. **Changes Take Effect:** Immediately
time_convert	Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since 00:00:00 UTC, January 1, 1970.	**Default:** local **Valid Values:** **[+] local** The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. **[+] utc** The time of log record generation is expressed as Coordinated Universal Time (UTC).

Name	Description	Value
		Changes Take Effect: Immediately

Load-Balancing Configuration

Deploying a Cluster

> **Important**
> Whenever you deploy a Knowledge Center Server instance, you must configure a Knowledge Center Cluster, even if you only plan on having one server.

Knowledge Center Cluster stores all of the settings and data that are shared by each of the Knowledge Center Server instances that reside within it. This makes it pretty easy to add additional servers as your knowledge needs grow.

Knowledge Center Cluster also serves as the entry point to all client requests sent to Knowledge Center Servers. The cluster application in Genesys Administrator needs to be configured to point to the host and port of the load balancer that will distribute these requests among your Knowledge Center Servers.

> **Important**
> If you only have one server deployed in your cluster, you can configure the cluster application to point directly to the host and port of that server.

Configuring Your Load-Balancer Solution

Let's take a look at how you might configure your load balancer to distribute requests between servers. This sample uses an Apache load balancer.

> **Important**
> Genesys recommends that you use a round-robin approach to balancing.

> **Important**
> If you need more information about load balancing in a Genesys environment, the

> Genesys Web Engagement Load Balancing page provides some useful background information.

Prerequisites

- Several Knowledge Center Servers should be installed. These servers will be used as cluster nodes (node1, node2, node3, and so on)
- You must have a Genesys Administrator application of type **Application Cluster**
- All Knowledge Center Server applications should be connected to the application cluster

Start

1. Install the Apache HTTP Server (http://httpd.apache.org/). The port and host of the installed load balancer should be used in the **Application Cluster** application in Genesys Administrator.
2. Enable these modules (in the *./conf/https.conf* configuration file):
   - *LoadModule proxy_module modules/mod_proxy.so*
   - *LoadModule proxy_ajp_module modules/mod_proxy_ajp.so*
   - *LoadModule proxy_balancer_module modules/mod_proxy_balancer.so*
   - *LoadModule proxy_connect_module modules/mod_proxy_connect.so*
   - *LoadModule proxy_ftp_module modules/mod_proxy_ftp.so*
   - *LoadModule proxy_http_module modules/mod_proxy_http.so*
   - *LoadModule proxy_scgi_module modules/mod_proxy_scgi.so*
3. Configure your proxy settings (http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html):

```
# Proxy
# ProxyPass / balancer://''knowledge_cluster''/
stickysession=JSESSIONID|jsessionid nofailover=Off
ProxyPass / balancer://''knowledge_cluster''/
<Proxy balancer://test_cluster>
    BalancerMember http://host_node_1:port_node_1 route=node1
    BalancerMember http://host__node_2:port_node_1 route=node2
</Proxy>
ProxyRequests On
<Proxy *>
    AddDefaultCharset off
    Order deny,allow
    Allow from all
    #Allow from .example.com
</Proxy>
```

4. In each node in your Jetty server configuration, set *./etc/jetty.xml* like this:

```
<Set name="sessionIdManager">
    <New id="hashIdMgr" class=
"org.eclipse.jetty.server.session.HashSessionIdManager">
        <Set name="workerName">node1</Set>
    </New>
```

```
</Set>
```

5. Restart Apache, then restart all of your nodes
6. All requests that are sent to Apache will be distributed to your cluster nodes. The current configuration supports stickysession mode based on JSESSIONID (http://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html#stickyness_implementation)

End

Here are couple of sample requests:

- Request to a specific node: *http://host_node_1:port_node_1/gks-sample-ui*
- Request to the cluster, which will be distributed to any appropriate node: *http://host_load_balacer:port_load_balancer/gks-sample-ui*

Security

Genesys Knowledge Center adheres to the standards described in the Open Web Application Security Project (OWASP) Top 10 — see the OWASP website for details about the Top 10 — and has adopted several methods of ensuring security, for example:

- Errors are logged locally to prevent information leakage through API requests.
- User sessions have a timeout option.
- Cross Site Request Forgery Protection

> **Important**
>
> Genesys performs security testing with OWASP Zed Attack Proxy (ZAProxy) to make sure the Genesys Knowledge Center solution is invincible to known attacks.

Genesys Knowledge Center includes additional security configurations that can be used with your Knowledge Center installation:

- Secure HTTP Communication — Load SSL certificates and configure Jetty to expose Knowledge Center API securely.
- Transport Layer Security (TLS) with Genesys Server — Configure TLS for connection between Knowledge Center servers and other Genesys server.
- Authentication — Enable authentication for the Knowledge Center Server and the CMS.
- Cassandra Security — Enable secure communication between Cassandra nodes and Knowledge Center CMS

Secure HTTP Communication

The Jetty web server supplied with the Genesys Knowledge Center Server and CMS includes a pre-configured, self-signed certificate. This allows you to use HTTPS out of the box in a sandbox deployment. In common case, you should use a certificate issued by a third-party Certificate Authority. The procedures on this page provide examples of ways to load SSL certificates and configure Jetty. These examples may vary depending on your environment.

> **Important**
>
> If Genesys Knowledge Center Server running in HTTPS mode - option in gks-security section should be configured in Genesys Knowledge Center CMS application for successful connection during publishing documents.

Loading an SSL Certificate and Private Key into a JSSE Keystore

> **Important**
>
> In a development environment, you can use self-signed certificates, but in a production environment you should use a certificate issued by a third-party Certificate Authority, such as VeriSign.

Prerequisites

- An SSL certificate, either generated by you or issued by a third-party Certificate Authority. For more information on generating a certificate, see http://wiki.eclipse.org/Jetty/Howto/Configure_SSL.

Start

1. Depending on your certificate format, do **one** of the following:
   - If your certificate is in PEM form, you can load it to a JSSE keystore with the keytool using the following command:
     ```
     keytool -keystore keystore -importcert -alias alias -file certificate_file
     -trustcacerts
     ```
     Where:

     keystore is the name of your JSSE keystore.

     alias is the unique alias for your certificate in the JSSE keystore.

     certificate_file is the name of your certificate file. For example, `jetty.crt`.

- If your certificate and key are in separate files, you must combine them into a PKCS12 file before loading it to a keystore.

  1. Use the following command in openssl to combine the files:
     `openssl pkcs12 -inkey private_key -in certificate -export -out pkcs12_file`

     Where:

     private_key is the name of your private key file. For example, `jetty.key`.

     certificate is the name of your certificate file. For example, `jetty.crt`.

     pkcs12_file is the name of the PKCS12 file that will be created. For example, `jetty.pkcs12`.

  2. Load the PKCS12 file into a JSSE keystore using keytool with the following command:
     `keytool -importkeystore -srckeystore pkcs12_file -srcstoretype store_type -destkeystore keystore`

     Where:

     pkcs12_file is the name of your PKCS12 file. For example, `jetty.pkcs12`.

     store_type is the file type you are importing into the keystore. In this case, the type is PKCS12.

     keystore is the name of your JSSE keystore.

> **Important**
>
> You will need to set two passwords during this process: keystore and truststore. Make note of these passwords because you will need to add them to your Jetty SSL configuration file.

End

Configuring Jetty

Start

1. Open the Jetty SSL configuration file in a text editor: ***jetty_installation*/etc/jetty-ssl.xml**.
2. Find the **<New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">** element and update the passwords:

```
<Configure id="sslContextFactory" class=
 "org.eclipse.jetty.util.ssl.SslContextFactory">
   <Set name="KeyStorePath"><path to keystore><Property name=
 "jetty.base"
 default="." />/<Property name="jetty.keystore" default=
 "etc/keystore"/></Set>
   <Set name="KeyStorePassword">OBF:<obfuscated_keystore_password>
 <Property name="jetty.keystore.password"
 default="OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4"/></Set>
   <Set name="KeyManagerPassword">OBF:
 <obfuscated_keymanager_password><Property name=
 "jetty.keymanager.password"
 default="OBF:1u2u1wml1z7s1z7a1wnl1u2g"/></Set>
   <Set name="TrustStorePath">"><path to truststore><Property name=
 "jetty.base" default="." />/<Property name="jetty.truststore"
 default="etc/keystore"/></Set>
   <Set name="TrustStorePassword"> OBF:
 <obfuscated_truststore_password><Property name=
 "jetty.truststore.password"
 default="OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4"/></Set>
   <Set name="EndpointIdentificationAlgorithm"></Set>
   <Set name="NeedClientAuth"><Property name=
 "jetty.ssl.needClientAuth" default="false"/></Set>
   <Set name="WantClientAuth"><Property name=
 "jetty.ssl.wantClientAuth" default="false"/></Set>
   <Set name="ExcludeCipherSuites">
     <Array type="String">
       <Item>SSL_RSA_WITH_DES_CBC_SHA</Item>
       <Item>SSL_DHE_RSA_WITH_DES_CBC_SHA</Item>
       <Item>SSL_DHE_DSS_WITH_DES_CBC_SHA</Item>
       <Item>SSL_RSA_EXPORT_WITH_RC4_40_MD5</Item>
       <Item>SSL_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
       <Item>SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
       <Item>SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</Item>
     </Array>
   </Set>
```

Note: You can run Jetty's password utility to obfuscate your passwords. See http://www.eclipse.org/jetty/documentation/current/configuring-security-secure-passwords.html.

3. Save your changes.

End

Choosing a Directory for the Keystore

The keystore file in the example above is given relative to the Jetty home directory. For production, you should keep your keystore in a private directory with restricted access. Even though the keystore has a password, the password may be configured into the runtime environment and is vulnerable to theft.

You can now start Jetty the normal way (make sure that **jcert.jar**, **jnet.jar** and **jsse.jar** are on your classpath) and SSL can be used with a URL, such as https://*your_IP*:8743/

Transport Layer Security (TLS) with Genesys Servers

Genesys Knowledge Center supports the Transport Layer Security (TLS) protocol to secure data exchanged with other Genesys components. For details about TLS, see the Genesys 8.1 Security Deployment Guide. You can configure TLS for Knowledge Center by completing the procedures on this page.

Configuring TLS for Genesys Servers

To configure the TLS parameters for Genesys servers, see Introduction to Genesys Transport Layer Security.

Configuring TLS for Genesys Knowledge Center Server

To enable TLS support for the Genesys Knowledge Center Server, you must do the following:

1. Have properly installed a trusted certificates for the Genesys server. For more information, please see Certificate Generation and Installation.
2. Configure TLS options for the Genesys Knowledge Center Server application.
3. Configure the appropriate connections between the Genesys Knowledge Center Server application and the necessary Genesys servers through secure ports. For example, by setting a secure Config Server port in the *Server Installation Folder*/server/setenv.bat file in the **PRIMARY_CFGSERVER_PORT** variable.

Configuring Secure Connections to Configuration Server

To configure a secured connection from Genesys Knowledge Center Server to Configuration Server use the following TLS-related configuration options in the **setenv.bat/sh** configuration:

Parameter Name	Acceptable Values	Purpose
PRIMARY_CFGSERVER_CONNECTION_MODE	TLS_UPGRADE or UNSECURED by default	Set this option to enable secured connection **Important** Incorrect setting of this parameter can lead to inability to establish a connection with the server
PROVIDER	PEM, JKS, MSCAPI, PKCS11	Type of used security provider

Parameter Name	Acceptable Values	Purpose
TRUSTED_CA	valid file name (including path)	Path to trusted CA PEM file or JKS truststore file or SHA-1 Thumbprint for MSCAPI storage. Specifies the name of the trusted store file which holds the public certificate to verify the server. Applicable for PEM and JKS trusted storage types only.
TRUSTSTORE_PASSWORD	n/a	Password for the JKS trusted storage. Provide password only if trusted CA is in the JKS format.
In case of enabled mutual TLS, configure the following options:		
CERTIFICATE	n/a	Client certificate file in PEM format or JKS keystore file or SHA-1 Thumbprint for MSCAPI storage.
PRIVATE_KEY	n/a	Unencrypted private key in PEM format or Certificate SHA-1 Thumbprint for MSCAPI storage. Ignored for JKS storage.
KEYSTORE_PASSWORD	n/a	Provide password if key storage is in the JKS format.
KEYENTRY_PASSWORD	n/a	Provide password if private key encrypted by its own password.

