Xpress Conversion Tool

1. Xpress Conversion Tool . . . 3
1.1 Overview . . . 4
1.2 How to Use the Xpress Conversion Tool . . . 7
1.3 Data Analysis . . . 22
1.4 Program Output . . . 23
1.5 SAP . . . 26
1.6 TWS for Distributed Data . . . 29
1.7 Generate Sample Data . . . 32
1.8 Keyword Mapping . . . 33
1.9 Log . . . 40
1.10 Job Control Language . . . 41
1.11 Support . . . 43

Xpress Conversion Tool

The Xpress Conversion Tool documentation contains the following pages:

- Overview
- How to Use the Xpress Conversion Tool
- Data Analysis
- Program Output
- SAP
- TWS for Distributed Data
- Generate Sample Data
- Keyword Mapping
- Log
- Job Control Language
- Support

The information on these pages also is located in the Xpress Conversion Tool.pdf.

Overview

- Overview
  - Supported Data Formats
  - Features
  - License
  - Linux
  - Program Specifications

Overview

The Xpress Conversion Tool (XCT) lets you convert information from your current job scheduler and transfer it to Universal Controller.

This document is a step-by-step guide on how to use the Xpress Conversion Tool. Stonebranch enables DIY transitions through tooling, making available a method to transfer data to Universal Automation Center (UAC).

Choose to convert all data at once or select subsets and slowly transfer your environment to UAC. Transfer your workload at your convenience.

Supported Data Formats

- AS/400 reports
- ASG-Zena
- Atos PC
- Automate 7
- Beta/42 reports
- CA-Autosys reports
- CA-7 reports
- CA Scheduler
- Comma-separated-value (.csv)
- Control-EM XML (distributed)
- Control-M XML (z/OS)
- Cron file / directory
- Cronacle (CPS)
- Custom / Home-grown XML
- D-Series
- JCL file / directory
- JobTrac reports
- Robot/Opal reports
- SAP Job definitions
- TWS for distributed
- TWS for z/OS
- UC4 XML reports
- Windows Scheduled tasks
- Windows XML

Features

- Translation of existing scheduler definitions
- Translation of existing file transfer definitions
- File mass-change facility

- Add JCL variables to UAC definitions
- Add Command Line variables to UAC definitions
- Agent variable/Agent cluster use
- Selective export
- Command scanning for variables
- Original definitions are archived as Notes

License

Enter license in the Global Options in the right pane and press [Enter].

The program must restart to activate the changes.

Linux

On Linux, a separate step sometimes is required to run the toolkit.

1. Verify that the default `OpenJDK` is installed. Enter the Terminal command: `java -version`

Ubuntu	OpenSuse

`Terminal Input:`
`java -version`
 	
`Terminal Output:`
`openjdk version "1.8.0_151"`
`OpenJDK Runtime Environment (build 1.8.0_151-8u151-b12-0ubuntu0.16.04.2-b12)`
`OpenJDK 64-Bit Server VM (build 25.151-b12, mixed mode)`
	`Terminal Input:`
	`java -version`
	
	`Terminal Output:`
	`openjdk version "1.8.0_131"`
	`OpenJDK Runtime Environment (IcedTea 3.4.0) (suse-11.1-x86_64)`
	`OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)`

2. Verify that the `openjfx package` is installed.

Ubuntu	OpenSuse

`Terminal Input:`
`sudo apt list --installed | grep -i openjfx`
 	
`Terminal Output:`
`libopenjfx-java/xenial,xenial,now 8u60-b27-4 all [installed,automatic]`
`libopenjfx-jni/xenial,now 8u60-b27-4 amd64 [installed,automatic]`
`openjfx/xenial,now 8u60-b27-4 amd64 [installed]`
	There is no default package from SUSE, but a package from the community is available.

3. Install the `openjfx` package.

Ubuntu	OpenSuse
`Terminal Input:` `sudo apt-get install openjfx`	Open Firefox in your SUSE VM and perform the "one click install": https://software.opensuse.org/package/java-1_8_0-openjfx?search_term=openjfx

4. Start the toolkit: `java -jar XpressConversionToolkit.jar`, or double-click the jar.

Program Specifications

Java 1.8	The Xpress Conversion Toolkit runs on Java. There is no installation for the toolkit itself. Double-click the jar to run the program. On Linux, it might be necessary to make the jar 'executable'. (File / Properties)

How to Use the Xpress Conversion Tool

- Overview
- Basic Workflow
- User Interface
    - Global Options
    - Specific Options
    - .csv Options
        - Usage
        - Variables
        - Expected Input Template
- Create Workflows Manually
    - Create Workflows from Tasks
- Sub-Selections
    - Export (sub) Selection
    - Delete (sub) Selection

Overview

The Xpress Conversion Toolkit (XCT) is a program that can help you migrate to the Universal Automation Controller (UAC).

It can produce different results depending on the chosen options. In general, it will produce all XML objects for the UAC environment.

The program uses different types of input files and commands to collect data from a system. Which input files are accepted can be found in the transitional listing of the corresponding scheduler product. For most input formats, a validity check will be performed.

During the conversion process, it is possible to enrich or change certain items like business servers, names, resources, and Email functionality.

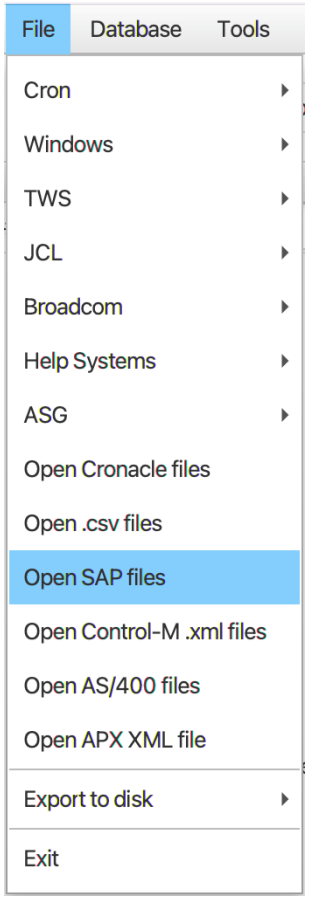
The following output format is available:

- Export to XML files (All)

Basic Workflow

Step 1	Set the Global options before reading the input file.
Step 2	

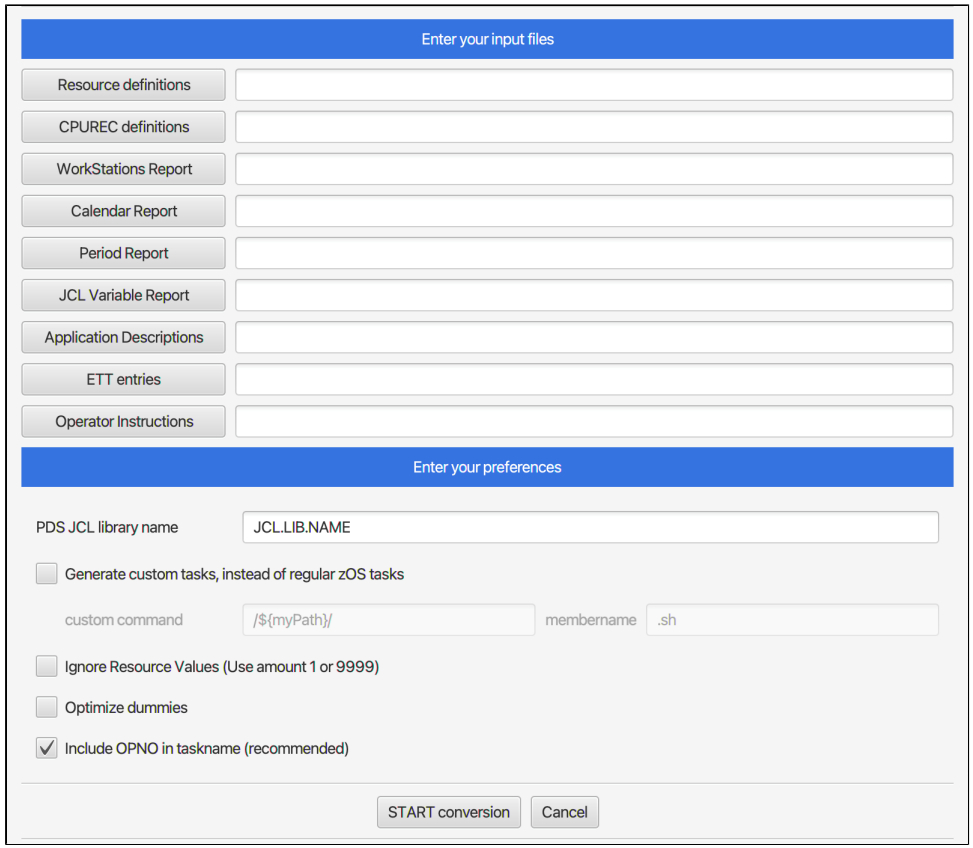
Open the input file.



