RSC Sync Engine / SQL Automation Engine rev 2.168
RSC SQL Watch Engine 1.04

Author - Geoff Dryer

Overview

The RSC SQL Automation Engine is a standalone executable that processes SQL statements for ARCHIBUS. The program is configured on a single server or workstation as a minimized application or a service. As an application it runs minimized on the task bar. Right clicking on the icon brings up the menu from which the application can be stopped.

The RSC Sync Engine is a standalone executable that permits data to be imported from Excel spreadsheets and moved between tables via a programmable interface. The most common use for this is employee synchronization but it can also be used for any business process.

The RSC SQL Automation Engine and the RSC Sync Engine are combined into a single engine. An RSC domain (Navigator) is included to access the configuration tables.

A sample Sybase project RSCSQLAUTO is included and referenced in this documentation

The program is configured on a single server or workstation as a minimized application or a service. As an application it runs minimized on the task bar. Right clicking on the icon brings up the menu from which the application can be stopped.

NOTE: Use the menu to stop the engine otherwise data may be lost. If running as a service ensure that the engine is idle before stopping.

The engine processes work orders via a timer. The timer and other parameters are configured in an INI file.

The engine writes all errors to a log file (RSC_SQL_AUTO_LOG.TXT). Errors can also be automatically emailed to a support person.

Installation

Installation and Configuration of the Files and Schema

NOTE: ARCHIBUS/FM is required in order to configure the ODBC Entry. A System DSN called "RSCSQLAUTO" is created.

If you have the installation files as separate zip files then perform the following:

1a) Unzip the following into the suggested directory structure:

RSCSQLAUTO.ZIP - Server Engine. C:\PROGRAM FILES\RSCSQLAUTO
SCHEMA.ZIP - AFM Schema files C:\PROGRAM FILES\AFMXX\SCHEMA\RSCSQLAUTO
SAMPLEDB.ZIP - Sample database C:\PROGRAM FILES\AFMXX\PROJECTS\RSCSQLAUTO
IMPORT.ZIP - XML files for custom fields and Navigator C:\PROGRAM FILES\RSCSQLAUTO\IMPORT
WEBCENTRAL.ZIP - Files for Web Central C:\PROGRAM FILES\RSCSQLAUTO\WEBCENTRAL

OR if you have the installer executable (e.g. RSCSQLAUTO69.EXE)
1b) run this executable and follow the prompts.

2) Open ARCHIBUS/FM and the project database.
Go to File - Data Transfer and import the following XML Files (C:\PROGRAM FILES\RSCSQLAUTO\IMPORT folder)

AFM_TBLS.XML
AFM_FLDS.XML
NAV1.XML through NAV5.XML

3) Run the Update Schema Wizard for the files listed in the TAB_LIST.TXT file.

4) Open ARCHIBUS/FM and the project database. Go to File - Data Transfer and import the following XML Files (C:\PROGRAM FILES\RSCSQLAUTO\IMPORT folder)

DATA1.XML through DATA8.XML - These files are sample definitions for an Employee Sync

Configuration of the Processing Engine

NOTE: ARCHIBUS/FM is required in order to configure the ODBC Entry. A System DSN called "RSCSQLAUTO" is created.

1) Complete the Installation and Configuration of the Files and Schema (see above)

2) From the install folder copy VBSENDMAIL.DLL MSWINSCK.OCX to the \WINDOWS\SYSTEM32 folder.

3) From the install folder copy RSCSQLAUTO.INI to the \WINDOWS folder.

4) Note: Skip this if you installed the SQL Automation Engine from the installation routine.
From a command prompt (in the \WINDOWS\SYSTEM32 folder) register the email program DLL files using the following commands:

REGSVR32 VBSENDMAIL.DLL
REGSVR32 MSWINSCK.OCX

5) Open ARCHIBUS/FM and the project database. Go to File - Run Basic Script and choose the file RSCREGSet.abs.

6) From the \WINDOWS folder open and edit the RSCSQLAUTO.INII file. The following parameters are supported:

```
[RSCSQLAUTO]
PATH=C:\Program Files\rscsqlauto          ;path where the application is installed [required]
LOG_FILE_DIR=C:\Program Files\rscsqlauto          ;path where the log file resides [required]
TIMERINTERVAL=.5          ;interval for processing   (minutes) [required]
SERVICE_NAME=RSC_SERVICE_MANAGE     ; Required for SQLWatch service
SUCCESS_EMAIL_SUBJECT=Archibus Automated Sync   ;Change to desired text (Sync Log)
DEBUG=0          ;default debug parameter [optional]
;DEBUG=1          ;verbose debug that echos all SQL statements [optional]
;DEBUG=-1          ;non-verbose debug with log entries showing selection of tasks to run [optional]
DSN=RSCSQLAUTO     ;the name of the System DSN ODBC entry pointing to the project database [DO NOT
MODIFY]
UID=afm               ;username to attach to the database   [optional]
PWD=afm          ;password to attach to the database   [optional]
PWD_ENCRYPT=afm   ;If PWD = PWD_ENCRYPT the PWD_ENCRYPT is enrypted and PWD is set to null
START_TIME=03:00        ;Required only for daily timed events
SERVICE_NAME=RSC_SERVICE_MANAGER  ;name of service for the SQL_Watch service to monitor
```

LOG_TABLE=N;logging to external debug file or internal process log table (rsc_proc_log)

;The following parameters are required only for email notification of processing errors
SMTP_HOST=127.0.0.1 ;SMTP server for processing emails
SMTP_PORT=25 ;SMTP port
SENDER_EMAIL=user@company.mail ;from email address for sending emails
ERROR_EMAIL=tech@company.mail ;email address to echo processing errors to [optional]
ERROR_EMAIL2=tech2@company.mail;secondary email address to echo processing errors to [optional]
ERROR_EMAIL_SUBJECT=MicroView FM Automated Work Processor Error ; [optional]
AUTHENTICATION=1 ;if the SMTP server requires authentication [optional]
POP_AUTHENTICATION=1 ;if the SMTP server requires 'POP' first authentication [optional]
POP_USERNAME=username ;username for required SMTP authentication [optional]
POP_PASSWORD=password ;password for required SMTP authentication [optional]

7) Start the engine by running RSCSQLAUTO.EXE. A blue icon with an "A" should show as a minimized icon. To stop the engine right mouse click and Pick Restore (This may take a few seconds to respond. Pick Exit from the dialog. This may take a few minutes to exist if it is in the middle of processing work orders.

Running the RSC SQL Automation Engine as a Windows Service

The RSC SQL Automation Engine can be configured to run as a Windows service. This has the advantage of keeping the program running when the host server console is logged off. By default the service runs as the SYSTEM user. Please note that the in the context of the user running the service the RSCSQLAUTO.INI file and the log file folder (application folder) must be accessible. When running as a service the application must be stopped using the Services application in Control Panel.

The additional Watch server (RSCSQLWatch) is an optional service that monitors and allows the main SQL Automation service to be remotely started and stopped via the Archibus view labeled Service Status. RSCSQLWatch also monitors for events that have not run and emails warning (See SQL Watch Engine) Installing the SQL Automation Engine service also installs the SQL Watch service.

NOTE: If running as a service, then ensure that the engine is idle before stopping.

Perform the following tasks to install the RSC SQL Automation Engine as a service:

1) Start the engine by running RSCSQLAUTO.EXE. A blue icon with an "A" should show as a minimized icon. Right mouse click and Pick Restore (This may take a few seconds to respond.

2) Pick Create Service. The engine will now terminate.

3) From the Services program in Control Panel start the service called "RSC_Service_Manager". Check the RSC_SQL_AUTO_LOG file to see that service has started. You can also optionally start the SQL Watch Service (RSC_Service_Manager_Watch)

Uninstalling the Service

The service can be uninstalled if required:

1) Open a command prompt in the application folder
2) Type the following line then press Enter:
    instsrv.exe "RSC_Service_Manager" remove

Installation of The Web Central Interface

Optionally you can install the Web Central interface. All of the same RSC views are duplicated for Web Central with the exception of the Synch Archive Table fields view and function. The installation files will be under \Program XXX\RSCSQLAUTO\Web Central

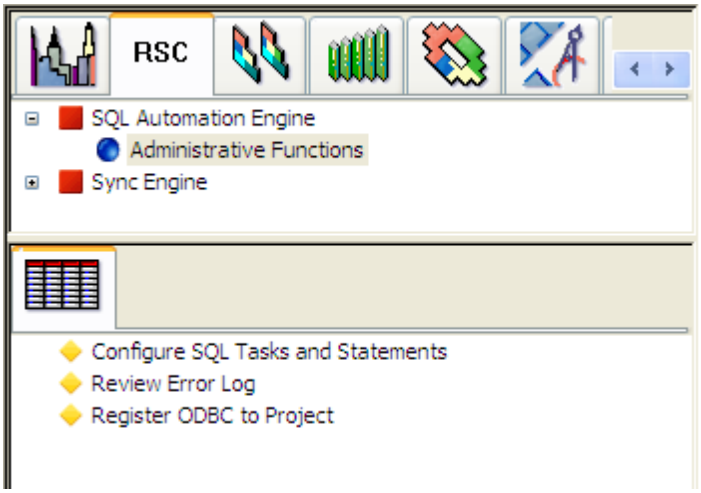
1) Copy contents of ab-products folder under \archibus\schema\ab-products
2) Copy contents graphics folder to \archibus\WEB-INF\lib
3) In Archibus Client Server use File - Data Transfer Import. Import wc1.xml through wc6.xml.
4) In Archibus Client Server use File - Data Transfer Import. Import afm_wf_rules.xml
5) Restart Web Central.

