

Update History for IBM iDoctor for IBM i – Heap Analyzer Heap Analysis Tools V5R4

Sep 17, 2008 – C00664

No Heap Analyzer-specific updates occurred this build for this release, but see the Job Watcher update history for general GUI enhancements

Nov 29, 2007 – C00584

Updated the compare function. There is now a stored procedure that gets called to replace the use of the HCOMPARE command in the GUI. You must select two object table dumps for the Create Comparison Report menu to be enabled when right-clicking on the selected collections. This will create a new file visible in the server side output files folder for the 1st collection selected of the 2.

Added a summary report for all object table dump in a library. Right click on a library and choose the "Create Object Table Summary" menu to perform this option. A new file QAIDRHASUM will be created in the library.

Added the QAIDRHASUM file to the server-side output files folder if it exists for any object table dump in a library.

The 'root finder' function of Heap Analyzer has been removed from the GUI. The collection wizard no longer provides this option and these types of existing collections are no longer visible.

Fixed a bug in Heap Analyzer install where it could not be installed if Job Watcher wasn't already installed on the system

Updated the iDoctor GUI to compile using the new compiler Visual Studio 2005.

From the Run SQL View, added a menu "Launch SQL in Run SQL Scripts..." that will take the current SQL statement, place it in a temporary file and launch it in iSeries Access for Windows Run SQL Scripts for the purpose of either running it within that interface or using visual explain. Note that iDoctor uses System naming convention (library/file) and iSeries Navigator uses SQL naming convention (library.file). You will have to change the default connection properties (Connection->JDBC Setup... Format tab, Naming Convention) in order for the queries created within iDoctor to be executable.

Added a button on the toolbar in the Data Viewer to normalize time interval based graphs. This will divide each value of the graph by the interval duration, which will make bars more consistent in height. This button is available on several Job Watcher graphs.

Added 2 new Preferences to the Graph views section of the Display tab. "Use normalize option" -when checked will cause the normalize option to be used when a graph is first opened (if it's available.) "Use variable width bar option" - when checked will cause the variable width bar option to be used when a graph is first opened (if it's available.)

Property sheets that are not modal (meaning you can go back to other screens while they are open) are now displayed inside of a view. This provides the following benefits/changes that users have been asking for: 1) These property sheets are no longer "always on top". You can click another view and it will move in front of the property sheet view. 2) Property sheet views can be resized to any size. 3) "Drill-down" actions from a property sheet view will now open on top of the property sheet view, like any other drill down in the Data Viewer.

Added a preferences on the misc tab called "Show non-modal property pages in a view". If you don't like the new property sheet views, you can uncheck this and go back to the way it was before.

Property sheet views that don't have a need for OK, Cancel buttons will no longer show these buttons (for the purpose of saving screen space). The user can press escape or the X to close the window just like any other view

Added a menu option called "Filter libraries..." from the component icon to filter the list of libraries shown. This setting applies to all components that display collection libraries. This library filter is shown next to the component's name in the tree when active. This can be helpful if there are many libraries on the system or if the connection in use is slow.

Added a scheduling preferences page that allows the user to change the default scheduled date and time, from 1 week in the future, to whatever the user prefers (in +days/hours from current date/time).

When opening a report from the list portion of a component tree/list window using the popup menu, added an option called "Edit" below the normal "Open table" or "Open graph" menu. This option allows a user to quickly open a user-defined or iDoctor-supplied table or graph such that the SQL editor is initially open without actually attempting to run the query. If the query will take a long time run, this can be used to open and modify the SQL statement first. This technique can also be used in graphs to modify the graph settings before running the query behind the graph

Added a new view, which processes database SQL statements in a separate thread. These requests can be queued up and are processed one at a time. This view is called the Remote SQL Statement Status View and is currently used by the V5R4/V6R1 JW summary program as well as the "Execute in batch" option from the SQL editor

The popup menu in the SQL editor has a new option, Execute in batch. This will run the SQL statement in a separate thread and show the status of execution in the remote SQL statement status view. If an SQL select statement is provided, you will be optionally prompted for an SQL table/library name in order to create a table using the SQL statement given

Added new icons to the remote command status view such that the icons change based on the status of the request (waiting to run, completed successfully, error). These same icons are used on the remote SQL statement status view.

