



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Knowledge Center Deployment Guide

Importing Data into the Knowledge Center Server (before 8.5.303)

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>
      </entry>
    </customFields>
  </document>
</documents>
```

```
        <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