Configuration and Testing

The overview below is based on the provided sample Sybase Archibus database RSCSQLAUTO.DB

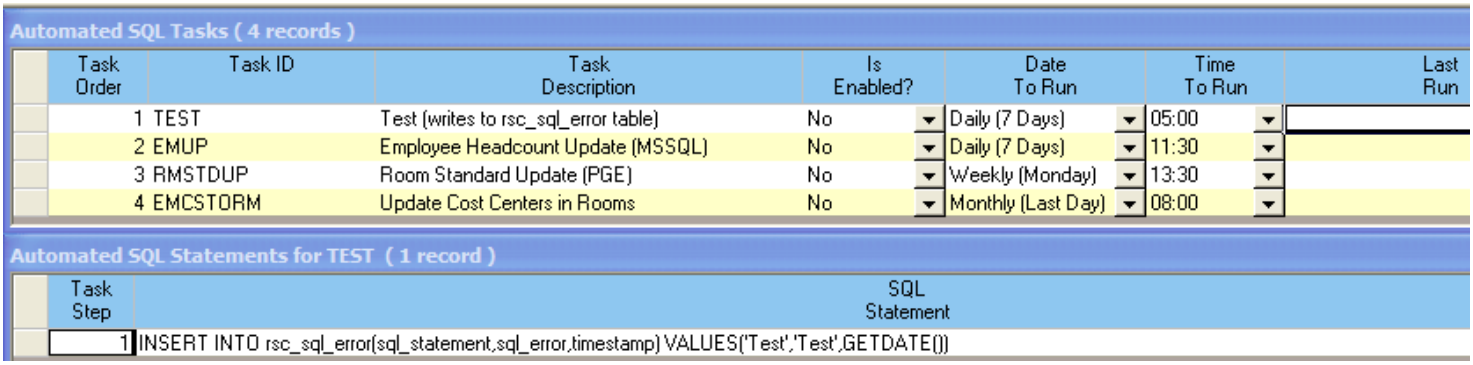
RSC SQL Automation Engine

The RSC SQL Automation Engine allows SQL statements to be run on a scheduled basis. If a single statement fails to execute (after 3 attempts) then an email is sent to the ERROR_EMAIL and ERROR_EMAIL2 addresses listed in the RSCSQLAUTO.INI file (C:\WINDOWS). On failure the routine moves to the next SQL statement and documents the failure in the Error Log table.



The RSC Domain contains links to the configuration tables and functions.

Configure SQL Tasks and Statements

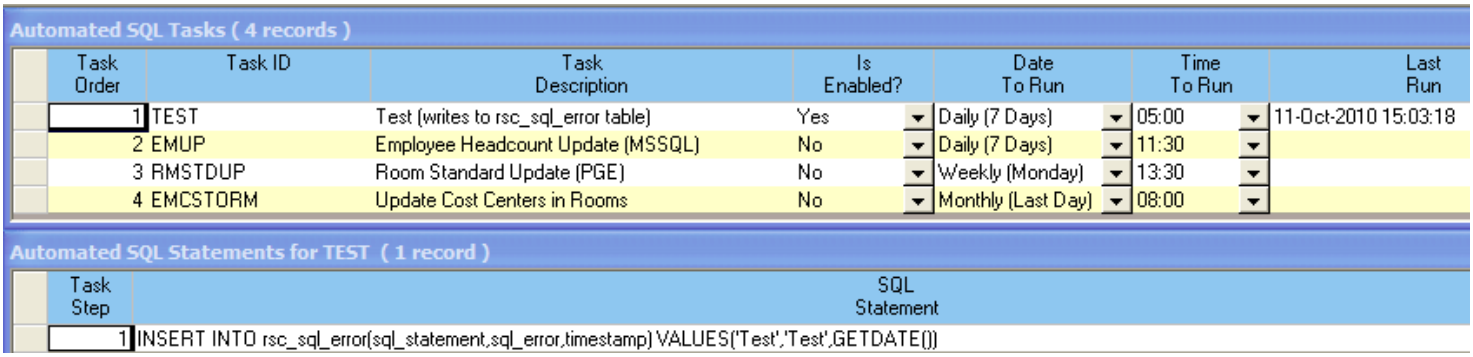


The Configure SQL Tasks and Statements item opens a two level table structure. The top pane is the list of tasks and the bottom pane is the list of statements to run for each task.

- The Date To Run and Time To Run fields define when the event will run.
- Is Enabled must be set to Yes for the event to run.
- If the events are scheduled to run at the same time then they are queued up via the Task Order.
- Once an event runs the Last Run field displays the compellation date and time.
- The Run Selected Task View Action will run the event via an ABS file. This is handy for debugging SQL Errors.

TIP: Blanking out the Last Run field will trigger an event again (assuming that it is past the scheduled time to run for that day.

Running Events



Automated SQL Tasks (4 records)

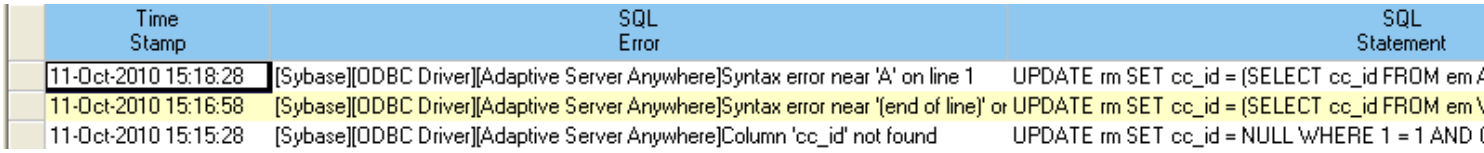
Task Order	Task ID	Task Description	Is Enabled?	Date To Run	Time To Run	Last Run
1	TEST	Test (writes to rsc_sql_error table)	Yes	Daily (7 Days)	05:00	11-Oct-2010 15:03:18
2	EMUP	Employee Headcount Update (MSSQL)	No	Daily (7 Days)	11:30	
3	RMSTDUP	Room Standard Update (PGE)	No	Weekly (Monday)	13:30	
4	EMCSTORM	Update Cost Centers in Rooms	No	Monthly (Last Day)	08:00	

Automated SQL Statements for TEST (1 record)

Task Step	SQL Statement
1	INSERT INTO rsc_sql_error(sql_statement,sql_error,timestamp) VALUES('Test','Test',GETDATE())

Once enabled, an event will run when the next cycle fires. The default cycle time is 1 minute and is set via the TIMERINTERVAL value in the RSCSQLAUTO.INI file. The shortest cycle time is 30 seconds (.5 minutes). The Last Run field will display the timestamp when the event completed.

Review Error Log



Time Stamp	SQL Error	SQL Statement
11-Oct-2010 15:18:28	[Sybase][ODBC Driver][Adaptive Server Anywhere]Syntax error near 'A' on line 1	UPDATE rm SET cc_id = (SELECT cc_id FROM em A
11-Oct-2010 15:16:58	[Sybase][ODBC Driver][Adaptive Server Anywhere]Syntax error near '(end of line)' or	UPDATE rm SET cc_id = (SELECT cc_id FROM em \
11-Oct-2010 15:15:28	[Sybase][ODBC Driver][Adaptive Server Anywhere]Column 'cc_id' not found	UPDATE rm SET cc_id = NULL WHERE 1 = 1 AND (

Debugging Errors

The Error Log displays a record for each failure running a SQL statement. The error and the original SQL statement are displayed. An error email is also sent to the ERROR_EMAIL and ERROR_EMAIL2 addresses listed in the RSCSQLAUTO.INI file (C:\WINDOWS).

If the SQL Engine does not appear to be functioning then:

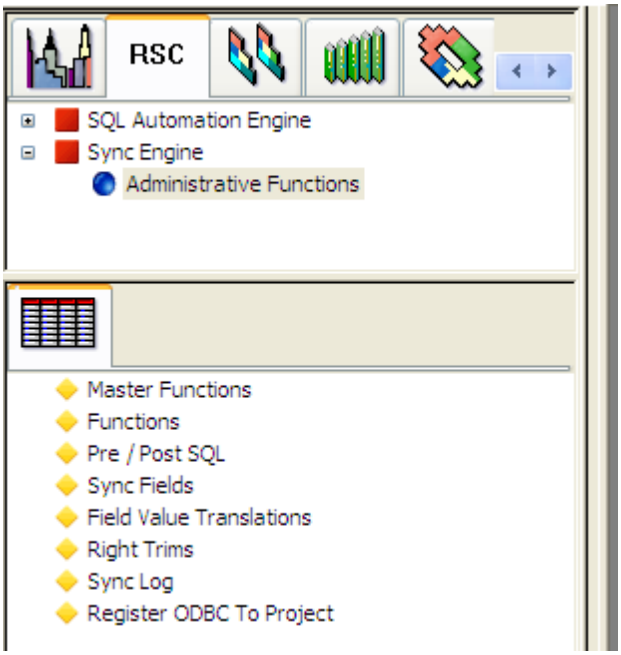
1) Check the RSC_SQL_AUTO_LOG.TXT file (C:\PROGRAM FILES\RSCSQLAUTO folder). This will indicate any errors or the last time the engine cycled.

2) If nothing appears to be happening then restart the SQL Automation Engine service. In Services it will be listed under RSC Service Manager.

Please note that this service should be running on a server or workstation but ONLY ONE INSTANCE IN TOTAL.

RSC Sync Engine

The RSC Sync Engine allows scheduled and automated import of EXCEL data and the movement of data between Archibus tables. All of the engine workflow is user definable in Archibus tables. If a single statement fails to execute then it will retry forever (as the most common error is database connection issues. On failure an email is sent to the ERROR_EMAIL and ERROR_EMAIL2 addresses listed in the RSCSQLAUTO.INI file (C:\WINDOWS). On recovery another email is generated. In the Error events are also documented in the RSC_SQL_AUTO_LOG.TXT file.



The RSC Domain contains links to the configuration tables and functions.

Master Functions

Master Function Name	Description	Is Enabled?	Last Run Date	Sync Log	Date To Run	Time To Run	Email Address 1
EMSYNC	Employee Sync	No			Daily (7 Days)	07:30	gdryer@webnotes.ca

Master Functions item is the top-level definition for sync events. All Functions belong to a Master Functions.

- The Date To Run and Time To Run fields define when the event will run.
- Is Enabled must be set to Yes for the event to run.
- Once an event runs the Last Run field displays the compellation date and time.
- Once an event runs the Sync Log displays the results from each Function.
- Email Address 1 and Email Address 2 allow users to be emailed with the results of a successful event.

TIP: Blanking out the Last Run field will trigger an event again (assuming that it is past the scheduled time to run for that day.