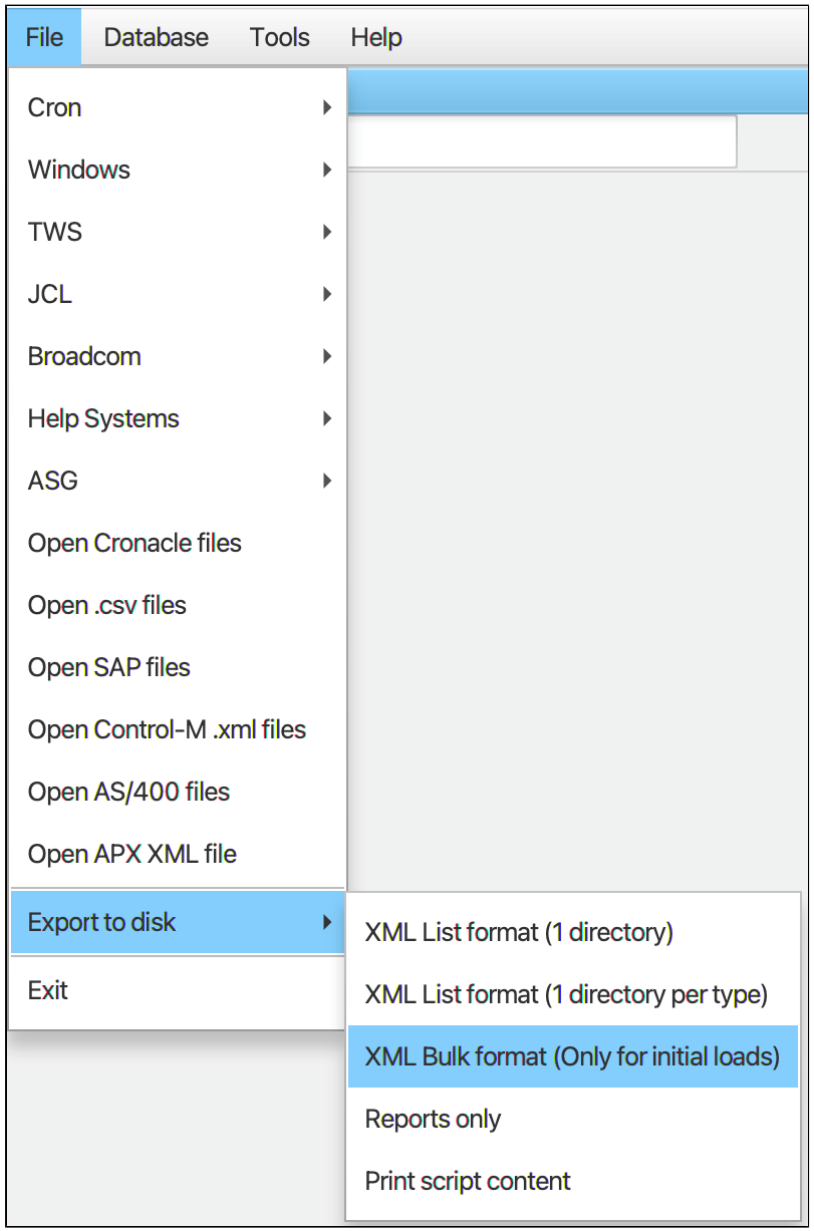
Additionally, scheduler-specific options can be set through a property file.

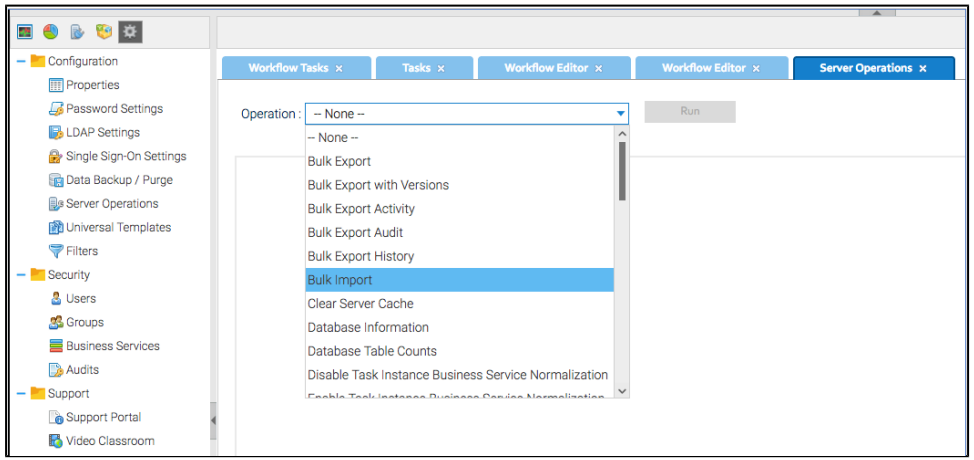
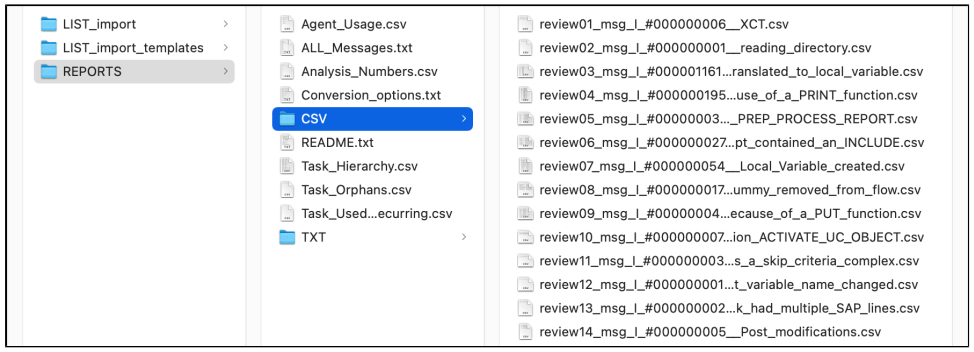
If you do not already have a property file, click the Example button to print one.

Alternatively, scheduler-specific options can be set through a dialog


Enter your input files
Resource definitions
CPUREC definitions
WorkStations Report
Calendar Report
Period Report
JCL Variable Report
Application Descriptions
ETT entries
Operator Instructions
Enter your preferences
PDS JCL library name
JCL.LIB.NAME
Generate custom tasks, instead of regular zOS tasks
custom command
/${myPath}/
membername
.sh
Ignore Resource Values (Use amount 1 or 9999)
Optimize dummies
Include OPNO in taskname (recommended)
START conversion
Cancel

Step 3 When the conversion has finished, write the data to disk.

