

ConfigInfo User Guide

Decision Resources, Inc.

Table of Contents

Table of Contents.....	2
Introduction	5
Support	5
Active	6
Sheet Name.....	6
Action Dropdown.....	6
FILTERDELETE	6
INVOKEMETHOD	6
INSERT	6
UPDATE	6
INSERTUPDATE.....	7
SELECT	7
FILTERSELECT	7
FILTERSYNC	7
IDO	8
Custom Load Method	8
Service URL	8
Config.....	8
User	9
Password.....	9
RecordCap.....	9
FILTERDELETE	9
INVOKEMETHOD	9
INSERT	9
UPDATE	9
INSERTUPDATE.....	9
SELECT	9
FILTERSELECT	10
FILTERSYNC	10
ChunkSize.....	10

Write To Text File.....	10
StartRow	11
FILTERDELETE	11
INVOKEMETHOD	11
INSERT	11
UPDATE	11
INSERTUPDATE.....	11
SELECT	11
FILTERSELECT	11
FILTERSYNC	11
EndRow	11
FILTERDELETE	12
INVOKEMETHOD	12
INSERT	12
UPDATE	12
INSERTUPDATE.....	12
SELECT	12
FILTERSELECT	12
FILTERSYNC	12
Filter	12
Order By.....	13
Exceptions for FILTERSELECT	13
Exceptions for FILTERDELETE and FILTERSYNC	13
LastRecordDate.....	13
SelectAsString	14
NoValidate	14
FILTERDELETE / INVOKEMETHOD	14
INSERT / UPDATE / INSERTUPDATE for New or Blank Data Tabs	14
INSERT / UPDATE / INSERTUPDATE for Existing Data Tabs.....	14
SELECT / FILTERSELECT / FILTERSYNC for New or Blank Data Tabs	14
SELECT / FILTERSELECT / FILTERSYNC for Existing Data Tabs.....	15
Secondary Table Props.....	15

Derived Props.....	15
Key Props	15
Exclude Props.....	15
FILTERDELETE / INVOKEMETHOD	16
INSERT / UPDATE / INSERTUPDATE	16
SELECT / FILTERSELECT.....	16
FILTERSYNC	16
Append.....	16
Calculate Formulas.....	16
Snapshot Settings	16
Values Only	17
Hide Sheet.....	17
Snapshot Local Directory.....	17
Snapshot SharePoint Directory.....	18
Snapshot SharePoint Directory.....	18
Snapshot SharePoint Site	18
Snapshot SharePoint ClientId	18
Snapshot SharePoint TenantId	18
Snapshot SharePoint Secret.....	18
Snapshot Exchange Folder.....	18
Snapshot Exchange Username.....	19
Snapshot Exchange ClientId.....	19
Snapshot Exchange TenantId.....	19
Snapshot Exchange Secret	19

Introduction

DecisionExcelerator is an application sold by Decision Resources, Inc. (DRI) which provides bidirectional transfer and manipulation of data between Mongoose-based ERPs, such as Infor CloudSuite Industrial (SyteLine), and Microsoft Excel. DecisionExcelerator can communicate equally well with both multi-tenant SaaS and on-premises editions of Mongoose.

DecisionExcelerator requires that Excel files be saved in the XLSX format (Excel 2007 and newer), and the file must reside in a folder accessible to DecisionExcelerator both for reading and writing. These files must contain a tab named "ConfigInfo" which stores the essential parameters for the processing of queries.

The table contained on this tab is used by the application to determine the active queries to be processed and what actions should be taken. Each row defines a separate query. Each column in the configuration table represents a setting that defines how the query behaves. These settings are explained in greater detail below. If optional settings are blank or missing, then they are safely ignored by DecisionExcelerator.

The "Blank ConfigInfo" button in the Excel Add-In provides a blank ConfigInfo tab that has been set up with a configuration table including all possible ConfigInfo settings. The information contained in this document is also provided by that button.

Passwords are visible on the ConfigInfo tab and therefore this file should be treated with care.

Support

Please direct all support requests to: DecisionExceleratorSupport@Decision.com.