Configuring TLS Options

For connections with other Genesys servers, configure **Connections** of the Knowledge Center Cluster (8.5.1+) application through secure ports. The Genesys Knowledge Center Server Node includes the following TLS-related configuration options in its security section.

Parameter Name	Acceptable Values	Purpose
tls	Boolean value. Possible values are "1"/"0", "yes"/"no", "on"/"off", "true"/"false". Example: • "tls=1"	Client: 1 - perform TLS handshake immediately after connecting to server. 0 – do not turn on TLS immediately but autodetect can still work.
provider	"PEM", "MSCAPI", "PKCS11" Not case-sensitive. Example: • "provider=MSCAPI"	Explicit selection of security provider to be used. For example, MSCAPI and PKCS11 providers can contain all other parameters in their internal database. This parameter allow configuration of TLS through security provider tools.
certificate	PEM provider: path to a X.509 certificate file in PEM format.	Specifies location of X.509 certificate to be used by

Parameter Name	Acceptable Values	Purpose
	Path can use both forward and backward slash characters. MSCAPI provider: thumbprint of a certificate – string with hexadecimal SHA-1 hash code of the certificate. Whitespace characters are allowed anywhere within the string. PKCS11 provider: this parameter is ignored. Examples: • "certificate= C:\certs\client-cert-3-cert.pem" • "certificate=A4 7E A6 E4 7D 45 6A A6 2F 15 BE 89 FD 46 F0 EE 82 1A 58 B9"	application. MSCAPI provider keeps certificates in internal database and can identify them by hash code; so called thumbprint. In Java, PKCS#11 provider does not allow selection of the certificate; it must be configured using provider tools. **Note:** When using autodetect (upgrade) TLS connection, this option MUST be specified in application configuration, otherwise Configuration Server would return empty TLS parameters even if other options are set.
certificate-key	PEM provider: path to a PKCS#8 private key file without password protection in PEM format. Path can use both forward and backward slash characters. • MSCAPI provider: this parameter is ignored; key is taken from the entry identified by "certificate" field. • PKCS11 provider: this parameter is ignored. Examples: • "certificate-key= C:\certs\client-cert-3-key.pem"	Specifies location of PKCS#8 private key to be used in pair with the certificate by application. MSCAPI provider keeps private keys paired with certificates in internal database. In Java, PKCS#11 provider does not allow selection of the private key; it must be configured using provider tools.
trusted-ca	PEM provider: path to a X.509 certificate file in PEM format. Path can use both forward and backward slash characters. MSCAPI provider: thumbprint of a certificate – string with hexadecimal SHA-1 hash code of the certificate. Whitespace characters are allowed anywhere within the string. PKCS11 provider: this parameter is ignored. Examples: • "trusted-ca= C:\certs\ca.pem" • "trusted-ca=A4 7E A6 E4 7D 45 6A A6 2F 15 BE 89 FD 46 F0 EE 82 1A 58 B9"	Specifies location of a X.509 certificate to be used by application to validate remote party certificates. The certificate is designated as Trusted Certification Authority certificate and application will only trust remote party certificates signed with the CA certificate. MSCAPI provider keeps CA certificates in internal database and can identify them by hash code; so called thumbprint. In Java, PKCS#11 provider does not allow selection of the CA certificate; it must be configured using provider tools.

Parameter Name	Acceptable Values	Purpose
tls-mutual	Boolean value. Possible values are "1"/"0", "yes"/"no", "on"/"off", "true"/"false". Example: • "tls-mutual=1"	Has meaning only for server application. Client applications ignore this value. When turned on, server will require connecting clients to present their certificates and validate the certificates the same way as client applications do.
tls-crl	All providers: path to a Certificate Revocation List file in PEM format. Path can use both forward and backward slash characters. Example: • "tls-crl= C:\certs\crl.pem"	Applications will use CRL during certificate validation process to check if the (seemingly valid) certificate was revoked by CA. This option is useful to stop usage of leaked certificates by unauthorized parties.
tls-target-name-check	"host" or none. Not case-sensitive. Example: • "tls-target-name-check=host"	When set to "host", enables matching of certificate's Alternative Subject Name or Subject fields against expected host name. PSDK supports DNS names and IP addresses as expected host names.
cipher-list	String consisting of space-separated cipher suit names. Information on cipher names can be found online. Example: • "cipher-list= TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA"	Used to calculate enabled cipher suites. Only ciphers present in both the cipher suites supported by security provider and the cipher-list parameter will be valid.
fips140-enabled	Boolean value. Possible values are "1"/"0", "yes"/"no", "on"/"off", "true"/"false". Example: • "fips140-enabled=1"	PSDK Java: when set to true, effectively is the same as setting "provider=PKCS11" since only PKCS11 provider can support FIPS-140. If set to true while using other provider type, PSDK will throw exception.
sec-protocol	String value. Possible values are "SSLv23", "SSLv3", "TLSv1", "TLSv11", "TLSv12". Example: • "sec-protocol=TLSv1"	Starting with PSDK release 8.5.1, an application can specify the exact protocol to send and accept secure connection requests on one or more of its connections.

See Configuring Trusted Stores below for details about configuration for a specific type of store (PEM,

JKS, MSCAPI).

Configuring Trusted Stores

PEM Trusted Store

PEM stands for "Privacy Enhanced Mail", a 1993 IETF proposal for securing email using public-key cryptography. That proposal defined the PEM file format for certificates as one containing a Base64-encoded X.509 certificate in specific binary representation with additional metadata headers.

PEM certificate trusted store works with CA certificate from an X.509 PEM file. It is a recommended trusted store to work on Linux systems.

Complete the steps below to work with the PEM certificate trusted store:

Start

1. Configure TLS for Genesys servers to use certificates signed by CA certificate **certificateCA.crt**.
2. Place the trusted CA certificate in PEM format on the Genesys Knowledge Center Server application host. To convert a certificate of another format to .pem format you can use the OpenSSL tool. For example:
   - Convert a DER file (.crt .cer .der) to PEM:
     ```
     openssl x509 -inform der -in certificateCA.crt -out certificateCA.pem
     ```
   - Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM:
     ```
     openssl pkcs12 -in certificateCA.pfx -out certificateCA.pem -nodes
     ```
     You can add **-nocerts** to only output the private key or add **-nokeys** to only output the certificates.
3. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Knowledge Center Server application.
4. Click the **Options** tab and navigate to the security section.
5. Set the **trusted-ca-type** option to PEM.
6. Set the **trusted-ca** option to the path and file name for your trusted CA in PEM format on the Genesys Knowledge Center Server application host.
7. Click **Save & Close**.

End

JKS Trusted Store

A Java KeyStore (JKS) is a repository of security certificates used, for instance, in SSL/TLS encryption. The Java Development Kit provides a tool named keytool to manipulate the keystore.

Complete the steps below to work with the JKS certificate trusted store:

Start

1. Configure TLS for Genesys servers to use certificates signed by CA certificate **certificateCA.crt**.

2. Import the CA certificate to an existing Java keystore using keytool:
   - Run the `keytool` command with option `-alias` set to `root`:
     ```
     keytool -import -trustcacerts -alias root -file certificateCa.crt -keystore
     /path/to/keysore/keystore.jks
     ```
   - Enter the keystore password in command line prompt - for example:
     ```
     Enter keystore password: somepassword
     ```
3. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Knowledge Center Server application.
4. Click the **Options** tab and navigate to the security section.
5. Set the **trusted-ca-type** option to JKS.
6. Set the **trusted-ca** option to the path and file name for your JKS trusted storage type on the Genesys Knowledge Center Server application host.
7. Set the **trusted-pwd** option to the password defined for your keystore in Step 2.
8. Click **Save & Close**.

