



stonebranch

Xpress Conversion Tool

© 2020 by Stonebranch, Inc. All Rights Reserved.

- 1. Xpress Conversion Tool 3
 - 1.1 Overview 4
 - 1.2 How to Use the Xpress Conversion Tool 7
 - 1.3 Program Output 25
 - 1.4 Database Tools 28
 - 1.5 Command Line Interface 39
 - 1.6 SAP 41
 - 1.7 TWS for Distributed Data 44
 - 1.8 Generate Sample Data 48
 - 1.9 Keyword Mapping 49
 - 1.10 Log 56
 - 1.11 Job Control Language 57
 - 1.12 Support 59

Xpress Conversion Tool

The Xpress Conversion Tool documentation contains the following pages:

- [Overview](#)
- [How to Use the Xpress Conversion Tool](#)
- [Program Output](#)
- [Database Tools](#)
- [Command Line Interface](#)
- [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
- Dollar Universe reports
- JCL file / directory
- JobTrac reports
- Robot/Opal reports
- SAP calendars
- SAP Job definitions
- TWS for distributed
- TWS for z/OS
- UC4 XML reports
- Windows Scheduled tasks
- Windows XML

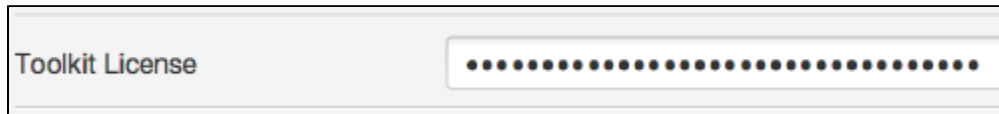
Features

- Translation of existing scheduler definitions
- Translation of existing file transfer definitions

- Enrichment of the converted data
- UAC database tools
- 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
- User preferences

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

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

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

3. Install the `openjfx` package.

Ubuntu	OpenSuse
Terminal Input: <code>sudo apt-get install openjfx</code>	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](#)
 - [Cron Options](#)
 - [Cron File Type](#)
 - [Cron Task Names](#)
 - [.csv Options](#)
 - [Usage](#)
 - [Variables](#)
 - [Expected Input Template](#)
- [Keyboard shortcuts](#)
- [Create Workflows Manually](#)
 - [Create Workflows from Tasks](#)
 - [Create Workflows from Triggers](#)
- [Sub-Selections](#)
 - [Export \(sub\) Selection](#)
 - [Delete \(sub\) Selection](#)
- [Enrich XCT Data](#)
 - [Context Menu Tasks](#)
- [Data Analysis](#)
 - [Analysis Walk-Through](#)

Overview

The Xpress Conversion Toolkit (XCT) is a program that can help you:

- Migrate to the Universal Automation Controller (UAC).
- Perform actions on the UAC database.

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.

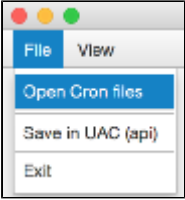
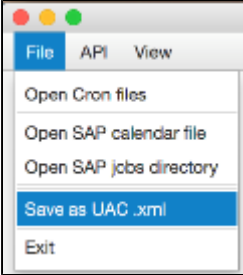
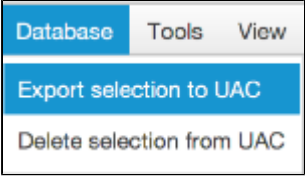
During the conversion process, it is possible to enrich or change certain items like business servers, names, resources, and Email functionality.

Depending on the input format, two output formats are available:

- Export to UAC using the RESTful API (Cron, Windows, and SAP calendars)
- Export to XML files (All).


Basic Workflow

Step 1	Set all desired options (Global and Specific) before reading the input file.
---------------	--

Step 2	<p>Open a file.</p> 
Step 3	<p>Export the data.</p>  <p>OR</p>  <p>The results of the export will be visible in the Log Pane.</p>

User Interface

The user interface is divided into four main sections:

1	<p>Menu bar with progress indicator and message field.</p> 
---	---

2 Left pane displaying objects (Tasks, Triggers, Credentials, etc.) to be generated.

<input type="checkbox"/>	*	name	summary	*	sys_class_name	number_of_tasks	agent_var	localVars	command	orphan
		flow2 LDATE_{TRDATE}			ops_task_workflow	4		0		true
		job4	text4		ops_task_unix		\$(AGT_VAR_hostB)	1	mkdir \$(dir)	false
		flow1 LDATE_{TRDATE}			ops_task_workflow	4		0		true
		job1	text1		ops_task_unix		\$(AGT_VAR_hostA)	0	ls	false
		job2	text2		ops_task_unix		\$(AGT_VAR_hostA)	0	ls -a	false
		job3	text3		ops_task_unix		\$(AGT_VAR_hostB)	0	ls -la	false
		flow3 LDATE_{TRDATE}			ops_task_workflow	4		0		true
		job5	text5		ops_task_unix		\$(AGT_VAR_hostA)	0	rm	true