Active

This setting must always be present on the ConfigInfo tab. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. Rows in the ConfigInfo settings table are only processed when they are marked as active. DecisionExclerator will only process the file if at least one query is active. This setting allows you to create an expansive ConfigInfo tab where only one, some, or all queries are active.

Sheet Name

This setting must always be present on the ConfigInfo tab. This setting stores the name of the data tab. The value entered here is not case sensitive.

Action Dropdown

ALTERNATE: ACTION

This setting must always be present on the ConfigInfo tab. This setting stores action to be taken by DecisionExclerator. The value entered here is not case sensitive.

FILTERDELETE

THERE IS NO UNDO!!

Requires a license key.

Cannot be ran in background.

Removes records from the system.

Uses the Filter setting on ConfigInfo to limit which records are deleted. A blank filter means EVERYTHING, so a filter expression is always advisable.

INVOKEMETHOD

METHOD and INVOKEMETHOD may be used interchangeably.

Requires a license key.

Invokes an IDO method.

Supply the IDO setting as either “SP!StoredProcedureName” (using the SP! Dynamic IDO), or else as IDOName-dot-MethodName.

INSERT

Requires a license key.

Adds records from Excel to the system.

UPDATE

Requires a license key.

Updates existing records in the system with values from Excel.

Requires a Filter column in the data tab.

Each row in the data tab:

- Finds all records matching the filter expression (up to RecordCap) and updates them.
- A blank filter expression means EVERYTHING, so the filter can be used to effect updates on one or more records.
- For example, you could flip the Extracted flag on EDI records with one Excel row by using the filter expression to specify all records for "today".

INSERTUPDATE

Requires license key.

Tries to update existing records in the system with values from Excel.

If records do not exist in the system, insert them instead.

This is a combination of INSERT and UPDATE actions in one step.

Because of the UPDATE component to this operation, a Filter column is required in the data tab.

SELECT

Without a license key, only the first twenty rows will run.

The data tab must have a FILTER column.

Loops through records on the data tab and retrieves the first matching system record for each filter expression.

FILTERSELECT

Without a license, will only retrieve up to twenty records.

Retrieves records from the system.

Uses the Filter setting in ConfigInfo. A blank filter expression means everything.

The Exclude Props setting does not affect FILTERSELECT. Instead, when writing records to the data tab, only those columns will be written to the data tab where the column name matches an IDO property name in the system. Consequently, you can prevent DecisionExcelerator from overwriting a column in the data tab by naming the column something other than an IDO property name.

FILTERSYNC

Requires a license key.

Must be ran against a data tab that has previously been populated with data.

Looks for recently changed records in the system and then finds and updates existing records in the data tab.

Only those columns that are named with IDO property names will be updated.

IDO

This setting must always be present on the ConfigInfo tab. This setting stores the name of the IDO collection to work with. This value is case sensitive. For INVOKEMETHOD, include both the IDO name and method name.

Custom Load Method

ALTERNATE: CUSTOMLOADMETHOD

This setting is optional. When supplied, it is only used by FILTERSELECT and FILTERSYNC. This setting allows you to override normal IDO Load behavior by calling a custom load method (CLM). Enter the custom load method name and any necessary parameters into this setting.

For example, the GetShipmentListSp method in SLShipments overrides the normal SLShipments load behavior and allows you to return a list of shipment IDs and related values per pick list ID. The method signature requires two parameters: a pick list ID, and the Infobar. In this example, the value entered into the Custom Load Method setting should be similar to this:

```
GetShipmentListSp('439',")
```

Notice that the parameters follow the method name and are encased in parenthesis. The parameter values are surrounded by apostrophes. When multiple parameters are required, they are entered in a comma-delimited list.

Typically, CLMs do not return any values. However, the Infobar parameter frequently appears in CLM signatures.

Service URL

ALTERNATE: SYTELINE WEB SERVICEURL

This setting must always be present on the ConfigInfo tab. This setting stores the URL to the Mongoose IDO Request Service. DecisionExclerator prefers the SOAP endpoint (IDOWebService.asmx) but will use the ASP.Net endpoint (RequestService.aspx) if the SOAP endpoint is unavailable.