iDoctor can now handle SQL statements that contain SQL table names

Opening a graph/table in the Data Viewer should no longer tie up the GUI and you should be able to cancel this action by closing the graph/table view being opened if desired. While a report is being opened you should be able to do other things in the GUI like view already open reports/graphs or go back to the main window and browse other collections. If you are currently opening a report, you can go to another view and drill down to another view while the 1st one is running if you wish. A maximum of 3 data viewer connections are used to accomplish this which means you can have a maximum of 3 queries running simultaneously from the same iDoctor client. If you exceed this the GUI thread may appear to hang and execution will no longer occur simultaneously for these extra requests. If you try to scroll a graph/table while another one is being loaded, you may experience delays in the GUI thread (if the same connection was used for the report being scrolled and the new one being opened) but it should work once the previous query finishes. As part of this change I've removed the preference "Show query cancel window when filtering or resorting a report" on the Misc tab as this option is no longer needed. The multithreaded behavior should now act the same whenever opening/requerying any graph/table in the Data Viewer.

Show the report/graph title on the view being opened before the SQL statement finishes.

Add/Remove programs in the Windows control panel now properly recognize the iDoctor GUI. You can use this option to uninstall the client.

Made a change how data viewer connections are handled, so that instead of only using a maximum of 3 connections if 3 simultaneous queries are executed, 5 connections are available to the Data Viewer's views where each view that is created uses a different connection by cycling through them. By doing this it will allow the user to open view A and view B, and scroll through view A while view B is running a query. Previously view A and view B would likely be using the same connection which meant trying to scroll view A while view B is still running a query would hang the client until view B finished it's query! The maximum value of 5 is a registry setting if you wish to change this:
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.PEX\DataViewer]
"QZDASOINITConnections"="5"

The toolbars in the main window and in the Data Viewer are now dockable (can move them to wherever you want)

When refreshing a table, the hourglass on the cursor is often lost. To resolve this, show a message on the table when SQL statements are running in the background similar to what is shown for graphs "The query behind this report is running, please wait..."

Since graph tool tips can sometimes get in the way of what you are doing, I've added the ability to turn them on/off by pressing CTRL+T while in a graph. The status bar in the data viewer now includes a message after the graph control memory used message that indicates if tool tips are currently enabled or disabled.

Fix a bug on the query definition -> member selection page where users could not change the member their query is running against.

Fixed a problem with graphs having some portion of their bars missing every 2000 or so intervals of continuous scrolling if the number of bars per page is set to > 100.

If an error is detected in a graph definition, such as a required field does not exist in the SQL statement, the error message will be drawn right onto the graph view now instead of shown in a popup window.

Fixed some problems in the manage query definitions interface such as getting SQL errors when copying many query definitions at once.

Fixed a bug in the query parsing that was causing too many CHKOBJs to be performed when checking for the existence of files. This would slow down the opening of reports and on slower connections could greatly decrease response times.

The hide/show borders option in the graph legend wasn't working if the graph definition was defined such that the field had the same border color as the fill color. If this is the case the color will switch to black and then back to the original color.

If an SQL error occurs when launching any graph, the sql editor/error window should be visible above the graph right away rather than requiring the user to open the editor and reexecute the query.

The option to save a graph as a JPEG (Save As... menu when right-clicking on a graph) has been removed from the GUI. It didn't work since the June builds when the graph control was recompiled to work with the latest MS compiler and I can't easily fix it. The preferred method to achieve this is use Alt-Print Screen which copies the current window to the clipboard. Then you can paste this image into Windows Paint or something similar and save the image to the desired format. Also removed the Copy menu (which copied the graph view only to the clipboard) because you can just use Alt-Print Screen to do this which will include the legend and the copy function did not

Add check in the install upfront for system value QALWOBJRST.

The Graph views section on the Preferences -> Clipboard Tab has been removed since the copy to clipboard of the graph image option was removed from the GUI in favor of telling users to use Alt-Print Screen instead

May 11, 2007 – C00542

Flyover text for long text strings in table views (like SQL statements) should no longer be cutting off text on the right side of the window.

Apr 3, 2007 – C00535

On the Display tab of the Preferences window added an option "Display patterns". When checked the graphs will display line hatchings and patterns instead of solid colors.

Added options to select the desired fill patterns from the legend and graph definition primary Y-axis fields panel. These selections are not savable in a user-defined graph and are only used for the current graph.

