

Release Notes: WOW 6.5.03

Enhancements, Changes, and Fixes From WOW 6.5.02 to WOW 6.5.03

Enhancements

E1) BasicRowFilter enhanced to allow overriding and CONTAINS filter. Added method getComparisonField(Row) to allow subclasses to provide their own logic if need be to return a different field to compare values with instead of the default. In addition, a new filter type CONTAINS was added to check to see if a Field's value contains the given value.

E2) ReportBreak text override. Ability to override the text used for report breaks. Instead of SUM(or other SQL functions) you could say Total Age or any text you want. There is the ability to change the text for both regular report breaks and overall report breaks.

E3) HTML Code Associations on Tabs. Enhanced tab operation support to allow HTML Code Associations to show up in tabs.

Fixes

F1) Replacement library not honored when creating Table object program. When attempting to get or create a Table object through Table.createTable(String, String, String), an incorrect Library is used if a replacement library was passed in for library name. Logic changed to see if there is a replacement library for the given library.

F2) ConnectionPool fixes. Previously connection was added to manager before it was even available. Also connections were checked back in after a timeout and new connection already created in its place. DBSystem added as connection listener. Connection listeners notified before and after insert/update/delete.

F3) Numeric getValueAs methods on Field modified for non numeric values. Fields may or may not hold a numeric value underneath the covers. When they don't, logic changed to get the internal value as a Number before converting it to a short, int, or double.

F4) Fields with empty string skipped in Excel download. Empty strings were not being displayed as fields at all so those fields got skipped. Instead a space is now used, so field is not skipped.

F5) Skype formatter missing 1 for US phone numbers. Modified because US phone numbers require a prefix of 1 and also modified generate of call button.

F6) Parent page and child ops disappear. When a child operation search was done, the next URI was showing only the child operation. It has been changed to point to the parent operations jsp file to keep the parent and other child operations showing.

F7) --Choose-- missing from HTML select for required fields that have a blank value and have possible values. --Choose-- is now included in the HTML select drop down when a field is required and has possible values.

F8) ReportBreak text doesn't show when Rows are not selectable or have any actions. HTML table column was not being included if no Rows are selectable and there are not any actions. This was causing ReportBreak text to not show up to the far left of the report break row.

F9) HTML table syntax issue with selected row. A couple HTML hidden inputs and JavaScript code were incorrectly being generated between a <tr> and <td> element. They were moved before the <table> element of that <tr>.

F10) SQL HASH not recognized as key word. SQL token HASH was not be recognized as a scalar token

F11) Incorrect HTML syntax for input disabled and readonly. Generated HTML inputs contained incorrect syntax for disabled and readonly elements. Should be disabled=disabled and readonly=readonly instead of disabled and readonly by themselves as before.

F12) Joined inserts, updates, and deletes not allowing option to exclude a given table. Logic was changed to allow subclasses to return 'null' for the row subclass to use for disjoint rows. The parent row will then ignore inserts, updates, or deletes for a given table if there is no row subclass.

F13) MS Access FieldDescriptors could not be created. MS Access returned empty string for the library name instead of null. Additional check added for this and library name set to --Default-- to allow proper create of Fds for files without libraries.

F14) XLS (Excel charting) missing last row of data. Last row of data was not being exported.

Changes

C1) Connection Listeners now managed by DatabaseManager instead of ConnectionPool. Connection Listeners now managed by DatabaseManager instead of ConnectionPool

C2) toUpperCase removed DataEngine.fixSQLMetaDataLibrary. Library names are already upper cases therefore no need to upper case them again.

C3) Field in charge of auto increment methods. Logic changed in a few places to let a field handle the incrementing of it's own value. Allows for a common entry point where a Field or Row subclass may do additional stuff when a Field's value needs to be incremented. By default, a Field uses it's FieldIncrementor as it did before to get its next value. Likewise, a Field is now asked if it is auto increment instead of it's FD.

C4) FieldIncrementor maps moved into DBSystem object. DBSystem object now holds FieldIncrementors used for a system rather than the FieldIncrementor class itself.

C5) CacheManager performance enhancements. Logic modified to no longer clone a cache key before look up.

C6) Row.initialize methods. Initialize methods on Row made public to allow custom programming to initialize a Row manually.

C7) Change start up message. Copy right year set to current year instead of hard coding.