Config

This setting must always be present on the ConfigInfo tab. This setting stores the Mongoose environment to connect to. This value is case sensitive.

User

This setting must always be present on the ConfigInfo tab. This setting stores the Mongoose automation username. Adequate permissions must be given to this user for accessing various IDO collections in the system.

Password

This setting stores the password for the Mongoose automation username. This setting is required in most cases.

RecordCap

ALTERNATE: RECORD CAP

This setting must always be present on the ConfigInfo tab, and must always be populated with a numeric value.

FILTERDELETE

The RecordCap setting has no effect.

INVOKEMETHOD

The RecordCap setting has no effect.

INSERT

Records are inserted in chunks of this size (can be any number greater than or equal to zero. Zero commits the full batch in one step).

UPDATE

RecordCap can be any number greater than or equal to zero.
Looks for up to this number of existing records.
Any found records will be updated.
Using RecordCap=0 (unlimited) can break if an enormous number of records are found.

INSERTUPDATE

RecordCap can be any number.
Looks for up to this number of existing records.
Any found records will be updated.
Using RecordCap=0 (unlimited) can break if an enormous number of records are found.

SELECT

The RecordCap setting has no effect. SELECT always returns the first found record.

FILTERSELECT

Looks for up to this number of records.

When RecordCap=0 (unlimited) or RecordCap > 500,000 data is retrieved in chunks of ChunkSize sorted by RowPointer.

RowPointer Required for Large Record Sets

If selecting a large record set (RecordCap = 0 or RecordCap > 500,000) where the IDO collection lacks a RowPointer property, DecisionExcelerator must attempt to retrieve records in a single chunk. Constraints in the system limit the maximum record cap for a single chunk to 500,000.

FILTERSYNC

The RecordCap setting has no effect.

ChunkSize

ALTERNATE: CHUNK SIZE

This setting is not required, but must be populated with a numeric value if used. FILTERSELECT for large record sets (RecordCap=0) retrieves data in chunks of ChunkSize sorted by RowPointer. If ChunkSize is missing or blank, ChunkSize defaults to 30,000.

Performance for large record sets depends on the overall size of the chunk being downloaded (number of columns and number of rows). If performance drops, try using a larger or smaller ChunkSize.

Write To Text File

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This setting only applies to FILTERSELECT. When this setting is enabled, the results from the FILTERSELECT query will be written to an external tab-delimited file rather than the data tab in the current file. The data tab is still required and is used to supply the list of IDO properties that will be retrieved by the FILTERSELECT query. One file will be generated per data tab in the current file. The resulting file will be overwritten to, or appended to, based on the Append setting in ConfigInfo. StartRow and EndRow settings are ignored when writing to the text file. The RecordCap setting controls the number of records that are retrieved and written to the file, with a RecordCap of zero meaning unlimited results. No snapshot actions will occur for a FILTERSELECT query where the Write To Text File setting is enabled.

The resulting file will be named following this convention, taking Config and IDO values from the ConfigInfo tab: {File Name}__{Data Tab Name}__{Config}__{IDO}__DataFile.txt

StartRow

ALTERNATE: START ROW

This setting must always be present on the ConfigInfo tab, and must always be populated with a numeric value.

FILTERDELETE

The StartRow setting has no effect.

INVOKEMETHOD

The IDO method defined in the IDO setting will be ran once per row between StartRow and EndRow (inclusive) where the records in the data tab between StartRow and EndRow represent method arguments.

INSERT

Records in the data tab between StartRow and EndRow (inclusive) will be inserted into the system.

UPDATE

DecisionExcelerator will attempt to update records in the system for each row in the data tab between StartRow and EndRow (inclusive).

INSERTUPDATE

An UPDATE or INSERT will be performed for every row in the data tab between StartRow and EndRow (inclusive).

SELECT

A SELECT will be performed for every row in the data tab between StarRow and EndRow (inclusive).