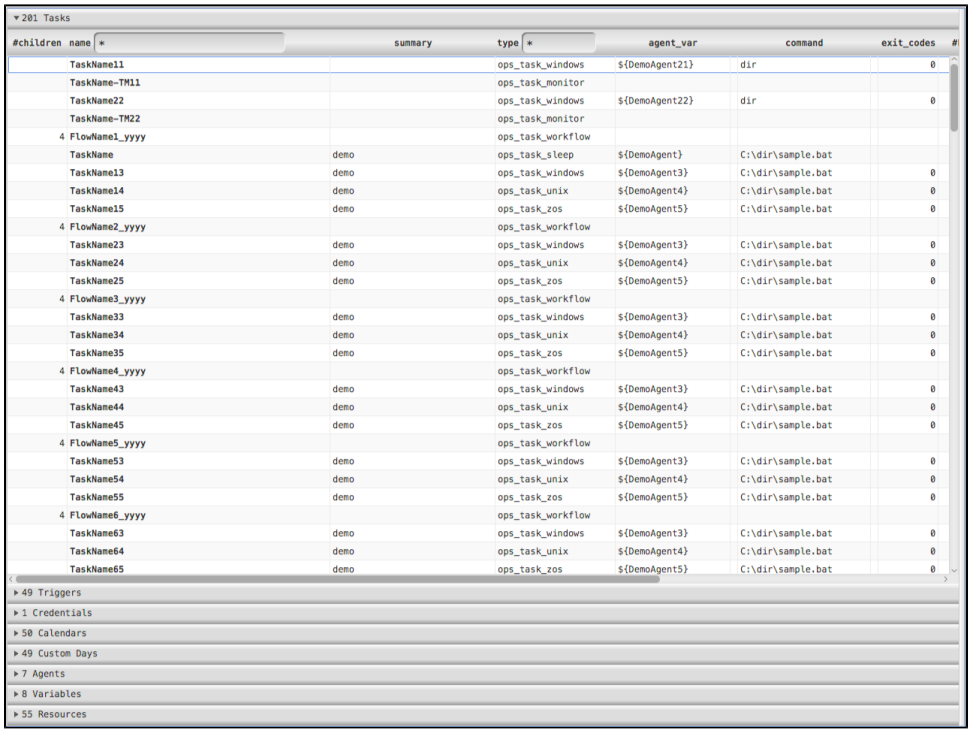
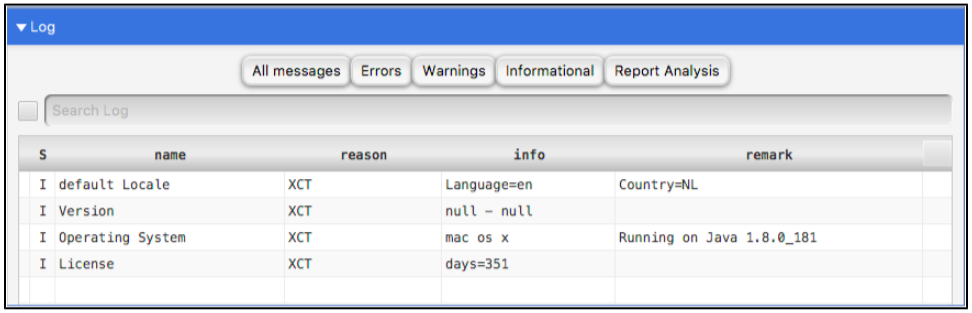


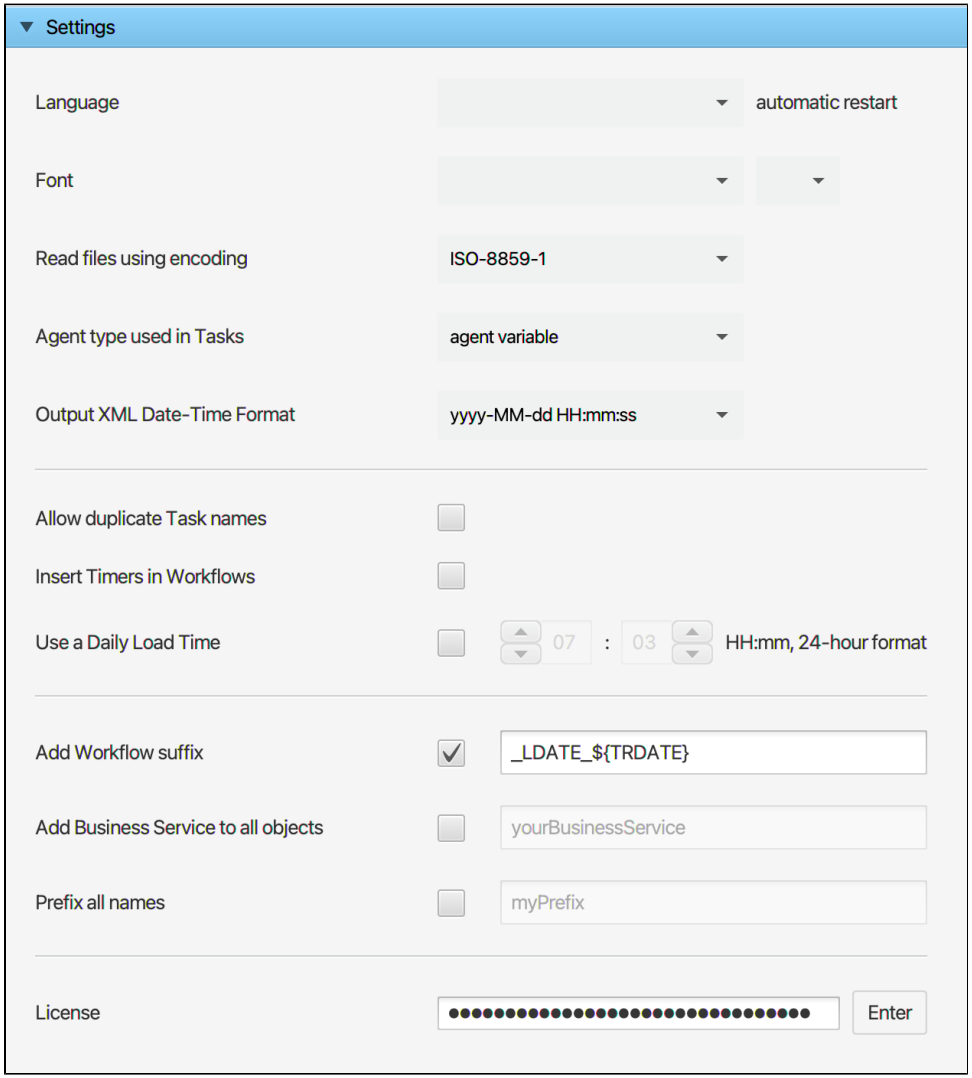
Step 4	Load the data into Universal Controller. 
Step 5	Review the reports created by the conversion tool. 

User Interface

The user interface is divided into four main sections:

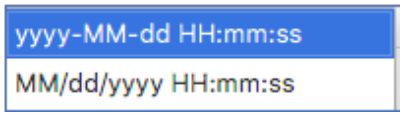
1	Menu bar with progress indicator and message field. 

2	Left pane displaying objects (Tasks, Triggers, Credentials, etc.) to be generated. 
3	Right pane displaying the log, a search-field, and the filter buttons. 

4	Right pane displaying the Global options. 

Global Options

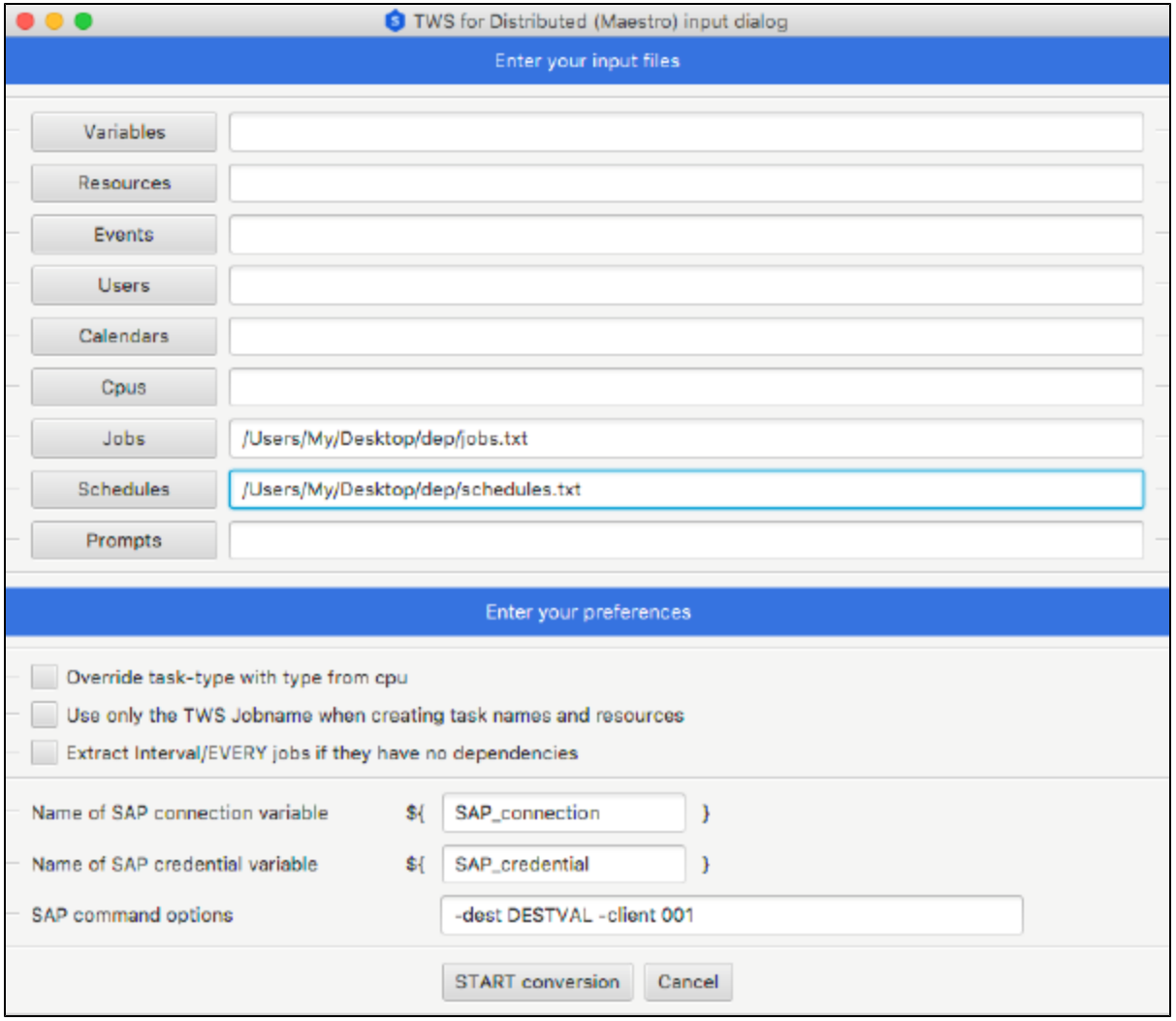
Global options are valid no matter which input format is used.