C8) DESecurityManager.checkAccess logic returns error. Previously if an error occurred attempting to check if user authorized to action on table, false was returned which led to confusion to developers as to why things were not working correctly. Logic change to instead throw a RunTimeExcpetion with the original exception.

C9) Check for keys using best row identifier. Sometimes database metadata doesn't return any primary keys when calling getPrimaryKeys. If and when this occurs, a secondary check will be made to getBestRowIdentifier to use as the primary keys. Helps eliminate user errors of forgetting to set key fields when creating FDs.

C10) Modified default logic for XLS charting to grab all rows. Previously XLS charting only took the rows from the current RowCollection. Now all Rows are obtained and written to the Excel file.

Release Notes: WOW 6.5.02

Enhancements, Changes, and Fixes From WOW 6.5.01 to WOW 6.5.02

Enhancements

E1) PV Search Allowed For Search Fields. Logic added to support PV Search operations for search parameter fields. Previously PV Search could only be used on Row detail screens.

E2) Disabling Prepared Statement Use. Some ODBC connections do not return proper result set metadata when using prepared statements. Method added to ConnectionRow to allow subclasses to override and disable the use of prepared statements. See ConnectionRow.isAllowPreparedStatements() below.

E3) SQLContext Allowing NULL Flag. Added ability to set a flag on the SQLContext property group that denotes NULL values are not allowed for SQL parameter values.

Fixes

F1) SQLServer 2005 JDBC driver. Modified creation of DBSystem object to recognize the SQLServer 2005 driver as a SQLServerSystem. NOTE: To use 2005 driver, it needs to be added to connection drivers PV list and correct JDBC jar needs to be in wow65/WEB-INF/lib.

SQL to SQL Server(2005) to connection drivers PV list -

MySQL Metadata System: INSERT INTO pjsys64.depvdata (sysalias, libnam, tblnam, fldnam, type, `key`, value, dspval, dspvalsqft, owid, dspord)

*VALUES ("", "", "", 0, "*DRIVER*", 'com.microsoft.sqlserver.jdbc.SQLServerDriver', 'SQL Server (2005)', 1, -1000, 0);*

iSeries Metadata System: INSERT INTO pjsys64.depvdata (sysalias, libnam, tblnam, fldnam, type, key, value, dspval, dspvalsqft, owid, dspord)

*VALUES ("", "", "", 0, "*DRIVER*", 'com.microsoft.sqlserver.jdbc.SQLServerDriver', 'SQL Server (2005)', 1, -1000, 0);*

To download latest SQL Server 2005 JDBC jar - <http://msdn.microsoft.com/data/jdbc/>

F2) PV Search Usage Id. Usage Ids were not properly being handle for PV Search.

F3) PV Search field name doesn't match. Field name in PV search results might not match name of field. In cases where this is true, logic was added to use the first field in the PV search results.

F4) Error getting complete Row or RowCollection with SQLServer (while attempting an update of a database record(s)). Logic changed to ask the DBSystem object for the table separator in both Row.getCompleteRow and RowCollection.getCompleteRowCollection. SQLServer uses two '.'s. Previously was always using '.'

F5) RowCollection.getSum() incorrect totals. Method was including ReportBreakRows. Logic changed to ignore ReportBreakRows.

F6) ReportBreaks not summing correctly between pages. Logic changes to ReportBreak, ReportBreakCollection, ReportBreakPropertyGroup, and ReportBreakCollection

F7) Excel downloads not including all pages. Download would only include the current page to the last page. Any previous pages were not included.

F8) Result set already closed. Some ODBC connections close their resultset after retrieving data. Logic changed in DataEngine to not manually close the resultset if already closed.

F9) Missing JDBC SQL Types. Added two return types, IBMPACKED and BCDDATE, that a KISAM ODBC driver was returning. Also modified logic so if an unrecognized type is found, instead of throwing an exception, the unknown type is logged and StringField was used for its field class so at least the data would show.

F10) FieldIncrementor not able to handle Double. Logic added to FieldIncementor to allow incrementing a Field's value when the value is a Double.

F11) NULL Library for FD. Logic added to set the FD's library name to --DEFAULT-- upon insert if the FD does not already have one. Some ODBC connections do not return any column metadata information for TABLE_SCHEM. Also changed to ask the FDs Table object for the table and library name as the FD might not have a library for some connections (as in ODBC) in which case the table will return --DEFAULT--.