Functions

Master Function	Function Name	Is Enabled	Function Type	Processing Order	Table Name From	Primary Key Field From 1	Table Name To	Primary Key Field To 1	From Table Restrict
EMSYNC	EXCEL_TO_EMSOURCE	No	Import	1			rsc_emsource		
EMSYNC	EMSOURCE_TO_EMDOWN	No	Fields	2	rsc_emsource	emsource_id	rsc_emdown	emdown_id	field1 <> '' AND field2 <> '' AND field3 <> ''
EMSYNC	EMDOWN_TO_EM	No	Fields	3	rsc_emdown	emdown_id	em	em_id	
EMSYNC	EM_TO_AEM	No	Archiv	4	em		rsc_aem		em.update_type = 'MODIFIED' OR em.update_type = 'WARNING'
EMSYNC	EM_TO_HEM	No	Archiv	5	em		rsc_hem		em.update_type = 'DELETE'

Update Non Mod Records	Update When Non Mod.	Validation Notes Field	Import File Name	Excel Sheet Name	Excel Row Start	Excel Column Start	Excel Number of Columns
Yes			c:\test2.xls	sheet1	3	1	11
Yes					1	1	1
No	update_type = 'NO_CHANGE'	update_comnt			1	1	1
Yes					1	1	1
Yes					1	1	1

The Functions table defines tasks that belong to the Master Function.

Function must have a unique name and belong to a Master Function.

Is Enabled must be set to Yes for the function to be included when the Master Function event runs.

Function Type choices are Import Excel, Fields and Archive

Import Excel - Import Data from an existing Excel file to the "Table Name To" table.
- Requires an "Import File Name" value, "Row Start",
"Column Start" and "Number of Columns" value.
- Uses "Excel Sheet" or the Active Sheet if NULL.
- Normally Excel files are read into the RSC_EMSOURCE table

NOTE: If the Import fails with a cannot find XXX file and the server is Windows 2008 or later then:
Create a folder on the server *c:\windows\syswow64\config\systemprofile\desktop*

Import ASCII - Import Data from an existing ASCII file to the "Table Name To" table.
- Requires and "Import File Name", "Field Delimiter" , "Row Start",
"Column Start", Number of Columns"
- Normally ASCII files are read into the RSC_EMSOURCE table
- Field Delimiter converts to the ASCII character if set as an integer. Eg. 9 = Tab.
- Special Function Variable 1: Record Delimiter
-If null then defaults to LF.
Options CR, LF, CRLF

Import XML - Import Data (eg. XML or JSON) from an existing XML file to the "Table Name To" table.
- Requires and "Import File Name", "Field Delimiter" , "Special Function Variable1",
"Special Function Variable 2"
-Field Delimiter:If "L" then it assumes a LineFeed character between records.
If null then it assume LineFeed - CarriageReturn.
- Special Function Variable 1: record start parameter e.g. <wd:Report_Entry>
- Special Function Variable 2: record end parameter e.g. </wd:Report_Entry>
- Special Function Variable 3: field value start character. Defaults to "<"
- Special Function Variable 4: field value end character. Defaults to ">"
- Normally data is read into the RSC_EMSOURCE table
- Record start and end supports special characters (var1, var2)
{CRLF} {CR} {LF} {TAB}

Export ASCII - Export Data from "Table Name From" to an ASCII (delimited file.
- Function Type = "Export ASCII"
- Function View Required Values:
- Function Name (The same function name is referenced in the Sync Fields view).
- Table Name From - Table to export data from.
- Import / Export File Name - Directory and file name to export to.
- Field Delimiter - Delimiter character between fields of data. Default is a comma.
- Special Function Variable 1 - Restriction to the source table. E.g. "bl_id IS NOT NULL"
- Special Function Variable 2 - Character to surround character fields. E.g. double quoted strings
- Special Function Variable 3 - Header line.
  Set to "Y" single character is you want the first record to list the fields
- Sync Fields Required Values (Master Function Name and Function Name match the
  Function View: Table Name From
  Field Name From
  Field Order

Export XML - Export Data from "Table Name From" to an XML formatted file.
- Function Type = "Export XML"
- Function View Required Values:
  Function Name (The same function name is referenced in the Sync Fields view).
  Table Name From - Table to export data from.
  From Table Restrict - Restriction in source table (optional)
  Skip If Table Empty - Name of source table to skip if empty
  File Prefix Text - XML parameters prior to record data.
  Can use character formatting (see below)
  File Suffix Text - XML parameters after record data.
  Can use character formatting (see below)

- Sync Fields Required Values (Master Function Name and Function Name match the
  Function View: Table Name From
  Field Name From
  Field Order
  Field Prefix Text - XML parameters prior to field value.
  Can use character formatting (see below)
  Field Suffix Text - XML parameters after field value.
  Can use character formatting (see below)

- Special character support for File and Field text (prefix / suffix)
  {CR} = Chr$(13) Carriage Return
  {LF} = Chr$(10) Line Feed
  {TAB} = Chr$(9) Tab

Export Crystal - Run Crystal Report Report (RPT) files. Export to formats including PDF.
- Requires CREXPORT.EXE program in the RSCSQLAUTO folder
- Function Type = "Export CR"
- Will email the attachment (Requires variable 4 and 5)
- Function View Required Values:
    - Special Function Variable 1 – Full path and name of source RPT file.
        e.g. D:\CAFM\TEMP\GENERIC_01_BLDGS_SUM.rpt
    - Special Function Variable 2 – Full path and name of output file.
        Supports embedded date and time variable
        e.g. D:\CAFM\TEMP\<yyyy_mmm>\GENERIC_01_BLDGS_SUM_<dd>.pdf
        Will automatically create one level of directory.
    - Special Function Variable 3 – Output type. Values include: “pdf” ,"doc","xls","xlsdata"
    - Special Function Variable 4 – Prototype Email Type.
        Must exist in Prototype Emails table
        Defines the Email Subject and Email Body.
    - Special Function Variable 5 – Email To – Email Address to send to.
    - Special Function Variable 6 – Email CC List – Email Address to CC
        Semi-colon separated list.

NOTE: .Net Runtime 4.0 and the Crystal Runtime for .Net (CRforVS_redist_install_64bit_13_0_5.zip) is required.

NOTE: .If you use the 64 bit Crystal Runtime then it will use 64 bit ODBC.
A 64 bit ODBC entry named RSCSQLAUTO will be required..

Export JSON - Export Data from "Table Name From" to an JSON formatted file (UTF-8 format).

- Function Type = "Export JSON"
- Function View Required Values:
  - Function Name (The same function name is referenced in the Sync Fields view).
  - Table Name From - Table to export data from.
  - From Table Restrict - Restriction in source table (optional)
  - Skip If Table Empty - Name of source table to skip if empty
  - File Prefix Text - JSON parameters prior to record data.
    - Can use character formatting (see below)
  - File Suffix Text - JSON parameters after record data.
    - Can use character formatting (see below)

- Import / Export File Name

- Sync Fields Required Values (Master Function Name and Function Name match the Function View:
  - Table Name From
  - Field Name From
  - Field Order
  - Field Prefix Text - JSON parameters prior to field value.
    - Can use character formatting (see below)
  - Field Suffix Text - JSON parameters after field value.
    - Can use character formatting (see below)

- Special character support for File and Field text (prefix / suffix)
  - {CR} = Chr$(13) Carriage Return
  - {LF} = Chr$(10) Line Feed
  - {TAB} = Chr$(9) Tab

- "Calculated Field Value"

  - <CURRENTDATE> Returns the current date.
  - <TIMESTAMP> Current date and time.
  - <RTRIM> Validates against the Field Value Translations table (see below)
  - <UCASE> Returns the result converted to uppercase
  - <LCASE> Returns the result converted to uppercase
  - <TRANS> Translate from the rsc_translation table

Fields

- Copy data between "Table Name From" and "Table Name To" tables
- Assumes that the tables have only a single primary key field
- "Primary Key Field From 1" and "Primary Key Field To 1" must be defined.
- Supports 1 or 2 Primary key fields to the “To Table”
- Special Function Variable 1 can hold an option SQL statement to define the read from the From Table(s)
  eg. SELECT em.em_id,em.em_std,emstd.description FROM em,emstd
  WHERE em.em_std = emstd.em_std
- NOTE: If the SQL pull is defined by SQL Function Variable 1 then the SQL Fields Calculated Field value syntax is different.
  e.g. SELECT pi_split_id FROM rsc_pi_split WHERE pi_em_id = <EMPLOYEE>
  Variables are evaluated from the source recordset e.g. <EMPLOYEE>.
- The Sync Fields table defines which fields will be transferred and other business rules.

Archive

- Copies all fields between "Table Name From" and "Table Name To" tables.
- If fields are missing from the "Table Name To" table then they will be skipped.
- Records can be filtered via the "From Table Restrict" value.
- Functions run in numeric order defined by "Processing Order".
- "Data Source From / To" Supports two separate database sources either being Primary or Secondary.
- If using an External datasource then the following parameters are required in the RSCSQLAUTO.INI
  EXTERNAL_DSN=dsn_name
  EXTERNAL_UID=username
  EXTERNAL_PWD=password
  EXTERNAL_TEST_SQL=
  SELECT table_name FROM afm_tbls WHERE table_name = 'bl'
  EXTERNAL_IS_AFM_DB=1 (0 if it is not an Archibus database

.

- "Table Name From" is the table from which data will be copied from.
  This is not required form "Function Type" = Import.

- "Primary Key Field From 1" is the first primary key field in the from table
  (currently only a single Primary key field is supported).
  This is required only for "Function Type" = Fields.

- "Table Name To" is the table from which data will be copied to.
  This is required for all "Function Types".

- "Primary Key Field To 1" is the first primary key field in the to table
  (currently only a single Primary key field is supported).
  This is required only for "Function Type" = Fields.

- The "From Table Restrict" restricts records chosen from the "From Table".
  Does not apply to "Function Type" = Import Excel.

  For Fields Functions this supports embedded SQL evaluated to form the restriction,
  [] surrounding with square bracket evaluates unquoted.(number)
  {} surrounding with braces evaluates quotes (string)
  Example :ID > [SELECT variable_value_int FROM rsc_variable_store
  WHERE varable_name = 'var1']

- "Update Non Mod Records" is a flag that enables or disables updating records that have not changed.
  If set to No then at least one field in the Sync Fields view must have "Check for Modification" = Yes.
  If set to No then the "Update When Non Mod" field must contain a field name and value

(valid SQL format) to set when a record is not modified.
This ends up being the only change to the record.
Normally this is used to set the UPDATE_TYPE field.

- Skip If Table Empty. Skips running the function if the entered table name contains no records. This prevents A function from running if a previous function did not run. If the table name is prefixed with NOT_EMPTY e.g. "NOT_EMPTY rsc_emsource" then the function is skipped if the table is not empty.

