

Datahaven Windows® SDK: Sample Content-Enabled Healthcare Information System in a Legacy Environment

This sample integration screenshot demonstrates that even a legacy greenscreen application can integrate with Datahaven to create seamless content management functionality. In this example, the controlling application is a legacy healthcare information system, or HIS, for which the vendor offers an optional Windows-based user interface. The vendor added the necessary calls from the user interface code to the Datahaven API to produce this integration. The integration design specifications call for content capture using stand-alone and copier-based scanning, inbound fax integration, as well as automated file imports of other medical content such as radiology images. Once captured, the content undergoes an initial data entry workflow process to deliver the content to the appropriate hospital personnel for action and associate the content with the patient account.

Datahaven supports multiple, dynamic views of its cabinet interface and the content stored in its repository that can be driven by any type of control (e.g., tab, radio buttons, checkboxes, etc.). Buttons were re-tasked to interact with the Datahaven API to perform a wide range of actions related to a specific document or other content object.

Navigation buttons can be turned on/off and displayed across the top or along the side of the document image viewer.

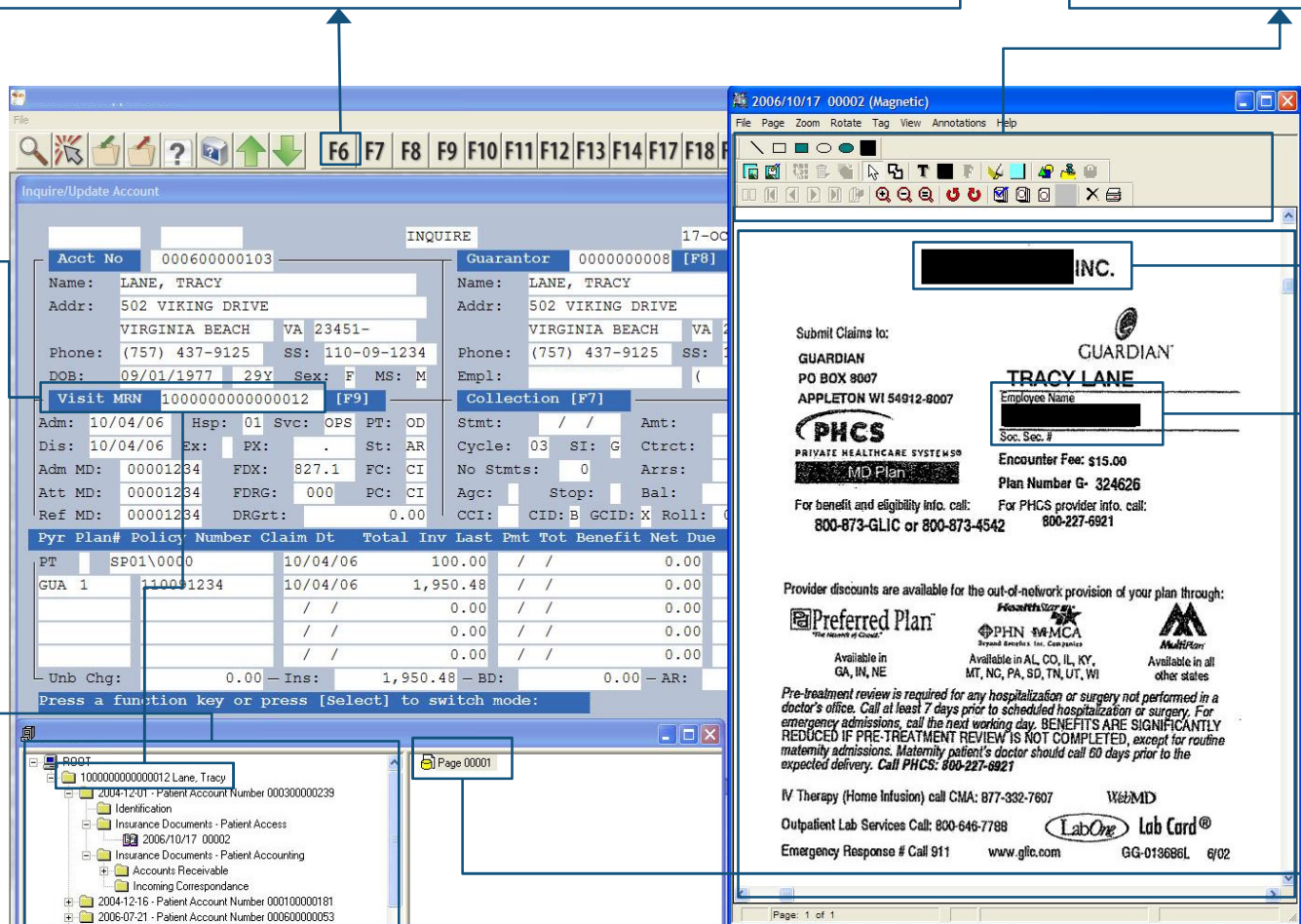
All views of the content stored in the Datahaven database are filtered windows into the much larger database containing all “unstructured” content and related metadata. Here, the Datahaven cabinet changes dynamically based on the patient record displayed by the HIS application. Also, Datahaven can display content from multiple originating systems and display this disparate content so that it appears to be “native” to this application. For example, Datahaven could display a copy of an invoice that originated in its content-enabled CRM system.

Page-level redactions allow for additional security to be applied to particular areas of a document.

The document/image viewer window can launch within a fixed window or sit atop other screens to maximize screen real estate. **Datahaven manages and allows access to any file type.** If Datahaven does not support the native viewing of a file type, it launches the application associated with the file type (MS Word for a .doc file) from the local machine.

Datahaven’s Autofile function automatically and intelligently creates and manages this database-driven folder structure, including the automatic naming of folders and files. This ensures that consistent data elements control the naming and filing of content and creates a consistent filing methodology, even over long periods of time. This also enhances the search functionality by allowing authorized users to “browse” or “climb the tree” to find related content in a parent or child folder.

Storage of document images at the page-level enables advanced functionality, including the ability to split, merge, and clone documents.



Display the Datahaven “file cabinet” component in its own window or, as here, as a **seamless window that is “bolted” to your application** that users can hide and resize automatically with your application’s window. You also have flexibility in how the relevant content from the Datahaven repository is depicted in the Datahaven file cabinet: (1) the entire contents of the repository, (2) the contents of an individual folder or folder tree, and (3) an individual file, list of files, or workflow queue. The display of an individual file, for example, could be used to enforce the display of only the oldest file in a queue. Here we see an example of a “tree view” that displays the folders and their content as it relates to a particular patient. Note also that authorized users can perform other actions on the displayed content, including tagging multiple documents for print, fax, or export (including document assembly).