< >

- ▶ 3 Credentials
- ▶ 3 Calendars
- ▶ 4 Agents
- ▶ 5 Variables
- ▶ 2 Resources

3 Right pane displaying the log, a search-field, and the filter buttons.

▼ Log

Find All messages Errors Warnings Informational Report Analysis

S	name	reason	info
I	Xpress Conversion Toolkit - 6.7.0		
I	Running on Java 1.8.0_231	mac os x	
I	Licensed days 306 days		

4 Right pane displaying the Global options.

▼ File Options

read files using encoding		ISO-8859-1	
Match case in list views and searches	<input type="checkbox"/>		
Generate agent clusters for tasks	<input type="checkbox"/>		
Allow duplicate task names	<input type="checkbox"/>		
Draw straight edges in workflows	<input checked="" type="checkbox"/>		
Insert Timers in workflows	<input checked="" type="checkbox"/>		
Remove objects not used	<input type="checkbox"/>		
Use a Daily Load Time	<input checked="" type="checkbox"/>	06 : 30	HH:mm, 24-hour format
Display system counters	<input type="checkbox"/>	01 , 01	Start-position, Length

Name Options

Add workflow suffix	<input checked="" type="checkbox"/>	<input type="text" value="-WF"/>	
Add Business Service to all objects	<input type="checkbox"/>	<input type="text" value="BusinessServiceName"/>	
Prefix all names	<input type="checkbox"/>	<input type="text" value="myPrefix"/>	
Change task name	<input type="checkbox"/>	position 1 to \${env}	

Task Monitor Options

Generate only when present in input	<input checked="" type="checkbox"/>		
Task monitor relative scope	<input type="checkbox"/>	From -32:00 To +01:00	

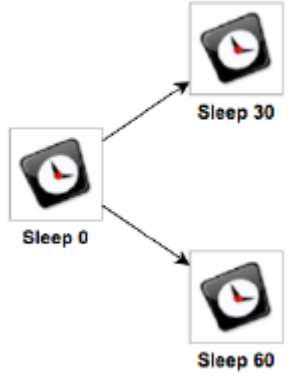
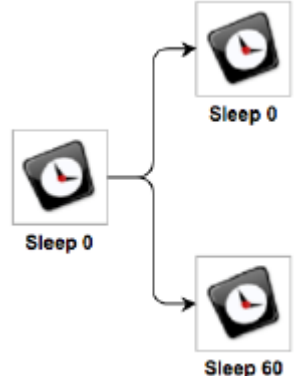

Output Options

Output XML files to one directory	<input checked="" type="checkbox"/>		
Use User-defined output directory	<input type="checkbox"/>		
Output XML DateTime format		<input type="text" value="yyyy-MM-dd HH:mm:ss"/>	

Xpress Conversion Toolkit License

Global Options

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

Option	Selected	Not Selected
File encoding	<ul style="list-style-type: none"> • ISO-8859-1 • UTF-8 • UTF-16 • IBM865 	
Match Case	<ul style="list-style-type: none"> • Case-sensitive search • Case-sensitive display 	<ul style="list-style-type: none"> • Ignore Case when searching. • Ignore Case in display.
Generate Agent Clusters	Generate an Agent Cluster for each task.	Generate an Agent Variable for each task.
Allow Duplicate Task names	Allow Duplicate Task names <ul style="list-style-type: none"> • JOB1 • JOB1 	Do not Allow Duplicate Task names <ul style="list-style-type: none"> • JOB1 • JOB1_dup1
Draw straight edges in workflows		
Insert Timers in workflows	When a time dependent job is encountered in the input, a sleep timer is inserted before the task.	When a time dependent job is encountered in the input, the time to wait option is set on the task. Note  The time is set on the task, not the vertex!
Remove objects not used	After conversion, an extra check is performed on referenced objects. When not used, they will be removed from the converted data.	Keep every object found in the input data.