FILTERSELECT

StartRow is overwritten at the end of each run to display the row number of the first row of new data on the data tab. For large record sets, StartRow will be set to row #2.

FILTERSYNC

StartRow is overwritten at the end of each run to row #2.

EndRow

ALTERNATE: END ROW

This setting must always be present on the ConfigInfo tab, and must always be populated with a numeric value.

FILTERDELETE

The EndRow setting has no effect.

INVOKEMETHOD

The IDO method defined in the IDO setting will be ran once per row between StartRow and EndRow (inclusive) where the records in the data tab between StartRow and EndRow represent method arguments.

INSERT

Records in the data tab between StartRow and EndRow (inclusive) will be inserted into the system.

UPDATE

DecisionExcelerator will attempt to update records in the system for each row in the data tab between StartRow and EndRow (inclusive).

INSERTUPDATE

An UPDATE or INSERT will be performed for every row in the data tab between StartRow and EndRow (inclusive).

SELECT

A SELECT will be performed for every row in the data tab between StarRow and EndRow (inclusive).

FILTERSELECT

EndRow is overwritten at the end of each run to display the row number of the last row of new data on the data tab.

FILTERSYNC

EndRow is overwritten at the end of each run to display the row number of the last row of new data on the data tab.

Filter

The Filter setting is optional. When supplied, it is only used by FILTERSELECT, FILTERSYNC, and FILTERDELETE. This setting provides a filter expression to limit the number of records retrieved (or deleted). Filter expressions rely on SQL syntax with the exception that the expression uses IDO property names instead of SQL field names.

The SQL key words that are known to work are: AND, OR, NOT, LIKE, and IN.

SQL functions such as LTRIM are not known to work.

It is possible to compare two IDO properties, such as QtyOrdered = QtyShipped

Order By

If the filter expression contains the phrase " ORDER BY " then everything after that phrase will be treated as the sorting precedence, which should be a comma-delimited list of IDO property names.

Exceptions for FILTERSELECT

When Selecting Small Record Sets (RecordCap between 1 and 500,000)

DecisionExceleator passes the "ORDER BY" sorting precedence directly to the system and retrieves sorted records from the system as a single step.

When Selecting Large Record Sets (RecordCap=0)

DecisionExceleator first retrieves data in chunks sorted by RowPointer. DecisionExceleator then sorts the retrieved records by the "ORDER BY" sorting precedence as a second step. Consequently, the IDO property names listed in the sorting precedence must also appear in the data tab.

When Selecting Large Record Sets (RecordCap>500,000)

To preserve data integrity, DecisionExceleator will limit the RecordCap to 500,000. DecisionExceleator will then pass the "ORDER BY" sorting precedence directly to the system and will retrieve sorted records from the system as a single step.

Exceptions for FILTERDELETE and FILTERSYNC

Retrieves data in chunks sorted by RowPointer. Any sorting precedence provided in the Filter setting has no effect.

LastRecordDate

ALTERNATES: LAST RECORD DATE, LASTMODIFIEDTIME, LAST MODIFIED TIME,

This is an optional setting. If the LastRecordDate setting is present on the ConfigInfo tab and the RecordDate column is present on the data tab, then FILTERSELECT and FILTERSYNC will update LastModifiedTime at the end of each run with the most recent RecordDate value from the new data.

The value in LastModifiedTime can then be used (via Excel formulas) anywhere a Filter is used. Here is an example of what that Excel formula might look like:

```
= "RecordDate > " & [@LastModifiedTime] & ""
```

The use case being supported here is that the results returned by subsequent runs of FILTERSELECT or FILTERSYNC will be limited by the RecordDate presented in the LastModifiedTime setting, thereby providing a “net change” mechanism.

SelectAsString

ALTERNATE: SELECT AS STRING

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This setting only applies to FILTERSELECT. The columns in the data tab are set to Text instead of General so that Excel does not serialize dates or turn long numbers into scientific notation. If the data tab already exists and is non-blank, DecisionExcellerator does not alter any column formats.

NoValidate

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This is a negative setting. Unchecked means *validate*, and the actions described below occur. Checked means *do not validate* and the actions below do not occur. NOValidate should be checked in most situations. Only leave NOValidate unchecked if you have a specific reason to do so.

