

DecisionExcelerator User Guide

Decision Resources, Inc.

Table of Contents

Introduction	5
Support	6
Support Disclaimer	6
Installation	7
IDORequestService Runtime URL.....	8
Intelligent Data Objects (IDOs)	9
IDO User Access	9
Security Considerations	9
User Modules and Permission	9
IDO Projects, Definition, and Properties.....	12
IDO Forms	12
IDO Projects.....	12
IDOs	13
IDO Properties	13
Property Types.....	14
Bound	15
Derived	16
Layers of Logic – User Interface, IDO, and Database.....	17
Excel Data File	18
Configuration Sheet.....	19
Snapshot Exports	20
Data Sheets.....	21
Running DecisionExcelerator	22
License Key	23
Excel Add-In	24
Create ConfigInfo.....	24
SyteLine Forms & Fields.....	24
Launch IDOQuery.....	24
Schedule Queries	24
Download Currency Rates	25

Calculate ADUs	26
Run Simulation.....	26
Copy Outlook Views.....	26
Appendix 1A: ConfigInfo Settings	27
Active	27
Sheet Name	27
Action Dropdown.....	27
SELECT	27
INSERT	27
UPDATE.....	27
INSERTUPDATE	27
METHOD	27
FILTERSELECT	27
IDO	28
Service URL	28
Config.....	28
User.....	28
Password.....	28
RecordCap.....	28
StartRow	28
EndRow.....	28
Appendix 1B: ConfigInfo Settings	29
Filter.....	29
LastModifiedTime.....	29
NoValidate	29
Append	29
Hide Sheet.....	29
Calculate Formulas	30
Values Only	30
Secondary Table Props	30
Derived Props	30
Key Props.....	30

NULL Props.....	30
Exclude Props.....	30
Select as String.....	30
Appendix 1C: ConfigInfo Settings	31
Snapshot Local Directory.....	31
Appendix 1D: ConfigInfo Settings	32
Snapshot SharePoint Directory.....	32
Snapshot SharePoint Username	32
Snapshot SharePoint Password	32
Appendix 1E: ConfigInfo Settings.....	33
Snapshot Exchange Folder.....	33
Snapshot Exchange Username	33
Snapshot Exchange ClientId.....	33
Snapshot Exchange TenantId.....	33
Snapshot Exchange Secret.....	33
Appendix 2: Net-Change Data	34
LastModifiedTime	34
Filter.....	34
Automation.....	34
Appendix 3: Inserting Current Operations.....	35
Appendix 4: Command Line Arguments	36
Example	36

Introduction

DecisionExcelerator is a set of tools, sold by Decision Resources, Inc. (DRI) that allow bidirectional transfer and manipulation of data between Infor CloudSuite Industrial (SyteLine) and Microsoft Excel.

Data is transmitted between Excel and SyteLine through the Mongoose IDO interface. Data transmission operations (queries) are defined within a configuration worksheet in Excel that control what data is transmitted. This toolset includes a selection of sample data files that can be referred to for example purposes.

The primary component of the DecisionExcelerator toolset is the file processor tool. The file processor tool executes queries defined in an Excel data file and either downloads SyteLine data into the worksheets within the same Excel file or else updates SyteLine data from values contained in the same Excel file. The file processor tool is intended to be run with user interaction. However, it is also possible to schedule the tool to run at regular intervals using Windows Task Scheduler.

An Excel Add-In is also provided in this toolset which provides some enhanced functionality that is limited exclusively to the Add-In and is not available within the file processor tool.

This toolset also includes an IDOQuery tool which can be used to browse data and is provided to help with the formulation of query definitions. The IDOQuery tool is analogous to the Infor .NET Web Service Test Utility and can perform the same basic functions.

As of this writing, DecisionExcelerator is only available on MS Windows.

DecisionExcelerator can communicate equally well with both multi-tenant SaaS and on-premises editions of SyteLine.

Support

The following pre-recorded demonstrations have been made available both to demonstrate the product and to set operation expectations:

- <https://youtu.be/QEIBeFkoBig>
- <https://youtu.be/hQrNyMqbfF8>

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

Support Disclaimer

The data files included in the DecisionExclerator installer are provided for example purposes only and are not guaranteed to work. The values and settings in these sample data files are for demonstration purposes only.

All data files are separate from the DecisionExclerator product and are not supported as product. **Support for data files is available as a billable service.** Please contact DecisionExcleratorSupport@decision.com if you need assistance using any data file.

Installation

DecisionExcelerator runs on any Windows 7+ or Windows Server 2008+ machine with access to a SyteLine utility server or SaaS service. Microsoft Excel is not required for DecisionExcelerator to run. However, the Excel Add-In will only run if Excel has previously been installed on the system.

An installer application is provided in a zipped file. Save and unzip this file then run the installer application.

The installer will install the Excel Add-in and will also create a folder named “DecisionExcelerator” in your “My Documents” folder. That folder will include a copy of this help file as well as a selection of sample data files which can be referred for example purposes.

Upon launching Excel for the first time following the installation of DecisionExcelerator, you will be prompted to allow the DecisionExcelerator Add-In to run. Thereafter, the add-in will run immediately.