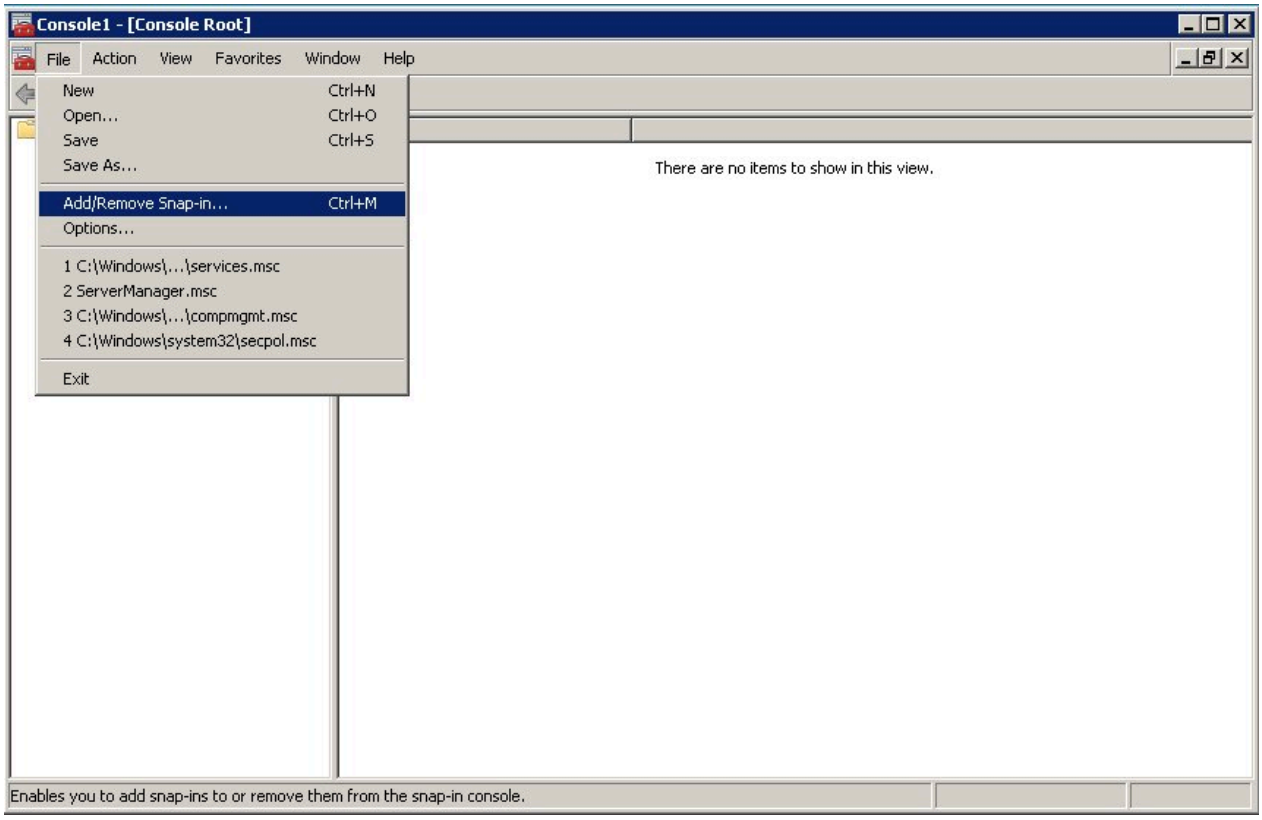
End

MSCAPI Trusted Store

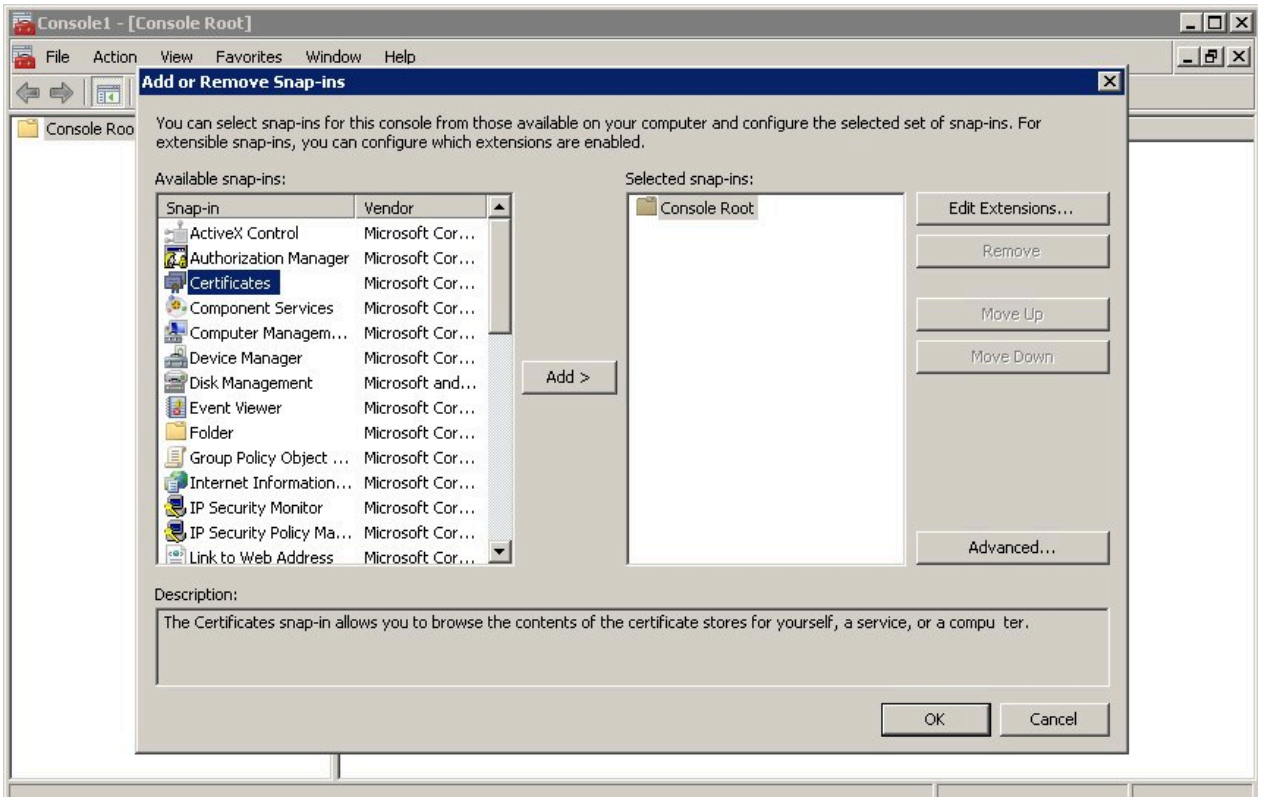
Complete the steps below to work with the MSCAPI certificate trusted store:

Start

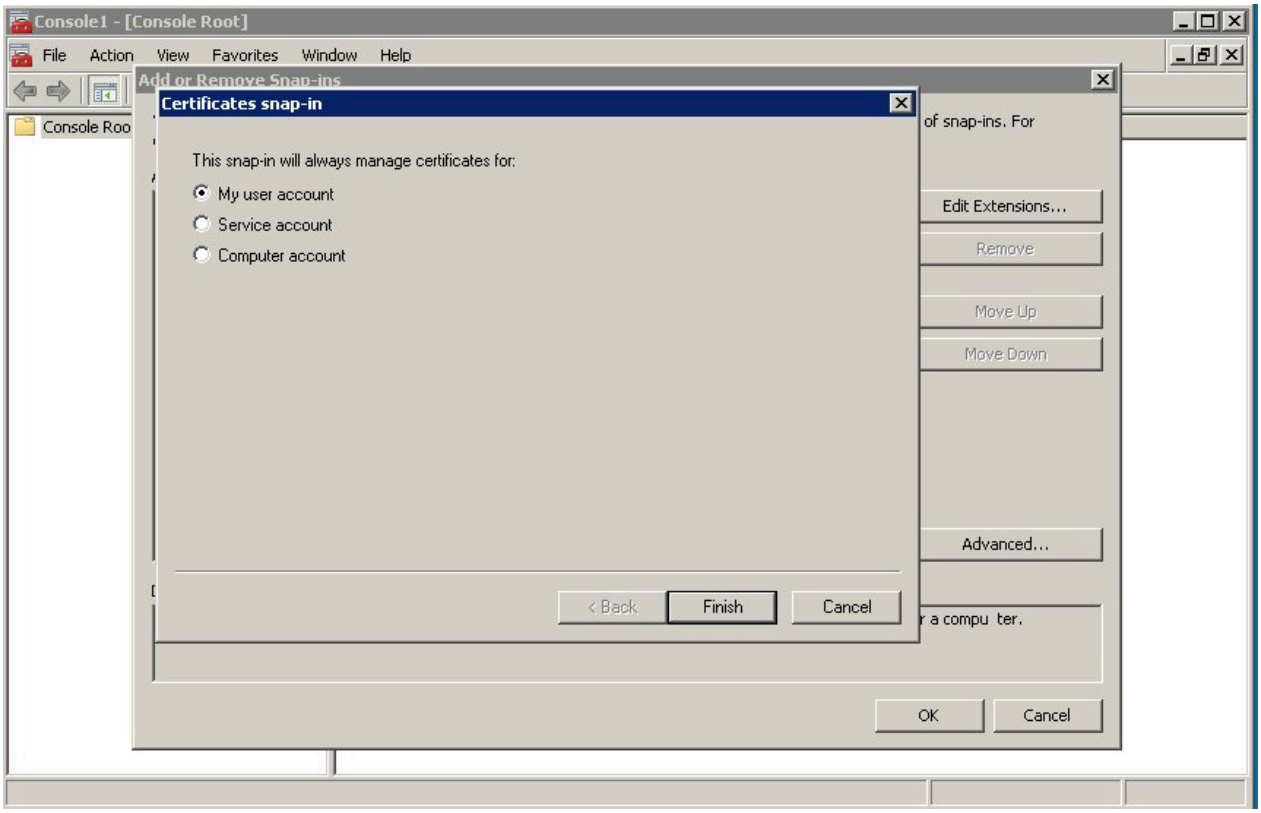
1. Configure and tune TLS for Genesys servers to use certificates signed by the same CA.
2. If the Knowledge Center Server is running on a different host, copy the trusted CA certificate to this host.
3. Import the CA certificate to WCS via Certificates Snap-in on the Knowledge Center Server host by launching the MMC console. Enter `mmc` at the command line.
4. Select **File > Add/Remove Snap-in...** from the main menu.



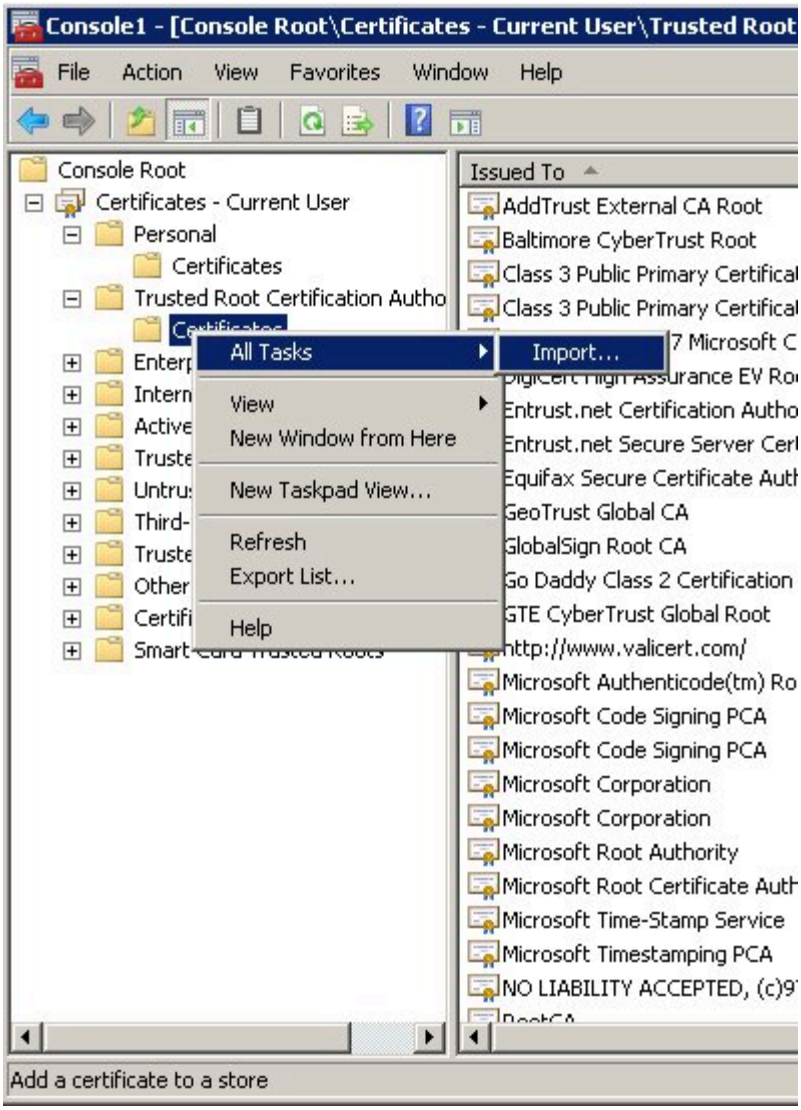
5. Select **Certificates** from the list of available snap-ins and click **Add**.



6. Select the account to manage certificates for and click **Finish**. It is important to place certificates under the correct Windows account. Some applications are run as services under the Local Service or System account, while others are run under user accounts. The account chosen in MMC must be the same as the account used by the application which certificates are configured for, otherwise the application will not be able to access this WCS storage.



7. Click **OK**.
8. Import a certificate. Right-click the "Trusted Root Certification Authorities/Certificates" folder and choose `All Tasks > Import...` from the context menu. Follow the steps presented by the Certificate Import Wizard. Oonce finished the imported certificate appears in the certificates list.



9. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Knowledge Center Server application.
10. Click the **Options** tab and navigate to the security section.
11. Set the **trusted-ca-type** option to MSCAP.
12. Click **Save & Close**.

End

Configuring TLS for a server running Windows

By default, Genesys Knowledge Server as a Windows Service runs without a TLS connection. To configure a secure connection from Genesys Knowledge Center Server to a Configuration Server while running as a Windows Service you need to update the installed default service for Genesys

Knowledge Center Server.

In order to do this you will need to:

1. Remove Genesys Knowledge Center Server Windows Service, which was configured in the installation package.
    1. Run the Windows Command Prompt (cmd.exe)
    2. Go to *<Knowledge Center Server installation folder>/server*
    3. Run the next command: **server.bat** remove
2. Configure a secure connection settings in **setenv.bat** to Genesys Configuration Server, as described in Configuring Secure Connections to Configuration Server.
3. Re-install the Windows Service for Genesys Knowledge Center Server, now with the secure connection configured to Genesys Configuration Server.
    1. Run Windows Command Prompt (cmd.exe)
    2. Go to *<Knowledge Center Server installation folder>/server*
    3. Run the next command: **server.bat** install

Authentication

You can enable secure communications with the Management and Reporting REST APIs by completing the procedures below to implement authentication. If you do enable authentication, then all API clients must use the authentication scheme and credentials. Three common clients of the API are the Genesys Knowledge Center Plugin for Administrator, Genesys Knowledge Center Plugin for Workspace Desktop Edition and Genesys Knowledge Center CMS.

Configuring Authentication in Genesys Knowledge Center

Complete the steps below to enable authentication for the Management and Reporting REST APIs.

Start

1. In Genesys Administrator, navigate to **Provisioning > Environment > Applications**, select the Knowledge Center Cluster application, and click **Edit...**.
2. Click the **Options** tab and scroll down to the **[security]** section.
3. Set the following options:
   - auth-scheme
   - user-id
   - password
4. Click **Save & Close**.

