Adagio

Release Notes

Adagio® GridView 9.2A (2014.08.29)

Upgrade Complexity from Version 9.1A **Easy** ●

Compatibility

Adagio

BankRec 8.1A – 8.1E
ChartView 9.2A
Checks 8.1A – 8.1B
Console 9.1A
DataCare 8.1A - 9.1A
ePrint 9.0A
FX 8.1A - 9.1A
GridView-RW 9.2A
Inventory 8.1A – 8.1E
Invoices 8.1A – 9.1A
JobCost 8.1A - 8.1D
Lanpak 9.2A
Ledger 8.1A – 9.1A
MultiCurrency 8.1A – 8.1B
OrderEntry 8.1A – 8.1F
Payables 8.1A – 8.1F
PurchaseOrders 8.1C, 8.1D
Receivables 8.1A – 9.1A
SalesAnalysis 8.1A – 9.1A
SalesCQ 8.1A - 8.1C
Time&Billing 8.1A – 8.1C

Other

Any product for which an Adagio Data Dictionary has been created
Excel Workbooks
Island Microsystems products
DSN / UDL data sources
MS Office 2007, 2010, 2013
MS Windows Vista, 7, 8, 8.1
Purchase Orders for Adagio 8.1B
Any product that uses the Adagio Data Source (ADS)

Note

The most up-to-date compatibility information is on the Product Compatibility Info link at www.softrak.com.

Enhancements

- GridView can now Combine or Append tables, making them appear as a single row set. In the same way that GETLINKEDVALUE() adds a column to every row, Combine or Append adds additional rows to a View from another table. Combine and Append are accessible from the Edit | Definition dialog on a second tab. Any Tables can be combined. Any fields in the subsequent table that do not exist in the original Table will return blank or zero. Fields with the same name will be automatically matched. You can manually match fields with different names.

  You can use this feature to create a common name and address file from Customer and Vendor tables, Append current sales records to a summary table or join 2 or more Customer or Inventory Item lists in different datasets. Referencing additional datasets consumes a lanpak for each additional dataset added. So for example, combining 3 customer master records will result in 2 additional lanpaks being consumed.

  The workspace `VendorCustomerCombine.gvs` in the sample data provides an example of this new feature.

- GridView can now export a View/Workspace to any ODBC compliant database. A new button "DB" will appear on the toolbar when View | Customize | Reset is performed. Click on the Down arrow button to select whether the Table will Add or Replace (only available with GridView-RW) an existing table with the same name (the name of the View). Click on the DB button to select the database where you want the rows delivered. The Select dialog can choose any Access database or DSN. If a Sort Sequence called "PrimaryKey" is defined in the View, then this will be made the Primary Key in the exported database. Tables are always exported in their entirety to new tables in the database. If a Table with the name of the View already exists, and "Add" is the requested operation, then (n) will be appended to the View name until a unique name is generated. (Tables cannot be updated in place.)

- Workspace and View tabs have been added to the View | Customize Dialog to provide a common place for setting program defaults.

- If multiple columns are selected that represent the key values into an external table, the table will be presented in the list of available links when "File | Insert Related Column…" is chosen. For example, if you select columns containing a general ledger account number and department code, you will be offered the GL Account Master file to link to.

- View and Workspace most recently accessed lists under the File menu are now saved by user, by company.
- Workspaces and Views are now auto-saved while they are being edited so that they may be recovered in the event of a program crash.
- **File | Shortcut | Create** and **File | Shortcut | Edit** menu choices have been added to facilitate the creation of GridView program shortcuts that open workspaces or send them to Excel using the "/XD" program parameters. These choices are especially useful when creating a series of GridView operations as part of an overnight automated process.
- Sorting a numeric or date column now displays the results descending first, then ascending. Text columns are sorted ascending first, then descending.
- Column Titles can now include formulas to allow them to change dynamically. Formulas cannot refer to database fields but may include include dates. (So, for example, this might be used to specify date ranges).
- Column hints (displayed when the cursor is over the column title), now display which other columns reference a calculation and which other columns a calculation uses.
- Minimized Views cannot be restored when a workspace is loaded without "/c" on the command line. This lets confidential information be referenced in a Workspace without being viewable.
- New codes have been added to allow printing View information in the header/footer of a report. These codes are:
    - `$T{format string}` Prints the Run Date formatted according to the format string at the time the View was printed.
    - `$Q{Column name}` Prints the contents of the first row of the named column. This allows you to include View values in the header/footer.