IDORequestService Runtime URL

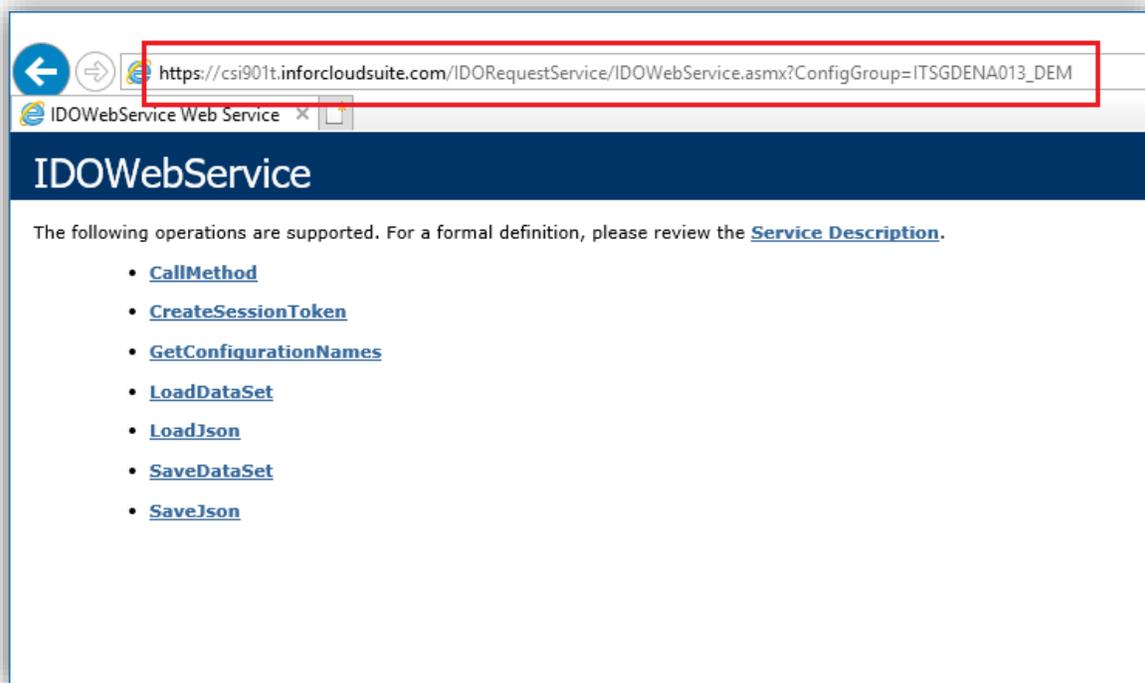
DecisionExclerator communicates with the IDO middleware using the SyteLine Runtime URL.

The main points to note are:

- Is the service running secured or unsecured (HTTP or HTTPS)?
- What is the name of the server hosting the IDO service?
- Is this a multi-tenant SaaS environment, where the ConfigGroup query string parameter is required?

Knowing the answers to these questions will help with providing the correct URL.

DecisionExclerator can also derive the runtime URL from the IDOWebService URL. Shown below is an example of where to find the IDOWebService URL.



Note that in some network architectures, it is possible that the target server may reside on a protected network segment not accessible from a user's computer. To test access, open a web browser on the computer that is to be running DecisionExclerator and attempt to browse to the address identified previously.

Intelligent Data Objects (IDOs)

SyteLine offers IDOs as an external touch point for accessing and manipulating SyteLine data from outside the Mongoose middleware. Much information about IDOs, their use, and SyteLine schema definitions can be found on the Infor Xtreme user support portal.

IDO User Access

Access to an IDO requires the use of a SyteLine user account, wherein a user ID, password, and config name are required to access SyteLine data just like when logging into SyteLine as a normal user.

Security Considerations

The Excel file used by DecisionExcellerator stores the configuration name, user ID, and password information in human readable text. This should be considered when providing access to the Excel files used by this application. A few possible methods for access are presented below for review.

- Log In as “SA”. (Major Security Issues)
- Create a new login for the sole and specific use in this application. (Less Security Issues)
- Use the login of the person running the application. (User is responsible)

In some instances, a password may be blank, however, that is not an ideal security scenario, and the use of passwords are recommended.

User Modules and Permission

User Setup may be required. Please work with the system administrator or person responsible for user setup and permission.

Note that in SyteLine version 10 the SyteLine user account will need to be configured to Allow Remote WinStudio. This can be configured either for the individual user account or can be changed globally in the Process Defaults form. When making changes to the Process Defaults form, you must also discard the IDO cache via the Unload Farm IDO Metadata form.

The screenshot shows the 'Users' configuration window. On the left, a table lists users with columns for 'User ID' and 'User Name'. The first row is '1' with 'DRI', the second is '2' with 'DRIDMA', and the third is '*' with an empty field. The main area is titled 'User' and contains the following fields:

- User ID: DRIDMA
- Super User
- User Description: [Empty text box]
- User Password: [Masked password field]
- Confirm Password: [Empty text box]
- Workstation Domain/ID: [Empty text box]
- Editing Permissions: None (dropdown menu)

Below these fields are buttons for 'User Modules', 'Row Authorizations...', and 'User Authorizations...'. At the bottom, there are tabs for 'Groups', 'Additional Info', 'Login Information', 'E-mail Address', and 'Source Control'. The 'Groups' tab is active, showing a table with columns for 'Group Name', 'Group Description', and 'Primary Group'.

	Group Name	Group Description	Primary Group
1 (n) ▶			<input type="checkbox"/>
*			<input type="checkbox"/>

Buttons for 'Row Authorizations...' and 'Group Authorizations...' are located at the bottom right of the 'Groups' tab.

Note the user has been assigned as “Super User”. This is the most straightforward configuration and provides the login access to all IDO’s as well as the SLParms IDO.

The SyteLineAutomation or Automation user module is required.

The screenshot shows the 'User Modules (Linked)' configuration window. It contains a table with columns for 'User Name' and 'Module Name'.

	User Name	Module Name
1 ▶	DRIDMA	SyteLineAutomation
2	DRIDMA	TLC_SyteLineDocTrak
*		

In some cases, a DocTrack license must be assigned to the user. If you observe a message that refers to LCDT IDO’s ensure the SyteLine user account has the DocTrack license assigned. This is necessary due to the number of IDO’s extended by DocTrack and the IDO extension methodology.

Users may be Super Users or limited to specific middleware object (IDO). While checking the Super User check box on the Users form is easy, this user can read and write to most areas of SyteLine, or Factory Track using the DecisionExcelerator. If the DecisionExcelerator spreadsheets are to be run by other than administrative users, or are to be distributed, it is advised to create users with limited authorizations.

In order to create new sheets, or to validate properties, the user must have access to the IdoTables and IdoProperties MIDDLEWARE on the Object Authorizations for User form, as well as the Object Name (IDO), and of course, the appropriate privileges.