Option	Selected	Not selected
Language	The selected Locale	Default Local or English
Font	The selected font family	Default font or Menlo
File encoding	• ISO-8859-1 • UTF-8 • UTF-16 • IBM865	n/a
Agent type used in tasks	• Agent variable • Agent cluster • Agent cluster variable	n/a
Output XML date-time format		n/a
Allow duplicate task names	Creates for each definition a unique task	Re-using task names when identical
Insert Timers in workflows	Uses Timers to handle time-dependent tasks	Uses Time-to-Wait to handle time-dependent tasks
Use a Daily Load Time	Mimics a Daily Plan time. All triggers are created with this time. 	Uses the original trigger time in the definition
Add workflow suffix	Add the suffix to each workflow name	No action
Add Business Service to all objects	Create and assign a Business Service to all objects generated.	No action

Prefix all names	Prefixes all object names with this prefix	No action

Specific Options

Scheduler-specific options will be asked for when using certain data input formats.



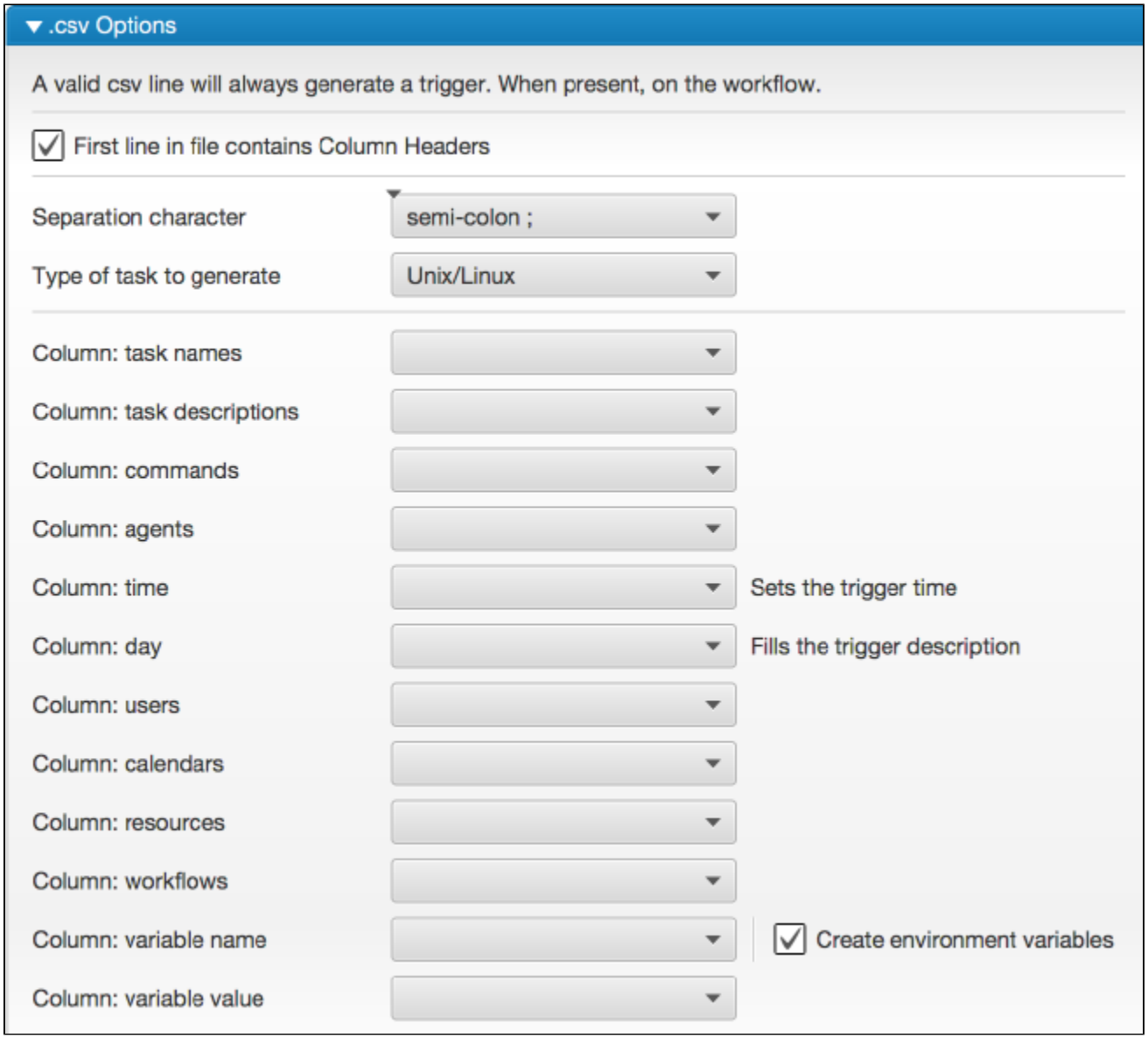
.csv Options

.csv options allow the user to choose from a set of drop-down boxes (combo boxes), whose columns should be interpreted. Every row is basically interpreted as a task.

The .csv can be used for the following purposes:

- Generate large amounts of (test-)data objects
- Parse columns, relate objects and POST them in UAC
- Delete objects from UAC.

Since the .csv content is free-format, not all UAC types can be extracted. It is meant to function as an easy way to get set-up quickly. Special triggers and complex dependencies must be added by the user.



The first three options must be set before reading a .csv file.

All column drop-down boxes will be filled with headers once the file has been read.

Changing a column selection will cause the program to re-interpret the data.

Usage

Step 1	Check the file for a header row, see which separation character is used (; or ,) and set the options.
Step 2	Set the desired task type (Unix, Linux, Windows, or z/OS). Note Note! The column DropDown-boxes are not filled yet!
Step 3	Read the file. This will fill the column drop-down boxes with headers so the user can choose.

The program tries to find the column with tasks by looking at the headers. Changing a column selection will cause the program to re-interpret the data.

The last entry in every column drop-down box represents the de-selecting of a column.

Variables

Variables exist in UAC as three different objects:

Global variables	Variables not connected to a specific task.
Local variables	Variables connected to a specific task.
Environment variables	Variables connected to a specific task.

Checking the box will cause the program to create task specific environment variables.

Unchecking the box will result in the task having a Local Variable defined. In addition, a Global Variable with the same name will be created.

Expected Input Template

The following worksheet is an example of how a .CSV could look like.