F12) Allow < right after ? Parameter. No need for an extra space after a parameter when using HTML operations. For example, previously had to put a space after the parameter name. ?? firstName some other html. Now the space isn't needed in an html operation. ?? firstNamesome other html

F13) Missing Field in Row for RowParameter. In some cases, custom programming might be pulling values from multiple associated Rows. Logic changed to only log a warning when a missing Field is not found in a Row rather than throwing an exception. Allows a programmer to use a second Row if needed to set any missing parameter values.

F14) Secured child operations showing up. For an ExecutionGroup, child operation security was not being honored. Logic modified to check if the user is allowed to view a child operation before it is included.

F15) Property Groups not carried over to getCompleteRow. When getting a complete row, property groups are now correctly moved into complete row.

F16) Field class parameters not ignored when guessing field class. Logic changed to ignore field class parameters when attempting to guess field classes.

F17) NULL ExecutingContext in Field.isReadOnly. Field.isReadOnly was passing NULL to Field.isReadOnly(ExecutingContext). Now the current ExecutingContext is passed instead.

F18) NULL ExecutingContext in Field.isValidLength. Field.isValidLength was passing NULL to Field.isValidLength(ExecutingContext). Now the current ExecutingContext is passed instead.

F19) NULL ExecutingContext in Field.isRequired. Field.isRequired was passing NULL to Field.isRequired(ExecutingContext). Now the current ExecutingContext is passed instead.

F20) NULL ExecutingContext in Field.isValidValue. Field.isValidValue was passing NULL to Field.isValidValue(ExecutingContext). Now the current ExecutingContext is passed instead.

Release Notes: WOW 6.5.01

Enhancements, Changes, and Fixes From WOW 6.5 to WOW 6.5.01

Enhancements

E1) Time Field allows CHAR SQL types. Enhancement made to TimeField to allow numeric CHAR SQL types to the TimeField Class. Previously only worked with TIME SQL type.

E2) Allow Tabbed operations to be directly executed from Next Operation and code. Tabbed Operation enhancement made to allow Tabbed Operations to be executed directly from code and from another operation's "Next Operation" setting. Previously the only way to invoke a Tabbed Operation was through the main navigation bar or through an association set on a Field.

Fixes

F1) Null no longer included in validation required string. Modified Field.getValidationRequiredString() to only include Field's value if it's not Null.

F2) DB Auto Increment value now set on field. Row.getAutoIncrementDatabaseDrivenField(ExecutingContext) previously never returned database driven auto increment field to set value after insert.

F3) NullPointerException occurs when attempting to execute a context without any context. NullPointerException handled when AbstractContext's code is set to null. WARN messages are now written to the log to warn developers of an AbstractContext attempting to be executed when it doesn't contain any code (HTML, SQL, etc...)

Release Notes: WOW 6.5

Enhancements, Changes, and Fixes From WOW 6.4.3 to WOW 6.5

Enhancements

E1) Auto Increment Initial Value. A field's default value can now be used as the auto increment's initial value. This can be set by overriding the Field subclass or by setting a default value on the Field's FD.

E2) WOW Show/Hide Side Steps. Functionality added to the WOW application builder to show and hide the side steps. Allows developers to gain more screen real estate.

E3) Table Authority. Access for running an SQL against a specific table is checked against the table descriptor before running. This allows developers to provide an additional level of security to their application files through WOW by using Table FDs.

E4) Go To Next Operation. Allows a developer to specify the next operation that should run after an insert, update, or delete of a given Row. This allows for simple flow control within WOW without having to add a lot of custom coding (just set Next Operation on an operation).

E5) Database Driven Auto Increment. Database identity fields previously caused a problem for WOW. A developer now has the ability to specify a Field as "Database Driven" auto increment, "WOW" auto increment or "None."

E6) Graph Formatters. These new field formatters allow the user to turn any numeric fields into vertical or horizontal bar graphs. This is done easily by simply selecting the desired formatter in the FD of the desired field.

E7) Execute Only. A new execution rule was added for operations providing the ability to just run an operation without prompting. This type of operation works well for associated inserts, updates, and deletes where the prompt parameters are actually set within code or grabbed from some other source (like the default value) where user prompting is not needed.