Add workflow suffix	Adds the suffix: LDATE_{TRDATE} (trigger date) Allows for targeting specific run days when using Task Monitors	No suffix added
Use Daily Load Time (TWS specific)	Allows for simulating a TWS daily plan. <ul style="list-style-type: none"> The trigger time will be set to this daily load time. The workflow will receive an extra timer stopping the workflow from running before the designated time. 	Any time trigger will receive the time found in the input.
Display system counters	<ul style="list-style-type: none"> Creates a list with counters for each system found. This String can be used in conversions to dispatch different agent variables or run-time directories for each system. 	<ul style="list-style-type: none"> No list No differentiation applied.
Add Business Service	Adds a Business Service to each object generated.	No Business Service added.
Prefix all names	Adds a custom prefix to each object name.	No custom prefix.
Change task name	Allows you to substitute 1 character in the name with a string. <ul style="list-style-type: none"> Old name = PJOB1 New name = \${env}JOB1 	No substitution.
Generate only when present in input	Generate Task Monitors only when the task-to-monitor is in the input data. If the task-to-monitor is not found a message will be generated.	Always generate a Task Monitor for each external dependency.
Check database for existing task monitors	Will check the UAC DB if the Task Monitor being generated is already in the DB. When found: <ul style="list-style-type: none"> The DB sysid will be adopted by the workflow. The XML will not be written during export to protect the DB version from overwriting. 	No check will be performed during the conversion.
Task Monitor relative scope	Can set a default scope for all Task monitors generated.	No default scope used.
Output XML files to one directory	Writes all XML files to the directory where your input files are. @XCT-dateTimeStamp. Allows for one-click loading the database.	Create a separate directory for each XML type allowing for more control when loading XML files into the database.
Output XML Date Time format	Sets a custom date time stamp in the XML output.	–

Specific Options

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

TWS for Distributed (Maestro) input dialog

Enter your input files

Variables

Resources

Events

Users

Calendars

Cpus

Jobs: /Users/My/Desktop/dep/jobs.txt

Schedules: /Users/My/Desktop/dep/schedules.txt

Prompts

Enter your preferences

Override task-type with type from cpu

Use only the TWS Jobname when creating task names and resources

Extract Interval/EVERY jobs if they have no dependencies

Name of SAP connection variable: \${ SAP_connection }

Name of SAP credential variable: \${ SAP_credential }

SAP command options: -dest DESTVAL -client 001

START conversion Cancel

Cron Options

Cron File Type

System Cron files usually contain users as the first "word" in the command.

- Set this option to System if you want the program to scan for users in the command and generate credentials.
- Set this option set to User if you do not know what type of file is being processed.

▼ Cron Options

Cron file type User System

Task name format cronTask# host name Add DateTime suffix

Cron Task Names

Cron task names can be generated in four ways, depending on the options used.

cronTask + sequence number	<ul style="list-style-type: none"> cronTask1 cronTask2
host name + sequence number	<ul style="list-style-type: none"> hostName1 hostName2

.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.

▼ .csv Options

A valid csv line will always generate a trigger. When present, on the workflow.

First line in file contains Column Headers

Separation character ▼
semi-colon ;

Type of task to generate ▼
Unix/Linux

Column: task names ▼

Column: task descriptions ▼

Column: commands ▼

Column: agents ▼

Column: time ▼ Sets the trigger time

Column: day ▼ Fills the trigger description

Column: users ▼

Column: calendars ▼

Column: resources ▼

Column: workflows ▼

Column: variable name ▼

Column: variable value ▼


Create environment variables

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). <div style="border: 1px solid gray; padding: 2px;"> <p>Note </p> <p>Note! The column DropDown-boxes are not filled yet!</p> </div>
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.

Column: variable name

17 variables ▼

 Create 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.

Keyboard shortcuts

Short cuts are defined to enable easy access to most panels.

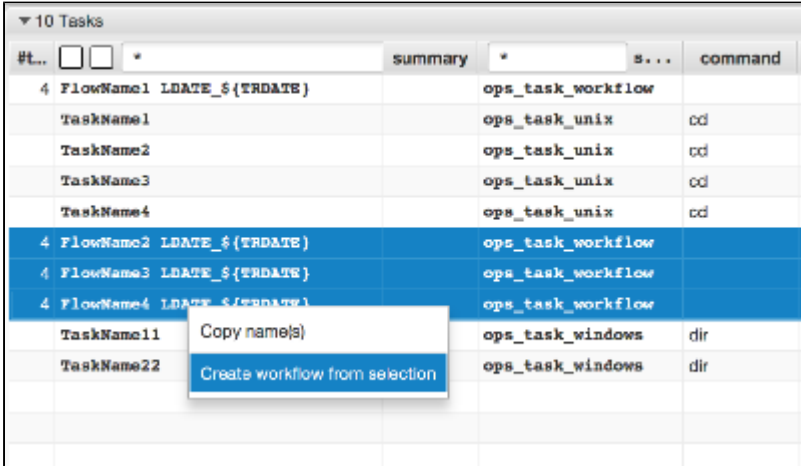
Shortcut	Panel
0	Tasks
1	Agents
2	Resources
3	Credentials
4	Triggers
5	BusinessServices
6	Calendars
7	CustomDays
8	Variables
l	Log
o	options

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 or Triggers pane.