End

Cassandra Security

Unauthorized access to Cassandra data is possible at several points:

- Direct access via "standard" interfaces: Thrift and CQL
- Access to data traveling through the network
- Access to data files that Cassandra stores on hard drives

Cassandra's default configuration provides mechanisms to secure direct interfaces (through authentication and authorization) and network traffic (through the use of TLS). The data stored on hard drives can be secured either by third-party commercial offerings or with some development investments.

Securing access interfaces

You can secure your access interfaces based on an authentication and authorization scheme. In other words, Cassandra needs to know:

- Who is trying to access the system
- Whether they are allowed to access the system at all
- If so, which data they should have access to

With the default setup, anybody is allowed to access all the data.

Authentication

Authentication (who) is managed by the authenticator parameter in the `cassandra.yaml` file.

Procedure

Start

1. Locate Cassandra configuration file *Cassandra installation directory/conf/cassandra.yaml*.
2. Change the authenticator option in the cassandra.yaml file to PasswordAuthenticator.

By default, the authenticator option is set to AllowAllAuthenticator.
`authenticator: PasswordAuthenticator`

1. Increase the replication factor for the system_auth keyspace to N (number of nodes).

If you use the default, 1, and the node with the lone replica goes down, you will not be able to log into the cluster because the system_auth keyspace was not replicated.

1. Restart the Cassandra client.

2. Start cqlsh using the superuser name and password.

```
./cqlsh -u cassandra -p cassandra
```

1. Create another superuser, not named cassandra. This step is optional but highly recommended.
2. Log in as that new superuser.
3. Change the cassandra user password to something long and incomprehensible, and then forget about it. It won't be used again.
4. Take away the cassandra user's superuser status.
5. Use the CQL statements listed previously to set up user accounts and then grant permissions to access the database objects.
6. Set the new user name and password to the values of the cassandra-keyspace **userName** and **password** options for the Knowledge Center Cluster application.

End

For more information about permissions see:

- Apache Cassandra Authentication
- Apache Cassandra Authorization

Knowledge Center Cluster Configuration

Prerequisites

The Knowledge Center Cluster applications are created and configured

Procedure

Start

For all Cassandra Resource Access Points:

1. Open the Knowledge Center Cluster configuration option, cassandra-keyspace section.
2. Set the userName option to the name of an already-created user.
3. Set the password option to the user's password.

End

Securing Network Traffic

The client-to-node and node-to-node traffic in your Cassandra deployment may require protection. They can both be secured by using SSL (Secure Sockets Layer) encryption.

Client-to-Node Encryption

Client-to-node encryption uses SSL to protect data that is traveling from client machines (Knowledge Center CMS nodes) to a database cluster. It does this by establishing a secure channel between the client and the coordinator node.

Prerequisites

- You must install Java Cryptography Extension (to enable 256-bit encryption).
- All nodes must have all of the relevant SSL certificates. See Preparing server certificates (Cassandra documentation).

> **Important**
>
> The Oracle Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6 must be installed when enabling client-to-node encryption.
>
> 1. Download the JCE:
>
> - JAVA 8
> - JAVA 7
> - JAVA 6
>
> - Unzip the downloaded file
> - Place the two jars from the zip file into `<java_jre_install_dir>/lib/security/` if running the jre or `<java_jdk_install_dir>/jre/lib/security` if running the jdk

Start

1. Configuring Cassandra nodes:
    a. On each Cassandra node edit Cassandra installation directory/conf/cassandra.yaml
    b. Set the following options under the client_encryption_options section:

```
# enable or disable client/server encryption.
 client_encryption_options:
     enabled: true
     optional: false
     keystore: <path to your JKS keystore, for example  c:\genesys\keystore.jks >
     keystore_password:  <password for JKS keystore>
     require_client_auth: false
     truststore: <path to your JKS trustore, for example  c:\genesys\truststore.jks>
     truststore_password: <password for JKS truststore>
     protocol: TLS
     algorithm: SunX509
     store_type: JKS

 cipher_suites:[TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_
```

3. Configuring Genesys Knowledge Center CMS:
    a. Navigate to the Application in Genesys Administrator
    b. Open Genesys Knowledge Center Cluster object
    c. Navigate to options tab
    d. Set the following options in the cassandra-security section:
        - enable-ssl = true
        - truststore-path = <path to your JKS trustore, for example c:\genesys\truststore.jks>
        - truststore-password = <password for JKS truststore>

        > **Important**
        > You can define truststore-path and truststore-password options in the Genesys Knowledge Center CMS application options in case you use different paths and passwords on every host.

    e. Open Cassandra Resource Access Point application object
    f. Open the properties for the port with an ID of default.
    g. Set this port to secured.

End

Node-to-Node Encryption

Node-to-node encryption uses SSL to protect data being transferred between cluster nodes. This includes node-to-node gossip communication.

Prerequisites

- You must install Java Cryptography Extension (to enable 256-bit encryption).
- All nodes must have all of the relevant SSL certificates. See Preparing server certificates.

Procedure

Start

1. On each Cassandra node edit Cassandra installation directory/conf/cassandra.yaml
2. Set the following options under the server_encryption_options section:

```
server_encryption_options:
    internode_encryption: all
                          # all—Cassandra encrypts all internodal traffic
                          # dc—Cassandra encrypts all traffic between datacenters
                          # rack—Cassandra encrypts all traffic between racks
    keystore:<path to your JKS keystore, for example  c:\genesys\keystore.jks>
    keystore_password:<password for JKS keystore>
    truststore:<path to your JKS trustore, for example  c:\genesys\truststore.jks >
    truststore_password:<password for JKS truststore>
    protocol: TLS
    algorithm: SunX509
    store_type: JKS
    cipher_suites: [TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA]
    require_client_auth: false
```

End

IP Geolocation

> **Important**
> Collecting information about a customer's location and the way it is stored may be subject to regulations or restrictions within your country or countries you operate in. Please check with your national legislation to ensure you are not in violation. This feature can be turned off if needed.

What is Geolocation

Geolocation is the identification or estimation of the real-world geographic location of an object. IP geolocation is the process geolocation that is based on the client's IP address as the into a physical location.

When geolocation is enabled it allows Genesys Knowledge Center to store client IP address and its relevant geolocation information in the History index. Stored information is mostly useful in the reports allowing to understand regional differences in the knowledge usage by the agent and customer.

IP geolocation is inherently imprecise. Locations are often near the center of the population. Any location provided should not be used to identify a particular address or household.

The actual location of the IP address is likely within some radius area around the latitude and longitude coordinates.

Configuring IP Geolocation

The Administrator is able to configure the precision of the geolocation for the Knowledge Center cluster (cluster/reporting/geo) as:

- off - disable the IP geolocation functionality: both IP and longitude and latitude are empty for historical records
- IP - only IP address is stored, Knowledge Center is not identifying geographic location of the customer
- country - IP and country name longitude and latitude of country are stored
- city (default) - IP, country name and and city longitude and latitude are stored

Described levels are defined in the Knowledge Center Cluster option geo that is located in section reporting.

Visualize the Geolocation Information

This stored data is used in the Kibana to visualize:

- a geo-map with requests heat indicators
- the top 10 countries

Activity Heatmap

How to Update the Geolocation Database

Geolocation functionality requires a special database to translate a client's IP address to the geographical location of the customer. When Genesys Knowledge Center Server is installed it provides the MaxMind GeoLite2 City database stored in **<installation directory>\linguatools\ geoip folder**. The folder storing the database can be changed in the **gks.yml** file:

```
...
path.geoip : <IP folder>/GeoIP/GeoLiteCity.dat
...
```

To update the database you need to:

1. Visit MaxMind GeoLite2 database download page. Note: Knowledge Center is not supposed to work with other MaxMind products (for example, GeoLite or GeoIP, please use GeoLite2).
2. Download the most recent version of the City database.**Note**: Please download database in MaxMind DB binary format.
3. Unarchive the database.
4. Store the database in the folder configured in the `gks.yml` file
5. Restart Genesys Knowledge Center Server.

Important

- These steps needs to be executed for every Knowledge Center Server node in the cluster.
- You can store the geolocation database in shared network location to ensure that it is updated for all Knowledge Center Server nodes.

UTF8

You can configure your Knowledge Center Servers and Knowledge Center CMS to support UTF-8 in Configuration Server, which in turn supports multi-language categories.

Configuring a UTF-8 Connection to Configuration Server

Complete the following steps for your Knowledge Center Servers.

Prerequisites

- Your version of Configuration Server supports UTF-8. For details, see the compliant versions for mandatory components.

Start

1. Navigate to the installation directory for your Knowledge Center Server and open the **setenv.bat** file for Windows — or the **setenv.sh** file for Linux — with a text editor. For example: *Path to installation directory*/server/setenv.bat.
2. Find the following string: **:: set JAVA_OPTS=%JAVA_OPTS% -Dgenesys.cfgServerUseUtf8=true**.
3. Remove the two colons (**::**) at the start. This converts the string from a comment to a command to use UTF-8. Your string should now look like this: **set JAVA_OPTS=%JAVA_OPTS% -Dgenesys.cfgServerUseUtf8=true**
4. Save your changes.

End

Supported Languages

Genesys Knowledge Center supports a search for the right answer in any language. It executes knowledge search involving different natural language processing techniques to come up with the best suggestions for the question asked. The level of the employed functionality depends on the language. Below is the table of the languages that has advanced search processing along with the level of support for every language.

Content Language	Product Version	Natural Language Processing Techniques Level
English	8.5.000+	
Danish	8.5.302+	
Finnish	8.5.302+	
French	8.5.100+	
German	8.5.100+	
	8.5.304+	
Italian	8.5.100+	
Norwegian	8.5.302+	
Portuguese	8.5.100+	
Spanish	8.5.100+	
	8.5.304+	
Swedish	8.5.302+	
	8.5.304+	

For the languages that are not listed in the table above, Genesys Knowledge Center provides keyword-based search over the knowledge.

Proper way of asking the question

Knowledge Center is ready to handle both natural language queries (when you express your question in human language) as well as keyword-based queries.

An example of queries:

```
Natural-language query: How to install Knowledge Center CMS?
```

```
Keyword query: install CMS
```

When it goes to the languages that support the natural-language techniques it doesn't much matters what type of the query is used. But for the language that basic keyword search is executed for (the one that is not listed in the table above) – using of keyword queries will give better confidence assessment comparing to the natural language queries. To mitigate this difference, you need to decrease the "out-of-domain" that will allow less confident results still make it in the final result set.

How To configure out-of-domain limit

Out-of-domain limit defines minimal confidence of the resulting document to appear in result set. It allows you to hide less relevant documents in search results. To change the out-of-domain limit you need to edit option in the properties of particular knowledge base.

Before 8.5.303:

- Follow the Editing Knowledge Base Options instructions of Knowledge Center Administrator Plugin
- Locate out of domain configuration option
- Set it to desired value

After 8.5.303:

- Follow the Editing Knowledge Base Option instructions of Knowledge Center CMS Administration.
- Locate out of domain configuration option
- Set it to desired value

Out of domain is measured from 0 (representing 0% confidence) to 1 (100% confidence). The magic value is 0.75 - it corresponds to the exact match of the search to the document w/o prove of learning signals. Every learning signal (for example, positive relevancy feedback) will improve it further.

Recommended out-of-domain limits:

Natural Language Processing Techniques Level	Out-of-domain value
None	0.20
	0.50
	0.46
	0.38
	0.35

For knowledge bases with multiple languages you need to set to the minimum value. For example, if you have knowledge base with English and French, individual recommended values will be 0.5 and 0.46. The value that is recommended for knowledge base is 0.46.

Knowledge Center in Production

This chapter provides you with information on the Knowledge Center once in production within your environment. It covers following topics:

Monitoring Knowledge Center

Knowledge Center provides access to metrics and other key performance indicators (KPIs).

It also gives you the ability to configure Message Server alarms when a KPI passes its threshold value.

> **Important**
> Monitoring Capability supported by both Knowledge Center Server and Knowledge Center CMS.

Knowledge Center Metrics

Starting with release 8.5.000.13, Knowledge Center integrates with the third-party Metrics Java library to keep track of several Knowledge Center metrics. The Metrics toolkit includes counters, timers, histograms, and gauges.