FILTERDELETE / INVOKEMETHOD

NoValidate has no effect.

INSERT / UPDATE / INSERTUPDATE for New or Blank Data Tabs

Pull down a list of IDO properties.

Create the data tab if necessary.

Populate column headers if necessary.

Populate property details on row two when populating column headers.

Reset StartRow to row C (row #3)

INSERT / UPDATE / INSERTUPDATE for Existing Data Tabs

Pull down all IDO properties.

Ensure that each column on the data tab has a valid IDO property name.

Evaluate each populated data cell for correct data type and/or string value length.

SELECT / FILTERSELECT / FILTERSYNC for New or Blank Data Tabs

Pull down a list of IDO properties.

Create the data tab if necessary.

Populate column headers if necessary.

Populate property details on row two when populating column headers.
Reset StartRow to row C (row #3)

SELECT / FILTERSELECT / FILTERSYNC for Existing Data Tabs

Pull down all IDO properties.

Ensure that each column on the data tab has a valid IDO property name.

Secondary Table Props

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This setting only applies to validation (NOValidate unchecked) for new (blank or missing) sheets. When collecting a list of IDO property names, include bound properties (PropertyType=0) that belong to a table other than the base table. Secondary properties are always included when validating existing sheets for INSERT, UPDATE, and INSERTUPDATE. Secondary now includes UETs.

Derived Props

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This setting only applies to validation (NOValidate unchecked) for new (blank or missing) sheets. When collecting a list of IDO property names to validate, include unbound and derived properties (PropertyType=1). Derived properties are always included when validating existing sheets for INSERT, UPDATE, and INSERTUPDATE. The Derived Props setting supersedes the Secondary Table Props setting so that Secondary Table Props are automatically included when including Derived Props.

Key Props

This setting is required by FILTERDELETE but has no effect on any other action. Enter a comma-delimited list of key IDO property names. FILTERDELETE collects key field values for all available records (depending on the ConfigInfo Filter column) and then deletes those found records by key.

Identifying the key properties on an IDO is not an exact science, though there tends to be an overlap between key properties and required form fields.

Exclude Props

This setting is optional. Enter a comma-delimited list of data tab column names.

FILTERDELETE / INVOKEMETHOD

The Exclude Props setting has no effect.

INSERT / UPDATE / INSERTUPDATE

The columns listed here will be skipped when writing records to the system.

SELECT / FILTERSELECT

The columns listed here will be skipped when writing records to Excel from the system.

FILTERSYNC

The Exclude Props setting has no effect. Prevent columns in the data tab from being overwritten by naming the column something other than an IDO property name.

Append

This setting is optional. Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. This setting only applies to FILTERSELECT when retrieving limited record sets (RecordCap > 0). When checked, new data is appended to the data tab starting on the first blank row on the data tab. Append DOES NOT CHECK FOR DUPLICATES in case you run the same query multiple times. When not appending (Append is missing or unchecked) the data tab is overwritten when FILTERSELECT runs.

Calculate Formulas

Enable this setting by entering an “X” into this setting. The “X” is not case sensitive.

Immediately prior to processing each active query in the ConfigInfo tab, the ConfigInfo tab will be recalculated regardless of whether this setting is enabled.

When Calculate Formulas is enabled, the current data tab will also be recalculated prior to processing the current active query.

This is especially useful for INSERT, UPDATE, SELECT, and INVOKEMETHOD data tabs where columns may contain Excel formulas that are intended to influence the data being sent to the ERP.

A popular use case is including a VLOOKUP formula in the Filter column on an UPDATE data tab.

Snapshot Settings

As noted previously, Excel files must be closed prior to processing them in DecisionExcelerator to avoid problems related to locked files. Consequently, when an Excel file is processed

routinely, it may be more feasible to distribute a copy of the Excel file so that the original file can remain closed for processing purposes. The snapshot feature provides a mechanism for creating file copies automatically.