Users		Object Authorizations for User						
User:	<input type="text" value="DRIRO_SLCustomers"/>	Object Type:	<input type="text" value="MIDDLEWARE"/>					
	Object Name	Delete Privilege	Edit Privilege	Execute Privilege	Insert Privilege	Read Privilege	Bulk Update	Update Privilege
1	IdoTables		Revoked	Revoked		Granted		Revoked
2	IdoProperties		Revoked	Revoked		Granted		Revoked
3	SLCustomers		Revoked	Revoked		Granted		Revoked

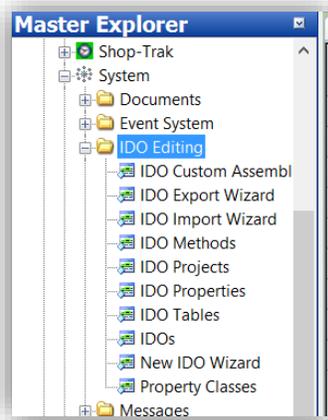
IDO Projects, Definition, and Properties

The sample data files included in this toolset provide examples of several common fields that are already loaded and ready to use. Should more fields be required, or if more information about a field is needed, there are 2 or 3 forms in that would typically be used. ***If you utilize the SELECT method for creating new worksheet templates this is generally not required.***

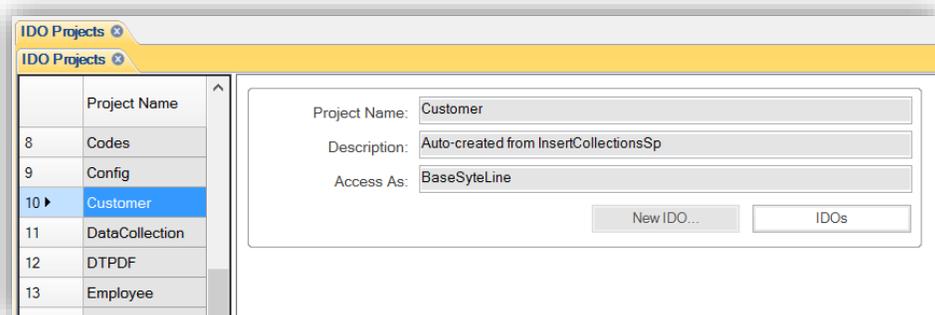
IDO Forms

These forms can be found under “System” and then “IDO Editing” in the Master Explorer.

The hierarchy in SyteLine for the link between forms is IDO Projects -> IDOs -> IDO Properties. Although IDO Properties does appear as a sub-collection in IDOs, the full form provides more detail and filtering options.



IDO Projects



IDO's

The screenshot shows the 'IDO Projects' window with the 'IDOs (Linked)' tab selected. A list of IDOs is on the left, with 'SLCos' selected. The main area displays the configuration for 'SLCos'.

Attributes:

- IDO Name:
- Project Name:
- Description:
- Custom Assembly Name:
- Extends:
- Ext Class Name:
- Ext Class Namespace:
- Replace
- Buttons: Referenced Property Classes, Advanced Attributes...

Revision Info:

- Revision Num:
- Locked By:
- Check Out
- Revision Date:
- Access As:
- Undo Check Out

Navigation: Tables... Properties... Methods... Filters... New IDO... New Table... New Property... New Method...

Property Table:

	Property Name	Sequence	Property Class	Property Type	Column
1 ▶	AckStat	0	AckStatus	Bound to Column	ack_s
2	ApplyToInvNum	0	InvNum	Bound to Column	apply
3	ApsPullUp	0	ListYesNo	Bound to Column	aps_p
4	BillToAddr_1	0	Address	Bound to Column	addr#

IDO Properties

The screenshot shows the 'IDO Properties' window with the 'IDOs (Linked)' tab selected. A list of properties is on the left, with 'CoNum' selected. The main area displays the configuration for 'CoNum'.

Property Attributes:

- Property Name:
- IDO Name:
- Property Class:
- Column Table Alias:
- Property Type:
- Column Name:
- Description:
- Sequence:
- Pseudo Key
- New Property...

Property Overrides: Implementation | Subcollection

Data:

- Data Type:
- Length:
- Decimal Position:
- Default Value:
- Column Data Type:
- *Required
- Read Only Record

Domain:

- Domain IDO Name:
- Domain Property:

Additional List Properties:

Formatting:

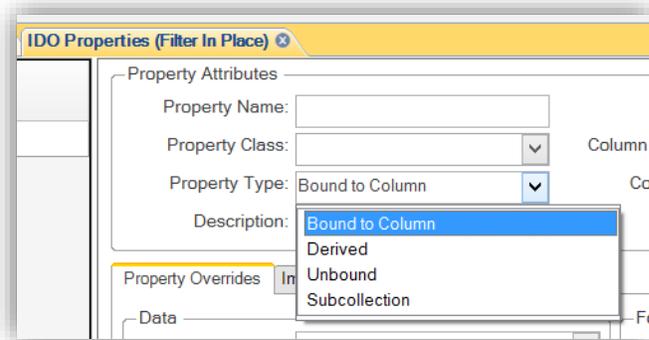
- Label String ID:
- Input Mask:
- Prompt Char:
- Justify:
- Date Format:
- Binary Format:
- IME Char Set:
- Boolean True:
- Boolean False:
- Display Decimal Position:
- Read Only
- Upper Case
- HTML

Property Types

Four types of IDO Properties can be access through SyteLine IDO.

- Bound to a column in a table
- Unbound
- Derived from a SQL expression. (The SQL expression can be used to perform calculations, subqueries, or calls to SQL functions.)
- A sub-collection.

DecisionExcelerator reads and writes to the SyteLine IDO interface and typically will be accessing only fields that are bound to a column in a table. A case may exist where there may be a requirement to access and read one of the derived fields thereby loading a calculated value into the target Excel spreadsheet. Unbound and sub-collections are unlikely to be used. Some value may be had in understanding the difference in these types with a focus on bound versus derived.



Bound

The example below shows the property CustPo in the SLCos IDO. Notice the Property Type is “Bound to Column”. Information on the top right of Property Attributes shows the table alias and the column name. This information is the binding of the property to the column. The tables and field names at the database level may differ slightly from the convention used at the IDO layer.