      The following table summarizes the codes that may be used in a View header/footer:

Specification Code	Description
$F	File title
$A	Application GVWin.exe
$P	Current page number
$N	Number of pages
$D	Date - can have parameters in {}
$R	View Title
$L	Filter name
$V	Filter value
$Y	Table name
$Z	Table description
$T	Run date - can have parameters in {}
$Q	Field Value - must have parameters in {}
$C	Report page number

- Subtotals are now inserted when Views are formatted as reports with a sort sequence. The fields used in the sort sequence will result in subtotals being generated when the contents of the field changes.

- A new option allows you to have the repeating values in a printed View suppressed on all rows but the first and total row. The workspace `CustomersAndTransactions.gvs` in the sample data demonstrates this new feature and the new Header and Footer codes available in this release.
- Filters can use the ROW() function to test the number of records that have been selected.
- GridView Views can now be the data source for other Views, in the same way an Excel workbook can act as a data source. This allows multi-pass processing of your data, or regular summarization to speed processing.
- Constraint violation checking has been enabled. Constraints are specified in calculated columns, after the calculated expression by appending a semicolon and the Constraint expression. "#" refers to the preceding formula. For example, if the calculated column contains:

  ```
  {Balance Outstanding};# < {Credit Limit}
  ```

  and the {Balance Outstanding} is greater than the {Credit Limit}, then the Cell will be formatted according to the "Constraint Violation" Style (under **Format | Styles**). This allows you to format a single cell in the View (rather than the whole row). However, there are limitations:

  - Only a single Constraint Violation Style is available.
  - The Constraint Violation Style is not stored with the View, but rather on the Workstation, so it must always be the same Style in different Views.
  - "Calculate after refresh" must be selected on the calculated column.
- Added command line parameter "/O" to allow generation of a sample COM script for external processing. If "/O" is specified on the command line, then right mouse clicking on a Filter name in the Edit Filter dialog results in a context menu that allows generation of a COM scripting example. The file `\Softrak\GVWin\GV_ComDoc.zip` is installed with GridView with examples of using the COM functionality in GridView.
- Added command line parameter "/XM [A | M | R]:<DSN Name>" to allow automatic export of a workspace or view to <DSN Name>. Options are A - Add to database, M - Modify existing table, R - Replace existing table.
- No Page Break" option added to Report Format Filters
- Custom colors can be selected for Styles and Formats
- The Filter Toolbox (Alt-F6) buttons have been rearranged for ease of access and are enabled/disabled depending on what Filters have been selected.
- The Defaults Dialog (F2) has been expanded to allow specification of all default values and to make clear which defaults apply to a Workspace and which to a View.
- You can now choose a Summary Filter in a Linked View.
- Compatible with Adagio Console and respects requests to shut down.
- Includes 64-bit version.

New FUNCTIONs

`SYSDATE()`

returns the system date/time, not influenced by the Run Date.

`FYEAR(`*`date`*`)`

Returns the Fiscal Year for a given date

`FPD(`*`date`*`)`

Returns the period as a string (e.g. " 2") in which a date falls

`FPERIOD(`*`date`*`)`

Returns FYEAR & FPD ("2013- 2") in which a date falls. This is useful when working with transactions in Ledger, or trying to create fiscal calendar summaries.

`ISERROR(`*`x`*`)`

Returns TRUE if the result of "x" is an error, otherwise FALSE.

`ISNUMBER(`*`x`*`)`

Returns TRUE if the result of "x" is an number, otherwise FALSE.

`ISSTRING(`*`x`*`)`

Returns TRUE if the result of "x" is a string, otherwise FALSE.

`FOLDER()`

Returns the path to the data space.

`FILEEXTENSION()`

Returns the extension of the data in the View.

`QUERYDATE("Run date?")`

Sets the run date for the Workspace. (The prompt must be exactly "Run date?".)

`ROW(optional range)`

Returns the index of the current row or the first row if a range is specified.

`COL(optional range)`

Returns the index (base 0) of the column of the formula.

`COLUMN("`*`title`*`")`

Returns the column number of the column with the specified title. Use this in RANGEREF() calls to keep them running properly when columns get moved around.

`CELLREF(`*`Column,Row`*`)`

Returns the contents of a Cell. *Column* zero based, *Row* is 1 base

Example: CELLREF(2,ROW()) will return the value from the current Row and the "third" column, more useful when Calculate after refresh is checked.

RANGEREF(*c0, r0, c1, r1*)

Returns a reference to a range of cells from column0, row0 to column1, row1. This would be used as an argument to the SUM() function to total a range of cells to another cell.