You will probably want to use Java Management Extensions (JMX) as your main way of reporting on these metrics. We show how to do that here. Or you may want to check out some of the other tools that are available.

You can also use REST—which is helpful for performance testing—or write your metrics to a log file or to the console.

Knowledge Center Alarms

Knowledge Center lets you use tools from the Genesys Management Layer for monitoring and controlling your applications. These tools can be an important factor in improving performance—especially alarms, which let you set performance thresholds for these key metrics:

- Garbage collection latency
- Heap memory usage

Alarm Configuration

Alarm name	Alarm description	Alarm Condition object					Related configuration option
Threshold type	**Selection mode**	**Application type**	**Detect Event ID**	**Cancel Event ID**			
Heap Memory Usage	Defines the heap memory usage threshold value. This is the ratio of used heap memory to maximum heap memory.	predefined	Select by Application Type	Knowledge Center Backend Server	100001	100002	HeapMemoryUsage.threshold
GC Latency	Defines the garbage collection latency threshold value, in milliseconds, in relation to the last time the garbage was collected within the configured time interval.				10005	10006	GcLatency.threshold

Viewing Metrics with JMX

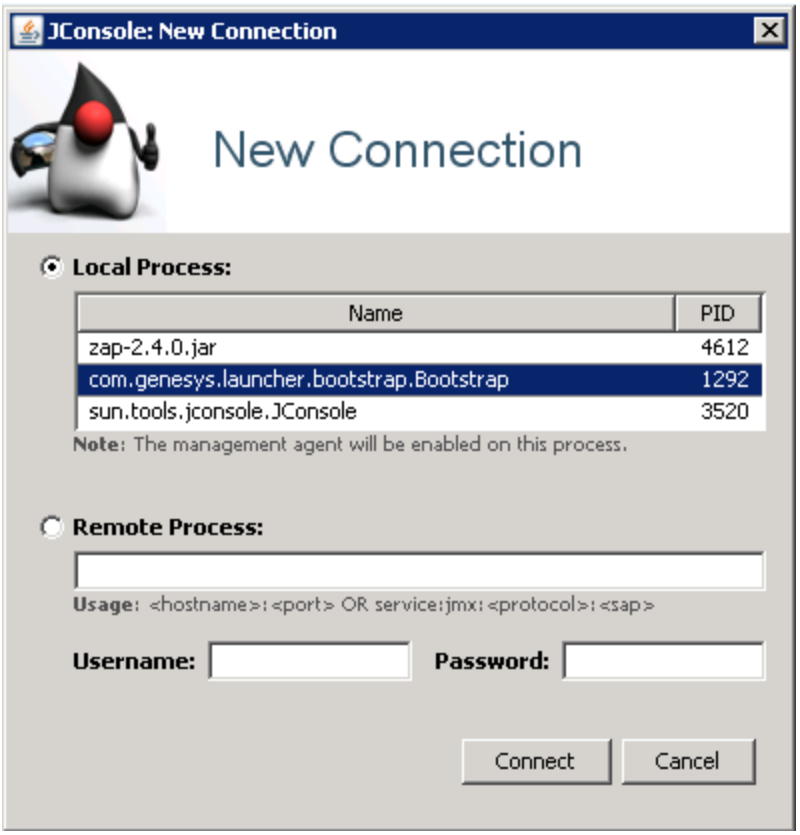
You can use JConsole to view metrics provided by your Knowledge Center Server. To do this, you can start Knowledge Center Server as a:

- Local java process
- Server on a remote host
- Windows service

Once you have connected, you can view your metrics in a JConsole JMX panel.

You may also want to look into some of the other tools that are available for viewing your Knowledge Center metrics.

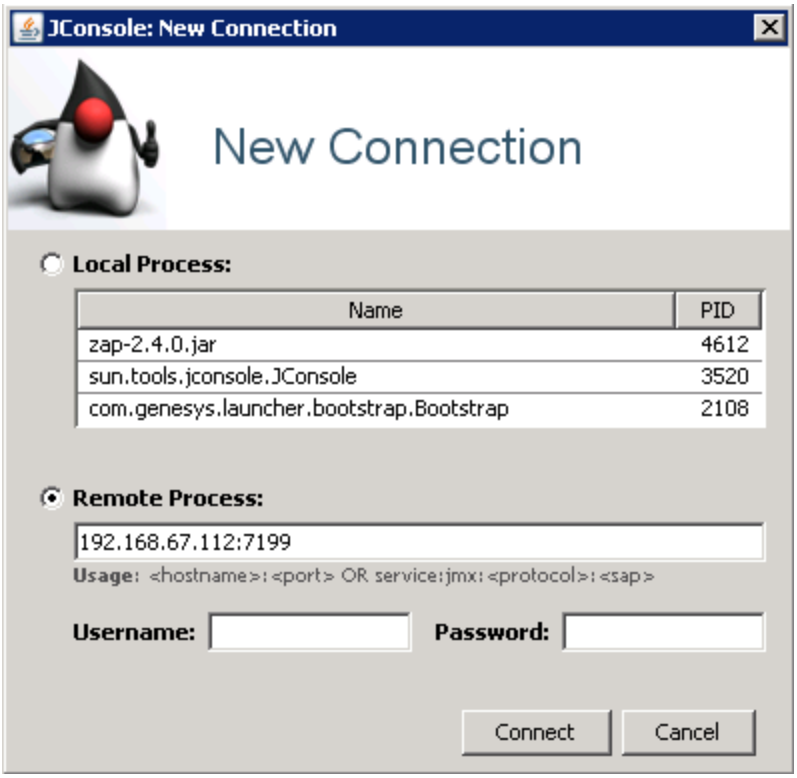
Connect to Knowledge Center started as a **local java process**.



1. Run **jconsole.exe** from the ***jdk*/bin** directory.
2. In the **New Connection** dialog, specify the Knowledge Center launcher java process.

   If the Knowledge Center Server was started via a BAT file in the same host where the JMX console is opened, this launcher process is the **com.genesys.launcher.bootstrap.Bootstrap** process from the **Local Process** list.

Connect to Knowledge Center Server started on a **remote host**.



If the Knowledge Center Server was started remotely as a server, follow these steps:

1. Run **jconsole.exe** from the ***jdk*/bin** directory.
2. Open **setenv.bat** and uncomment all of the lines under the line that begins:

   ```
   :: Uncomment for enabling JMX
   ```

3. Save your changes.
4. Restart the Knowledge Center Server application.
5. Specify *host*:*JMX port* in the **Remote Process** section, as shown in the screenshot on the left.

Connect to Knowledge Center started as a **Windows service**.

If Knowledge Center Server is started as a Windows service, you should first stop the service, reinstall it, and restart it, as shown in these steps:

1. Stop the service.
2. Open **setenv.bat** and find the service name in the line that says `SVC_NAME=`.

3. Run this command to remove the service:

```
server.bat -service <service name> remove
```

4. Open **setenv.bat** and uncomment all of the lines under this one:

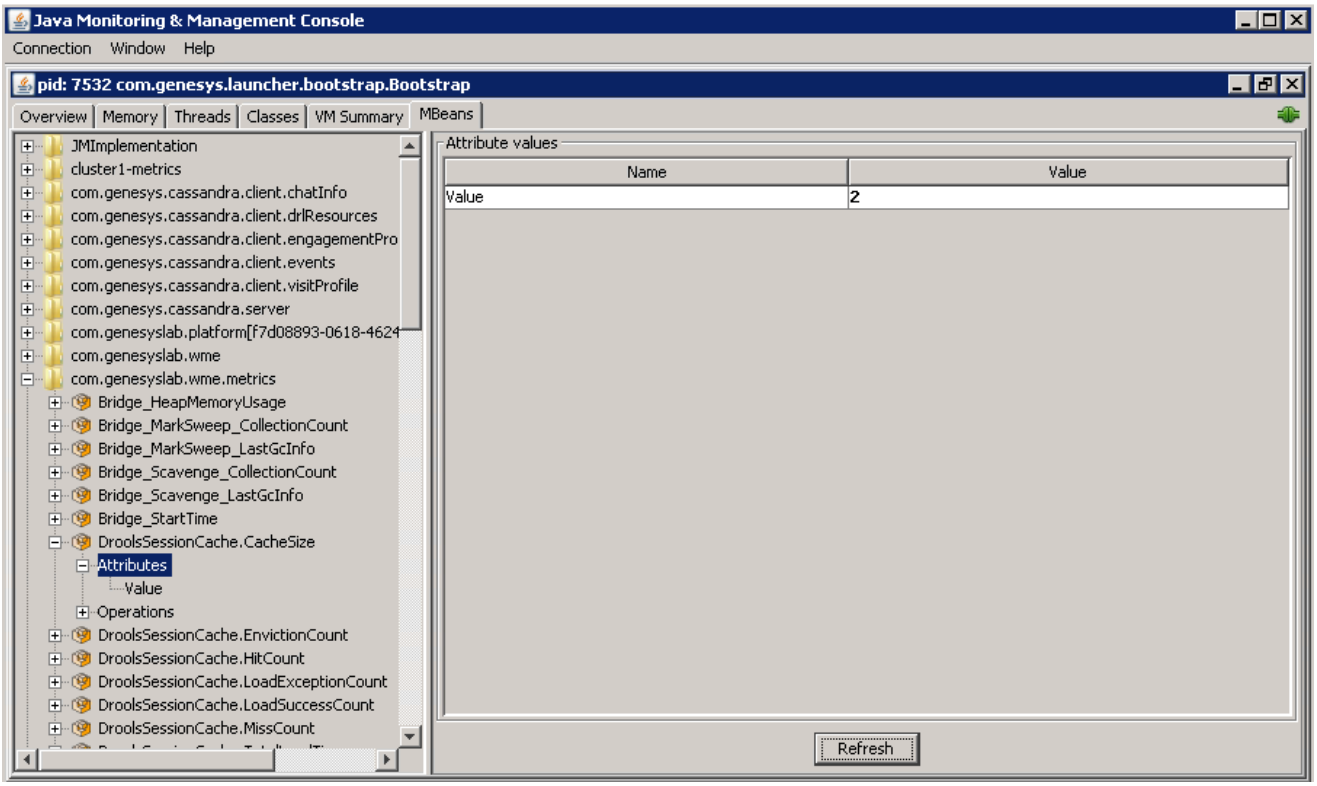
```
:: Uncomment for enabling JMX Remote. Memorize JMX port.
```

5. Save your changes.
6. Run this command to install the service:

```
server.bat -service <service name> install
```

7. Start the service.
8. Specify *host*:*JMX port* in the **Remote Process** section, as shown in the above screenshot.

Open the JMX panel to view the metrics.



1. Click **Connect** in the **New Connection** dialog. The JMX panel opens.
2. Open the **MBeans** tab and expand **com.genesyslab.wme.metrics**. All of the Knowledge Center metrics are there.
3. To refresh the metrics, click **Refresh**.

Other Tools

We have just explained how to use the JConsole tool bundled with Oracle Java (TM) to view your metrics, but there are several other tools you can use to do this:

- The EJTools JMX Browser
- Panoptes
- jManage
- MC4J
- Zabbix

Sample UI

Overview

Knowledge Center comes with a Sample UI, hosted on a sample website, which provides basic access to your installation of Knowledge Center and your configured knowledge base content. You can use it to test and demonstrate what Knowledge Center can do or as an example of how to integrate Knowledge Center access into your existing website.

The Sample UI is based on independent and easily configurable components. Its website was created using Bootstrap and works on all web browsers that support Bootstrap. See the Bootstrap documentation for details.

