

Implementing a Knowledge Base System in a Windows Environment

By Harris Kern's Enterprise Computing Institute

Knowledge bases are extremely powerful tools that can significantly improve your ability to deliver services to your customer base. A knowledge base contains solutions and workarounds to known problems and allows your Service Center agents to access those solutions to resolve customer problems. Knowledge bases provide many potential benefits to the Service Center. When properly implemented, a knowledge base documents the collective knowledge of your Service Center agents and leverages that knowledge by making it available to all agents. This ability alone overcomes a common problem in many Service Centers: communicating solutions to everyone who needs them. In addition to your internal knowledge, product-specific and third-party multi-product knowledge base content is available for purchase so that you can leverage external knowledge as well.

When properly implemented and used, the knowledge base stops agents from redundantly troubleshooting problems. The knowledge base documents problems and symptoms as well as solutions and workarounds. Typically, an agent receives a call and if he or she does not know the solution, they generally begin troubleshooting or analyzing the problem, and then move on to identifying a solution. If a knowledge base is available, the process is slightly different. The agent receives a problem and if he or she does not know the solution, the next step is to check the knowledge base, not start troubleshooting. Without the knowledge base, the agent thinks, "I'm not with the problem. Let's figure out what's causing it and how to fix it." With a knowledge base, the agent thinks, "I'm not familiar with the problem. Let's see if anyone else has encountered it, and if so, what they did to resolve it."

Another significant benefit of knowledge bases is that if agents use it properly, they can consistently apply solutions and workarounds. Consistency reduces support costs because it reduces "one-offs" (custom solutions). If new solutions or better workarounds become available, they too can be consistently applied.

There are different types of knowledge bases, but one fact that is consistent across them all is that the information that comes out is only as good as what goes in. The old adage "garbage in, garbage out" certainly holds true with knowledge bases, so if you are going to implement one, you must be prepared to deal with the overhead required to make sure that what goes in is good. You certainly do not want to propagate bad solutions or workarounds. Someone will have to validate knowledge base reports prior to releasing them into the knowledge-base. You also have to make sure that they are consistently formatted and well written for ease of use and understanding. Finally, to be effective, every common problem and solution must be captured, documented, and updated as necessary. These three tasks, gathering, documenting, and testing represent a significant amount of work and cannot be overlooked or underestimated if you want to implement a knowledge base that will actually be used.

Another important consideration is training for agents. It is a fact that given a good knowledge base, the agent who is best at finding and retrieving the information will be more productive and successful at resolving customer problems than other agents, regardless of their technical knowledge.

As mentioned previously, there are different types of knowledge bases. The primary difference is based on how information is retrieved and thus how it is stored. The most common knowledge bases are document-or text-based storage and retrieval systems. In this model, a standard document is used to document a single problem and solution, and all the known symptoms. It is important that these documents all use the same format. A text retrieval engine searches the database for the keywords entered by the Service Center agent, which makes the system simple to use. The benefits of this approach are that it is easy to design and to add, update, and remove knowledge base documents as necessary. The potential problems occur as the database grows in size. A search may return hundreds or thousands of documents that match the agents' search criteria. The agent must then look through these to find the correct document, which can dramatically increase the time it takes to resolve the issue. This could cause agents to avoid using the system, thus eliminating the benefits of having a knowledge base.

The other type of knowledge base system uses expert-based knowledge to lead the agent to the correct information. Generally, the knowledge base information is organized or accessed hierarchically. The knowledge base retrieval model attempts to stimulate the troubleshooting approach. The agent selects from a list of choices and/or questions to navigate through the hierarchy until he or she reaches the desired information. For example, the agent may start by selecting a product or type of product (Microsoft PowerPoint or network printer) at the highest level of the hierarchy. Based on that choice, a new set of options is presented, which, upon selection, navigate deeper into the hierarchy. For example, if the agent chose network printer at the highest level of the hierarchy, the next level may present a list of network printers to choose from. Or, if the agent selected Microsoft PowerPoint at one level of the hierarchy, the next level may ask the agent what type of Microsoft PowerPoint problem he or she is having and present a list of the most common problems. The agent selects the problems and continues navigating deeper into the hierarchy. This is a good approach for agents, because they do not require much training to use it effectively. The downside of this approach is that it requires much more design planning on the part of the Service Center. It also requires significant maintenance to provide the right options for navigating the hierarchy and accessing new solutions. As the customer environment being supported becomes more complex, so does the hierarchy required to support it. The approach works best for common and well-known problems.

Variations of the hierarchic approach are often referred to as case-based systems, decision tree systems, and troubleshooting systems. They all use a hierarchy but use different selection criteria to navigate the hierarchy or decision tree. A fourth variation of the approach uses a network structure rather than a hierarchy. The benefit is that an agent can navigate through the data rather than up and down the hierarchy; thus if the information is not found, the agent doesn't have to back all the way out and start over. AN additional benefit is that because the system uses network architecture, it is easier to add new information, because there is no rigid hierarchic structure to redesign. The downside to this approach is that because of the tool's inherit flexibility; more process rigidity must be implemented by the organization to ensure that the data does not get out of control.