- "Validation Notes Field" is the field in the "Table Name To" where validation errors are written

- "Import File Name" is the full path and file name for a source Excel file.
This is required only for "Function Type" = Import Excel.

- "Excel Sheet name" is the Excel file sheet name.
This is required only for "Function Type" = Import Excel.

- "Excel Row Start" is the starting row where the data is read from.
This is required only for "Function Type" = Import Excel.

- "Excel Column Start" is the starting column where the data is read from.
This is required only for "Function Type" = Import Excel.

- "Excel Number of Columns" is the number of columns of data to read.
This is required only for "Function Type" = Import Excel.

Archive By Record

- Same as the Archive function but it copies one record at a time to support multiple data sources.
- Copies all fields between "Table Name From" and "Table Name To" tables.
- If fields are missing from the "Table Name To" table then they will be skipped.
- Records can be filtered via the "From Table Restrict" value.
- Same parameters as the Archive function

While Loop - Routine to perform a SQL statement inside of a While Loop

- Function Name – Specified name fir this function

- Function Type – While Loop

- Function Variable1 - holds the SELECT Statement
e.g. SELECT bl_id, name FROM bl WHERE name IS NOT NULL

- Function Variable 2 through 6 holds the UPDATE or INSERT statement(s) to process for each records selected
e.g. INSERT INTO rsc_emsource (field1,field2) VALUES (<bl_id>,<name>)

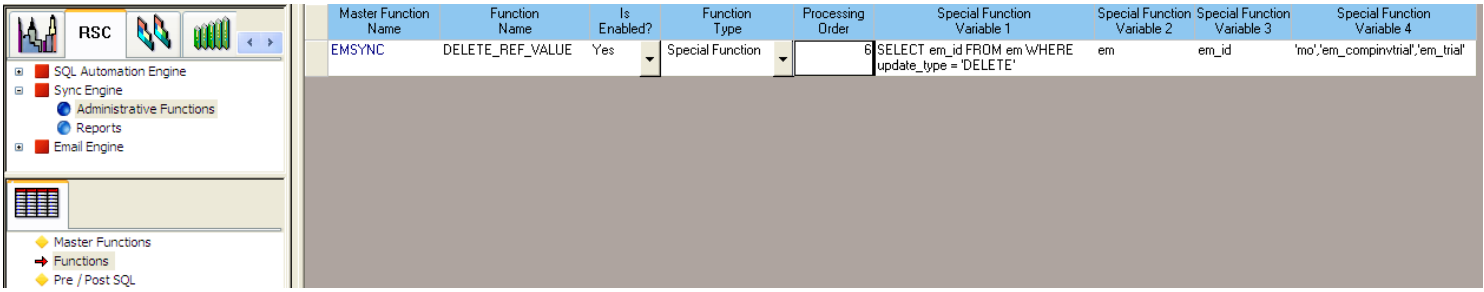
- Datasource From and Datasource To can be different

Special Functions

Special Functions are a Function Type that calls specialized routines to perform particular tasks

Special Function utilize calling variables defined in the fields labeled Special Function Variable (1 to 4)

DELETE_REF_VALUE



DELETE_REF_VALUE performs cascading deletion of records

This is required when the referencing field is part of the primary key of the table. To select the tables to applied "Special Function Variable 4", apply the following restriction to the Archibus fields table (afm_flds)

Validating Table = table name e.g. Em
AND Primary Key > 0
AND Validate Data? = Yes

Required Variables:

Function Name - DELETE_REF_VALUE

Function Type - Special Function

Special Function Variable 1 - Restriction that defines the records to remove.
This SQL statement must retrieve a single field value.

Special Function Variable 2 - Parent table name.

Special Function Variable 3 - Parent table field name. This must be the primary key of the parent table.

Special Function Variable 4 - A list of child tables to exclude. Format 'table1','table2','table3'.

Special Function Variable 4 - % Threshhold for deletion of records.
Prevents deletion of records is the records in restriction
exceeds the total number of records. Set to 100 to allow all records to be deleted.

This is required when the referencing field is part of the primary key of the table, To search for tables apply the following restriction to the Archibus fields table (afm_flds)

Validating Table = table name e.g. Em
AND Primary Key > 0
AND Validate Data? = Yes

UPDATE_REF_VALUE

Master Function Name	Function Name	Is Enabled?	Function Type	Processing Order	Table Name From	Primary Key Field From 1	Table Name To	Primary Key Field To 1	From Table Restrict	Update Non Mod Records
EMSYNC	UPDATE_REF_VALUE	Yes	Special Function	4	rsc_emdown	emdown_id	em	em_id	rsc_emdown.combinedname <> ''	Yes
EMSYNC	EM_TO_AEM_RENAMED	Yes	Archive	5	em		rsc_aem		em.update_type = 'RENAMED'	Yes

Function Name	Validation Notes Field	Special Function Variable 1	Special Function Variable 2	Special Function Variable 3	Special Function Variable 4
UPDATE_REF_VALUE	update_comnt	emnumber	em_number	combinedname	'em_compinvtrial','em_trial'
EM_TO_AEM_RENAMED					

UPDATE_REF_VALUE performs cascading updates of records

An example of the application of this function would be to rename employee (em table) records when an employee name changes. This function can compare records by the employee number (EMNUMBER) and then rename all references to the em record before renaming the parent record.

The function will mark the record as RENAMED. The Archive function should be called after UPDATE_REF_VALUE to create a record of the change.
The original parent record is marked as REN_DELETE

Required Variables:

Function Name - UPDATE_REF_VALUE

Function Type - Special Function

Table Name From - Table name where new list of records are sourced from

Primary Key Field From 1 - Primary key field in the source table

Table Name To - Destination table name where records are written to (parent table)

Primary Key Field To 1 - Primary key field in the destination table

From Table Restrict - Restriction in gathering records from the source table

Update Non-Mod Records - Always set to Yes

Validation Notes Field - Field name in the destination table where notes regarding the rename are stored.

Special Function Variable 1 - Match field in the source table. This is compared to the match field in the destination table to determine if a record was renamed.

Special Function Variable 2 - Match field in the destination table.

Special Function Variable 3 - Field in the source table that represents the primary key field in the parent table.

Special Function Variable 4 - A list of child tables to exclude. Format 'table1','table2','table3'.

Special Function Variable 5 - Duplicated Suffix.
Removes Duplicate suffix values eg. "_DUP" from the value.

Must be set to NA if no used.

This is required when the referencing field is part of the primary key of the table. To search for tables apply the following restriction to the Archibus fields table (afm_flds)

Validating Table = table name eg. Em
AND Primary Key > 0
AND Validate Data? = Yes

Special Function Variable 5 – Character string to be recognized as suffix for duplicate records (multi-seated employees eg. _DUP. **NOTE:** Set to NA if there is no suffix.

IMPORT_WEB_FILE

IMPORT_WEB_FILE - Performs download of file from Web URL.

An example of the application of this function would be to download a list of employees from an outsourced HR company.

The download file is overwritten.

Required Variables:

Special Function Variable 1 – Source URL e.g. microviewfm.com
Special Function Variable 2 – Source file name eg. "/download/MVHVEL.PDF
Special Function Variable 3 -.Download file name e.g. C:\TEMP\MVHVEL.PDF
Special Function Variable 4 – Port e.g. 80 for HTTP, 443 for HTTPS
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection

INI File variables (RSCSQLAUTO.INI):

PROXY=192.10.10.10 – name or IP of the proxy server

IMPORT_WEB_FILE_REPEAT

IMPORT_WEB_FILE_REPEAT - Performs repeated download of files from Web URL using a variable parameter read from a table.

An example of the application of this function would be to download employee photos from an outsourced HR company.

The download file(s) are overwritten.

Required Variables:

Table Name From - Table name where <field_value> is sourced from

Primary Key Field From 1 - Field that maps to <field_value>
From Table Restrict - Restriction in gathering records from the source table

Special Function Variable 1 – Source URL eg. microviewfm.com
Special Function Variable 2 – Source file name
e.g. "/api//employees/<field_value>/photo/small" (field value is evaluated)
Special Function Variable 3 -.Download file name
e.g. "d:\cafm\pic\<field_value>.jpg" (field value is evaluated)
Special Function Variable 4 – Port e.g. 80 for HTTP, 443 for HTTPS
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection

INI File variables (RSCSQLAUTO.INI):

PROXY=192.10.10.10 – name or IP of the proxy server

IMPORT_FTP_FILE

IMPORT_FTP_FILE - Performs download of a file from an FTP site

- FTP only. Use IMPORT_SFTP for secure FTP

The download file is overwritten.

Required Variables:

Special Function Variable 1 – Source URL e.g. ftp.microviewfm.com
Special Function Variable 2 – Source file name e.g. "/download/MVHVEL.PDF
Special Function Variable 3 -.Download file name e.g. C:\TEMP\MVHVEL.PDF
Special Function Variable 4 – Port e.g. 21
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection

EXPORT_FTP_FILE

EXPORT_FTP_FILE - Performs upload of a file to an FTP site

- FTP only. Use EXPORT_SFTP for secure FTP

The upload file is overwritten.

Required Variables:

Special Function Variable 1 – Remote URL e.g. ftp.microviewfm.com
Special Function Variable 2 – Remote file name e.g. "/download/MVHVEL.PDF
Support embedded date parameters.eg. _<mm_dd_yy hh:nn>.
Special Function Variable 3 -.Source file name e.g. C:\TEMP\MVHVEL.PDF.
Special Function Variable 4 – Port e.g. 21
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection

IMPORT_SFTP_FILE

IMPORT_SFTP_FILE - Performs download of a file from an SFTP site

- SFTP only. Use IMPORT_FTP for non-secure connections
- Requires registering WINSCP software. See Below

The download file is overwritten.

Required Variables:

Special Function Variable 1 – Source URL eg. ftp.microviewfm.com
Special Function Variable 2 – Source directory and file name on SFTP server
e.g. "/download/MVHVEL.PDF
Special Function Variable 3 -.Download directory and file name on local server
e..g. C:\TEMP\MVHVEL.PDF
Special Function Variable 4 – Port eg. 22
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection
Import / Export File Name - HostKey. See WINSCP Installation Instruction below.