#...	summary	s...	command
4 FlowName1 LDATE_\${TRDATE}		ops_task_workflow	
TaskName1		ops_task_unix	cd
TaskName2		ops_task_unix	cd
TaskName3		ops_task_unix	cd
TaskName4		ops_task_unix	cd
4 FlowName2 LDATE_\${TRDATE}		ops_task_workflow	
4 FlowName3 LDATE_\${TRDATE}		ops_task_workflow	
4 FlowName4 LDATE_\${TRDATE}		ops_task_workflow	
TaskName11		ops_task_windows	dir
TaskName22		ops_task_windows	dir

Step 2 In the Create Workflow dialog that displays, enter a name for the new workflow.



3 tasks selected

Give the workflow a UNIQUE name: flow11

Cancel OK

Create Workflows from Triggers

The same procedure works on the Triggers pane. Every task attached to the trigger will be copied to the new workflow.

Example 1: Trigger descriptions usually contain all run information.

Sort the description field on triggers to 'cluster' triggers with the same description. Then use the create workflow function to build workflows from the sorted list.

Example 2: Use the create workflow function to 'nest' workflow that were already created by the parser.

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

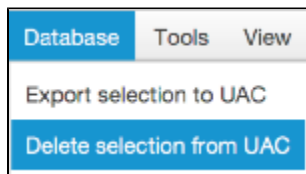


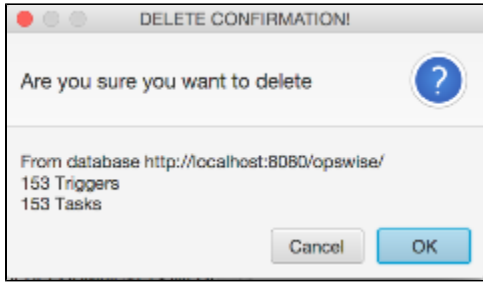
This option is set to UNCHECKED when an export is started.

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.






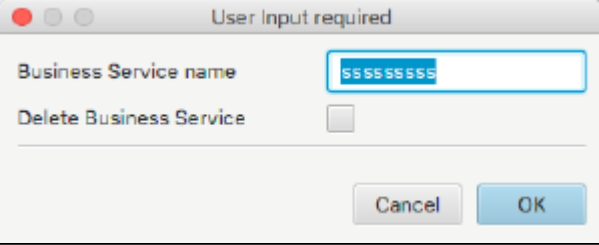
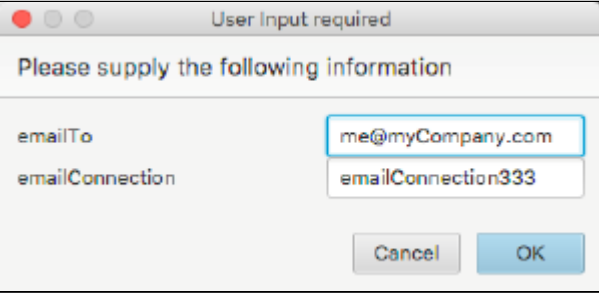
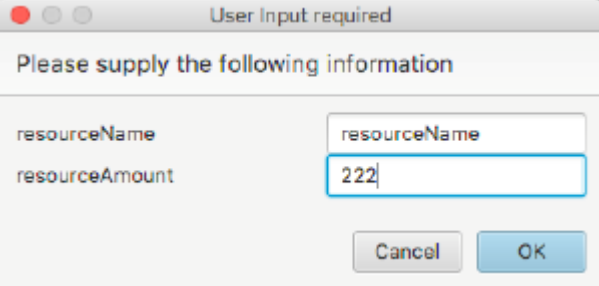
Enrich XCT Data