The screenshot displays the 'IDO Properties (Linked)' window. On the left, a list of properties is shown, with 'CustPo' selected. The main panel is divided into several sections:

- Property Attributes:**
 - Property Name: CustPo
 - Property Class: CustPo
 - Property Type: Bound to Column
 - IDO Name: SLCos
 - Column Table Alias: co
 - Column Name: cust_po
 - Sequence: 0
 - Pseudo Key
 - New Property...
- Property Overrides:**
 - Data:**
 - Data Type: String
 - Length: 22
 - Decimal Position: []
 - Default Value: []
 - Column Data Type: CustPoType
 - Required
 - Read Only Record
 - Domain:**
 - Domain IDO Name: []
 - Domain Property: []
 - Additional List Properties:** []
- Formatting:**
 - Label String ID: sCustPO
 - Input Mask: []
 - Prompt Char: []
 - Justify: []
 - Date Format: []
 - Binary Format: []
 - IME Char Set: []
 - Boolean True: []
 - Boolean False: []
 - Display Decimal Position: []
 - Read Only
 - Upper Case
 - HTML

Derived

The example below shows the property UseFixedSchedule in the SLJobRoutes IDO. Notice the Property Type is “Derived”. Also note that many but not all derived fields start with “Der”. Information on the top right of Property Attributes shows no table alias or column name but instead the tab labeled “Implementation” is enabled. On that tab we can see the expression used to calculate the field.

In this example we know that in the SyteLine application front end the field for Used Fixed Schedule Hours is a checkbox which enables another field where a value can be added for hours. On this form we see how however this field is calculated based off the Schedule Hours field. This may be unexpected and appear the opposite from the user interface. Note that this same functionality does exist in other places such as Offset Hours and other forms in SyteLine.

The screenshot displays the 'IDO Properties (Linked)' window. On the left, a list of properties is shown, with 'UseFixedSchedule' selected at index 40. The main panel shows the configuration for this property:

- Property Name:** UseFixedSchedule
- Property Class:** ListYesNo
- Property Type:** Derived
- IDO Name:** SLJobRoutes
- Column Table Alias:** (empty)
- Column Name:** (empty)
- Description:** (empty)
- Sequence:** 0
- Pseudo Key:** (unchecked)
- New Property...** (button)

The 'Implementation' tab is active, showing the following SQL expression:

```

CASE
  WHEN JshSchedHrs >= 0.00
  THEN 1
  ELSE 0
END
  
```

Layers of Logic – User Interface, IDO, and Database

SyteLine is a multi-tier application. Processing logic exists at more than one tier or layer. The IDO interface is an external touch point to a middle tier. Some logic, validation of fields, or calculations happen at this level but not all. Some of what is seen from the user experience point of view may be occurring at the database level or at the form level. For most forms in SyteLine, some amount of logic happens in all three layers.

An important example of this to consider when loading data through DecisionExcelerator can be seen in the forms for Customer Order Lines and Purchase Order Lines. In both forms, the field labeled Item Description appears when a stocked Item is chosen. The Item's description is pulled from the Item information and this is happening at the form level reacting to user input. This information does not automatically populate at the database level, nor does it populate automatically at the IDO level. The DecisionExcelerator operates at the middle layer and must therefore collect and upload this information or the field will be left blank (NULL) and would need subsequent correction.

Excel Data File

As noted previously, Excel is not required for DecisionExcelerator to run.

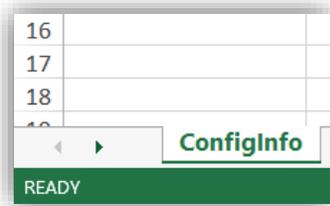
DecisionExcelerator only works with Excel files that are saved in the XLSX format (Excel 2007 and newer).

The Excel file must reside in a folder accessible to DecisionExcelerator both for reading and writing.

DecisionExcelerator actively edits the file during processing so the file must be closed prior to processing to prevent blockages due to file locking. Also, the folder permissions must be set so that the file can be written to and modified by DecisionExcelerator.

Configuration Sheet

The configuration information is stored in a worksheet named "ConfigInfo".



This sheet contains essential parameters for the processing of a query. The table shown below is used by the application to determine the active sheets to be included and how they will be processed. Each row defines a separate query. Passwords are visible on this sheet and therefore this file and this sheet should be treated with care.

Active	Sheet Name	Action dropdown	IDO	SyteLine Web ServiceURL	Config	User
	Jobs	FILTERSELECT	SLJobs	https://csi901t.inforcloudsuite.com/ID	DECISION_DEM_DALS	DRIAutomation
	EDIPricingInSL	FILTERSELECT	SLitemprices	https://csi901t.inforcloudsuite.com/ID	DECISION_DEM_DALS	DRIAutomation
	MatlTran	FILTERSELECT	SLMatltrans	https://csi901t.inforcloudsuite.com/ID	DECISION_DEM_DALS	DRIAutomation
	SLLedgers	FILTERSELECT	SLLedgers	https://csi901t.inforcloudsuite.com/ID	DECISION_DEM_DALS	DRIAutomation
	EDIContractPricing	INSERT	SLitemCustPrices	https://csi901t.inforcloudsuite.com/ID	DECISION_DEM_DALS	DRIAutomation

Each column in the configuration table represents a setting that defines how the query behaves. These settings are explained in Appendix 1. If settings are blank or missing, then they are ignored by DecisionExcelerator. Those who have used the DMA application in the past will recognize some of these settings.

Snapshot Exports

When running DecisionExcelerator, the snapshot activity will run after all active queries are processed. The following criteria must be met for snapshot activity to occur:

- Snapshot is only available for sheets that are marked Active in ConfigInfo settings.
- Snapshot is only available for sheets that use the FILTERSELECT action.
- Snapshot is only available when the worksheet corresponding to Sheet Name exists in the file.
- Snapshot will only occur when that worksheet has at least one row of data.

Note that snapshots will copy the entire file to the specified location. The ConfigInfo worksheet will be removed from the exported copy. However, all other sheets will remain included in the file. Consequently, you may desire to use the Hide Sheet action to make one or more worksheets invisible in the exported copy.