EXPORT_SFTP_FILE

EXPORT_SFTP_FILE - Performs upload of a file to an SFTP site

- SFTP only. Use IMPORT_FTP for non-secure connections
- Requires registering WINSCP software. See Below

The upload file is overwritten.

Required Variables:

Special Function Variable 1 – Source URL eg. ftp.microviewfm.com
Special Function Variable 2 – Source directory and file name on local server
eg. C:\TEMP\MVHVEL.PDF
Special Function Variable 3 -.Remote directory and and file name on SFTP server
e.g. \download\MVHVEL.PDF
Support embedded date parameters.eg. _<mm_dd_yy hh:nn>.
Special Function Variable 4 – Port eg. 23
Special Function Variable 5 – Username for secure connect
Special Function Variable 6 – Password for secure connection or private key file
For Private key use: KEY_FILE:D:/CAFM/STORE/pkey_name.ppk
Import / Export File Name - HostKey. See WINSCP Installation Instruction below.

Registering the WINSCP Software

WINSCP is a shareware Secure FTP client software. It is written in DotNet and requires registering the DLL before use

1) Run \RSCSQLAUTO\REGWINSCP.BAT. Verify the the DotNet library registers correctly.
2) Run \RSCSQLAUTO\WINSCP.EXE. Open a test connection the destination SFTP server. Copy the HostKey when prompted.
3) In the EXPORT_SFTP_FILE or IMPORT_SFTP_FILE functions add the HostKey to the Import / Export File Name field.

NOTE: After the initial setup you can find the HostKey by connecting to the SFTP session and then going to Commands – Server/Protocol Information.

EXPORT_IMAGES

EXPORT_IMAGES - Extract images stored in a table field. Repeats for each record

- Normally uses the rsc_emsource.image_field. Use the sql_addimagefld.abs to add the field.
- The files are overwritten

Required Variables:

Table Name From - Source table holding image data e.g. rsc_emsource
Primary Key Field From 1 - Source field holding image data e.g. image_field
From Table Restric - restriction on source data field (not required) e.g. image_field IS NOT NULL

Special Function Variable 1 – field that holds the name of the file. (<name>.jpg) e.g. field1
Special Function Variable 2 – extraction folder (requires trailing backslach) e.g. d:\cafm\temp\
Special Function Variable 3 -.file extension e.g .jpg.
Special Function Variable 4 – Perform Base 64 conversion e.g. Y

PROC_FLOOR_COVERING

PROC_FLOOR_COVERING - RSC Floor Coverings customization (PGE),

- Keeps the rsc_rm_flooring table in sync. Corrects percentages.
- Normally scheduled to run once per day.

Required Variables:

None.

DELETE_UNREF_VALUE

DELETE_UNREF_VALUE performs deletion of records in tables from record values gathered in another table.

This is used to remove references to a record that are part of the primary key in a referenced table.

Required Variables:

Function Name - DELETE_UNREF_VALUE

Function Type - Special Function

Special Function Variable 1 - Restriction that defines the records to remove.
This SQL statement must retrieve a single field value.
e.g. “SELECT em_id FROM em WHERE update_type = 'DELETE'”

Special Function Variable 2 – table(s) and field(s) to delete
e.g. “em_trial.em_id”. Records are line separated (ctrl-lf)

Special Function Variable 3 - % Threshhold for deletion of records.
Prevents deletion of records is the records in restriction
exceeds the total number of records. Set to 100 to allow all records to be deleted.

FILE_RENAME_DATE

FILE_RENAME_DATE renames a file to a new name. Can include the current date and time.

Required Variables:

Function Name – FILE_RENAME_DATE

Function Type - Special Function

Special Function Variable 1 – File name to rename that includes full path.
e.g “c:\import\em_import.txt”
Note: You can also use the * wildcard character. to rename multiple files.

Special Function Variable 2 – File name to rename to that includes full path.
e.g “c:\import\em_import_<mm_dd_yy hh:nn>.txt”
Can include embedded timestamp format surrounded by <> characters.
Note: You can also use the * wildcard character. to rename multiple files.

FILE_RENAME Identical functionality as FILE_RENAME_DATE.
This exists in order support two rename operations within the same function.

UPDATE_STATISTICS

UPDATE_STATISTICS writes records to the rsc_sql_statistics table based on counts for Update Types

Required Variables:

Function Name – UPDATE_STATISTICS

Function Type - Special Function

Table Name From - The name of the table to read statistics from.
e.g. em

Primary Key Field From 1 - The field to extract counts from (count for each unique value).
e.g. update_type

Special Function Variable 1 – Function name to display in the statistics table.
e.g. Employee Update.

Special Function Variable 2 – field values to exclude. This is comma delimited list in single quotes.
e.g. 'NO_CHANGE','NO_UPDATE'

ARRAY_2D_INSERT

ARRAY_2D_INSERT Insert an array of records based on 2 lists of records based on 2 select statements. These can be from separate tables. If there are 10 records returned from the first select and 5 returned from the second then a total of 50 records will be created.

Required Variables:

Function Name – ARRAY_2D_INSERT

Function Type - Special Function

Table Name To - The name of the table to write records to.

Primary Key Field To 1 – Field to write field value from first select.

Primary Key Field To 2 – Field to write field value from second select.

Special Function Variable 1: First select statement
e.g. SELECT project_id FROM project

Special Function Variable 2: Second select statement
e.g. SELECT field2 FROM rsc_emsource

COMPLETE_MP

COMPLETE_MP – Auto-complete Move Projects for SpaceView

Required Variables:

Function Name – COMPLETE_MP

Function Type - Special Function

Table Name To - The name of the table to write records to.

Special Function Variable 1: Restriction on the MP table
e.g. move_status = 'Approved' AND DateDiff(d, date_start, GETDATE()) = 0

CLOSE_WR

CLOSE_WR – Auto-close Completed Work Requests

NOTE: This function sets the activity_log.external_req = 5 (close). It requires SVMSLA.java version 1.6 or later (RSC-SVMoblie-workflow.JAR). This is a scheduled Workflow in Archibus.

Required Variables:

Function Name – CLOSE_WR

Function Type - Special Function

Special Function Variable 1: Restriction on the WR table
e.g. DATEDIFF(day, date_completed, GETDATE()) > 30 AND status = 'Com'

FILE_TIMESTAMP

FILE_TIMESTAMP – Update timestamps of files in the afm_dwgs (or any other) table

Required Variables:

Function Name – FILE_TIMESTAMP

Function Type - Special Function

Table Name From - source table for records and table where date and time fields reside.
Each record in this table represents a single file - REQUIRED.
E.g. afm_dwgs

Primary Key Field From 1 - 1st primary key of the from table.
Necessary as it may be an external DB – REQUIRED
E.g. dwg_name

Primary Key Field From 2 - 2st primary key of the from table.
Necessary as it may be an external DB

From Table Restrict: - restriction applied to the source table records

Special Function Variable 1 - date_field - field to hold date and time – REQUIRED
e.g. rsc_date_map

Special Function Variable 2 - time_field - field to hold date and time – REQUIRED
e.g. rsc_time_map

Special Function Variable 3 - source_dir - directory start search point – REQUIRED
Can contain embedded variables.
<bl_id and <fl_id> are special as they are extracted from the Space Hierarchy field
e.g. D:\CAFM\APPS\CADNOW\<bl_id>-<fl_id>\

Special Function Variable 4 - file_name - file name to search for - REQUIRED
Can contain embedded variables.
<bl_id and <fl_id> are special as they are extracted from the Space Hierarchy field
e.g. roomlist.txt

WRITE_MAPS

WRITE_MAPS – Publishes drawing via CADNow executable.

Drawings are published where rsc_date_dwg > rsc_date_map (afm_dwgs table)

Required Variables:

Function Name – WRITE_MAPS

Function Type - Special Function

Table Name From - source table for records and table where date and time fields reside.
Each record in this table represents a single file - REQUIRED.
e.g. afm_dwgs

Primary Key Field From 1 - 1st primary key of the from table.
Necessary as it may be an external DB – REQUIRED
e.g. dwg_name

Special Function Variable 1 – Path and name of CADNOW executable
e.g. d:\cafm\apps\cadnow\CADNOW.EXE

Special Function Variable 2 – Root drawing folder. Root folder where it searches for drawings from.
e.g. d:\cafm\afmdata\dwgs\

ZIP_FOLDER

ZIP_FOLDER – Compress a folder of files (and subfolders into a compressed (ZIP) file.

Required Variables:

Function Name – ZIP_FOLDER

Function Type - Special Function

Special Function Variable 1 – Path for top level folder.
e.g. d:\cafm\afmdata\maps\

Special Function Variable 2 – Path and filename for zip file..
e.g. d:\cafm\afmdata\mapzip\maps.zip

Special Function Variable 3 – Maximum minutes to wait for completion.
Note – It will continue to zip in the background after this timeout period
e.g. 10

RUN_SHELL

RUN_SHELL – Shell out and run a command (from the command prompt).

Required Variables:

Function Name – RUN_SHELL

Function Type - Special Function

Special Function Variable 1 – Command line parameters..
e.g. del c:\temp*.bak

IMPORT_EMAILS

IMPORT_EMAILS – Import emails and place the emails in the rsc_email_import table.

Rsc_email_import table

Email_from – address from
Email_text_subject – email subject
Email_text_body – email body
Processed – default 0. Used as a flag for post processing of email records.
Email_type – Prototype Email Name. Used as a flag for post processing.
Field1 to field5 – Used to store intermediate data.

Requires installing the EAGetMail client on the server:

1) Run EAGetMail.exe and install under \RSCSQLAUTO\EAGetMail folder.

Required Variables:

Function Name – IMPORT_EMAILS

Function Type - Special Function

Import/Export File Name – Prototype Email Name eg. ACT_LOG_IMPORT

Special Function Variable 1 – Server Type
IMAP – Exchange
DAV
EWS
POP3

Special Function Variable 2 – Use SSL
TRUE – value for exchange
FLASE

Special Function Variable 3 – Server URL / name
Eg. imap10.exchangecentral.net

Special Function Variable 4 – Server Port
Eg. 993

Special Function Variable 5 – User account name
Eg. scheduleview@rsc2lc.com

Special Function Variable 6 – User account password.

WRITE_PDFS

WRITE_PDFS – Publishes drawing via PDF Files.

Requires installation of AnyDWGToPDF software on server.