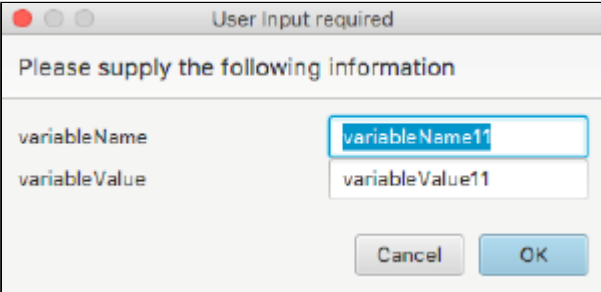
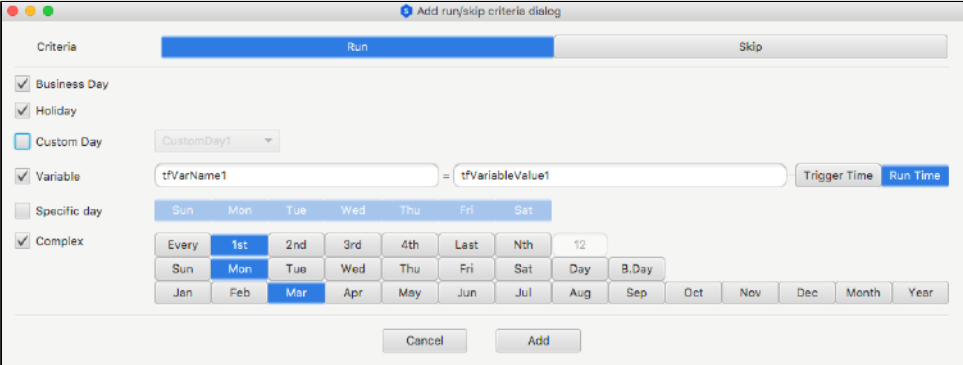
When the data is in the XCT, you have the option to enrich/change this data.

Depending on what type of input was used, extra options are available if you right-click the tasks list.

Context Menu Tasks

Function	Description	Available when Using File Input	Available when Using Database GET
Copy name to clipboard	Copies the name of the task and puts it on the clipboard.	Y	Y
Log parents	Retrieves the names of all parents and prints these in the log. (The log can copy to your clipboard.)	Y	N
Log predecessors	Retrieves the names of all predecessors and prints these in the log. (The log can copy to your clipboard.)	Y	N
Create workflow	Creates a workflow from the selected tasks. You will be prompted for the name. 	Y	N

<p>Add / Delete a Business Service</p>	<p>Add / Delete a Business Service to the selected tasks while preserving existing Business Service references. You will be prompted for the name.</p> 	<p>Y</p>	<p>Y</p>
<p>Add Email action</p>	<p>Creates an Email action for the selected tasks using a template connection. You will be prompted for an Email address and the connection name.</p> 	<p>Y</p>	<p>Y</p>
<p>Add Trigger</p>	<p>Adds a default daily trigger to the task.</p>	<p>Y</p>	<p>N</p>
<p>Add Virtual Resource</p>	<p>Creates a resource for the selected tasks. You will be prompted for the name.</p> 	<p>Y</p>	<p>Y</p>

<p>Add Local Variable</p>	<p>Creates a local variable for the selected tasks. You will be prompted for the name and value.</p> 	<p>Y</p>	<p>Y</p>
<p>Add end-dummy task to workflow</p>	<p>This will include dependencies to every 'last' vertex.</p>	<p>Y</p>	<p>N</p>
<p>Set Run / Skip Criteria</p>	<p>Creates Run/Skip criteria for each vertex associated with the selected tasks.</p> 	<p>Y</p>	<p>N</p>
<p>Delete from database</p>	<p>See DELETE in Database Tools. As opposed to the DELETE function in the Database Menu, which deletes all XCT objects from the UAC database in sequence, this DELETE is performed on the selected task(s) only!</p>	<p>Y</p>	<p>Y</p>
<p>Add System Operation</p>	<p>Creates a System Operation for only the selected tasks.</p>	<p>Y</p>	<p>N</p>

The resulting action will be performed ONLY on the selected (blue) tasks.

▼ 502 Tasks				
ta...	*	name	summary	* sys_class_name
4	FlowName1			ops_task_workflow
	TaskName12	demo		ops_task_sleep
	TaskName13	demo		ops_task_windows
	TaskName14	demo		ops_task_unix
	TaskName15	demo		ops_task_zos
4	FlowName2			ops_task_workflow
	TaskName22	demo		ops_task_sleep
	TaskName23	demo		ops_task_windows
	TaskName24	demo		ops_task_unix
	TaskName25	demo		ops_task_zos
4	FlowName3			ops_task_workflow
	TaskName32	demo		ops_task_sleep
	TaskName33	demo		ops_task_windows
	TaskName34	demo		ops_task_unix
	TaskName35	demo		ops_task_zos
4	FlowName4			ops_task_workflow
	TaskName42	demo		ops_task_sleep
	TaskName43	demo		ops_task_windows
	TaskName44	demo		ops_task_unix
	TaskName45	demo		ops_task_zos
4	FlowName5			ops_task_workflow
	TaskName52	demo		ops_task_sleep
	TaskName53	demo		ops_task_windows
	TaskName54	demo		ops_task_unix
	TaskName55	demo		ops_task_zos
4	FlowName6			ops_task_workflow
	TaskName62	demo		ops_task_sleep
	TaskName63	demo		ops_task_windows
	TaskName64	demo		ops_task_unix
	TaskName65	demo		ops_task_zos