FILTERSELECT removes all existing data from a worksheet prior to running the new query unless the Append action is selected. Without using Append, you will only get exported snapshots when the query returns data. However, you may get inconsistent results when using the Append action.

Data Sheets

The first row of the worksheet must contain the names of the IDO properties that do or will exist in that sheet. These property names must exactly match how they appear in SyteLine (case sensitive). Begin at the first column (A1) and proceed to the right. DecisionExcelerator stops processing at the first empty or blank column. You may place unused property columns after a blank column.

Naming must precisely match (case sensitive) for both the IDO name and IDO property names.

UETs can be accessed with DecisionExcelerator but property name definitions require special formatting. UET's must follow the default Syteline naming convention of appending the "Uf_" prefix to the property name. In addition, the table alias must be included in the property name. For example, a UET on the Items table would be included in the template with a property name of: itmUf_fieldname.

If Extended IDO properties are included in the list of properties to update or retrieve, the log will record that an IDO Property defined in the data file is not part of the standard base IDO and that these extended IDO properties and should be closely reviewed.

The data in some fields are space padded or fully expanded to the full field length. Examples are Customer Order Number (CoNum) and Customer Number (CustNum). DecisionExcelerator generally handles those well, but careful consideration should be made when working with these key properties. If you are loading data with an alpha prefix that will be fully expanded in the SyteLine form, you should carefully prepare and pad the data. It is best to format these fields as text and space pad or fully expand your source data – A000001 instead of A1.

The general rule to follow is to ensure your data conforms to the data type as specified in the IDO or returned by the SELECT process.

Running DecisionExcelerator

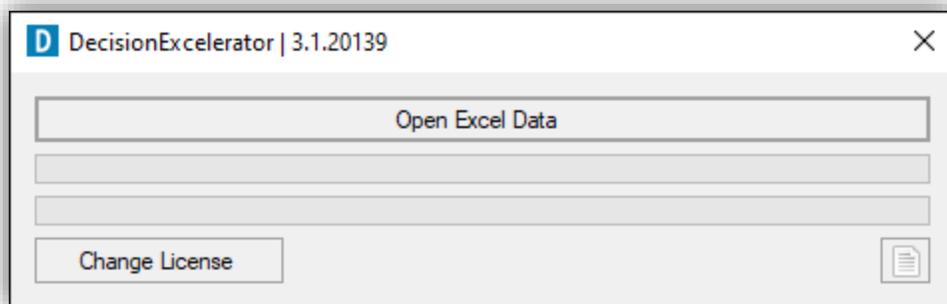
When the installer runs, it will place a shortcut on the desktop. You can also find the executable in the “DecisionExcelerator” folder in “My Documents”. This folder will be created by the installer.

DecisionExcelerator accepts command line arguments (see Appendix 4), so that it can be launched directly from the shortcut, or it can be called from the command line.

Look for this icon:



This is what the DecisionExcelerator looks like when it runs. The version number on the title bar will reflect the current installed version.



As DecisionExcelerator runs, the text on the application will change to keep you updated with the program’s progress. Pressing the Show Log button in the lower right corner will display this same information in a pop-up dialog. There are also two progress bars that will fill up as the program moves towards completion. The top progress bar tracks the progress of an entire file while the bottom progress bar tracks the progress of an individual (active) sheet. Additional information will be populated in a log file that will be placed into the same directory as the data file. Both provide sources of valuable information in case the active queries do not behave as expected.

License Key

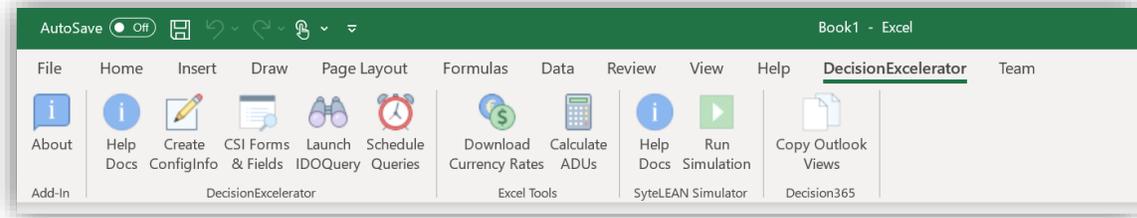
When you obtain a license key from DRI, you press the Add License button and enter it there. You will need to update the key file prior to processing a file. When processing a file without a key entered, DecisionExcelerator will limit the number of processed data rows to 20, regardless of the row count indicated by the StartRow and EndRow settings in the ConfigInfo table.



The license key is stored in user-specific settings on the computer. Therefore, if DecisionExcelerator is used by another user or on another machine, then the license key will have to be reapplied.

Excel Add-In

The DecisionExcelerator Excel Add-In appears as a ribbon in Excel.



Create ConfigInfo

If you wish to create a new data file, this button will create a blank, properly formatted ConfigInfo worksheet in the current workbook. You can then populate the configuration table with your values.