Drawings are published where the drawings date is later than the PDF date

Required Variables:

Function Name – WRITE_PDFS

Function Type - Special Function

Function Variable 1 - Full path and executable name of the AnyDWG to PDF software [D:\CAFM\RSCSQLAUTO\AnyDWGtoPDF\dp.exe]

Function Variable 2 - Full path to the source DWG folder [D:\CAFM\AFMDATA\DWGs]

Function Variable 3 - Full path to the output PDF folder [D:\CAFM\AFMDATA\PDFs]

Function Variable 4 - Full path to the backup PDF folder [D:\CAFM\AFMDATA\PDFs\Old]

All folders must exist.
If the PDF is being updated then it is backed up to the backup PDF folder

WR_PARTS

WR_PARTS – Adds a part to work requests by parsing the part_id quantity, and parts warehouse from a field in the WR table (normally the description). This was written to be used with the submit request functionality is SpaceView.

Required Variables:

Function Name - WR_PARTS

Function Type - Special Function

Table Name From - Table to pull records from ‘wr’

Primary Key Field From 1 – ‘wr_id’

Primary Key Field From 2 – Not required

From Table Restriction – Restriction to pull records.
e.g. wr.prob_type = 'WAREHOUSE' AND
(SELECT COUNT(part_id) FROM wrpt WHERE wrpt.wr_id = wr.wr_id) = 0

Special Function Variable 1 – Field to parse part data from
e.g. description

Special Function Variable 2 – Start of String before the part code <LF> ends the line
e.g. Item:

Special Function Variable 3 – Start of String before the part quantity <LF> ends the line
e.g. Quantity:

Special Function Variable 4 – Start of String before the warehouse <LF> ends the line
e.g. Warehouse:

Parts are removed to the PT table inventory.

WS_DP_PCT_USE

WS_DP_PCT_USE – Populates the RMPCT table with records that represent shared use of rooms by department

Required Variables:

Function Name - WS_DP_PERCENT_USE

Function Type - Special Function

Table Name From - Table to pull records from
e.g. rsc_emdown

Primary Key Field From 1 – Primary key of source data table
e.g. emdown_id

From Table Restriction – Restriction to pull records.
e.g. comments IS NULL AND division IS NOT NULL AND department IS NOT NULL AND mailstopdown <> '0'

Special Function Variable 1 – Field for bl_id
e.g. buildingdown

Special Function Variable 2 – Field for fl_id
e.g. floordown

Special Function Variable 3 – Field for rm_id
e.g. rooomdown

Special Function Variable 4 – Field for dv_id
e.g. division

Special Function Variable 5 – Field for dp_id
e.g. department

Special Function Variable 6 – Field for % of space used
e.g. mailstopdown

The RMPCT table is updated using standard Archibus rules for Workplace Transactions. NOTE: Apply a custom routine for Reconcile otherwise it will likely overwrite the data.

RMPCT_INCLUDE_CALCS

RMPCT_INCLUDE_CALCS – Update rmpct.rsc_include_calcs for SpaceView Reports Default value is Yes Review the code for the algorithm

Required Variables: None

Pre / Post SQL

Function Name	Pre / Post To Field Proc	Processing Order	SQL Statement
EMDOWN_TO_EM	Pre	1	UPDATE em SET update_type = NULL, update_comnt = NULL
EMDOWN_TO_EM	Post	2	UPDATE em set update_type = 'DELETE' WHERE update_type IS NULL
EMDOWN_TO_EM	Post	1	UPDATE em set update_type = 'WARNING' WHERE update_comnt LIKE '%Validation%'
EMSOURCE_TO_EMDOWN	Pre	1	DELETE FROM rsc_emdown
EXCEL_TO_EMSOURCE	Pre	1	DELETE FROM rsc_emsource

The Pre / Post SQL table holds SQL statements to run before and after the Functions that perform field processing. This is useful for clearing tables prior to processing.

- "Function Name" is required
- "Pre / Post To Field Pro" is either Pre (before) or Post (after)
- Sql statements run in numeric order defined by "Processing Order".
- The SQL statement do not support embedded field names.

Sync Fields

Function Name	Table Name From	Field Name From	Field Order	Table Name To	Field Name To	Field Type	Apply To	Check For Modification	Exclude If Match
EMDOWN_TO_EM	rsc_emdown	combinedname	1	em	em_id	Char	Both	No	None
EMDOWN_TO_EM	rsc_emdown		2	em	date_refresh	Date	Both	No	None
EMDOWN_TO_EM			3	em	update_type	Char	Update	No	None
EMDOWN_TO_EM			4	em	update_type	Char	Insert	No	None
EMDOWN_TO_EM	rsc_emdown	buildingdown	5	em	bl_id	Char	Both	Yes	None
EMDOWN_TO_EM	rsc_emdown	floordown	6	em	fl_id	Char	Both	Yes	None
EMDOWN_TO_EM	rsc_emdown	division	7	em	dv_id	Char	Both	Yes	None
EMDOWN_TO_EM	rsc_emdown	department	8	em	dp_id	Char	Both	Yes	None
EMDOWN_TO_EM	rsc_emdown	fullname	9	em	name_full	Char	Both	No	None
EMDOWN_TO_EM	rsc_emdown	date_sourcedown	10	em	date_import	Date	Both	No	Current Record
EMSOURCE_TO_EMDOWN	rsc_emsource	field1	1	rsc_emdown	emnumber	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field2	2	rsc_emdown	lastname	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field3	3	rsc_emdown	firstname	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field8	4	rsc_emdown	fullname	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field4	5	rsc_emdown	division	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field5	6	rsc_emdown	department	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field6	7	rsc_emdown	statusdown	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field9	8	rsc_emdown	buildingdown	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field10	9	rsc_emdown	floordown	Char	Both	No	None
EMSOURCE_TO_EMDOWN	rsc_emsource	field11	10	rsc_emdown	date_sourcedown	Date	Both	No	None

Calculated Field Value (from table)	Valid Check SQL (from table)	Validated Table Insert SQL
<lastname>, <firstname> <emnumber>		
<CURRENTDATE>		
MODIFIED		
NEW		
	SELECT bl_id FROM bl WHERE bl_id = <buildingdown>	
	SELECT fl_id FROM fl WHERE bl_id = <buildingdown> AND fl_id = <floordown>	
	SELECT dv_id FROM dv WHERE dv_id = <division>	INSERT INTO dv (dv_id) VALUES (<division>)
	SELECT dp_id FROM dp WHERE dv_id = <division> AND dp_id = <department>	INSERT INTO dp (dv_id,dp_id) VALUES (<division>,<department>)
<RTRIM>		
<SELECT trans_to FROM rsc_translation WHERE trans_from = {field9}>		
<SELECT trans_to FROM rsc_translation WHERE trans_from = {field10}>		

Sync Fields table values are required when the Function Type is "Fields". Fields are copied from "Table Name From" to "Table Name To"

NOTE: The primary key field (in the To Table) should be first in the processing order for fields.

- "Table Name From" and "Field Name From" are omitted for calculated fields.

- "Table Name From" and "Table Name To" must match with the parent Master Function definition.

- "Field Order" is required when processing multi-level validated field (e.g. dv_id, dp_id).

- "Table Name To" and "Field Name To" are always required.

- "Field Types" choices are Char, Numeric, Date or Time

- "Apply To" indicates if the field value is updated for an existing record (Update), new record (Insert), or both (Both).

- "Check For Modification is used to define which fields constitute a change to a record. If the fields are the same then the record is not updated, "Update Non Mod Records" in the Functions view must be set to No for this to apply. "Update When Non Mod" in the Functions view must be set for this to apply.

- "Exclude If Match" allow record updates in inserts to be skipped based on a field matches the current value
  Current Record checks only the single matching record in the "Table Name To" (by the primary key).
  Any Record looks for the field value in any record in the "Table Name To" .

- "Calculated Field Value" describes text, field names, or SQL statements that define the value of the field.
  Field values are from the "Table Name From"

  Pure text (not surrounded by { } or < > is returned unevaluated.
  Example: MODIFIED returns that string to the field value.
  Field names are enclosed greater than and less than signs. They can also include other text.
  Example: <lastname>, <firstname>

  Field names can also be embedded in a SQL Statement.
  Example: <SELECT trans_to FROM rsc_translation WHERE trans_from = {field10}>

  <CURRENTDATE> Returns the current date.

  <TIMESTAMP> Current date and time.

  <RTRIM> Validates against the Field Value Translations table (see below)

  <UCASE> Returns the result converted to uppercase

  <LCASE> Returns the result converted to uppercase

- "Valid Check SQL" describes a select statement to evaluate a foreign key.
  If the statement returns null then it returns null to the field value.
  Validation errors are returned to the "Validation Notes field"
  The SQL statement supports embedded field values from the "Table Name From" encoded as <name>.

- "Valid Table Insert SQL" is the SQL select statement to insert a new validating record in a foreign key table.
  If not null then foreign key values are inserted automatically
  The SQL statement supports embedded field values from the "Table Name From" encoded as <name>.