E8) SQLContextPropertyGroup. For some complex SQL statements, the WOW parser is not always able to determine the tables to use for Field FD's. In these cases, a SQLContext property group can be used to specify the appropriate tables. NOTE: This replaces the StoredProcedure property group since SQLContext can be used for stored procedures (as well as any other SQL operation).

E9) Possible Value Search for HTML Text Areas. Added support for possible value search on fields using the "Text Area" display component. Previously, possible value search could only be used on standard input fields.

E10) Row.getCreationContext(). Logic has been added to Row to track what SQLContext the Row was created from. This allows for a Row to get additional information when initializing, validating, and logging. Example: if an error occurs during initialization, the Row can find out what operation it is being creating from.

E11) Row.setCreationContext(SQLContext). See Row.getCreationContext().

E12) Row.setCreationOperation(Operation). Method added to set the Operation the given Row was created from. When getOperation called, the creation operation will be returned.

E13) RowCollection.getLastRow(). Method added to return the last Row in the RowCollection.

E14) ApplicationManager.getApplication(int, String, String). Convenience method to ApplicationManager to allow retrieval of an Application for a given system, user library, and application id.

Changes

C1) Log4J Configuration. Modified the log4j configuration to include the date.

C2) Message JSP for Default Theme. Modified the default theme's template to allow a message JSP override through the layout property group. Previously the default message JSP was the only one that could be used.

C3) RowCollection.isRefreshing(). Method changed from private to public. Allows other classes to ask a RowCollection if it is refreshing.

C4) RowCollection.setPersistent(boolean). Method changed from default to protected.

C5) RowCollection.setRefreshing(boolean). Method changed from private to protected.

C6) Direct Execution of Operations. Execution Groups and Associated Execution Groups can now be run via URL similar to SQL and HTML Operations.

C7) Possible Value Search. Property group was added to allow multiple fields to be populated from the possible values rather than just one field.

Fixes

F1) Timestamp value not cleared in MySQL. Timestamp field not being set to "null" or zero timestamp when a timestamp Field value is cleared or blanked out (MySQL only). This is a result of the way MySQL handles a "null" timestamp parameter through JDBC. In order to clear a timestamp value or set a timestamp database value to null, the zero date needs to be passed instead of null. '0000-00-00' instead of "null".

F2) Overall Report Breaks show on every screen. They only should show on the last page. To add, previously the last page only showed totals for the last page, not the whole query. This has also been fixed. NOTE: also fixed for Excel download.

F3) Default parameter HTML syntax error. default_param.jsp was missing a " on the HTML button for search, which caused "enter" key to not function properly.

F4) PDF download not including all results if on last page. When downloading to PDF, if user was on last page of results, only that page would be included. Logic changed to also make sure there aren't any previous result pages.

F5) RowKey not serializable. RowKey class was not serializable causing issues in saving session objects.

F6) != not registering as SQL comparison. Previously != was not registering as a valid SQL comparison requiring SQLs to always use <> even though != is valid SQL syntax.

F7) NullPointerException after SQL error where no application exists. A

NullPointerException was occurring when an attempt was made to get a user message from OperationMessageManager when there was no current application.

F8) New window functionality for HTML associations. Browser properties were not being used properly for HTML associations. They would always open in the same window rather than a new window if specified.

F9) CLOB and BLOB support for Oracle. CLOB and BLOB support existed in future versions but were disabled for Oracle. They are now enabled.

F10) Log Off showing on Top Menu. Log off link was showing on top menu even for unsecured applications.

F11) Exception JSPs throwing StringIndexOutOfBoundsException.

<dataengine/jsp/display_exception.jsp> and <dataengine/jsp/main_exception_handler.jsp> were throwing a StringIndexOutOfBoundsException due to the original exception not having a message value larger than 2000.

Release Notes: WOW 6.4 SP3

Enhancements, Changes, and Fixes From WOW 6.4.2 to WOW 6.4.3

Enhancements

E1) Operation Actions. These are similar in appearance to RowCollection actions, except when the user clicks an Operation action button, the Operation.handleAction() method is invoked (instead of RowCollection.handleAction()). Also, RowCollection actions can only appear when a RowCollection is being displayed to the user, but Operation actions can be shown in more places.