SyteLine Forms & Fields

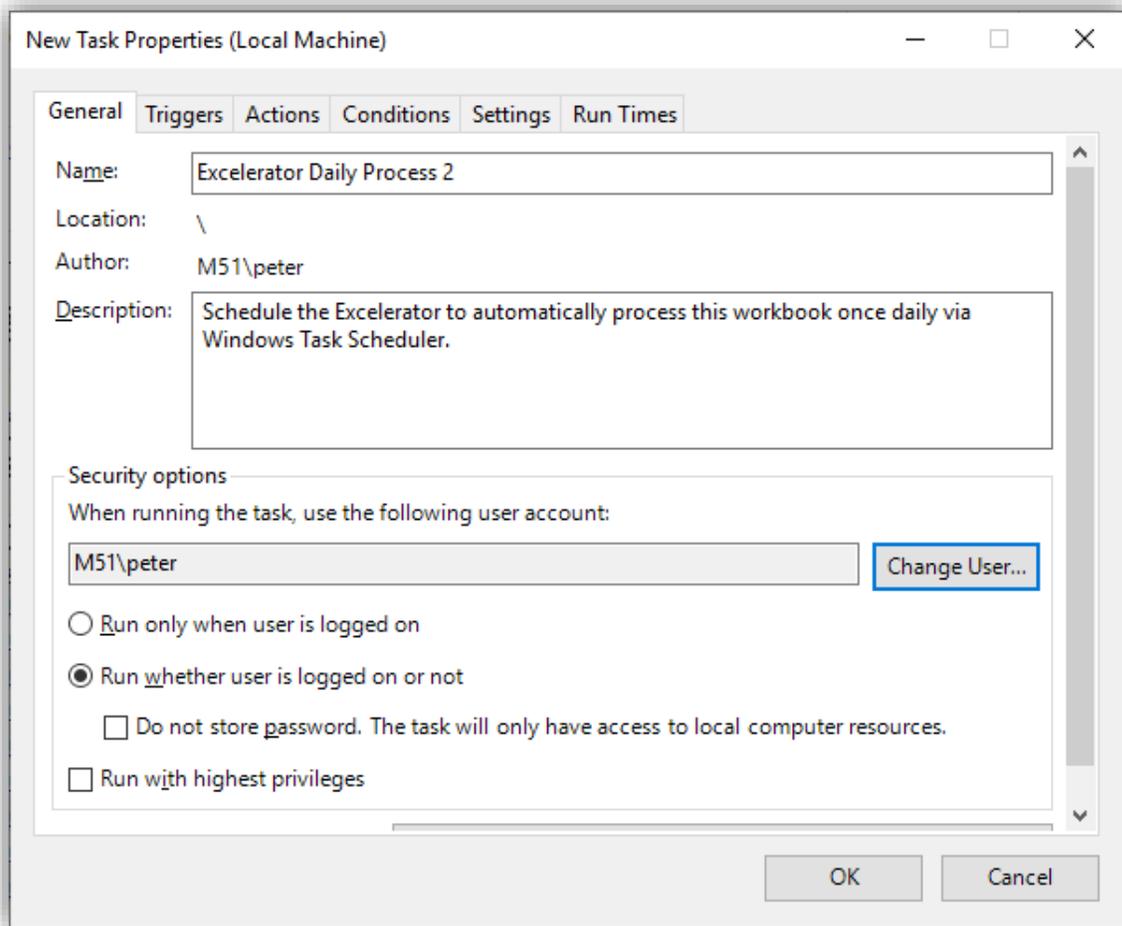
When designing queries in data files, it is necessary to understand the IDO collections and Properties that you are working with. This button displays a dialog of SyteLine Forms and form fields and shows the underlying IDO collection and properties upon which the form operates. Upon selecting a specific form, you can then view the fields displayed on that form, and their corresponding IDO Properties. The information gleaned here can help guide you as you construct a query.

Launch IDOQuery

DecisionExcelerator ships with an IDOQuery tool that is similar in nature to the Infor .NET IDO Web Service Test Utility and can be used to validate IDO requests and the resulting data sets.

Schedule Queries

When running DecisionExcelerator with command line arguments (see Appendix 4), DecisionExcelerator does not require user interaction to run. As a result, it can be scheduled to run at regular intervals via Windows Task Scheduler. Such a feature might be valuable if you wanted to download a table of data from SyteLine daily where that data could serve as the foundation for a dashboard. Pressing the Schedule Queries button will open a task scheduling dialog. Confirming your scheduling options will cause DecisionExcelerator to generate a Windows Scheduled Task which will launch DecisionExcelerator at the specified interval. Note that by default the scheduled task will cause DecisionExcelerator to process the data file that you have open currently.



Download Currency Rates

The Download Currency Rates button will download international currency exchange rate information from the internet. This information will be formatted in such a matter that it could then be loaded directly into the CurrencyRates IDO (Currency Rates form).

	A	B	C	D	E	F
1	FromCurrCode	ToCurrCode	BuyRate	SellRate	EffDate	UserCode
2	USD	AED	3.6732	3.6732	07/17/2019 04:11:35 PM	DRI
3	USD	AFN	80.250369	80.250369	07/17/2019 04:11:35 PM	DRI
4	USD	ALL	108.804886	108.804886	07/17/2019 04:11:35 PM	DRI
5	USD	AMD	476.450056	476.450056	07/17/2019 04:11:35 PM	DRI
6	USD	ANG	1.78005	1.78005	07/17/2019 04:11:35 PM	DRI

Calculate ADUs

Pressing Calculate ADUs will run a SyteLEAN simulation against SyteLine and will produce an Average Daily Usage (ADU) calculation for the top 100 most active items in your inventory. Contact DRI for more information about this feature.

Run Simulation

This button launches the SyteLEAN Simulator Utility. This utility provides an interactive demonstration of the capabilities of the SyteLEAN product. Press the corresponding Help Docs button for further information.

Copy Outlook Views

This button launches the Outlook View Copy Utility. This utility provides the means to distribute Outlook folder view definitions. Views can be selected from a mailbox folder within a mail profile. The selected view definition will be exported to the local computer as an XML file. Subsequently, XML files containing view definitions can be imported into a mailbox folder. If you have multiple mail profiles set up on your computer, this utility provides the means to copy views from one profile to another. This utility is primarily useful when building workbenches for the Decision365 product.

Appendix 1A: ConfigInfo Settings

The settings listed below are the basic set of required settings without which DecisionExceleator will not run.

Active

X = True. Activates this query.

Sheet Name

Name of the data worksheet (Case Sensitive).

SELECT/FILTERSELECT: Where retrieved SyteLine records will be deposited.

INSERT/UPDATE: Where changed values can be found.

METHOD: where parameters can be found.

Action Dropdown

SELECT

Retrieve records from SyteLine. Filter column required on data worksheet.

INSERT

Insert records into SyteLine. When inserting records into SLJobRoutes, DecisionExceleator will validate and/or update the job number on the data worksheet and will also invoke the PreSaveCurrOperSp and PostSaveCurrOperSp procedures so that Job Operations will be populated correctly.