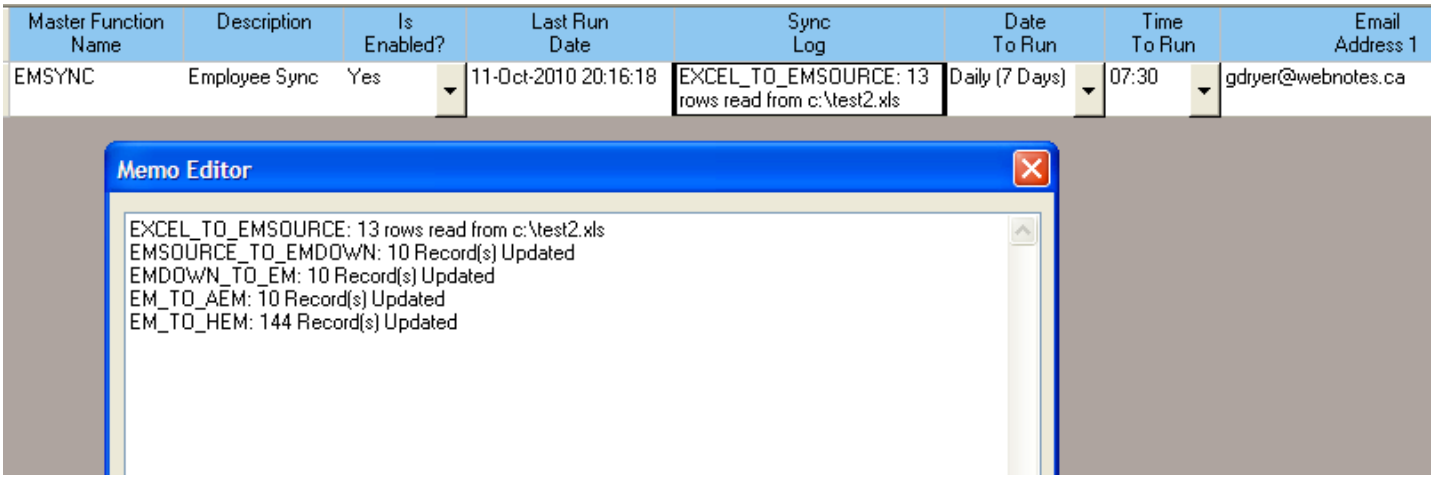
Field Value Translations

Translate From	Translate To
1B	HQ
V85	19

The Field Value Translation table holds string translations and is used in conjunction with a Calculated Field value

- If the translation is not found then the field returns the original value from "Field Name From".
- The "Calculated Field" value in the Sync Fields table requires a value.
  Example: <SELECT trans_to FROM rsc_translation WHERE trans_from = {field10}>

Running Events



Once enabled, an event will run when the next cycle fires. The default cycle time is 1 minute and is set via the TIMERINTERVAL value in the RSCSQLAUTO.INI file. The shortest cycle time is 30 seconds (.5 minutes). The Last Run field will display the timestamp when the event completed. The Sync Log will display the results of each function.

Sync Log

Master Function Name	Description	Run Timestamp	Sync Log	Email Address 1
EMSYNC	Employee Sync	11-Oct-2010 20:16:18	EXCEL_TO_EMSOURCE: 13 rows read from c:\test2.xls	gdryer@webnotes.ca
EMSYNC	Employee Sync	11-Oct-2010 20:01:09	EMDOWN_TO_EM: 10 Record(s) Updated	gdryer@webnotes.ca
EMSYNC	Employee Sync	11-Oct-2010 19:59:30	EM_TO_AEM: 0 Record(s) Updated	gdryer@webnotes.ca

The Sync Log displays a history of all Master Functions (events).

RSC Email Engine

The RSC Email Engine performs email notifications either driven from email events (a SQL restriction) or directly queued emails populating the RSC_EMAIL_QUEUE table from an external application.

There are three tables that provide the email functionality:

1) Prototype Emails (RSC_EMAIl_DEF) This defines email types and their associated Email Subject and Email Body (overridden if defined in the email event.

2) Email Events (RSC_EMAIL_EVENTS) Links a prototype email to a scheduled function.

a) If the Email Subject and the Email Body Record are both NULL then an email is sent for all records matching the restriction. The Email Subject and Email Body must be set in the Prototype Email. The Post Email SQL should alter the parent record to prevent the email from being sent again. This supports parameterized variables e.g. "UPDATE wr SET option1 = 'email req' WHERE wr_id = <wr_id>"

b) If the Email Subject and the Email Body Record are both set then a single email is sent for all records matching the restriction although each record can be reflected in the email body. See below for further details.

3) Email Queue (RSC_EMAIL_QUEUE) Email Events create queue records which are processed from this table. Records remain as a history. An external application can add records to this table directly. See below for further details.

Email Queue

The Email Queue table (RSC_EMAIL_QUEUE) is the table for the email events. This table can be populated by an external application.

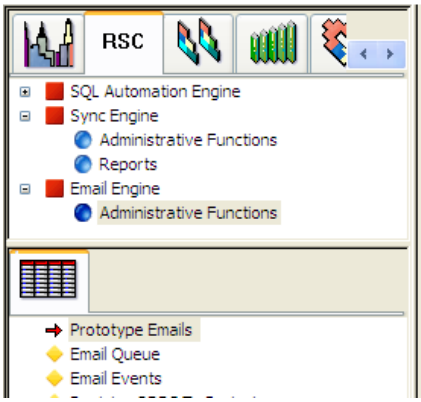
These fields are automatically generated:

auto_id - Auto-increment Primary Key
auto_date - Date record was generated (auto generated)
auto_time - Time record was generated (auto generated)
processed - Processed (indicates that email has been sent 1 = sent)
error_message - Error returned by email engine on failure

These fields are supplied values when the record is created (external application):

email_type - Email Definition (points to RSC_EMAIL_DEF table) CHAR 24 (uppercase)
email_to - Email Address (single entry only max 100 addresses including cc and bcc) VARCHAR 50
email_cc - CC List (one or more recipients delimited with a semicolon) VARCHAR 2000
email_bcc - BCC List (one or more recipients delimited with a semicolon) VARCHAR 2000
email_text_subject - Email Subject (If not null overrides value from RSC_EMAIL_DEF table) VARCHAR 200
email_text_body - Email Body (If not null overrides value from RSC_EMAIL_DEF table) VARCHAR 4000
attachment URL to attachment (this is a URL that will be added to the end of the email_text_body) VARCHAR 150
update_sql_success - SQL statement to issue after email sent (prevents repeat) VARCHAR 250
update_sql_failure - SQL statement to issue on failure to send email (prevents repeat) VARCHAR 250
source_table_name - Name of table of source record. See RSC_EMAIL_DEF table below.
source_primary_key_value - Primary key value for the source record. See RSC_EMAIL_DEF table below.

Prototype Emails



Email Type	Email Subject	Email Body	Email CC List (;) Delimited	Email BCC List (;) Delimited
MO_APPRV_MGR1_NEW	Default Subject	Default Body		
MO_COORD_NEW	Default Subject	Default Body		
MOVE_NOTICE	Hello <em_id>	The employee to move is <em_id>. This person currently is		
NOTIFICATION	Notification from Faciities	This is a notification from the Facilities Department		

The RSC_EMAIL_DEF table provides prototype email definitions for the email events and single externally driven (batch emails.

Email Events

- The Master Function and Function point to the scheduled event (Pre and Post SQL are also supported)
- The Function Name points to the Email Type In te Prototype Emails table.
- The Email Subject, Email Body are defined by the Prototype Email if they are null.
- The Email Record Restriction sources records to process from the Email Data Table
- Post Email SQL On Success is a statement that is run if the email is sent. Email variables eg <name> are supported.
- Post Email SQL On Failure is a statement that is run if the email fails. Email variables eg <name> are supported.
- The Email Data Table and Email Table Primary Key point to the record in which the following fields derive their data from:
  Email To Field - Field where email to is derived from. It assumes that the source table has a field named EMAIL.
  Email To Field 2 - Field where second email to is derived from. Generates a separate email without cc or bcc.
  Email To Field 3 - Field where third email to is derived from. Generates a separate email without cc or bcc.
  Email To Field 4 - Field where fourth email to is derived from. Generates a separate email without cc or bcc.
  Email To Fixed - Fixed email value. Generates a separate email without cc or bcc.
  Email CC Field - Field where CC email list is derived from. This is sent with the first email.
  Email BCC Field - Field where BCC email list is derived from. This is sent with the first email.
  Email BCC List (;) Delimited is a fixed BCC list
  Email CC List (;) Delimited is a fixed CC list or
  if it begins with SELECT then it is a SQL statement that supports email variables eg. <name>.
  e.g SELECT em_id FROM cf WHERE is_supervisor = 1
  AND work_team_id = <work_team_id>
  AND em_id IS NOT NULL
  AND work_team_id IS NOT NULL
  The first value returned becomes the Email To and the rest of the values form a CC list.
  It is assumed that the SQL returns em_id values from which the email values are sourced.

Externally Driven (Batch) Emails

- The Email Queue record must reference the RSC_EMAIL_DEF record via the Email Type field.
- The Email Subject can contain embedded SQL (see below). This requires that the Email Queue record has values for the Source Table and Source PKEY variable.
- The Email Body can contain embedded SQL (see below). This requires that the Email Queue record has values for the Source Table and Source PKEY variable.
- Email CC and Email BCC fields override values if the are NULL in the email queue record.

Embedded SQL in the Email Subject and Email Body fields

- Pure text (not surrounded by { } or < > is returned unevaluated.
- Field names are enclosed greater than and less than signs. They can also include other text. Example: <lastname>, <firstname>
- Field names can also be embedded in a SQL Statement. Example: <SELECT bl_id FROM em WHERE em_id = {em_id}>

Debugging Errors

If there is a processing error (fatal) then an error email is sent to the ERROR_EMAIL and ERROR_EMAIL2 addresses listed in the RSCSQLAUTO.INI file (C:\WINDOWS).

If the SQL Engine does not appear to be functioning then:

1) Check the RSC_SQL_AUTO_LOG.TXT file (C:\PROGRAM FILES\RSCSQLAUTO folder). This will indicate any errors or the last time the engine cycled.

2) If nothing appears to be happening then restart the SQL Automation Engine service. In Services it will be listed under RSC Service Manager.

Please note that this service should be running on a server or workstation but ONLY ONE INSTANCE IN TOTAL.

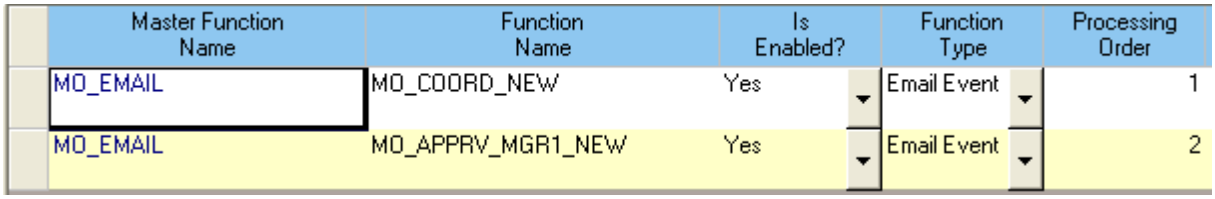
Configuring Email Events

The Email Events table configures emails to be fired and queued via the scheduler.

1) Under Sync Engine - Administrative Functions - Master Functions, create a Master Function and apply it to a schedule

Master Function Name	Description	Is Enabled?	Last Run Date	Sync Log	Date To Run	Time To Run
MO_EMAIL	Move Order Emails	Yes			Daily (7 Days)	08:00

2) Under Sync Engine - Administrative Functions - Functions, create one or more Functions of Function Type "Mail Event"



Master Function Name	Function Name	Is Enabled?	Function Type	Processing Order
MO_EMAIL	MO_COORD_NEW	Yes	Email Event	1
MO_EMAIL	MO_APPRV_MGR1_NEW	Yes	Email Event	2

3) Under Email Engine - Administrative Function - Email Events, create 1 or more email events.