After you install your Knowledge Center Servers and configure the Knowledge Center Cluster, you can access the Sample UI sandbox via the following URLs:

- If you have configured a load-balancer for your cluster: *http://host_load_balacer:port_load_balancer/gks-sample-ui*
- If you use a Knowledge Center Cluster with a single node: *http://gkc_server_host:gkc_server_port/gks-sample-ui*

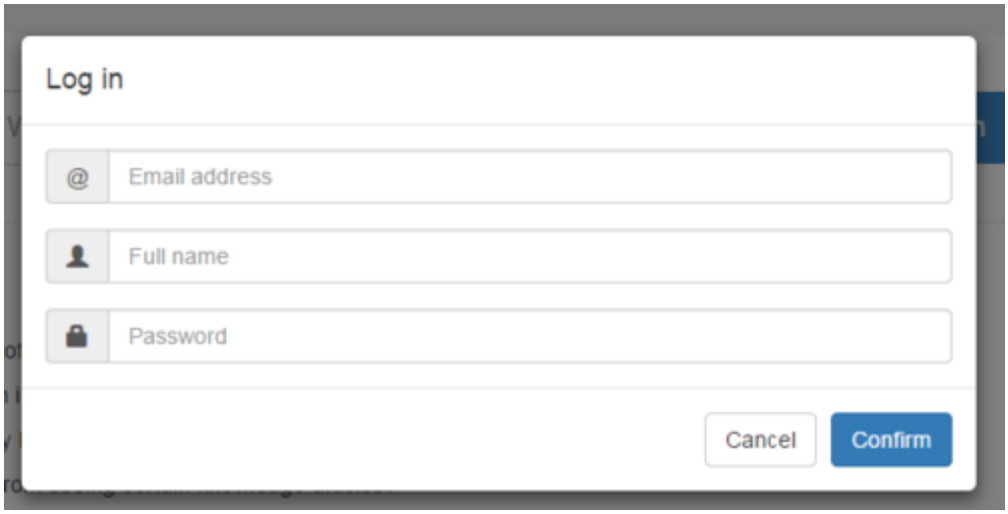
The Sample UI is pre-configured to show all Active and Public knowledge bases configured in Knowledge Center Server in language en (English).

Authorizing

You can use the Sample UI to:

- Browse the site, either as an anonymous user or by authorizing yourself as a customer.
  To authorize, click the **Log in** link, enter your credentials, and click **Confirm**

> ### Important
> This is not a real site authorization, as Knowledge Center server will only use an email as a *customerId* to identify sessions in History records.



Sample UI Login

- To log out, click the link with your customer name and select "Logout"

Sample UI Logout

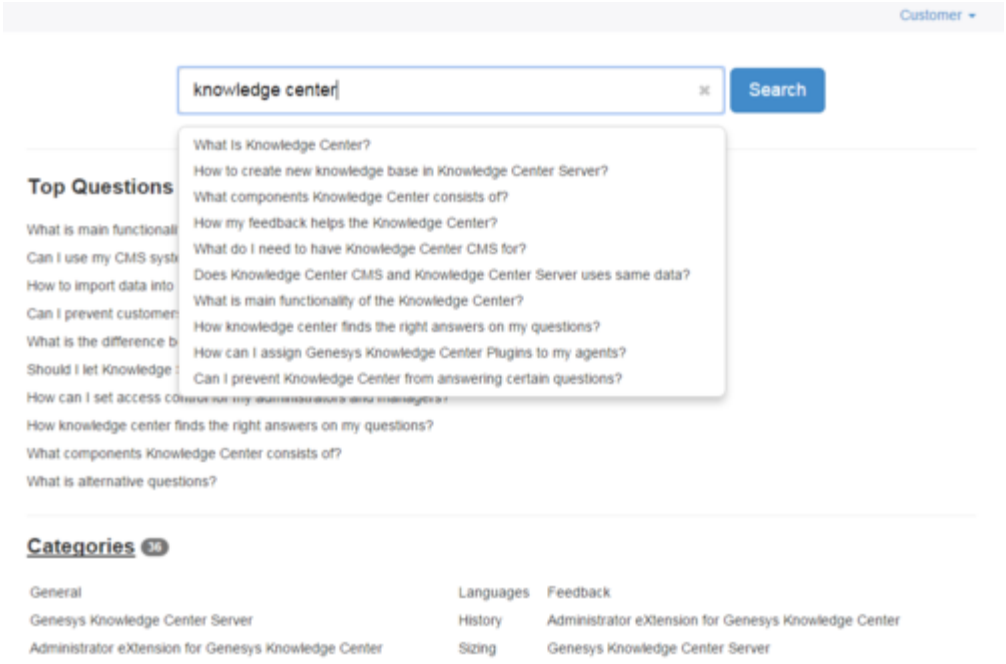
Searching

Search for any QNA document using the search bar.

Conduct a search

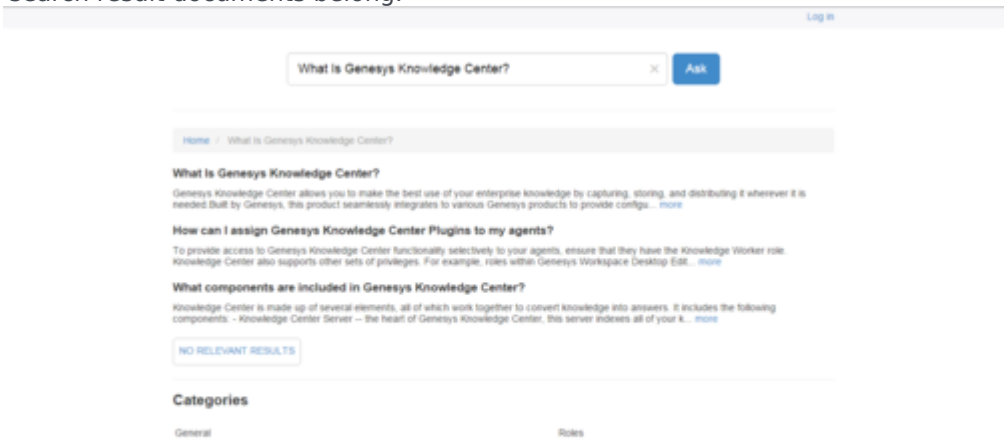
Start

1. Enter a question in the search bar and **Search** or press **Enter**.



Sample UI Search

2. Review search results. You can use the **No relevant result** button to let Knowledge Center know that your search was unsuccessful. At the bottom of the page, there is a list of categories to which your search result documents belong.

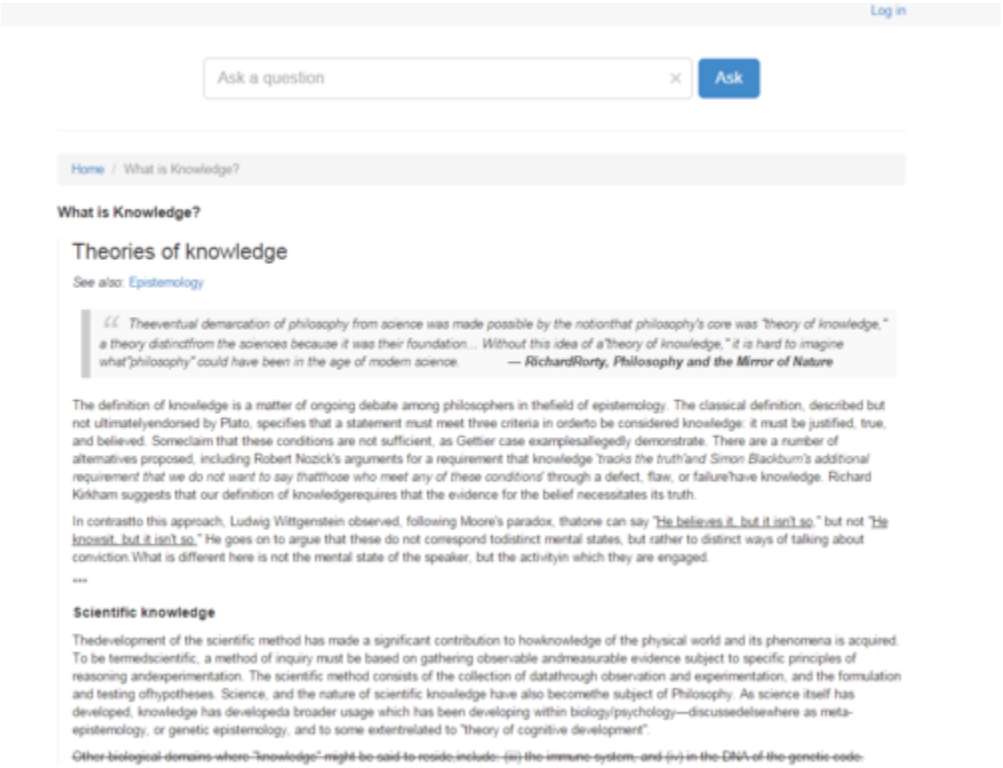


Sample UI Search Results

End

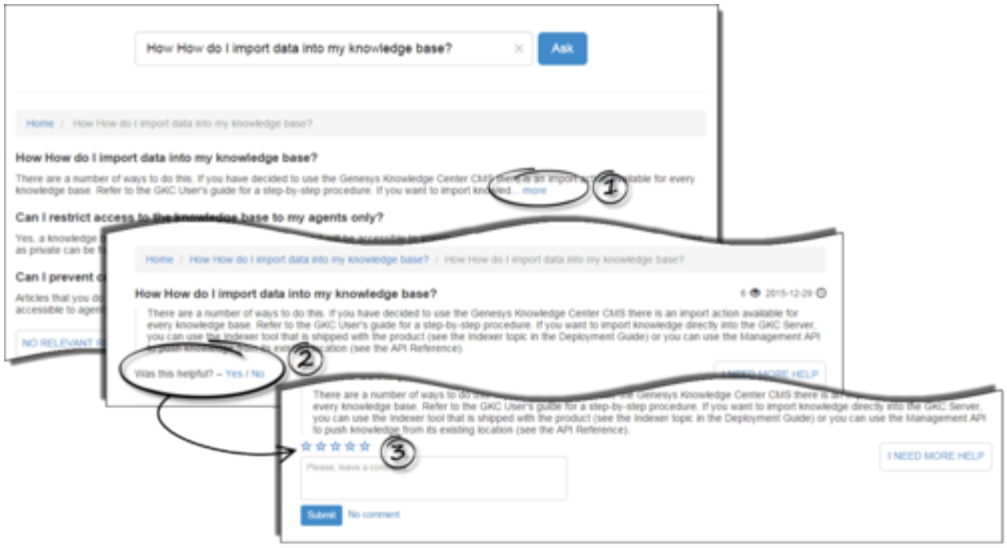
Open and Review a Document

> **Important**
> Documents can be in plain text or rich text



Example of Rich Text

- To expand the document, click the **more** link.
- Send feedback about the relevance of a search, using the **Yes/No** link to Like or Dislike the quality of the search. If you like or dislike an answer, you are asked to provide a star-rating and a comment (optional) to improve the Knowledge article.

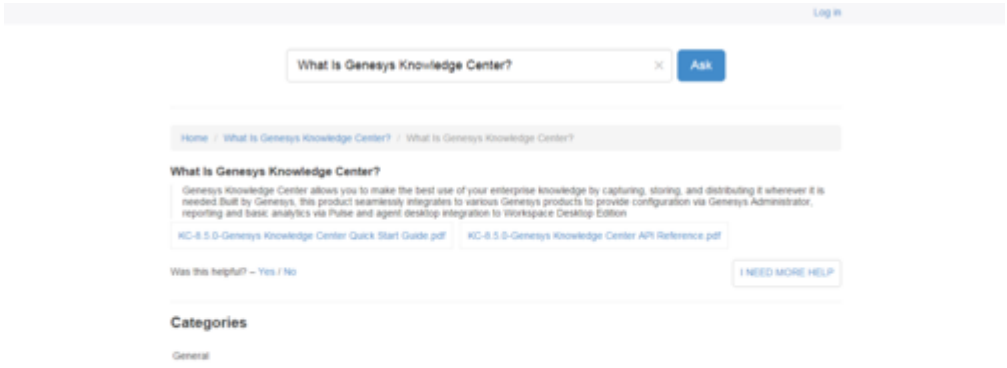


Negative Feedback Comment Field

- Click the **I need more help** button to send a request for proactive help from Genesys Web Engagement.

> ## Important
>
> This feature has been created only for use in conjunction with Genesys Web Engagement. No real message will be sent without integrating your Knowledge Center installation with GWE.