E2) SkypeTelephoneFormatter. Allows a direct call to a phone number using the popular internet telephony software, Skype. The formatter class is planetj.formatters.SkypeTelephoneFormatter.

E3) Joined Inserts/Updates/Deletes. The standard row collection actions (inserts, updates, and deletes) are now available with joined results. Previously, if two or more tables were joined in the SQL, these three actions were not allowed.

E4) FD ?? Parameter Support. In a field's field descriptor, the Auto Update and Default Value properties now accept the ?? parameter allowing the user to pull in values from a source row.

E5) Display Update Table Property. The Boolean property "display update table" has been added to the TableDisplay property group to control whether or not the "Update Table" button should be included when generating a results table. For example, TableDisplay { display update table: false; } will hide the Update Table button.

E6) Row.create(SQLContext, List, boolean). Added the static convenience method, Row.create(SQLContext, List, boolean), to allow for creating a new Row from properties in a given SQLContext.

E7) SQLContext Tracking Logic. Logic [getCreationContext(), setCreateContext(SQLContext)] has been added to Row to track what SQLContext the Row was created from. This allows for a Row to get additional information when initializing, validating, and logging. Example: if an error occurs during initialization, the Row can find out what operation it is being creating from.

E8) ConvertUtility.convertToDouble(String). Added convenience method ConvertUtility.convertToDouble(String). Normally a NumberFormatException occurs when attempting to parse a String that contains a comma (e.g. "1,000"). This method removes the comma and then calls Double.parseDouble(String) to convert the String value to a double.

E9) Javascript Cookie Methods. JavaScript convenience methods were added for creating a cookie, removing a cookie, and checking if cookies are allowed: readCookie('<cookie name>'), eraseCookie('<cookie name>'), cookiesAllowed().

Changes

C1) Operation Display Group Default. Rather than "Default" being the default value, if the developer does not set a "Display Group" on an operation, the operation's label is used by default.

C2) RowParameter. RowParameter was modified to provide better logging and user messaging when a developer incorrectly uses a "Usage Id" RowParameter. (e.g. ??^first_name is invalid. Usage Ids must be numbers. ??^10 for example).

C3) HTMLField.appendDisplayValue. HTMLField.appendDisplayValue(String, Field, StringBuffer, HttpServletRequest, HttpServletResponse) was modified so it only appends an HTML checkbox for a BooleanField if the Field's display component is default.

C4) PhoneNumberFormatter. PhoneNumberFormatter can now handle long distance numbers consisting of 11 digits. (e.g. 1-760-432-0600).

Fixes

F1) Field.isPrepareFieldValue. Fix added for Field.isPrepareFieldValue(ExecutingContext) to check and return Row.isPrepareFieldValue(ExecutingContext) when Field has a Row.

F2) Possible Value. Fix for row details screen that originated from a horizontal parameter search. "-- All --" option was being include on details screen instead of "--None--" or "--Choose--".

F3) AS/400 Casing. Addition changes made to a fix put into 6.4.02 that uppercases the user id AS/400 uses when signing in so it may be referenced using triple question mark in SQL operations.

F4) Number Field isValueEqual comparison. Previously any number field could not compare a String representation of a number to its value. For example, this would cause "1" not to be equal to the fields value when it was 1, because of the String value. NumberField was modified to allow for such a comparison.

F5) NullPointerException. NullPointerException fix. Preparing a new row for insert from an operation was causing a white screen in some cases.

Release Notes: WOW 6.4 SP2

Enhancements, Changes, and Fixes From WOW 6.4.1 to WOW 6.4.2

Enhancements

E1) Possible Value Paging. Previously, when showing possible values for a pick list, any rows after the specified row count (defaulted at 50) in the Possible Values operation would not be shown. Now, these extra rows are accessible by 'next and previous' links included in the pick list. For example, let's say your possible values operation returns 100 records. You could set the row count to 10 and only 10 records at a time would be shown in the pick list.

E2) Please Wait Page. Some operations take a while to run and return results. For these operations, it might be helpful to set up a 'please wait' page for the user, so they know that the request is still being processed. Sometimes users are impatient and/or think the request failed, so they click the button again. This helps prevent that.