Data Analysis

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	https://docs.stonebranch.com/confluence/display/UC65/How+to+Use+the+Xpress+Conversion+Tool#HowtoUse+the+Xpress+Conversion+Tool-BasicWorkflow
2	Go to the Log Pane and press Report Analysis	https://docs.stonebranch.com/confluence/display/UC65/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: <ul style="list-style-type: none">• 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](#)
- [Output Directories](#)
- [Import Generated XML](#)

Program Output

The conversion program can load the UAC database either of two ways:

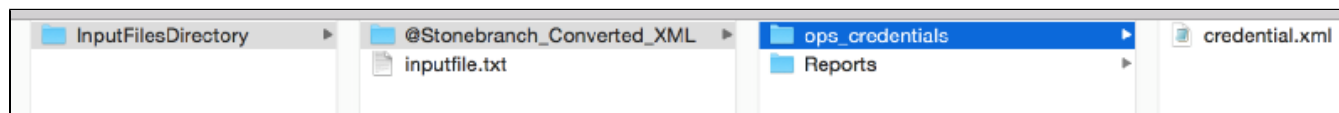
- Using the UAC RESTful API.
- Writing to file XML.

Both methods have their pro's and con's, but the API is recommended. Use it especially when you are doing a phased transition instead of a big-bang.

Method	Pro's	Con's
UAC RESTful API	<ul style="list-style-type: none"> • Easy loading. • Does not overwrite existing objects. 	<ul style="list-style-type: none"> • No Script and Credential support *
Write to file XML	<ul style="list-style-type: none"> • Supports all object types. • Overwrites existing objects. 	Manual load step.

* Credentials and Scripts are written to disk in a directory called `//InputFileDirectory/@Stonebranch_Converted_XML/`

(`InputFileDirectory` represents the directory where your input file resides.)



Output Directories

The conversion program can write to file:

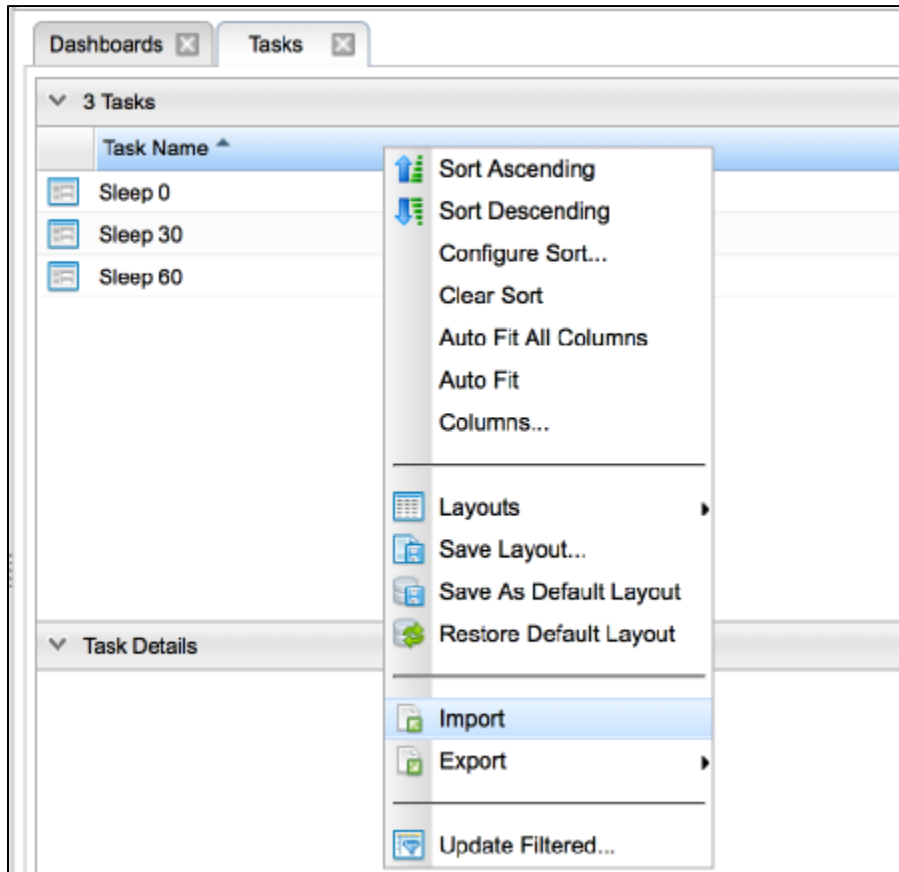
- In one directory.
- To multiple typed directories.