APPLYMASK(mask, string)

Applies a mask to a string (like Crystal Reports for Adagio). Masks use lowercase "l" as a placeholder with all other characters being insertion characters. "<" and ">" may delimit the string and will not be inserted if they are at the beginning / end of the mask respectively. Mask may be a literal string or the contents of a database field or formula.

APPLYMASK() is useful to format phone numbers or item numbers:

Example	Result
`APPLYMASK("(lll) lll-llll",{Phone})`	Formats a 10 digit phone number
`APPLYMASK("lll.lll-lllllllll",{Item #} )`	Formats an item number

`APPLYMASK(`
`GETLINKEDVALUE(`
`"GVICGlobal","ICGLOBAL",`
`"Item Format",0),`
`{Item #})`
	Formats an item number according to the formatting information stored in the database.

TABLEFIND(text, CaseFlag, [ReturnItem,] "View", "Column" [, "Column2…"])

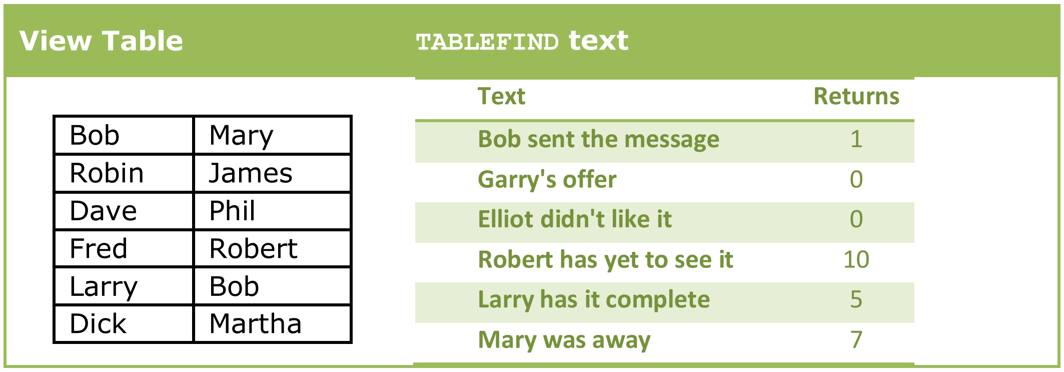
This new function allows you to lookup a value from another table with an arbitrary number of rows. The sample workspace TABLEFIND.gvs uses the TABLEFIND() function to locate all transactions in the GL where the staff member's name is in the description. The parameters are:

Argument	Usage
Text	The field you want to search
CaseFlag	An integer 1 for case sensitive search, otherwise 0.
ReturnItem	A number in this position causes the function to return the value that matched, rather than its index position. If this parameter is not specified, then the index of the item is returned.
"View"	A text string containing the name of the View to search (NB: Do not include the .GV extension)
"Column"	A text string with the name of the column containing the values to look for. You can specify an arbitrary number of columns

Returns

0 if the string is not found, otherwise an integer value of the row the element was located in. If more than one column is specified, then the value returned is the column index times number of rows + row number

For example:



View Table

Bob	Mary
Robin	James
Dave	Phil
Fred	Robert
Larry	Bob
Dick	Martha

TABLEFIND text

Text	Returns
Bob sent the message	1
Garry's offer	0
Elliot didn't like it	0
Robert has yet to see it	10
Larry has it complete	5
Mary was away	7

MATCH(*value, array, [Type]*)

Used to lookup values in a table where the values being inspected are approximate. For example, you need to find the commission payable on a dollar value of sales, and the commission levels "step" between discrete values.

Value - The value that you want to match in Array. For example, when you look up a book in the library, you are using the name of the book or author as the lookup value, but the "location" is the value you want.

The Value argument can be a value (number, text, or logical value) or a cell reference to a number, text, or logical value.

Array - The range of cells being searched.

Type - Optional. Number -1, 0, or 1. The Type argument specifies how the Value is matched with values in Array. The default for this argument is 1.

Returns is dependent on the type specified.

1 or omitted - Returns the largest value that is less than or equal to Value. The values in the Array argument must be placed in ascending order, for example: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE

0 – Returns the first value that is exactly equal to Value. The values in the Array argument can be in any order.

-1 – Returns the smallest value that is greater than or equal to Value. The values in the Array argument must be placed in descending order, for example: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.