E3) Possible Value Search Operation. Some possible value operations might return a large number of choices. Rather than having a user scroll through a list of say 500 possible values, you can set up a Possible Values Search operation. Instead of the possible values showing up in a pick list or drop down, a button is included next to the field that users can click to go retrieve that field's value. This function opens a new window and allows the user to select the value for the field. As a side note, the new window can allow for searching. For example you might be looking for a contact. You could allow the user to search by city and name to narrow the amount of possible values shown.

E4) Numbered Paging. When showing results users usually don't know how many results there are. Numbered paging allows the developer to specify whether or not the user can see page numbers, the current page of results, the total number of results, or even link directly to another page, directly to the first page, directly to the last, and control what kind of descriptive text is shown.

E5) HTML Code Association Operation. This highly anticipated enhancement combines the abilities of the HTML Code operation and the Association operation. Users can now create fully dynamic HTML pages that can easily take advantage of row parameters and all other association features. This enhancement is ideal for just about any situation requiring data to be displayed in a non-table format. See Full Field Rendering below for more details.

E6) Full Field Rendering. This enhancement is part of the HTML Code Association operation listed above. It allows row parameters (fields) in an HTML Code Association operation to fully render all their specified field descriptor properties. For instance, full field rendering will allow a row parameter (field) in an HTML Code Association to show its association hyperlink rather than just its plain text value.

E7) SQL COMMENT ON Statement. Added support for SQL syntax for COMMENT ON statements for tables and table columns.

E8) SQL LABEL ON Statement. Added support for SQL syntax for LABEL ON statements for tables and table columns.

E9) Paging Justification. Ability to control which side of the page paging links show up on (left or right).

E10) Added method - Row.createNamesActionRequest(String, ExecutingContext). Method added to allow easy creation of named row action requests that can be used on custom JSPs.

E11) Added method - RowCollection.createNamesActionRequest(String, ExecutingContext). Method added to allow easy creation of named row collection action requests that can be used on custom JSPs.

E12) Added method – RowCollection.getMaxValue(String). Method added to RowCollection object to allow a developer to get the maximum value for a specified column.

Changes

C1) Detail Button JSPs. When Row detail pages are generated, the button JSPs are always included to allow for any custom Row actions to be shown on the detail page regardless if any actions are set to true on the DetailsDisplay property group.

C2) Distributed Objects Removed. Any custom Row or RowCollection subclass having any insert, update, and delete methods that overrode the parent class and threw a DistributedException will need to be changed. There was old code in the WOW framework that dealt with a distributed feature (that was never fully developed) that has been removed.

C3) Preparing Field Values. When a field value is about to be displayed, normally its value is “prepared” to allow the proper HTML rendering. In some cases, a field's value should stay as is. For example, if the field contained HTML code, we wouldn't want to modify its value when generating. For that logic, a field could control whether or not its value was prepared. A lot of times though, rather than a separate field class, that logic would be nice to control via a field's formatter. A hook was added in to ask a field's formatter class if its value should be prepared.

C4) Default TextArea component. Previously when creating field descriptors, any time WOW encountered a character field with length greater than 50 the field descriptors display component was defaulted to Text Area. This number was increased to 100.

Fixes

F1) WOW Start Up Message. Start up message in log file changed to reflect the actual version of WOW being run.

F2) Database types added. INTEGER UNSIGNED was added as a valid database type. Previously INTEGER UNSIGNED was an unrecognized type and caused an error when reading in a records value.

F3) Repeatable Actions. Some magic requests and actions are not repeatable, such as an insert. Users, however, would hit the refresh button on the web page after the insert, which

causes the browser to send the exact same request to the web server. This was causing duplicate records to insert. Modifications were made to ensure some actions and magic request can not be repeated.

F4) Number Field isValueEqual comparison. Previously any number field could not compare a String representation of a number to its value. For example, this would cause "1" not to be equal to the fields value when it was 1, because of the String value. NumberField was modified to allow for such a comparison.

F5) TableParameters. Field descriptors were modified so possible values could now be set for a table FD. In addition, two fixes were added. One that prevented a "NullPointerException" from occurring when attempting to execute an SQL operation containing a table parameter. And another that prevented table parameters from running on any system that required a library to be specified in the SQL.

F6) User Id Value not being propagated. Using operation system sign on, when an attempt was made to get the user id from the signed in user object, no value was found. A fix was applied to return the operation system's user id. For example, the iSeries user profile that was used to sign into the application.