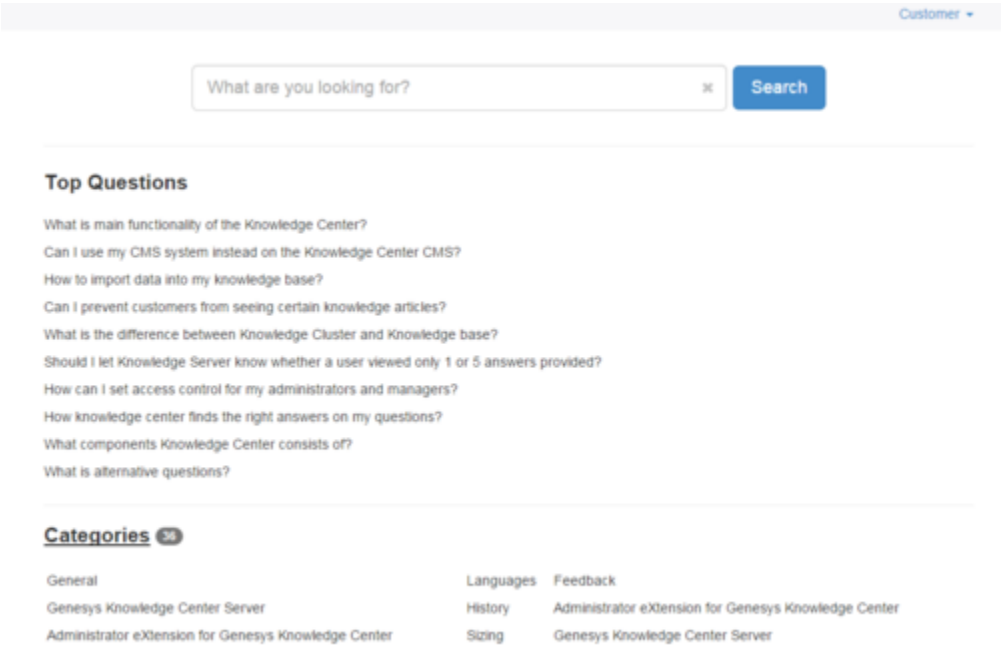
- Click attachment names to open any attachments in the document. Attachments will open in a new window.



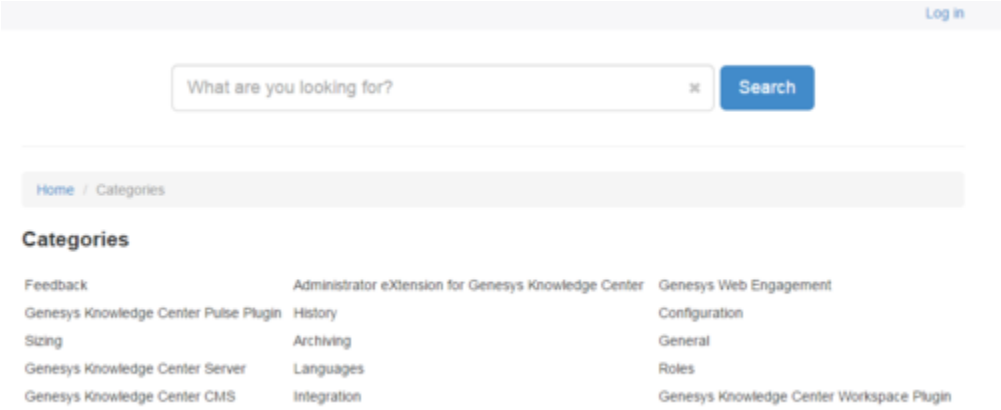
Opening Attachments

Browsing

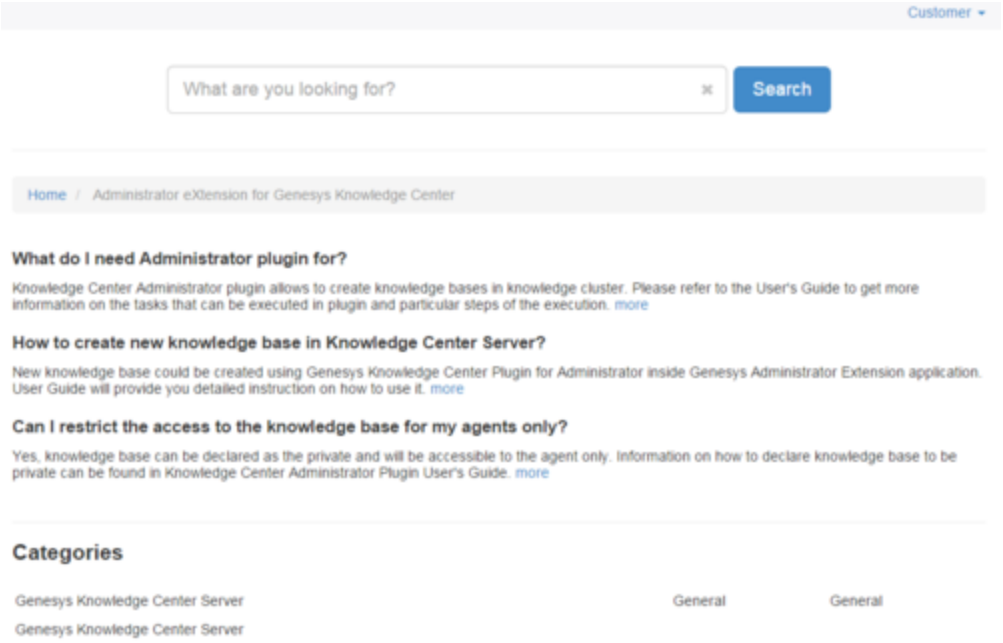
To browse Categories click the "Categories" link from main page.



Sample UI Main Questions



Sample UI Categories



Sample UI Document Categories

Importing Data into the Knowledge Center Server (before 8.5.303)

> **Important**
> The content of this page only applies to 8.5.30x.xx versions prior to 8.5.303.14.

Using Indexer to Import Data

If you are not going to use a CMS you can use the indexer to import data for use with Genesys Knowledge Center.

The indexer is installed during the installation of Knowledge Center Server. It is located inside your Knowledge Center Server installation folder in the *\server\tools\indexer* subdirectory.

Options

POSIX-like options	GNU-like long options	Required	Default	Description
-h	--host	yes	none	Genesys Knowledge Center Server url
-f	--file	no	`./`	file or directory that contains indexed data for import
-t	--transformer	no		file that contains a *.xsl transformer
-u	--user	yes	none	username of the agent the operation is executed on behalf of
-a	--authorization	no	none	Credential for Basic Authorization on Knowledge Center Server (if enabled)
-sbt	--subTenantId	no	—	sub-tenant identifier
-l	--loop	no	`false`	read "-f (--file)" file

POSIX-like options	GNU-like long options	Required	Default	Description
				or folder infinitely

Usage

```
java -jar gks-indexer-${version}.jar
--host "http://<host>:<port>/gks-server"
--file "<import_file_or_folder_path_and_name>"
--transformer "<processing_XSLT>"
--user "<agent_name>"
--authorization "<name>:<password>"
```

XML example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<documents kbsId="knowledgeFAQ" lang="en">
    <document>
        <lang>en</lang>
        <question>Question from indexer</question>
        <answer>Answer from indexer</answer>
        <categories>
            <category>
                <name>Testing</name>
            </category>
            <category>
                <name>Actors</name>
            </category>
        </categories>
        <validTo>2015-12-31 00:00:00</validTo>
        <alternatives>
            <alternative>Alternative question from indexer</alternative>
        </alternatives>
        <media>
            <media>media indexer</media>
            <media>media indexer</media>
        </media>
        <tags>
            <tag>tag indexer</tag>
            <tag>tag indexer</tag>
        </tags>
        <url>indexer url</url>
        <customFields>
            <entry>
                <key>numfield</key>
                <value>123</value>
            </entry>
            <entry>
                <key>strfield</key>
                <value>Hello GKS</value>
            </entry>
            <entry>
                <key>datefield</key>
                <value>2015-12-31</value>
            </entry>
            <entry>
                <key>unexistingField</key>
```

```
                <value>I'm an unexisting custom field</value>
            </entry>
        </customFields>
        <created>2014-10-01 00:00:00</created>
        <modified>2014-12-31 00:00:00</modified>
        <attachments>
            <attachment>
http://www.mydomain.com/Document.pdf
            </attachment>
        </attachments>
    </document>
</documents>
```

> **Important**
>
> For importing Rich Text in HTML format via indexer use **<![CDATA[** and **]]>** tags inside <answer> or <description> fields.

JSON example

```
{
    "kbsId": "knowledgebaseId",
    "lang": "en",
    "documents": [{
        "answer": "answer_1",
        "categories": [{
            "id": "cat_1_id",
            "name": "cat_1"
        }, {
            "id": "cat_2_id",
            "name": "cat_2"
        }],
        "created": "2013-09-08 13:15:33",
        "id": "document_1_id",
        "media": [
            "application",
            "audio"
        ],
        "modified": "2014-01-03 22:03:19",
        "question": "question_1",
        "tags": [
            "tag1",
            "tag2"
        ],
        "url": "genesys.com"
    }, {
        "answer": "answer_2",
        "categories": [{
            "id": "cat_1_id",
            "name": "cat_1"
        }, {
            "id": "cat_3_id",
            "name": "cat_3"
        }],
        "created": "2010-03-09 11:15:21",
        "id": "document_2_id",
        "media": [
            "video",
            "text"
        ],
        "modified": "2013-08-01 05:54:52",
        "question": "question_2",
```

```
        "tags": [
            "tag3",
            "tag4"
        ],
        "url": "genesys.com"
    }]
}
```

Importing Sample Data

You can use the Import Tool to add sample QNA data to your knowledge base. This tool is located in the *./server/tools* directory in the Knowledge Center installation folder. It comes with the following resources:

- **knowledgeFAQ.xml**—List of basic QNA data, provided with the Knowledge Center Server indexing tool
- **gks-indexer-tool.jar**—Java-based indexing tool
- **importFAQ.bat**—Simple data import script

Data Import Syntax

> **Important**
>
> Users must have **Knowledge.AUTHOR** privileges in order to use the Administrator plugin.

Use the following syntax to import data:

```
java -jar gks-indexer-${version}.jar
--host "http://<host>:<port>/gks-server"
--file "<import_file_or_folder_path_and_name>"
--user "<agent_name>"
--authorization "<name>:<password>"
```

The authorization parameter is only required if you have enabled the security option for Knowledge Center Cluster.

Sample Import Script

Here is an example of what your import script might look like:

```
java -jar <Path to GKC Server>\server\tools\gks-indexer-tool.jar
--host "http://sample.com:9092/gks-server"
--file "<Path to GKC Server>\server\tools\knowledgeFAQ.xml"
--user "gkc_admin"
```

If it works, this script will import sample QNA data into the knowledge base.

Sample QNA Data

Here is an example of the data stored in the XML file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<documents kbsId="knowledgefaq" lang="en">
```

```
    <document>
        <question>What Is Knowledge Center?</question>
        <answer>The Genesys Knowledge Center ultimate goal is to convert
your knowledge into the answers on the question your clients
or agents have. It delivers set of the component for administration,
authoring and using the knowledge. The heart of the system is the
Knowledge Center Server that aimed to find the best answer on
the question you have asked.</answer>
        <categories>
            <category>
                <name>General</name>
            </category>
        </categories>
    </document>
</documents>
```

Fields that can be used with Indexer

Kind	Name	Type	Mandatory	Description
Attribute	kbsId	String	Yes	Knowledge base identifier
Attribute	lang	String	Yes	Language identifier
Node	documents	Nested XML	Yes	Markup of documents for indexing, each document is wrapped with inner node "document".