	A	B	C	D	E	F	G	H	I	J
1	taskname	taskdescription	command	user	agent	workflow	calendar	resource	varName	varValue
2	job4	text4	mkdir ${dir}	user1	hostB	flow2				
3	job1	text1	ls	user1	hostA	flow1	bankholiday	res1	PATH	/user/me/deze
4	job2	text2	ls -a	user1	hostA	flow1		rezouze	Email	jos@me.com
5	job3	text3	ls -la	user1	hostB	flow1		rezouze		
6	job4	text4	mkdir	user1	hostB	flow1				
7	job1	text1	ls	user1	hostA	flow2	bankholiday			
8	job2	text2	ls -a	user1	hostA	flow2				
9	job3	text3	ls -la	user1	hostB	flow2				
10	job1	text1	ls	user1	hostC	flow3	workdays			
11	job1	text1	ls	user1	hostC	flow3				
12	job1	text1	ls	user1	hostC	flow3				
13	job1	text1	ls	user1	hostC	flow3				
14	job5	text5	rm	user5	hostA		workdays	rezouze	WHODIR	/user/pointer

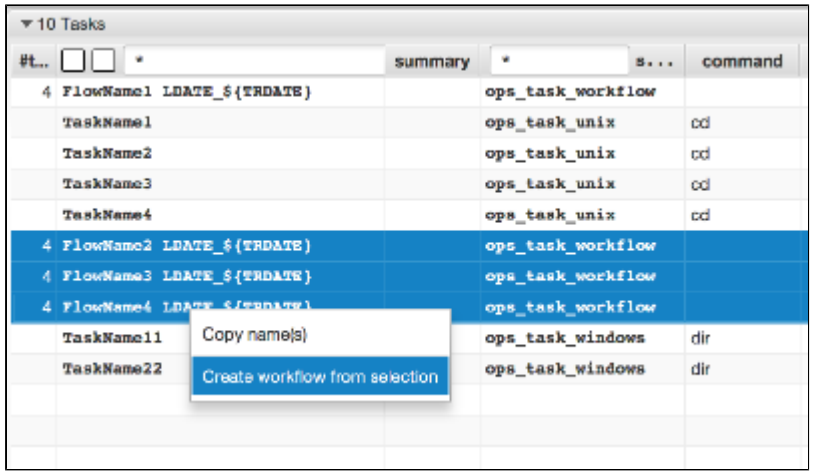
Note

The columns can be in any order. Use the combo-boxes to choose a column.

Create Workflows Manually

Most data types will generate workflows automatically, but when the input data does not contain workflow information (such as Cron or Windows), the following procedure could be used.

Create Workflows from Tasks

Step 1	Right-click on the Tasks pane. 
Step 2	In the Create Workflow dialog that displays, enter a name for the new workflow. 

Sub-Selections

The program can export sub-selections of the parsed data. Selections can be made in the left pane in the tasks section.

Input fields can be found in:

- Name column
- sys_class_name column (task type)

Note

The tasks section will be visible only after a file has been processed.

Export (sub) Selection

Enter a sub-selection by typing characters in the input field and pressing [enter]. The corresponding Triggers, Business Services, etc. will automatically update. Basically, only what is displayed goes to the exporter.

Note

The program does not allow incomplete workflow exports, even when they are partially displayed.

Related tasks do not necessarily occur in sequence in input files. Therefore, an extra checkbox is provided to display all workflows and its children tasks in a structured way.

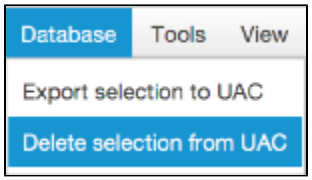
Note

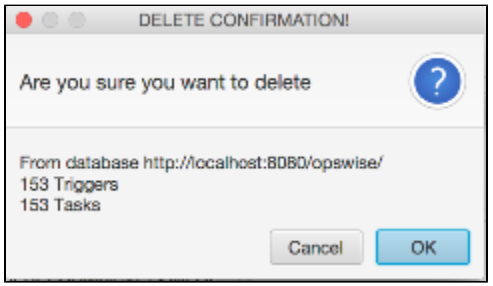
This option is set to UNCHECKED when an export is started.

Delete (sub) Selection

Enter a sub-selection by typing characters in the input field and pressing [enter]. The corresponding Triggers, Business Services, etc. will be updated automatically.

Select **Delete Selection from UAC** in Database to end delete requests to UAC.





Data Analysis

- Overview
- Analysis Walk-Through

Overview

The XCT can provide a data analysis of the input data. It will show numbers/totals pertaining to the input as well as the output.

It enables you to assess the quality and quantity of the parsed and generated objects. This can be done in three ways:

1. A visual inspection of the data by using the XCT user interface.
2. Generating message reports on disk.
3. Using the Report Analysis button in the log.

Analysis Walk-Through

Step	Action	Help
1	Run a conversion using file input	Basic Workflow
2	Go to the Log Pane and press **Report Analysis**	Log This will present you with a condensed view of all headers, counters, and totals. The view is also available as a report on disk called "Analysis_Numbers.csv".
3	Go to File Menu and press **Print Log Analysis**.	This will create a number of reports: • Analysis_Numbers.csv • Informational reports • Warning reports All messages will be grouped by type to allow for further automated/manual processing.

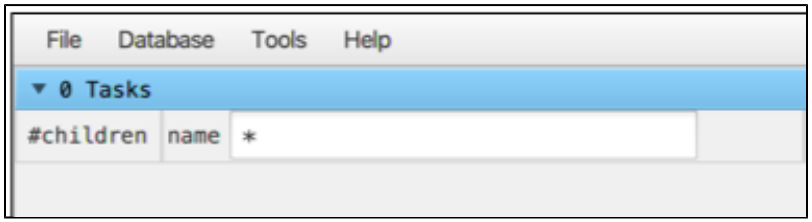
Program Output

- Program Output
- Import Generated XML

Program Output

The conversion program can export its results in a number of ways:

- All objects (XML, Templates and Reports)
- A selection of objects using wild-cards



When exporting to XML, there are three possibilities:

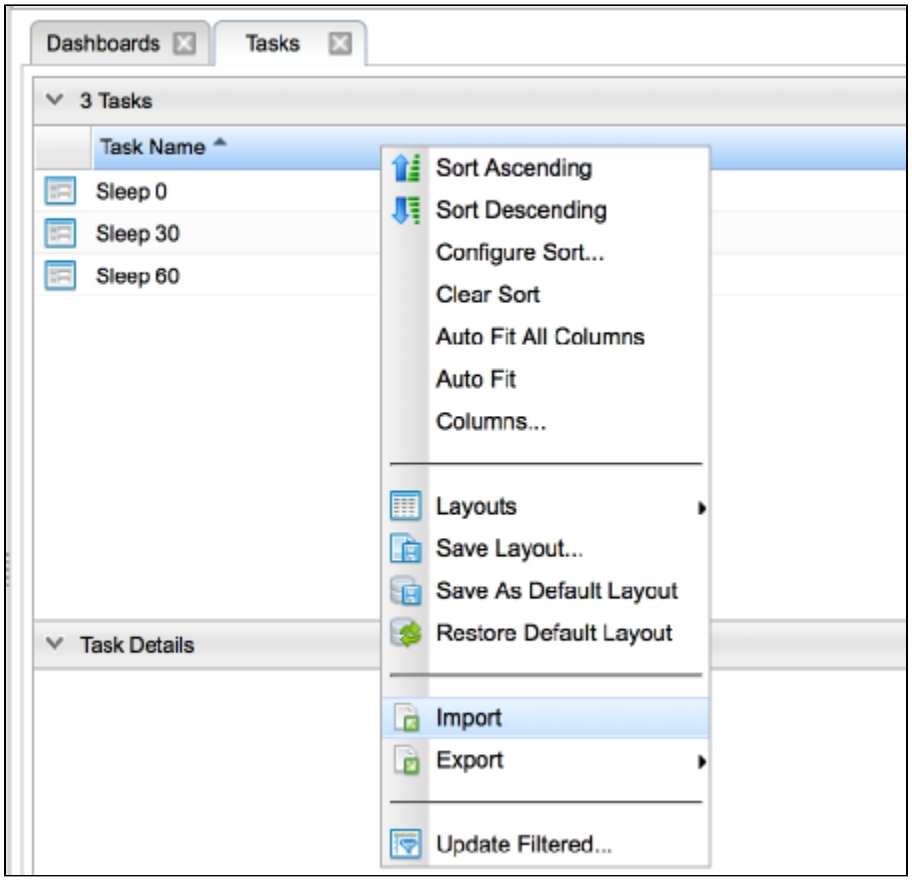
Format	Output	File Structure	Intended For
LIST	1 directory	1 file per object	Phased conversion
LIST	1 directory per object type	1 file per object	Phased conversion
BULK	1 directory	1 file per object-type	Initial load of the database

Additionally, you can print only the reports, or print the content of all the scripts that were generated during the conversion.

