

Datahaven Web SDK: Sample Content-Enabled Issue Management Application in a Web-Based Environment

This web-based sample integration illustrates the use of the Datahaven Web SDK to content-enable this open source issue tracking application. Among other things, Datahaven enables this application to support input from desktop scanners, digital copiers, inbound fax servers, email, FTP servers, etc. Once captured and classified in the system, users have access to the full spectrum of content. More importantly, access to content is context-sensitive so that they see the content that relates to the specific record displayed by the controlling application.

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.). For example, this application displays the entire folder structure related to a particular issue reported by a customer.

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. Datahaven can draw content from multiple originating systems and display this disparate content so that it appears to be “native” to this application. Here, the view displayed of the Datahaven cabinet changes dynamically based on the vendor record displayed by this application, but also can display content originally authored in another system.

Developers can use any type of control (e.g., radio buttons, checkboxes, etc.) to suppress the view or present a desired view of the Datahaven cabinet and the content of the Datahaven repository.

Datahaven’s Autofile function uses pre-defined metadata to automatically and intelligently create and manage this database-driven folder structure, including the automatic naming of folders and files. This ensures 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.

Developers can use as many of the available Datahaven controls (e.g., cabinet, viewer, indexing, workflow, etc.) as needed to produce the desired end-user experience for your application.

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 to eliminate the ability of workers to “cherry pick” what files they may prefer to process. Here we see an example of a “tree view” that displays the folders and their content as it relates to a particular customer. Note also that authorized users can perform other actions on the displayed content, including tagging multiple documents for print, fax, or export.

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

Datahaven supports recognition of numerous barcode formats to automate data entry and drive automated filing procedures with its Autofile technology.

The document/image viewer window can launch within an IFRAME or in its own window with a fixed or resizable window so that it can 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.

Storage of document images at the page-level enables advanced functionality, including the ability to split, merge, and clone documents as well as allow for faster retrieval times for large documents.

The screenshot displays a web browser window titled "Add/Edit Issue - Windows Internet Explorer" with the URL "http://company.com/dew/issues.aspx?id=53". The application interface includes a navigation menu with options like "issues", "search", "queries", "admin", "reports", "settings", "logoff", and "about". The main content area shows details for "Issue ID: 53", including a short description "Installation initialization error", reported by "lmeeks" on 2008/11/21 1:52 PM. It lists project details, priority (medium), assigned user (lmeeks), and status (in progress). A file cabinet view below shows a tree structure for "Issue 53" with folders like "Research" and "Inbound Information", and files such as "2008/11/21 00001 - Error Detail.jpg" and "2008/11/21 00001 - Error Screenshot.msg". A table at the bottom lists issue entries with columns for ID, Date, User, and Note. An inset window shows a Microsoft Windows XP error dialog box with the message: "Could not initialize installation. Could not extract Wse0132.dll to 'C:\DECLINE-I\WARREN-1\Temp\SLC4.tmp'. CRC does not match. Try function."