

# Extract Data Items from ci360:Direct onprem logs

Raja Marla

Last updated: 12/05/2025

## Table of Contents

Release History .....	3
Installation package contents .....	3
Description .....	3
Assumptions.....	4
Customization Instructions .....	4
Scheduling.....	4
Windows: .....	4
Linux.....	6
Program outputs .....	6
Last-Update-file (Flat file) .....	6
Outlib.runtask .....	7
Outlib.export.....	7
Outlib.node .....	7
Outlib.calc_items .....	8
Outlib.segmap_data.....	8

## Release History

01/20/2024     Improvements

- Capture calc items and their definitions
- Capture the data item selection criteria (in addition to their name)

Bug fixes

- Capture data item names longer than 50 characters. Newer limit is 100 characters

12/05/2025     Save generated datasets to a repository (SAS datasets in a library of users' choice)

Sample reports that show how to use the datasets to generate reports

Sample scripts (Windows/Linux) to execute the tool via task scheduler on windows or cron on linux.

## Installation package contents

	File name	Description
1	Utl_onpremdiext.sas	The main program that extracts data items from the onprem direct log
2	Utl_onpremdiext_rpt.sas	Sample report(s) can be generated using this code. They can be customized to your implementation needs
3	Run.sh	Batch script to execute utl_onpremdiext.sas on Linux
4.	Run.cmd	Batch script to execute utl_onpremdiext.sas on Windows
5	Sample_reports.html	Sample reports generated by executing utl_onpremdiext.sas
6	Email_error.sas	Generates an email alert upon error This file needs to be customized to specify correct smtp email server name and the email address of the administrator.
7	Onprem_Direct_Log_Parser_Data_Items.pdf	This file

## Description

This utility program (utl\_onpremdiext.sas) reads the Engage:Direct onprem logs (but never the current days' log) and produces SAS datasets that help cross-reference each segment map/direct task executed with the data items in the Information Map.

The program is designed to be run once a week (or once a day or another frequency). It saves the last processed file date and continues processing from the next date. At each invocation the program

processes a maximum of thirty-one files (configuration options). Processing too many files may result in longer execution times.

The output datasets (described below) are based on the current execution.

Note:

Every execution of each task/segment is preserved in the output dataset. It is normal for a single task/segment to be changed over time and re-executed. Thus, many historical versions may exist for each segment/task. This program does not attempt to remove the historical versions.

## Assumptions

1. The onprem logs are named onprem\_direct.log.yyyy-mm-dd.
2. All the onprem logs to be read are from the same directory/folder

## Customization Instructions

Refer to utl\_onpremdiext.sas and update the variables prefixed “c\_”. Additional description is included in the sas program.

The variable `c_extract_dataitem_values` controls whether to retrieve data items values from criteria nodes. If you are only interested in data item names rather than data item values, you may choose to set `c_extract_dataitem_values=N`

Also – update the file `email_error.sas` to specify the correct smtp server name and the email address that should be alerted upon error.

## Scheduling

You can maintain a history of data item usage in CI360 Direct, by executing the script periodically and saving the resultant data. This will enable analysis of data items which are used more often vs those that are used less or not-at-all and access optimization at the database level.

Scheduling steps vary depending on the platform you use – Windows or Linux, as described below.

### Windows:

Start the Microsoft Windows task scheduler and use the menu option Action->Create task...and specify a name, for eg., `Extract_data_items`, and specify a service account that you wish to use to execute the task and choose the option to run regardless of whether the user is signed on. Create a new trigger and specify the execution time and frequency. A sample set up is shown below.

**New Trigger**

Begin the task: On a schedule

**Settings**

☐ One time    Start: 12/ 6/2025    4:00:11 AM    ☐ Synchronize across time zones

☒ Daily

☐ Weekly

☐ Monthly

Recur every: 1 days

**Advanced settings**

☐ Delay task for up to (random delay): 1 hour

☐ Repeat task every: 1 hour for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