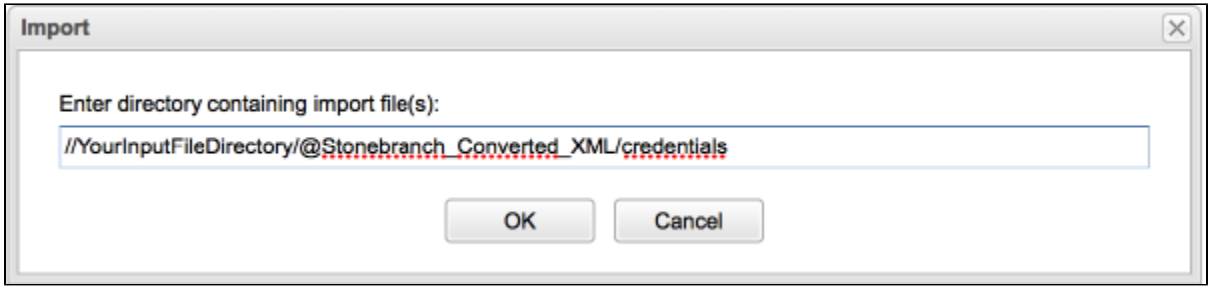
Import Generated XML

The following example describes the loading of a credential, but the same procedure is valid for every UAC object type.

You can import XML files in UAC by right-clicking on one of the column headers in almost every screen.



A pop-up will appear asking you to enter the full path name of the converted output.



Paste the path string in the text field and then click OK.

The UAC's console will let you know whether or not the import succeeded.

SAP

- SAP
- SAP Job Definitions
  - Example
  - EVENTPARM

SAP

The conversion program supports:

- SAP Job definitions

These reports can be generated through a Stonebranch Utility.

Read the directory containing the SAP Job definition files. The corresponding values in the CALENDARID keyword will hook up to the calendars and their custom days. The conversion program recognizes different SAP calendar records by their lengths. Records that do not match are discarded.

SAP Job Definitions

Job definition files should be in a separate directory when using the conversion program.

Every file will be converted into a workflow, instead of a regular task, to enable the adding of extra functionality, such as like restart, dependencies, skip criteria, and run criteria.

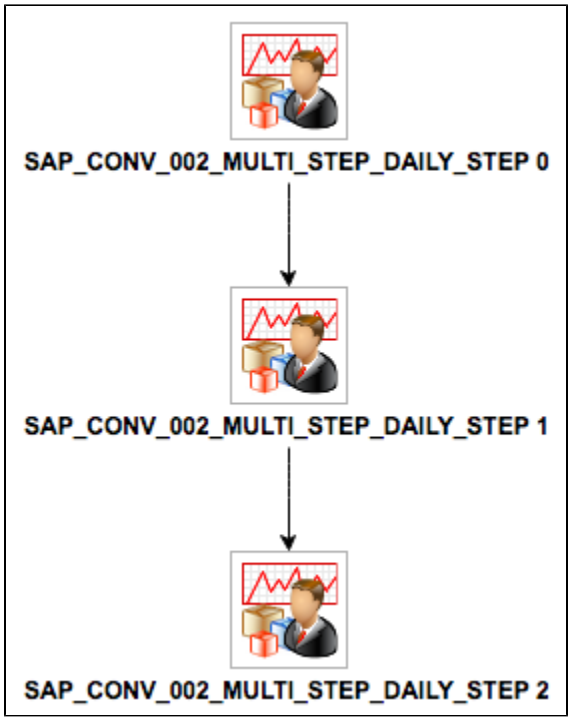
Every ABAP step in the file then is converted into a separate task and automatically receives a dependency to its predecessor ABAP step.

Example

The following job definition file:

```
/* Job Header statement. */
JOBNAME = "SAP_CONV_002_MULTI_STEP_DAILY"


/* ABAP Step statement. */
ABAP_STEP = "STEP 0"
   ABAP_PROGRAM_NAME  = "ZNBU_ABAP_STEP_001"



/* ABAP Step statement. */
ABAP_STEP                   = "STEP 1"
   ABAP_PROGRAM_NAME = "ZNBU_ABAP_STEP_002"



/* ABAP Step statement. */
ABAP_STEP                   = "STEP 2"
   ABAP_PROGRAM_NAME = "ZNBU_ABAP_STEP_001"
```

Converts to the following workflow (SAP_CONV_002_MULTI_STEP_DAILY):



EVENTPARM

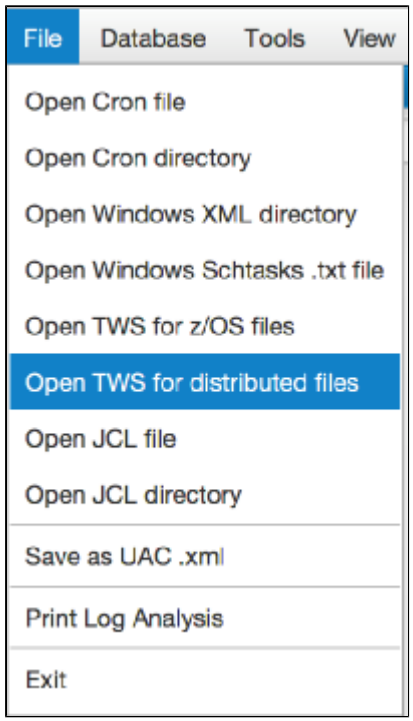
If an EVENTPARM is present, a Task Monitor will be created pointing to the task in the EVENTPARM value.

TWS for Distributed Data

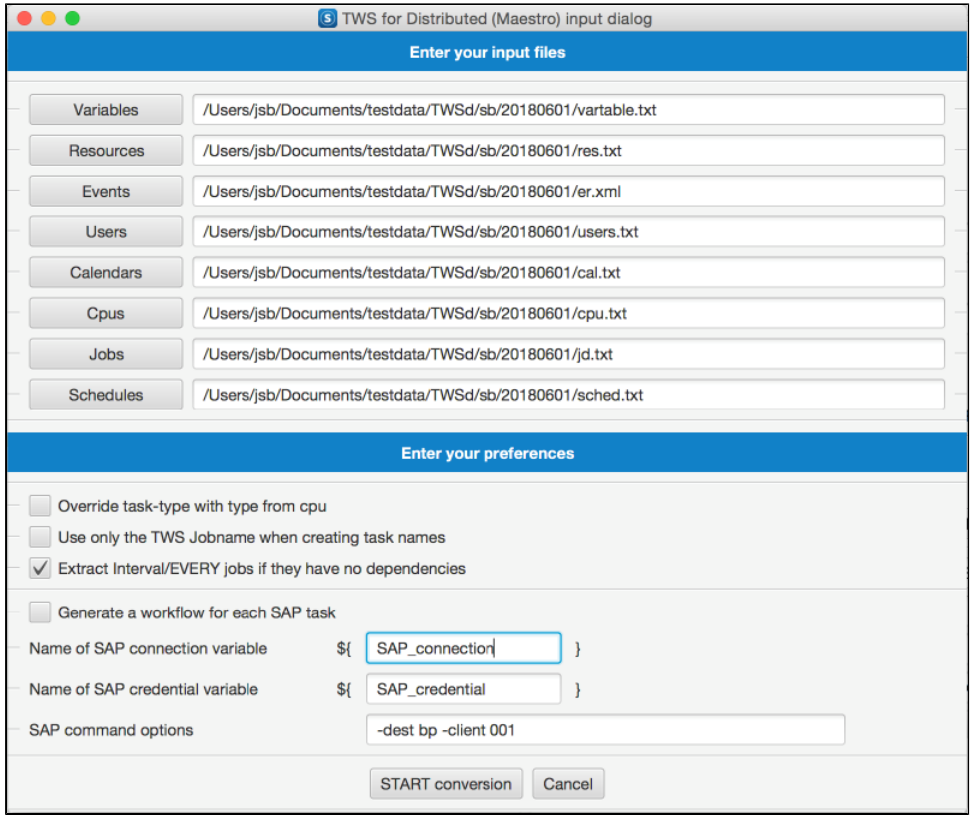
- TWS for Distributed Data

TWS for Distributed Data

When transitioning TWS for distributed data, use the following workflow.

Step 1	Set global options.
Step 2	Open the TWS-d files input dialog. 

The input dialog will display.



Step 3

Set the TWS-d input file locations.

Use the buttons or text fields on the input dialog to enter the file locations. The file locations that you enter will be saved over sessions.