See [Global Options](#) for selecting this checkbox.

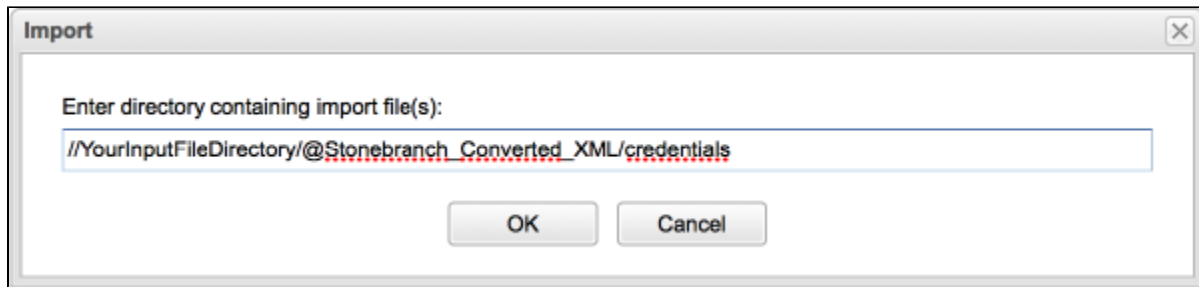
Import Generated XML

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 press [OK](#).

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

Database Tools

- Database Tools
 - Available Tools
 - GET
 - POST
 - PUT
 - DELETE
 - VERIFY
 - Transfer from/to UAC
 - Test API XML
 - Add Notes
- Audit Trail

Database Tools

The goal of database tools is to allow for multiple changes (mass update) at the same time.

For example, if you want to add a local variable to 800 selected tasks, you could achieve this with a few clicks.

These functions use the RESTful Application Program Interface (API) to make the calls.

These tools make it easy to modify data and PUT it back in the database without concerning yourself with:

- Where to PUT the data (the 'resource'-part in the URL)
- What sequence to use when PUTting the data (A referenced calendar needs to be present)
- Using a PUT or POST when loading
- What vehicle to use when making the call (Pearl, script, Power*, web-service, REXX, Java)

Available Tools

	UAC	
#GET	GET data from UAC to XCT	
#POST	POST data in UAC.	Available when retrieved with the GET request.
#PUT	PUT data in UAC.	Available when retrieved with the GET request.
#DELETE	DELETE data in UAC.	Using the XCT list.
#VERIFY	Verify data in UAC.	Using the XCT list.
#Transfer	Copy data from UAC1 to UAC2.	
#Test API	Experiment with the API.	

#Add Notes	Add text files as a UAC Note to tasks.	
------------	--	--

Most functions use the log to display messages and the progress bar to indicate progress.

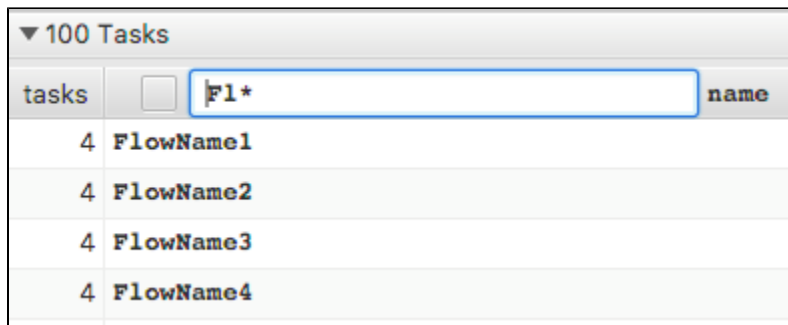
The log automatically will open when using these functions.

GET

The request will prompt you for credentials and some wildcards. It will retrieve all tasks and triggers matching the selection pattern.

The task will be displayed in the XCT after the completed transfer of all objects.

You then can make further sub-selections using the task filter field in the XCT for modifying or PUT back.



▼ 100 Tasks	
tasks	name
<input type="checkbox"/>	F1*
4	FlowName1
4	FlowName2
4	FlowName3
4	FlowName4

When Trace is checked then all RESTful calls will result in a log message.

When include references is checked then referenced objects like calendars, resources, etc. will also be retrieved.

UAC database Login

Enter credentials for the GET request

Connection URL

User name

User password

Trace

Include references

task selection

trigger selection

Cancel GET

Use Trace only in case of trouble!