Each element of type of "document" of XML markup of field "documents" is XML of structure (for type of QNA)

Name	Type	Mandatory	Default	Description
id	String	No	Autogenerate	Document identifier
created	StringDate yyyy-MM-dd HH:mm:ss	No	now	Date of document creation
modified	StringDate yyyy-MM-dd HH:mm:ss	No	now	Date of document last modification
validTo	StringDate yyyy-MM-dd HH:mm:ss	No	never	Date of document expiration
url	String	No	null	Absolute url for retrieving full document content
question	String	Yes (in case of FAQ item)	—	Text of FAQ question
answer	String	Yes (in case of FAQ items)	—	Text of FAQ answer

Name	Type	Mandatory	Default	Description
answerContentType	String	No	html-autodetect	Content type of content in node answer. Valid values: text/plain, text/html
tags	Nested XML	No	Empty	XML Markup that consist of nodes <tag>, each of which contains string value TAG that related to this document
media	Nested XML	No	Empty	XML Markup that consist of nodes <media>, each of which contains string value of MEDIA that related to this document.
categories	Nested XML	No	Empty	XML Markup that consist of set of nodes of <category> and describes all categories, that are related to this document
alternatives	Nested XML	No	Empty	XML Markup that consist of set of nodes <alternative> that contains text of question, that match to theme of content of this document
attachments	Nested XML	No	Empty	Markup of that consists of set of nodes <attachment> that wraps absolute URL to content of attachment to this document
customFields	Nested XML	No	Empty	XML Markup that consists of set of nodes <element> that describes names and values of customFields of this document

Each element of type of "document" of XML markup of field "documents" is XML of structure (for type of ARTICLE)

Name	Type	Mandatory	Default	Description
id	String	No	Autogenerate	Document identifier
created	StringDate yyyy-MM-dd HH:mm:ss	No	now	Date of document creation
modified	StringDate yyyy-MM-dd HH:mm:ss	No	now	Date of document last modification
validTo	StringDate yyyy-MM-dd HH:mm:ss	No	never	Date of document expiration
url	String	No	null	Absolute url for retrieving full document content
title	String	Yes	—	Title of article
description	String	Yes	—	Description of article
summary	String	Yes	—	Summary of article
answerContentType	String	No	html-autodetect	Content type of content in node answer
tags	Nested XML	No	Empty	XML Markup that consist of nodes <tag>, each of which contains string value TAG that related to this document
media	Nested XML	No	Empty	XML Markup that consist of nodes <media>, each of which contains string value of MEDIA that related to this document.
categories	Nested XML	No	Empty	XML Markup that consist of set of nodes of <category> and describes all categories, that are related to this document
alternatives	Nested XML	No	Empty	XML Markup that consist of set of nodes <alternative> that contains text of question, that match to theme of

Name	Type	Mandatory	Default	Description
				content of this document
attachments	Nested XML	No	Empty	Markup of that consists of set of nodes <attachment> that wraps absolute URL to content of attachment to this document
customFields	Nested XML	No	Empty	XML Markup that consists of set of nodes <element> that describes names and values of customFields of this document

Each item of type category of XML markup of "categories" consists of nodes:

Name	Type	Mandatory	Default	Description
id	String	—	AutoGenerate	Identifier of category
name	String	Yes	—	Name of category

Each item of type element of XML markup of "customFields" consists of nodes:

Name	Type	Mandatory	Default	Description
key	String	Yes	—	Name of document custom field
value	String	Yes	—	Value of document custom field

Importing Data into the Knowledge Center Server

Indexer Tool

If you are not going to use a CMS you can use the indexer to import data for use with Genesys Knowledge Center.

The indexer is installed during the installation of Knowledge Center Server. It is located inside your Knowledge Center Server installation folder in the \server\tools\indexer subdirectory. **Command line:**

```
java -jar gks-indexer.jar  [parameters]
```

Parameters:

Short Parameter	Qualified Parameter	Mandatory	Example	Description
-u	--user	n/a	--user gkc_super	Name of internal user with authoring permissions
-a	--authorization	n/a	-a user:password	Username and password for basic authorization
-f	--file	Y	--file c:\xml	Path to file or directory with files for importing
-h	--host	Y	--host http://gks/gks-server:8080	target knowledge server url
-o	--overwrite	n/a	-o	For replacing all existing documents with documents from importing file
-tenant	--tenantId	n/a	--tenantId 1	Target tenant identifier
-sbt	--subTenantId	n/a	--subTenantId default	Target subtenant identifier
-sk	--sslKeys	n/a	--sslKeys c:\keys\sslkey	Path to trust store
-sp	--sslPassword	n/a	--sslPassword topsecret	trust store password

Short Parameter	Qualified Parameter	Mandatory	Example	Description
-t	--transformer	n/a	--transformer c:\transformers\ transformer.xsl	Path to XSL transformer

Knowledge File Structure

field	type	mandatory	format	description
knowledge	Object	Y	n/a	Container of documents and categories for indexing. **There are two mandatory attributes of a node of "knowledge":** • kbId - Knowledge base identifier • lang - language identifier • version - version identifier (current: "8.5.304")
knowledge.categories	Array	N	n/a	Knowledge base categories directory
knowledge.categories[]	Object	Y	n/a	Knowledge category
knowledge.categories[].id	String	Y	n/a	Category identifier
knowledge.categories[].categoryParentId	String	N	n/a	Parent category identifier. Omit for root categories.
knowledge.categories[].name	String	Y	n/a	Category name
knowledge.documents	Arrray	N	n/a	Documents for indexing. Omitting of this field means that indexer must not touch already indexed documents at all.
knowledge.documents[].id	String	N	n/a	Document identifier. Server

field	type	mandatory	format	description
				may generate identifier automatically in case when documents[].id is omitted
knowledge.documents[].templateId	String	Y	n/a	Document template identifier
knowledge.documents[].validFrom	Date	N	YYYY-MM-DD	Document start date
knowledge.documents[].validTo	Date	N	YYYY-MM-DD	Document expiration date
knowledge.documents[].media	Array	N	n/a	List of media channels that this document is related to.
knowledge.documents[].media.media	String	N	n/a	Document media channel value
knowledge.documents[].tags	Array	N	n/a	List of tags related to this document
knowledge.documents[].tags.tag	String	Y	n/a	Document tag value
knowledge.documents[].url	String	N	n/a	Document external url
knowledge.documents[].title	String	Y	n/a	Document title
knowledge.documents[].title.id	String	Y	n/a	Document title name
knowledge.documents[].title.value	String	Y	n/a	Document title value
knowledge.documents[].content	Array	Y	n/a	Document content
knowledge.documents[].content[].docField	Object	Y	n/a	Document content field
knowledge.documents[].content[].docField.id	String	Y	answer, description, body	Document content field name
knowledge.documents[].content[].docField.value	String	Y	n/a	Document additional content field value **Important** For importing Rich Text in HTML format via indexer use <![CDATA[and]]> tags inside <value> field.
knowledge.documents[].additional	Array	N	n/a	Document additional content
knowledge.documents[].additional[].docField	String	Y	n/a	Document

field	type	mandatory	format	description
				additional content field
knowledge.documents[].additional[].docField.id	String	Y	n/a	Document additional content field name
knowledge.documents[].additional[].docField.value	String	Y	n/a	Document additional content field value **Important** For importing Rich Text in HTML format via indexer use <![CDATA[and]]> inside <value> field.
knowledge.documents[].alternatives	Array	N	n/a	Alternative names/ questions for the document
knowledge.documents[].alternatives[].alternative	String	N	n/a	Relevant text item
knowledge.documents[].attachments	Array	N	n/a	Document attachments
knowledge.documents[].attachments[].attachment	String	N	n/a	Document attachment URL
knowledge.documents[].categories	Array	N	n/a	Document category identifiers
knowledge.documents[].categories[].category	Object	N	n/a	Document category object
knowledge.documents[].categories[].category.id	String	Y	n/a	Document category identifier
knowledge.documents[].customFields	Array	N	n/a	Document custom attributes
knowledge.documents[].customFields[].entry	Object	N	n/a	Document custom attribute item
knowledge.documents[].customFields[].entry.key	String	Y	n/a	Document custom attribute name
knowledge.documents[].customFields[].entry.value	VAR	N	n/a	Document custom attribute value

Example of content of indexing file (v2)

```
<?xml version="1.0" encoding="UTF-8">
<knowledge kbId="knowledgefaq" lang="en" version="8.5.304">

    <categories>
        <category>
            <id>c1</id>
            <name>category 1</name>
        </category>

        <category>
            <id>c2</id>
            <name>category 2</name>
            <categoryParentId>c1</parentId>
        </category>
    </categories>

    <documents>
        <document>
            <id>doc1</id>
            <templateId>basefaq</templateId>
            <validFrom>2017-02-20</validFrom>
            <validTo>2017-02-21</validTo>
            <media>
                <media>m1</media>
                <media>m2</media>
            </media>
            <tags>
                <tag>t1</tag>
                <tag>t2</tag>
            </tags>
            <url>doc1url</url>

            <title>
                <id>question</id>
                <value>document question</value>
            </title>
            <content>
                <docField>
                    <id>answer</id>
                    <value>answer body</value>
                </docField>
            </content>
```

```
    <alternatives>
        <alternative>document alt1</alternative>
        <alternative>document alt2</alternative>
    </alternatives>

    <attachments>
        <attachment>a1</attachment>
        <attachment>a2</attachment>
    </attachments>

    <categories>
        <category><id>c1</id></category>
        <category><id>c2</id></category>
    </categories>

    <customFields>
        <entry>
            <key>strField</key>
            <value>some string</value>
        </entry>
        <entry>
            <key>numField</key>
            <value>123</value>
        </entry>
    </customFields>
</document>

<document>
    <id>doc2</id>
    <templateId>basefaq</templateId>
    <validTo>2017-02-21</validTo>
    <media>
        <media>m1</media>
        <media>m2</media>
    </media>
    <tags>
        <tag>t1</tag>
        <tag>t2</tag>
    </tags>
    <url>doc2url</url>

    <title>
```

```
            <id>question</id>
            <value>document question</value>
        </title>
        <content>
            <docField>
                <id>answer</id>
                <value>faq answer</value>
            </docField>
        </content>
        <alternatives>
            <alternative>document alt1</alternative>
            <alternative>document alt2</alternative>
        </alternatives>

        <attachments>
            <attachment>a1</attachment>
            <attachment>a2</attachment>
        </attachments>

        <categories>
            <category><id>c1</id></category>
            <category><id>c2</id></category>
        </categories>

        <customFields>
            <entry>
                <key>strField</key>
                <value>some string</value>
            </entry>
            <entry>
                <key>numField</key>
                <value>123</value>
            </entry>
        </customFields>
    </document>

    <document>
        <id>doc3</id>
        <templateId>basearticle</templateId>
        <media>
            <media>m1</media>
        </media>
```

```
            <title>
                <id>title</id>
                <value>document title</value>
            </title>

            <content>
                <docField>
                    <id>description</id>
                    <value></value>
                </docField>
            </content>

            <additional>
                <docField>
                    <id>summary</id>
                    <value>document summary</value>
                </docField>
            </additional>

            <customFields>
                <entry>
                    <key>cf1</key>
                </entry>
            </customFields>
        </document>
    </documents>
</knowledge>
```

Importing Sample Data

In the `./server/tools` directory in the Knowledge Center installation folder, you can find a sample knowledge base along with the indexer tool:

- knowledgeFAQ.xml — Sample knowledge base describing some of the questions related to Knowledge Center
- gks-indexer.jar—Java-based indexing tool
- importFAQ.bat—Simple data import script

> **Important**
>
> Users must have Knowledge.AUTHOR privileges in order to use the Administrator plugin.

To import a sample knowledge base you need to:

1. open importFAQ.bat file:
   - ensure that --host parameter is pointing on one of your Knowledge Center Servers or the load balancer in front of the cluster (recommended)
   - ensure that --user parameter is set to valid used with Knowledge.AUTHOR privileges (knowledge by default)
2. save changes if any
3. open knowledgeFAQ.xml:
   - ensure that kbId attribute is set to desired knowledge base you would like to import data to (knowledgeFAQ by default)
   - ensure that language is set to properly configured one and added to the knowledge base
4. save changes if any
5. run importFAQ.bat