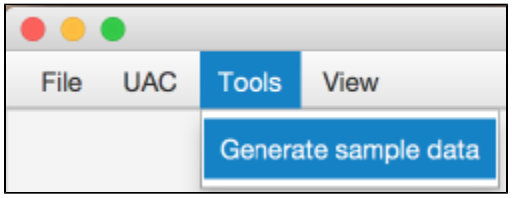
Step 4	Set TWS-d preferences: 1. Override task type with type from cpu. Normally, a task type is determined by the job definition. In some cases, the determined type can be overridden by the type of operating system it runs on. **Use only on direction of Stonebranch engineers.** 2. Use only the jobname when creating task names. Normally, a task name is created from the cpu name and jobname: `CPU#jobname`. Checking the box will result in task names without `CPU#`. 3. Extract EVERY jobs. Interval jobs can be taken out of a workflow to enable interval processing. This will give the extracted job it's own interval trigger. This can result in empty workflows if all jobs are interval jobs. 4. Generate a workflow for each SAP job. This option can be set to prepare the converted TWS-d data for merging with SAP transitions. **Use only on direction of Stonebranch engineers.**
Step 5	Set SAP preferences. 1. Name of SAP connection variable The name of SAP connection variable can be set for each transition. 2. Name of SAP credential variable The name of SAP credential variable can be set for each transition. 3. SAP command options Free format text field to enter specific SAP command options to be included in the generated SAP task.
Step 6	Start conversion. After starting the conversion, all files will be processed in sequence. A progress bar at the top right corner informs you of the progress.
Step 7	When conversion has finished...

Generate Sample Data

- Generate Sample Data

Generate Sample Data

You can generate sample data in the conversion program for testing the UAC connection, try different program options, or just get used to the user interface.



The result will be similar to the following.

10 Tasks

name	agent_var	sys_class_name	localVars
FlowName1 LDATE_${TRDATE}		ops_task_workflow	0
TaskName1	${AGT_VAR_DemoAgent1}	ops_task_unix	1
TaskName2	${AGT_VAR_DemoAgent2}	ops_task_unix	1
TaskName3	${AGT_VAR_DemoAgent3}	ops_task_unix	1
TaskName4	${AGT_VAR_DemoAgent4}	ops_task_unix	1
FlowName2 LDATE_${TRDATE}		ops_task_workflow	0
FlowName3 LDATE_${TRDATE}		ops_task_workflow	0
FlowName4 LDATE_${TRDATE}		ops_task_workflow	0
TaskName11	${AGT_VAR_DemoAgent21}	ops_task_windows	0
TaskName22	${AGT_VAR_DemoAgent22}	ops_task_windows	0

You could change some global options and press **Generate sample data** again to see the result of the changed options.

Use the UAC POST Menu to export all objects to UAC.

When using the **Save as XML** menu option, a dialog will display asking for a directory to save the XML files in.

Keyword Mapping

- Automic UC4
- Autosys
- CA-7
- Control-EM Distributed
- Cron
- TWS Distributed
- TWS z/OS
- Windows Scheduled Task
- Windows XML

Automic UC4

Keyword	Results In
CALE	Calendars, Custom days
CALL	Actions
HOSTG	Agent clusters
JOBF	Task
JOBG / JOBP	workflow
JOBS_R3	Task
JOBS_MVS	Task
JOBS_ZOS	Task
JOBS_NSK	Task
JOBS_PS	Task
JOBS_UNIX	Task
JOBS_SQL	Task
JOBS_WINDOWS	Task
JOBS_CIT	Task
JOBS_OS400	Task
JOBS_GENERIc	Task
SCRI	Scripts
USRG / USER / Login	credentials

VARA	Variables

Autosys

Keyword	Results In
insert_job	Creates a default Unix, Windows, File Monitor, or Workflow task.
box_name	Creates a task parent or new workflow.
condition	Creates a dependency.
box_success	Sets a message.
command	Creates a command.
machine	Creates an agent for the task.
owner / #owner	Creates a credential.
timezone	Translates to UAC time zone.
description	Creates a description.
date_conditions	Creates a default trigger.
days_of_week	Sets the trigger date details, updates the trigger description.
run_calendar	Creates a trigger or calendar (see Autosys options).
exclude_calendar	Creates a calendar.
start_times	Creates trigger(s).
start_mins	Forms a repeating trigger together with run_window.
run_window	Restricts time on a trigger.
watch_file	Sets the file to watch for.
watch_file_min_size	Sets file monitor type.
std_in_file	Scans value for variables.
std_out_file	Scans value for variables, sets redirect command.
std_err_file	Scan value for variables, sets redirect command.
max_run_alarm	Set late finish on the task.
min_run_alarm	Sets early finish on the task.
term_run_time	Sets late finish.
box_terminator	Sets a message.

job_terminator	Sets a message.
n_retrys	Sets a retry maximum.
chk_files	Creates an extra file monitor for the task, creates a dependency.
auto_hold	Set start held = true

When an undefined keyword is encountered, a message will be set.

CA-7

CA7 Field	UAC
Job Name	Task name
JCL ID	Variable: JCLID_###
JCL Member	JCL Location of the job
User ID	Business Service
Main ID	Agent variable: AGT_xxxx
Last-Run Date/Time	Message if date is > 2
Override Of JCL Required	Y, add manual job preceding this job
JOB MARKED AS MAINT ONLY	N, all DSN= statements are file dependencies. Y, none of the DSN= statements are file dependencies *Requirements and Network Connections* section
MANUAL VERIFICATION REQD	Y, add manual job preceding this job
JOB SET FOR HOLD IN REQQ	Y, put the job on hold
AUTO-GENERATION OF 7 RMS	N, set parameter: OPSDSDEL / NO
JOB SET FOR EXEC ON MAIN	N, set Run/Skip for the job to always skip
OWNER=	Credential
JCLLIB=	Set Jcl_location
ARFSET=	Message
CLASS=	Virtual resource
PRTY=	Set Res_priority
DONT SCHEDULE BEFORE	Message
DONT SCHEDULE AFTER	Message
Prose	Note

Schedules	Triggers
SUBTM	Insert Timers
NWK	Message
DSN=	Workflow + trigger tree
DSN=	File Monitor (trigger)
DSN=	Composite File Monitor trigger
USR=	Manual task
JOB= requirement	Mutually Exclusive when paired
JOB= requirement	Message when containing a ?

Control-EM Distributed

Control-EM	UAC
JOB	• Task (Unix/Linux/Windows, Timer) • File Monitors • File Transfers • Universal Task
PARENT_NAME	Workflow Task
PARENT_FOLDER	
TABLE_NAME	
PARENT_NAME	
PARENT_NAME	
GROUP	
APPLICATION	
NODEID	Agents / Agent Cluster
CYCLIC	Time Trigger
QUANTITATIVE	Resources
External dependencies	Task Monitor
AUTOEDIT2	Variables
VARIABLE	Variables

Cron

Keyword	Results In
minute	Minute field of the Cron trigger
hour	Hour field of the Cron trigger
Day of month	field of the Cron trigger
Month of year	field of the Cron trigger
Day of week	field of the Cron trigger
Shell command	Command field of the Cron task

TWS Distributed

TWS-d	UAC
Job	Task
Job streams / Schedules	Workflow Task
Workstations	Agents / Agent Cluster
Calendars	Calendar and custom days
Prompts	Manual Task
Run cycles	Time Trigger
Resources	Resources
External dependencies	Task Monitor
Parameter	Variables
Variable tables	Variables
Users	Credentials
SAP jobs	SAP task
Opens	File Monitor
Event Rules	Task / Web-Api / system action

TWS z/OS

TWS-z	UAC

Application Description	Workflow task
Job/operation	Task
Workstations	Agents
Calendars	Calendar and custom days
JCL prep workstation	Manual Task
Run cycles	Time Triggers
Resources	Resources
External dependencies	Task Monitor
Variable tables	Variables
Users	Credentials
SAP jobs	SAP task
NV* workstations	Application Control Task / Web-Api
Events	Task / Web-Api / System Action

Windows Scheduled Task

Keyword	Results In
HostName	Task
Author	Business Service
Task To Run	Task command
Comment	Task summary
Idle Time	Message
Run As User	credential
Stop Task If Runs X Hours and X Mins	Set Late finish
Schedule Type	Trigger, set start time
Start Time	Trigger description, set start time
Start Date	Trigger description
End Date	Trigger description
Days	Set custom days
Months	Trigger description

Repeat: Every	Set Repeat Interval
Repeat: Until: Time:	Message
Repeat: Until: Duration	Message
Repeat: Stop If Still Running	resource

Windows XML

Keyword	Results In
Author	credentials
Description	Task description
URI	Task name
CalendarTrigger	Trigger
TimeTrigger	Trigger
Repetition	Trigger description
WeeksInterval	Trigger description
DaysOfWeek	Trigger description, set custom day
Command	Task command
Arguments	Task parameters