The record quick view window no longer has a checkbox to "allow multiple rows" to be compared. This is in effect always on now. If the user selects multiple rows in a table (up to 30) and uses the record quick view menu, they can be viewed vertically.

The window that prompts a user for a member has been revamped. Instead of the user needing to specify the member for every single file found in the query, a list of all tables found in the query is built and then sent to a single dialog. This window allows the user to pick a member from a list and then indicate that it should be used for all files found in the query.

Query parsing should be a bit faster. Removed redundant checking for the existence of file/members in the query.

Fixed some flickering problems in the list of fields on the field selection panel of the query definition interface.

If the query behind the graph is changed/reran and it returns no data, the graph will now display the message "The query behind this graph returned no data" rather showing the graph data from the local cache.

If the graph definition window is resized, the coordinates are remembered for next use.

For readability purposes, the field column in the list for the graph definition -> primary Y-axis fields window now only displays the short field name instead of the long description and the field name.

In the open file window, pressing the enter key will no longer close the window, but will instead trigger the browse button to refresh the list of files

Added a menu option to the legend to hide/show a bar's black border. This can offer the effect of emphasizing a field in the graph

The legend can now be hidden/shown using the toolbar when the legend has focus. Previously this button only was enabled when the graph had focus.

The legend will no longer force lower case characters in field descriptions.

The creation date shown for user-defined queries and graphs should now show the creation time as well as the date

Fixed some multi-threaded timing issues that would cause canceling a long running query to crash the client. As a side effect of this the amount of time the user has to cancel the query is reduced, once the query has executed and it is to the point of building the columns and fetching rows; that part cannot be canceled.

Fields removed from the legend (or dropped into the 'available fields' section) will now retain their properties, such as the field description, and color. These fields are temporary and are listed first among the 'available fields'. If added back to the graph, the field description, color/pattern will become visible again instead of being randomly set.

Feb 12, 2007 – C00520

Heap Analyzer users should now be able to open a Data Viewer, and then open files and write custom queries from the Heap Analyzer component view.

The iDoctor supplied queries/graphs folders should now contain graphs and queries previously only visible for V5R3 collections.

Updated the install so all commands issued are library qualified with QSYS. Though unlikely this could cause a potential problem depending on what was in the library list.

Add the ability to delete an object root finder in the GUI and using the DLTHEAPWCH command.

The Data Viewer now has a new button on the toolbar which provides alternate views for the graph with current focus. Look for the button that looks like a line graph next to the legend on/off button. Clicking this brings up a list of any available alternate views for the current graph. This mostly applies to Job Watcher graphs.

For all components/releases, when throwing an error message after a remote program call in iDoctor, one of the CPF messages was not being returned in the error window.

For all releases, in the client portion of the install program, updated a few remote commands that were issued that did not qualify the library as QSYS, such as the ALCOBJ command. Though unlikely this could cause a potential problem depending on what was in the library list.

Added a preference on the display panel of the preferences interface for the default legend size (as a percentage of the graph window). This preference allows you to show more or less of the legend (from 0 to 50%) every time a graph is opened.

Removed the "reuse these settings" checkbox from the field selection page of the query definition interface. This function has become obsolete since users can now create their own user-defined queries.

When showing a graph of a query consisting of only a single record, the graph would often not be displayed until the screen was refreshed or clicked on by the user.

Fix a problem with calculating record counts on some types of advanced queries.

Fix a SQL parsing error that would happen if a select statement started with an expression having more than 1 set of parentheses. No field name would be displayed in the column heading.

When writing your own queries and defining expressions in the field list of the select statement, you can now leave off the AS <field> after the expression. The GUI will display nnnnn as the short name and the expression as the field description. This fixes the inability to define expressions without an AS clause in iDoctor.

When writing your own queries in iDoctor you can now define field descriptions right into the SQL statement (surrounded by double quotes) and have them be displayed as multiline headings in the table view. Example: FIELD AS "My description"

Updated the query definition's parsing code to ensure that only the outer most where, order by, group by and having clauses are shown in the query definition interface. Previously the parsing would become messed up if inner select statement's where, group by, or order by clauses were shown in the query definition interface and then modified. Modifying the inner selects requires the user to change the sql statement manually (not supported from the graphical interface).

Fix a bug on the group by query definition panel. If fields are built from expressions and used in the group by and then later updated through the interface, the expression's alias name would be passed into the sql statement instead of the group by's expression itself which is incorrect SQL syntax.