UPDATE

Update existing records in SyteLine with values from Excel. Filter column required on data worksheet.

INSERTUPDATE

Attempts to update existing records in SyteLine with values from Excel. Filter column required on data worksheet. If records cannot be updated because they do not exist, the values will instead be inserted as new records in SyteLine. Unlike the INSERT action, the insert portion of INSERTUPDATE is just a simple data insertion and possesses no special logic for SLJobRoutes.

METHOD

Invoke IDO method using parameters from Excel and return output parameters to Excel. Expert use only. Contact DRI support.

FILTERSELECT

Retrieve records from SyteLine. Filter column in ConfigInfo used. Filter column on data worksheet is ignored if it exists.

IDO

Name of the SyteLine IDO (Case Sensitive).

Service URL

IDORequestService Runtime URL – you can supply the IDOWebService URL instead as needed.

Config

Name of configuration for login. Determines “site”.

User

SyteLine username for login.

Password

Password for login (Case Sensitive).

RecordCap

SELECT: The number of records to retrieve. Should be (EndRow – StartRow)

INSERT/UPDATE:

1 = Insert all records individually (rollback will affect single record)

>1 = Insert up to this many records as a single transaction (rollback will affect
all records in the transaction)

For SELECT queries, when running the query for the first time, if the RecordCap is greater than 200 rows, the query will only return the top 200 rows. Run the query a second time to get the full RecordCap amount. This is a safety feature to prevent poorly designed queries from running endlessly.

StartRow

The row on the data sheet where data records do or should begin.

EndRow

The row on the data sheet where data records do or should end.

FILTERSELECT: Will be updated after the query runs to reflect the end row for the data that was returned.

Appendix 1B: ConfigInfo Settings

The settings listed below are optional settings that alter how the query behaves. Because these settings are optional, you may remove the setting column entirely from ConfigInfo and the query will still run.

Filter

An IDO filter expression used by SELECT queries. The filter expression can also be used to sort the returned data. AT THE END of the filter expression, add the words ORDER BY followed by a comma-delimited list of IDO properties which represent a sort precedence. For example:

```
PMTCode = 'P' ORDER BY ProductCode,Item
```

LastModifiedTime

FILTERSELECT: The maximum RecordDate value returned by the query. Otherwise the runtime of the query (in UTC).

See Appendix 2 for an explanation of usage.

NoValidate

X = True. Prevents validation of the data sheet.

When running SELECT, INSERT, or UPDATE queries for the first time, DecisionExceleator can do several things for you including: create the data worksheet if necessary, populate IDO property column headers, and validate the length of values in string fields per the IDO property definition.

When creating IDO property column headers, row 1 will contain the IDO property name. Row 2 will contain some meta data about the property, then the data records will begin in Row 3. For this reason, StartRow should be 3 when not using NoValidate.

The property metadata includes Data type, table alias, and data length.

For example: String/clf/6.

Append

X = True. For FILTERSELECT only. This will find the first empty row after the StartRow and ADD to it. The rows before StartRow are not considered.

Hide Sheet

X = True. The sheet will be hidden after the query runs. **Hiding does not work properly in Office365 web and mobile applications.**

Calculate Formulas

X = True. Forces all cell formulas to recalculate after all queries have run (recalculate without first opening the workbook).

Values Only

X = True. Replaces cell formulas with their calculated values.

Secondary Table Props

X = True. Includes extended IDO properties in SELECT and FILTERSELECT queries.

Derived Props

X = True. Includes derived IDO properties in SELECT and FILTERSELECT queries.

Key Props

FILTERDELETE (required). These are the key IDO properties required to delete an IDO record.

NO UNDO - expert use only.

NULL Props

A list of field names that exist in the data tab. These are not required to be valid IDO property names. The value for each row will be set to "NULL" and will not be included in the IDO query.

Exclude Props

A list of field names that exist in the data tab. These are not required to be valid IDO property names. The value for each row will remain unchanged and the field will be excluded from the IDO query.

Select as String

X = True. Applies only to FILTERSELECT. Dates and numbers will be downloaded as text. These will display nicely in Excel without extra formatting. However, they cannot be used in formulas.

Appendix 1C: ConfigInfo Settings

The settings listed below are optional settings that determine if and how copies (or snapshots) of DecisionExceerator data files are exported. Because these settings are optional, you may remove the setting column entirely from ConfigInfo and the query will still run.

Snapshot Local Directory

A copy of this file will be saved to the local folder indicated. The file name will be appended with DateTime. The file will contain all the sheets from this file except for ConfigInfo (this sheet). Data tabs can optionally be hidden using Hide Sheet. If multiple folders are listed below, this entire workbook will be copied to all those locations. You must have Read/Write permissions to this location. BLANK: No copy.

Appendix 1D: ConfigInfo Settings

The settings listed below are optional settings that allow DecisionExcelerator snapshots to be uploaded to a SharePoint directory. Because these settings are optional, you may remove the setting column entirely from ConfigInfo and the query will still run.

Snapshot SharePoint Directory

A copy will be saved to SharePoint Online using the path indicated with DateTime appended to the file name. The file will contain all the sheets from this file except for ConfigInfo (this sheet). Data tabs can optionally be hidden using Hide Sheet. If multiple paths are listed in the ConfigInfo table, the entire workbook will be copied to all those locations. Only works with SharePoint Online and URL must reference sharepoint.com.

BLANK: No copy.

Snapshot SharePoint Username

This setting is only required when Snapshot SharePoint Directory is provided. The user entered here must have permission to write to the SharePoint location.

Snapshot SharePoint Password

This setting is only required when Snapshot SharePoint Directory is provided. If the Office365 account associated with Archive Username requires MFA, then this password needs to be an app-specific password.

Appendix 1E: ConfigInfo Settings

The settings listed below are optional settings that allow DecisionExcelerator snapshots to be uploaded to an Exchange shared mailbox. Because these settings are optional, you may remove the setting column entirely from ConfigInfo and the query will still run.

Snapshot Exchange Folder

Exports a snapshot to Exchange. Only works with Office365 using OAuth over the Exchange Web Services API.