Log

Log

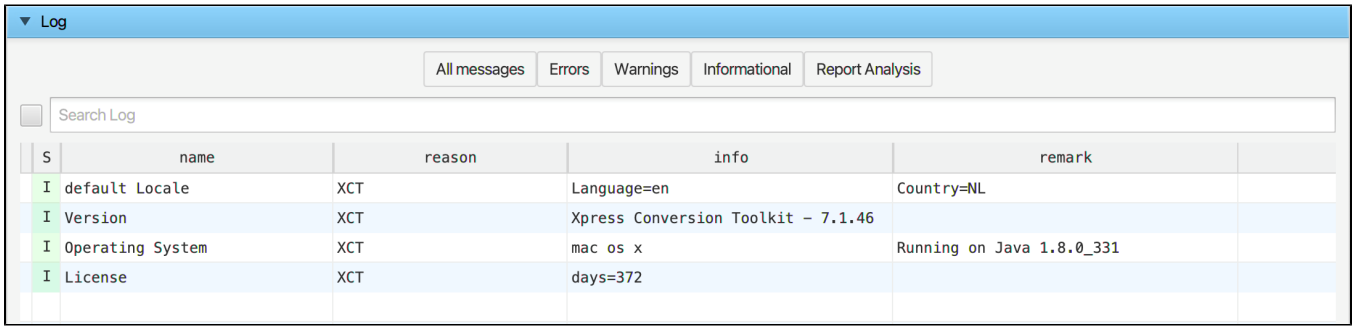
The log is divided into 6 columns, which can be sorted like any other listview in the program.

- Date time stamp
- Message severity
- name column: displays the object that generated the message
- reason column: displays the reason the message occurred
- info column: displays extra information for the message
- remark column: displays extra information for the message

An additional search field and buttons are provided to find specific messages.

When the Find checkbox is selected, all search hits will be highlighted.

When the Find checkbox is not selected, only search hits will be shown.



The complete log will be exported to the */XCT_timeStamp/@Report_* directory when the XML output is created.

Job Control Language

- Overview
- How to Perform a JCL Transition
- Output

Overview

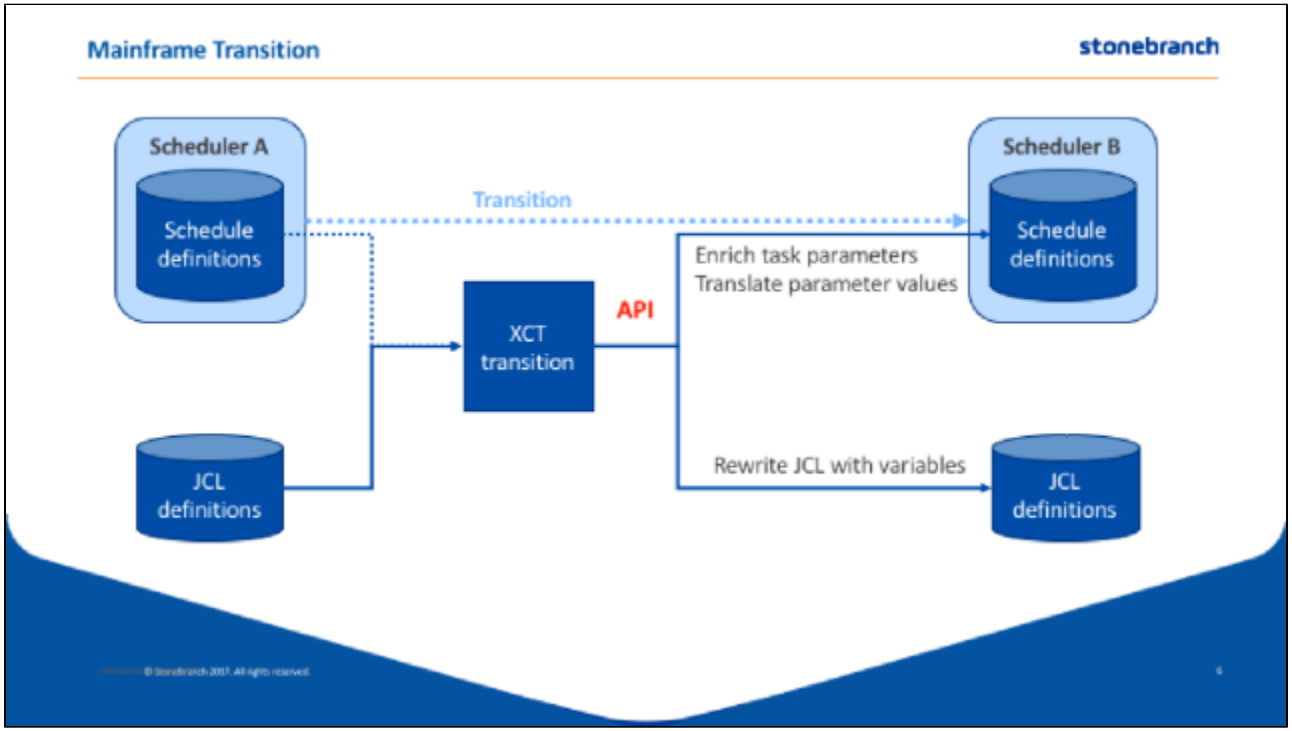
Mainframe scheduler transitions are, by nature, large and complex, but more importantly, there is always a second component that must be converted: JCL, Job Control Language.

This language uses many variables, some of which are provided by the scheduler, operating system, or otherwise. When doing transitions, this information must somehow be made available to UAC.

Since JCL libraries tend to be large, an automated solution is available in the Xpress Conversion Toolkit.

Each JCL member must be:

1. Translated to a new member that UAC can handle.
2. Parameters must be added to existing or new z/OS tasks.

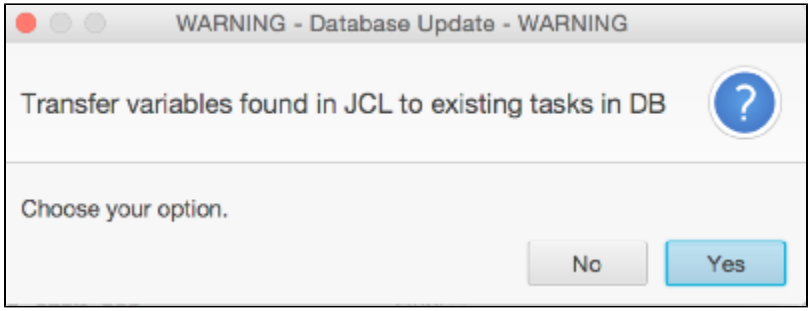


How to Perform a JCL Transition

1. Make sure to review all available options in the Options pane. If you plan to use the API, make sure you enter a valid user/password combination.
2. File Menu, *Open JCL directory* or *Open JCL file*.

After processing the input data, you will be given a choice:

- Update the database with parameters/variables found in the JCL (Yes)
- Do not update (No)



Output

The Xpress Conversion Toolkit will write all jobs to new a new directory called *@output*, created in the input directory.

The content of the JCL members will be translated to UAC JCL.

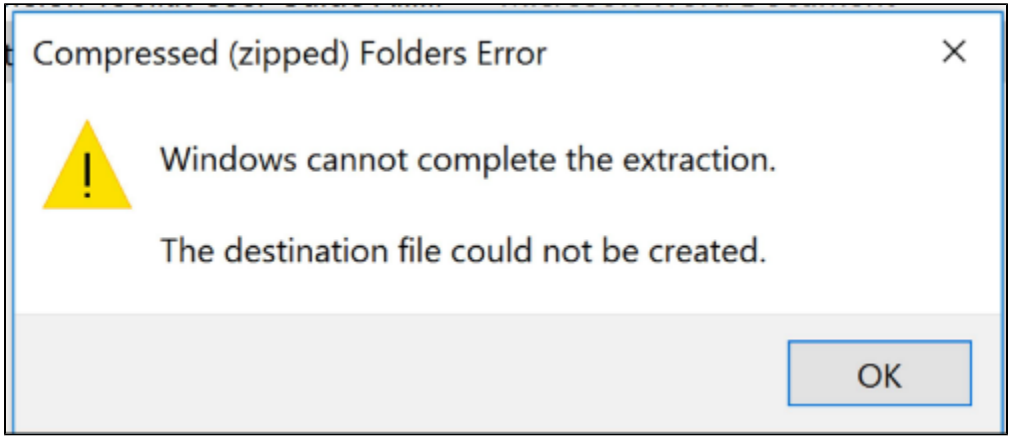
Scheduler-specific variables will be translated, as much as possible, and created as parameters on the z/OS task via the web API.

Support

Support

Contact: support@stonebranch.com

If the following message appears, the toolkit has been stored in a compressed windows folder.



Please copy the toolkit a new (not compressed) folder.