Function Name	Email Data Table	Email Table Primary Key	Email To Field	Email Record Restriction	Email Subject	Email Body Header	Email Body Record	Email Body Footer
MO_APPRV_MGR1_NEW	mo	mo_id	apprv_mgr1	mo_type = 'New Hire' AND date_completed IS NULL AND	New Moves In Progress	Employee ID|Location Bl	<em_id>|<to_bl_id>	Please Review.
MO_COORD_NEW	mo	mo_id	mo_coord	mo_type = 'New Hire' AND date_completed IS NULL AND	New Moves In Progress	Employee ID|Location Bl	<em_id>|<to_bl_id>	Please Review.

- Email Data Table is the table from which records will be processed

- Email Table Primary Key is the primary key field of the table (Only tables with a single primary key field can be chosen).

- Email To Field is is a field from which the email address will be extracted.
  If this is a a foreign key to the EM table then it looks up the email.
  If it is not foreign keyed to EM then it evaluates the field (field value.
  A single email is sent for all records grouped by this field.

- Email To Field 2 is is a field from which the email address will be extracted. This creates a separate email
  If this is a a foreign key to the EM table then it looks up the email.
  If it is not foreign keyed to EM then it evaluates the field (field value).
  A single email is sent for all records grouped by this field.

- Email To Field 3 is is a field from which the email address will be extracted. This creates a separate email
  If this is a a foreign key to the EM table then it looks up the email.
  If it is not foreign keyed to EM then it evaluates the field (field value).
  A single email is sent for all records grouped by this field.

- Email To Field 4 is is a field from which the email address will be extracted. This creates a separate email
  If this is a a foreign key to the EM table then it looks up the email.
  If it is not foreign keyed to EM then it evaluates the field (field value).
  A single email is sent for all records grouped by this field.

- Email To Fixed is is a field from which the email address will be extracted. This creates a separate email
  The field represents the value, not the field name.
  A single email is sent for all records grouped by this field.

- Email CC Field is the field name that stores a list of CC addresses.
  This is added only to an email going to "Email To Field".
  The field parses ctrl, ctrl - linefeed or commas and converts then to semi-colons.

- Email BCC Field is the field name that stores a list of CC addresses.
  This is added only to an email going to "Email To Field".
  The field parses ctrl, ctrl - linefeed or commas and converts then to semi-colons.

- Email Record Restriction is the SQL restriction that chooses the records to be processed.
  e.g. "mo_type = 'New Hire' AND date_completed IS NULL AND move_cancelled = 0"

- Email Subject is the subject line of the email. If blank then it uses the value in the Prototype emails table.

- Email Body Header is the start of the body of the email

- Email Body Record is processed for each record. This is grouped by the "Email To" Field.
  The text is parsed for record values that are defined inside <value> field
  e.g. <bl_id> finds the current record value for the bl_id field.

NOTE: If “Email To” is not set then it creates a single grouped email. The email is sent to “Email To Fixed”.

- Email Body Footer is the end of the body of the email

- SQL On Success. SQL statement to issue after email sent This needs to alter the source record so the email will not be queued again.

- SQL On Failure. SQL statement to issue if an email is not successfully sent Optionally this needs to alter the source record so the email will not be queued
update_sql_failure SQL statement to issue on failure to send email (prevents repeat) VARCHAR 250

After the event runs the results by function are displayed in Master Function table, Sync Log field.
Records to be emailed are placed in the Email Queue for processing.

Demo Email Mode

Email Events and more precisely any email processed from the email queue can be configured to be run in a demo mode in which all emails are directed to a single email address. To enable Demo Mode add the following parameter to the RSCSQLAUTO.INI file (normally in C:\WINDOWS).

DEMO_EMAIL=gdryer@rsc2l.com

With DEMO_EMAIL set all emails will be directed to the set address and all CC and BCC addresses will be excluded.

HTML Formatted Emails

Email Subject, Email Body Headers, Email Body Record and Email Body Footer within Email Events can also support HTML tags. This requires setting a parameter in the RSCSQLAUTO.INI file (normally in C:\WINDOWS).

HTML_EMAIL = 1

NOTE: If configured for HTML email then it is assumed that all email events will use HTML tags

Supported Tags are as follows:

<html> </html>
<head> </head>
<body> </body>
<table> </table>
<tr> </tr>
<td> </td>
<hr> </hr>
<td colspan>

Example:

Email Body Header:

```
<html><head></head><body><table>
<tr><td>Employee</td><td>Start Date</td>
<td>Manager</td><td>Division</td><td>Department</td></tr>
<tr><td colspan="5"><hr></td></tr>
```

Email Body Record:

```
<tr><td><em_id></td><td><date_to_perform></td>
<td><mo_coord></td><td><to_dv_id></td><td><to_dp_id></td></tr>
```

Email Body Footer:

```
</table></body></html>
```

Alternate Email Addresses

Email Events can support alternate email addresses. This is supported if the field em.email_alt (employee table) is populated. This is useful when the admin person should receive the email instead of the manager or if emails should be temporarily be redirected.

The alternate email is used when emails are looked up from the employee table. Fixed email addressed are NOT redirected.

RSC SQL Watch Engine

The additional Watch server (RSCSQLWatch) is an optional service that monitors and allows the main SQL Automation service to be remotely started and stopped via the Archibus view labeled Service Status. RSCSQLWatch also monitors for events that have not run and emails warning.

Remote Access to the RSC SQL Automation Service

When direct access to the server running the SQL Automation Service is not possible, the SQL Watch Engine allows remote access the the SQL Automation Service. The view under Sync Engine - Administrative Functions - Service Status shows the current status of the service. and allows the user to choose a command (Current Command). Commands Are:

Stop - Stop the SQL Automation Engine Service
Start - Start the SQL Automation Engine Service
Verbose Debug - Show all SQL in log file or log table
Debug Errors Only - Show only errors in the log file or log table
Log To Table - redirect log output to the log table (rsc_proc_log)
Log File Only - Log SQL and error output only to the log file (RSC_SQL_AUTO_LOG.txt)

After setting the command the SQL Watch service will perform the command in a maximum of 30 seconds. The Current Command will display None when the command has been completed.

The Current Status and Last Check fields indicate the status of the SQL Automation Engine as of the last test (every 30 seconds).

Process Run Watch

The other function the the SQL Watch service is to report when a scheduled task fails to run. This is written to the log file and an email is sent to ERROR_EMAIL1, ERROR_EMAIL2 and ERROR_EMAIL3 as defined in the RSCSQLAUTO.INI file

Daily Weekly and Monthly events are checked 4 hours after they should have started or the value of the INI variable CHECK_DAILY_EVENT_HOURS. eg CHECK_DAILY_EVENT_HOURS=4.

Please note that Monthly, weekly and daily events (to run only on one day a week) are only checked on the day that they should have run,

Hourly events are checked after one half the time interval. E.g every 12 hour events are checked after 6 hours.

Events scheduled to run at an interval less than one hour are not checked.

Checks tasks, and emails go out every 10 minutes.

Using the GNU Decryption With the SQL Automation Engine

The RUN_SHELL special function can used to call the GNU GPG command line to encrypt or decrypt files. NOTE: Make sure that the RUN_SHELL function Special Function Variable 1 value DOES NOT contain a return character at the end of the line

Install GNU for Windows

1) Download and install GPGforWin from https://www.gnupg.org/download/ (Windows GpG4Win)

2) Run setup with default options other than the install folder. Include GPA in the install option.

3) Run GPA application
- Pick Generate key now.
- Give it a name, email and passphrase
- After generation click on the key and choose Export
- Export to a folder D:\CAFM\AFMDATA\GnuPg (Send backup to this folder)
- Save the passphrase to a file in D:\CAFM\AFMDATA\GnuPg

4) Send the public key to the host. e.g. Workday.

Using GNU To Decrypt a File

Below is an example command line to decrypt a file:

d:\cafm\gnupg\pub\gpg.exe --homedir "d:\cafm\afmdata\gnupg" --batch --passphrase Zynga_Workday_Secret01 --output "d:\cafm\afmdata\download\emsync\current_workers.xml" --decrypt "//10.0.0.51\FileManager\zynga\upload\WD_ARCHIBUS_Current_workers.xml"

gpg.exe needs a path to where the executable was installed

--homedir points to the GNU configuration, passphrase and keys. You need to create this folder and copy the files from the user's home folder in order to run as the system user under the SQL Automation Engine. The default user's home folder will be c:\users\xxxx\AppData\Roamin\gnupg

--passphrase is the passphrase text chosen when generating the keys.

--output is the file to decrypt to

--decrypt is the file to decrypt

Using GNU To Encrypt a File

d::cafm\gnupg\pub\gpg.exe --homedir "d:\cafm\afmdata\gnupg" --recipient "Client_Name Test Key <gdryer@rsc2lc.com>" --output "C:\temp\Archibus full_encrypted.csv" --encrypt "C:\temp\Archibus full.csv"

--homedir points to the GNU configuration, passphrase and keys. You need to create this folder and copy the files from the user's home folder in order to run as the system user under the SQL Automation Engine. The default user's home folder will be c:\users\xxxx\AppData\Roamin\gnupg

gpg.exe needs a path to where the executable was installed

--recipient is the name of the key as displayed in User Name column from the Key Manager program

--output is the file to encrypt to

--encrypt is the file to encrypt from

Importing Public Keys

- To receive encrypted files and then decrypt them, you need to send the third party your Public key.
- To send encrypted files to a third party you need to encrypt then with a Public key that they provide.

1) Save the public key file (.ASC) into the Homedir eg D:\CAFM\AFMDATA\GnuPg
2) Open the KeyToool (GPA) program and go to Key Manager
3) Go to Import and import the Private Key from the provided .ASC file
4) Right mouse click on the Key and choose Set Owner Trust.
5) Set the Owner Trust to Ultimate.
6) Note the that User Name will be the name required in the --recipient value for the encryption command line.
7) Exit GPA
8) Copy all files from C:\Users\name\Application Data\GnuPg to the home folder D:\CAFM\AFMDATA\GnuPg.

GLD