When DecisionExcelerator has finished processing the active queries in the current file, it will process the snapshot settings below and then produce a copy of the current file reflective of those settings.

The default name of the snapshot file is the current file name suffixed by a timestamp. For each of the snapshot settings discussed below (Local Directory, SharePoint Directory, and Exchange Folder), you can override this behavior and supply a custom file name. If the value in those settings ends with a slash (back slash for Local Directory and Exchange Folder, forward slash for SharePoint Directory) then the default file name will be used. If the value does not end with a slash, then everything after the last slash will be used as the file name.

All of the settings listed below are optional. When using SharePoint or Exchange snapshot settings, the corresponding OAuth settings must also be supplied.

Values Only

Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. Prior to generating snapshot output, replace any Excel formulas with their calculated results on the data tabs named in active queries where Values Only is also enabled. The effect of this action is only seen in the snapshot file and does not affect the current file being processed.

Hide Sheet

Enable this setting by entering an “X” into this setting. The “X” is not case sensitive. Prior to generating snapshot output, hide the data tabs named in active queries where Hide Sheet is also enabled. The ConfigInfo tab is always hidden in snapshot copies. The effect of this action is only seen in the snapshot file and does not affect the current file being processed.

Snapshot Local Directory

Enter a folder path on the local system, or a UNC path to the local network. This action runs after Calculate Formulas, Values Only, and Hide Sheet. This action saves a copy of the current file to the path specified here. A timestamp is appended to the file name so that this snapshot action never overwrites existing snapshots. The user profile running DecisionExcelerator must have read/write permissions to the network location specified here. If there are multiple active queries bearing different Snapshot Local Directory paths, this file will be copied once to each distinct location.

Snapshot SharePoint Directory

The simplest way to achieve uploading snapshots to SharePoint is to synchronize a SharePoint folder to your local computer, then use Snapshot Local Directory to deposit a snapshot file into that folder on your local system, and then allow OneDrive to synchronize that file up to SharePoint in the cloud.

Alternately, you can use Snapshot SharePoint Directory, and the corresponding Snapshot settings below. These settings utilize Microsoft APIs to upload a snapshot file directly to SharePoint.

Please review the SharePoint Snapshot User Guide document for information about preparing Microsoft 365 to receive Snapshot files. The same setup is required for both SharePoint and Exchange Snapshot files.

Snapshot SharePoint Directory

Enter a SharePoint Online Directory URL. This action runs after Calculate Formulas, Values Only, and Hide Sheet. This action uploads a copy of the current file to the URL specified here. A timestamp is appended to the file name so that this snapshot action never overwrites existing snapshots. If there are multiple active queries bearing different Snapshot SharePoint Directory paths, this file will be uploaded once to each distinct URL.

Snapshot SharePoint Site

Required if Snapshot SharePoint Directory is used. Represents the name of a SharePoint site to upload to within the Office365 tenant identified by the OAuth settings below.

Snapshot SharePoint ClientId

Required if Snapshot SharePoint Directory is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot SharePoint TenantId

Required if Snapshot SharePoint Directory is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot SharePoint Secret

Required if Snapshot SharePoint Directory is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot Exchange Folder

Enter the full name to an Office365 Exchange Online directory (using UNC syntax). This action runs after Calculate Formulas, Values Only, and Hide Sheet. This action creates a memo bearing an inline attachment of this file and uploads the memo to the Exchange Online (Office365)

folder specified here. The memo's subject line (and attached file's name) will be the name of the snapshot file. A timestamp is appended to the file name so that this snapshot can be identified apart from any existing snapshot memos.

Please review the SharePoint Snapshot User Guide document for information about preparing Microsoft 365 to receive Snapshot files. The same setup is required for both SharePoint and Exchange Snapshot files.

Snapshot Exchange Username

Required if Snapshot Exchange Folder is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot Exchange ClientId

Required if Snapshot Exchange Folder is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot Exchange TenantId

Required if Snapshot Exchange Folder is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.

Snapshot Exchange Secret

Required if Snapshot Exchange Folder is used. Represents Exchange Online (Office365) OAuth 2.0 credentials.