FILTERNAME([*type*])

Takes an optional numeric parameter for use with Summary Filters

- **FILTERNAME(1)** returns the name of the detail record Filter
- **FILTERNAME(2)** returns the name of the Summary Filter

Problems Fixed

- Row formatting works more reliably in Summary Filters
- Column formats are more reliably saved.
- Calculated columns cannot be deleted while they are used in other calculations, Filters, Row Formats or Sort Sequences. Column hints (displayed when the cursor is over the column title), now display which other columns reference a calculation and which other columns a calculation uses to assist in determining where a calculation is used.
- Excel workbooks referenced in a View were not released until GridView was closed. They are now released when all Views are closed.
- Tiling Windows now properly positions Views when a toolbar is docked at the side of the desktop
- ExcelDirect now always names the target sheet.
- ExcelDirect now correctly formats columns containing percent values.
- ExcelDirect now formats columns containing character data as text columns to preserve the presentation of gl account and phone numbers (for example).
- Error messages are clearer when a View into a composite table fails to open due to missing secondary files.
- The "Display total" option is now suppressed when a column is formatted as a percent
- You can now press Escape to cancel the initial load of a View when "Refresh on open" is enabled.
- Summary Filters would revert to the default column for Top/Bottom N selection if the Filter was edited.
- GridView now remembers the Folder most recently used when a View is saved.
- Row formatting is now applied even if the View load is cancelled by pressing Escape.
- If AutoRefresh or Refresh on Open was enabled for a View using one of the QUERYxxx() functions, the user would be prompted for values twice.
- Workspace window positions are now saved for Filter Toolbox, Row View, and User Notes.
- Clicking the "Most recently accessed" list on the Login Dialog no longer results in an error when there are no entries in the list.

Enhancements and revisions in earlier versions

Refer to the Readme help file for details of enhancements and revisions in earlier versions or Historical Enhancements under Products on Softrak's web site.

Installing

To install Adagio GridView, you need the serial number and install code from the download email sent to you by Softrak Systems.

If you are installing an upgrade, you need the serial number and install code from the upgrade download instruction email. A previous version of Adagio GridView should already be installed on your computer before installing the upgrade. If a previous version is not already installed, you can still install the upgrade. After entering the upgrade serial number and install code, you will also have to enter the serial number and install code from your previous version.

Read the 'Upgrading' sections below before installing an upgrade.

To install Adagio GridView, an upgrade, or Service Pack, run the program file you downloaded. Once the install program is running, you can click the View button for detailed installation instructions in the Readme help.

If Adagio GridView is installed on a network, you must also do a workstation install at each station running Adagio GridView. Adagio GridView does not support the auto-workstation install feature found in other modules. You must explicitly do a workstation install. Refer to the manual or Readme help for workstation install instructions. Note: If you are updating multiple applications at the same time, it may be faster to run the All Workstation install to update all applications at once on the station. Refer to "Installing on a Network" in the Readme for instructions.

Upgrading

Upgrading to this Version from 9.1A Upgrade Complexity: Easy ●

You need to reset the toolbar to show new buttons added in 9.2A. Select **View | Customize** from the menu; then select each of the Toolbars' options and click the Reset button.

Adagio GridView 9.2A requires Adagio Lanpak to be version 9.2A or higher. Do not install this upgrade before you have installed Lanpak 9.2A if you require multiple users in Adagio GridView. If you have any prior version Lanpaks installed but no 9.2A Lanpaks, the Adagio GridView install will warn you.

Adagio GridView-RW 9.2A (if used) is required for use with GridView 9.2A. Earlier versions of GridView-RW are not compatible. Do not install this upgrade if you do not also have your GridView-RW 9.2A upgrade available to install.

Adagio ChartView 9.2A (if used) is required for use with GridView 9.2A. Earlier versions of ChartView are not compatible. Do not install this upgrade if you do not also have your ChartView 9.2A upgrade available to install.

Views and Workspaces in 9.2A use a new file format. Views and Workspaces created with earlier versions will be automatically converted. **Views and Workspaces created with 9.2A cannot be used in earlier versions.** If 9.2A Views or Workspaces are opened with earlier versions, error messages will be displayed saying "a newer version of GridView is required to open this file".

Upgrading from earlier versions Upgrade Complexity: Intermediate ■

Views and Workspaces created with earlier versions can be used with 9.2A and will be automatically converted. Some adjustments may be required, as described below. **Views and Workspaces created with 9.2A cannot be used in earlier versions.** If you open a 9.2A View with an earlier version of GridView, an error message is displayed (for example: 'MyViewName.gv' was created with a newer version of GridView. A newer version of GridView is required to open this file.).