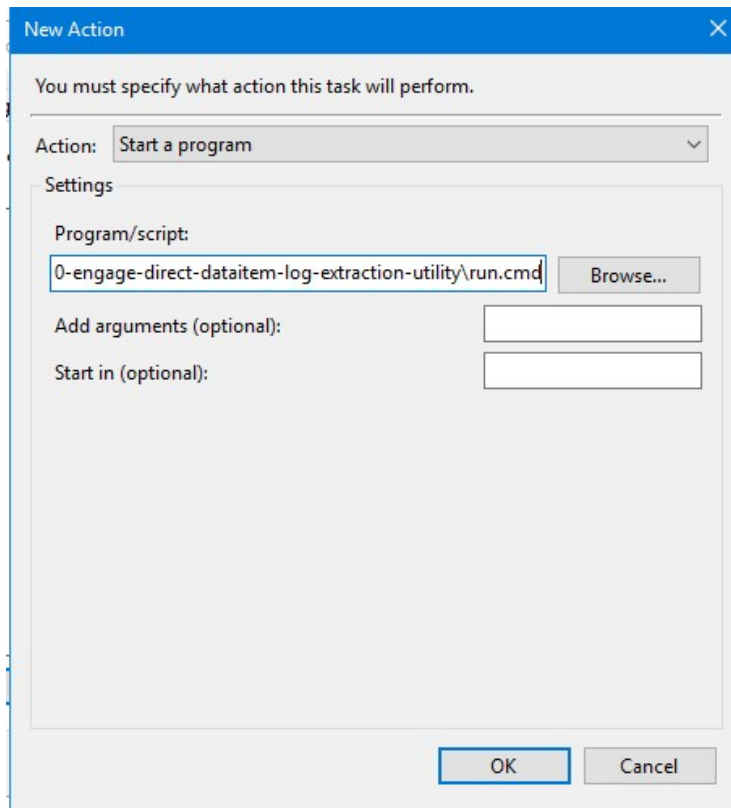
☒ Stop task if it runs longer than: 4 hours

☐ Expire: 12/ 6/2026 12:06:11 PM    ☐ Synchronize across time zones

☒ Enabled

OK Cancel

Navigate to the “Actions” tab and create a new... action. And specify the location of batch file run.cmd. A sample screenshot is shown below



Click Ok, Ok to save the task. You may be prompted for the service account password.

Linux

Edit the crontab to specify a schedule. An example to execute the script at 4am every day would be:

```
0 4 * * * /userdata/dev/common/projects/ci360-engage-direct-dataitem-log-extraction-utility/run.sh utl_onpremdiext.sas
```

Program outputs

Last-Update-file (Flat file)

The location/name of file is specified at the top of program as a configuration option. It contains a single record, containing a date specified in the format yyyy-mm-dd. If you wish to

process historical files on the initial execution, you will need to create this file with a single record (If you wish to process files from 01-Feb-2022, then specify the date 2022-01-31.)

If the file does not exist, then one would be created after the process termination. In this case, only the one log file for the previous day would be processed.

#### Outlib.runtask

This SAS dataset is appended after each execution with the data from the logs processed during that run. It contains one row for each task/segment executed in the log files.

Name	Type	Length	Format
task_name	Character	100	
type	Character	15	
externalCode	Character	15	
modifiedByUserName	Character	50	
runtask_id	Character	40	
log_filename	Character	100	
task_runtime	Date	8	E8601DT.
countsOnly	Numeric	8	
businessContextU...	Character	36	
file_record_nbr	Numeric	8	

#### Outlib.export

This dataset relates each task/segment with the list of data items exported to one or more DM task outputs.

Name	Type	Length	Format
externalCode	Character	15	
runtask_id	Character	40	
dataitem_name	Character	100	
outputName	Character	60	
outputSubjectId	Character	50	
task_runtime	Date	8	E8601DT.

#### Outlib.node

This dataset relates each DM task/segment nodes with the corresponding data items from the information maps. At the present time, “node” refers to “criteria/segment or split” nodes.

Name	Type	Length	Format
externalCode	Character	15	
runtask_id	Character	40	
segmentMapN...	Character	50	
dataitem_name	Character	100	
node_name	Character	60	
outputSubjectId	Character	50	
task_runtime	Date	8	E8601DT.
dataitem_value	Character	200	
operator_name	Character	20	
node_type	Character	20	

#### Outlib.calc\_items

This data captures the calc item name and the expression. Calc item names on the “node” table are represented by an id. The node.id is a foreign key to the calc\_items.ci\_id.

Name	Type	Length	Format
ci_id	Character	36	
ci_name	Character	100	
ci_type	Character	20	
ci_expression	Character	2048	
ci_related_items	Character	100	

Note – calculated items can contain other calculated items and data items. While we capture the calculated item expressions, we do not recursively search the expressions for other calculated items or data items.

#### Outlib.segmap\_data

This dataset stores the segment maps associated with each task occurrence. This will allow for the reporting of segment map names in conjunction with associated data items.

Name	Type	Length	Format
runtask_id	Character	40	
ordinal_root	Numeric	8	
id	Character	36	
token	Character	32	
clientId	Character	32	
segmentMapOccur...	Character	36	
clientData	Character	1312	
idToSegmentId	Character	237	
externalCode	Character	6	
segmentMapName	Character	12	
parentExternalCode	Character	6	



## Upgrade considerations

The table structure may change with each new version of this utility. One of the following approaches is recommended:

- Rename the current SAS datasets with the \_old suffix, and let the first execution of this process recreate the tables. Afterwards, copy the data from \_old tables to the new tables
- Review the new table structure from the above document, and update your old SAS datasets to add the new columns to match the new structure.