The path to the folder should be provided here. For example:

```
\\Public Folders - SFT0000041@SFT0000041.onmicrosoft.com\All Public Folders\Public  
DecisionExcelerator Output\
```

Blank: no export.

Snapshot Exchange Username

This setting is required when a Snapshot Exchange Folder is provided. This is the email address of the Exchange user who will be responsible for writing to Exchange. This user must have permission to write to the folder.

Snapshot Exchange ClientId

This setting is required when a Snapshot Exchange Folder is provided. This value is defined when setting up an App Registration in Azure Active Directory for a corresponding Office365 account.

Snapshot Exchange TenantId

This setting is required when a Snapshot Exchange Folder is provided. This value is defined when setting up an App Registration in Azure Active Directory for a corresponding Office365 account.

Snapshot Exchange Secret

This setting is required when a Snapshot Exchange Folder is provided. This value is defined when setting up an App Registration in Azure Active Directory for a corresponding Office365 account.

Support for configuring Office365 OAuth access is available as a billable service.

Please contact DecisionExceleratorSupport@decision.com if you need assistance.

Appendix 2: Net-Change Data

The Filter and LastModifiedTime settings in ConfigInfo can be used together as part of a FILTERSELECT query to gather net-change data where the query returns a full data set during the initial run and then only returns new or modified data during subsequent runs.

LastModifiedTime

To make use of LastModifiedTime, first ensure that the LastModifiedTime field exists in the ConfigInfo table. The field can be left blank. Next, ensure that the RecordDate field is included in the data tab where the results from the FILTERSELECT query will be written.

When the query runs, DecisionExceerator will download the corresponding RecordDate for each record. After writing the data to the data tab, DecisionExceerator will find the MAX(RecordDate) and will write that value to the LastModifiedTime field in ConfigInfo. If a value has already been populated in the LastModifiedTime field, the old value will be overwritten. The RecordDate value will be written to the LastModifiedTime field using the format: "yyyy-MM-DD HH:mm:ss.fff". As such, the value written to LastModifiedTime can immediately be used without modification in the Filter field.

Filter

During the initial run of the query, the Filter field in ConfigInfo should be left blank, or populated with whatever filter represents a complete data set. On subsequent query runs, utilize the LastModifiedTime field to ensure that you are only downloading records from SyteLine that have a RecordDate that is newer than the maximum RecordDate that was previously downloaded. Shown below is a formula that might be used to achieve this.

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

NOTE: In Excel, formulas are not interpreted as formulas when the cell format is set to TEXT. Therefore, you must ensure that the cell format for the Filter field in ConfigInfo is set to GENERAL. This will allow Excel to evaluate the formula correctly.

Automation

When saving the Excel file at the end of the query run, the formulas in the ConfigInfo table will be recalculated. This means that you could then, for example, run the Excel file hourly and both the LastModifiedTime and Filter fields in ConfigInfo would be continuously updated. You could then schedule DecisionExceerator to run as a Windows Scheduled Task with the ability to download only new or modified (net-change) data at regular intervals. When combined with Snapshots, this presents the ability to download and export net-change data as part of a reporting or data migration strategy.

Appendix 3: Inserting Current Operations

A multi-step process is required to insert Current Operations. You can query existing Current Operations by running a FILTERSELECT query on the SLJobRoutes collection. However, writing to the SLJobRoutes collection is not sufficient by itself to create new Current Operations.

DecisionExcelerator can create Current Operations in a single step using a custom IDO method.

An IDO Custom Assembly named DRI_SLJobRoutes can be found in the same zip archive folder as the DecisionExcelerator installer. After installing DecisionExcelerator, a copy of this IDO Custom Assembly can be found in the “DecisionExcelerator” folder under “My Documents”.

Installing the IDO Custom Assembly into SyteLine will create an IDO collection of the same name. This IDO will contain an IDO method named “InsertCurrentOperations”.

An example for calling this IDO method can be found in the Demo10-InsertCurrentOperations data file that is installed into the “DecisionExcelerator” folder under “My Documents” by the DecisionExcelerator installer. All the fields that are populated in the “Operations” data tab should be considered required values.

This IDO Custom Assembly is provided as-is and is only known to work with SyteLine version 10.

Support for importing, maintaining, and using the DRI_SLJobRoutes IDO is available as a billable service.

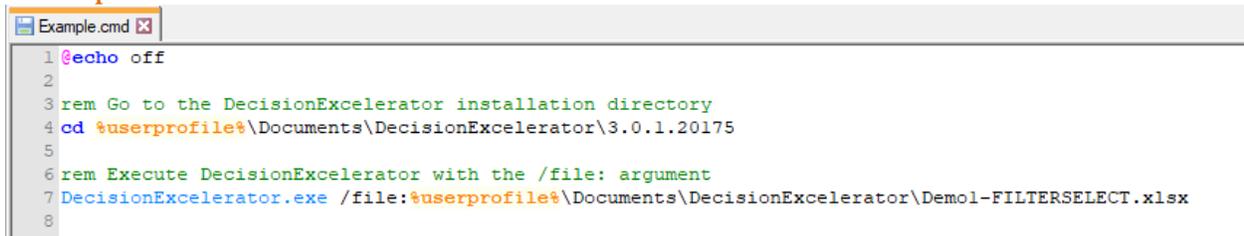
Please contact DecisionExceleratorSupport@decision.com if you need assistance importing or using the DRI_SLJobRoutes IDO.

Appendix 4: Command Line Arguments

It is possible to pass the name of the Excel file to be processed as a command line argument. When doing so, the normal User interface does not display, and no user interaction is required. One example where the command line is useful is when scheduling DecisionExcelerator to run as a Windows Scheduled Task.

The command line argument is **/file:** followed immediately with the path to the Excel file.

Example



```
1 @echo off
2
3 rem Go to the DecisionExcelerator installation directory
4 cd %userprofile%\Documents\DecisionExcelerator\3.0.1.20175
5
6 rem Execute DecisionExcelerator with the /file: argument
7 DecisionExcelerator.exe /file:%userprofile%\Documents\DecisionExcelerator\Demol-FILTERSELECT.xlsx
8
```