The query definition parsing code should now be able to handle expressions in a where clause that have been defined with redundant sets of parentheses.

When reconstructing a query after working with the record selection window, an extra space added between the from clause and the where clause has been removed.

Support added to create user-defined pie charts. The Pie chart is now a selectable option on the general tab of the graph definition interface

Made a change in the query definition parsing such that if the parsing cannot be done successfully, the query definition interface menu options are greyed out. This interface is not intended to be used for all possible types of SQL syntax. This should help prevent users from trying to use the interface after a complex query has been ran and encountering problems because the original SQL statement was not parsed correctly.

Add the create alternate view menu that is shown on all graph view popup menus to all table view popup menus. This will allow a user to either make a copy of their current table view in the same data viewer (perhaps for comparison purposes), or to take their current table view and begin working with the raw sql statement more easily using the table view with SQL editor.

Flyover field descriptions defined in user-defined graphs within the flyover tab of the graph definition interface are now saved with the graph definition.

When defining flyover fields in the graph definition interface, added a check to ensure no more than 10 can be added to the list of flyover fields.

Sept 1, 2006 – C00500

Updated the icons for several of the iDoctor components. Also made it so reports /graphs opened in the Data Viewer will match the icon for the component being opened. Added icons to the iDoctor components window and made it wider to make it easier to see any status error messages.

Made a change so that opening a file or using the SQL query window within Heap Analyzer does not require a valid JW or PA access code. Heap Analyzer now has full functionality in the Data Viewer.

June 19, 2006 – C00496

The flyovers for long cells in table views should look a little cleaner now. It won't wrap the flyover text unless the cell length is at least 100 bytes.

The GUI will now set the QPTFOVR data area to '0' if the user answers in the GUI to NOT override the PTF checking. Previously the GUI would set the data area to a '1' the first time the user overrides PTF checking and did not provide a way to set it back to a '0'. QPTFOVR will also be set back to 0 by the GUI if the GUI detects the PTFs are installed so the user does not get a warning in the collection job log.

May 8, 2006 – C00492

Updated the interface used for copying/transferring a collection so that it has a consistent look for all components that copy or transfer collections.

The menu User-defined reports ->Select local database.... has been renamed to User-defined reports ->Set local database...

Added a new menu option under the user-defined reports menu for all components that allows a user to manage their query or graph definitions. The menu is User-defined reports ->Query definitions ->Manage... OR User-defined reports ->Graph definitions ->Manage...

Added a new manage definitions interface, that lets a user work with their current local query/graph definitions database. This interface allows a user delete or to make copies of their definitions and set them to another release or component. The interface also allows a user to export definitions to the server. These definitions exported to the server can then be imported by other users, by using the menu User-defined reports ->Query definitions ->Import from server... OR User-defined reports ->Graph definitions ->Import from server...

When opening a data viewer, the component the data viewer was opened from will be remembered. When opening a table from the open file window or creating an SQL editor/table split view, and when a query is defined and saved, the default component selected on the save query or save graph window will match the component the data viewer was created from. The default version will be the version of the system, rather than blank if it is not known

The SQL statement within the Query tab in a report's properties is now editable. After making the change the user must refresh the report manually (F5) to pick up the changes in the SQL statement. This applies to both table views and graph views.

In the record selection page of the query definition interface, if a field is defined as an expression but is also a regular field name in the file, the user should now be able to filter on this field when the field was not listed previously in the list of fields.

Fix a bug with the legend where fields were not draggable if certain small fonts were used.

February 28, 2006 – C00479

Add a position indicator in the toolbar of the data viewer. This allows a user to enter the desired record/bar position and press enter or press the button next to the edit box to have the graph or table scrolled to the desired point.

Add popup menu options to build table views from any graph either with the SQL edit split view or without. The menu options are Create alternate view -> Table Create alternate view -> Table with SQL editor

When first opening a table view the widths of the columns should be resized better than before. Sometimes field data longer than the header was not completely visible.

When resizing columns in a table view, they should now be remembered again after performing a refresh.

Rename adjust scale popup menu for graphs to adjust primary scale. Add adjust secondary scale menu option for graphs. This will work just like the adjust scale menu for the primary axis but instead works on the secondary axis. This allows a quick way for user to see above 100% utilization in the CPU utilization graphs (as an example).

February 14, 2006 – C00476

Heap Analyzer V5R4 released.