The Trace function will produce many messages, including the xml decks that are being transferred.

R	GET: HTTP://LOCALHOST:8080/OPSWISE/		
R	-----		
I	GET	triggers retrieved	0
I	GET	tasks retrieved	48

POST

The request will prompt you for credentials.

It will take all visible entries in the XCT and try to POST the data in UAC.

This means you can retrieve a set of data and put only a subset back in the Controller.

Use the task filter field in the XCT to filter the tasks that you want to POST.

UAC database Login

Enter credentials for the POST request

Connection URL

User name

User password

Trace

Cancel POST

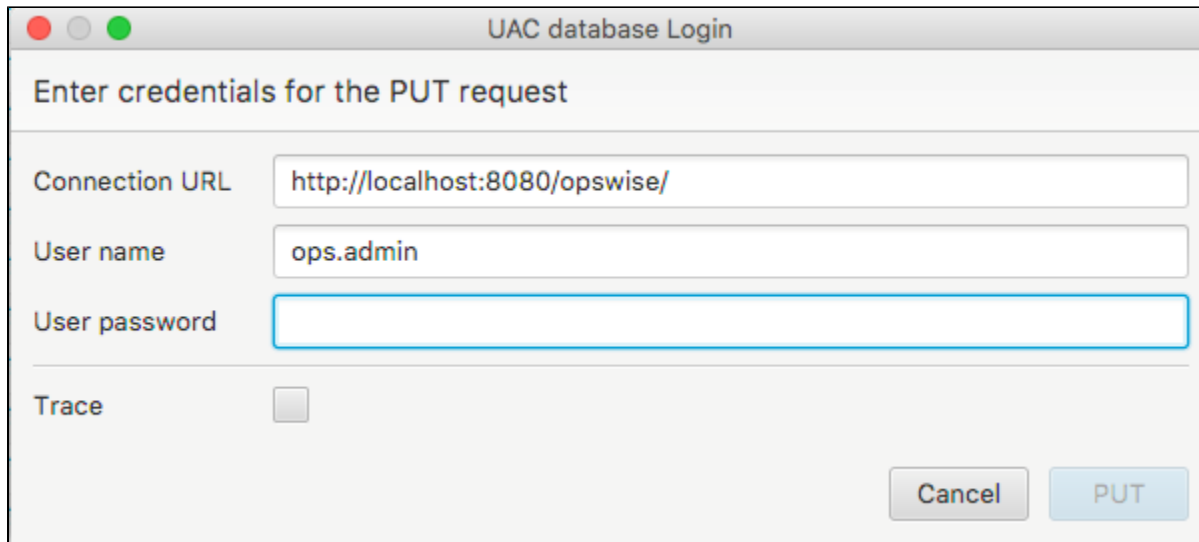
PUT

The request will prompt you for credentials.

It will take all visible entries in the XCT and try to PUT the data in UAC.

This means you can retrieve a set of data and put only a subset back in the Controller.

Use the task filter field in the XCT to filter the tasks you want to PUT.



UAC database Login

Enter credentials for the PUT request

Connection URL

User name

User password

Trace

Cancel PUT

DELETE

The request will prompt you for credentials.

It will take all visible entries in the XCT and try to DELETE the data in UAC.

This means you can retrieve a set of data and only DELETE a subset in the Controller.

Use the task filter field in the XCT to filter the tasks you want to DELETE.

Since a DELETE request requires only the objects name, this function can be used whether the XCT data has been created using:

- The GET request
- A regular conversion through input files

UAC database Login

Enter credentials for the DELETE request

Connection URL

User name

User password

Trace

Cancel DELETE

VERIFY

The request will prompt you for credentials. It will GET the data and PUT it back unchanged.

The PUT, however, passes the UAC validator, thus making sure that all properties, references, and dependencies have been set correctly.

The PUT will fail if an error is detected, and it will generate a message in the log.

It is typically used after a conversion when the converted data has been loaded in UAC, no matter what conversion tool was used.

UAC database Login

Enter credentials for the VERIFY request

Connection URL

User name

User password

Trace

Transfer from/to UAC

The request will prompt you for credentials and some wildcards for the source UAC Controller (see [#GET](#)).

A second prompt will appear for the target UAC Controller.

UAC database Login

Enter credentials for the XFER request

Connection URL

User name

User password

Trace

Overwrite

Cancel XFER

This will allow you transfer objects from one UAC to another; for example, a cloud Controller.

Test API XML

The request will prompt you for credentials.

UAC database Login

Enter credentials for the POST request

Connection URL

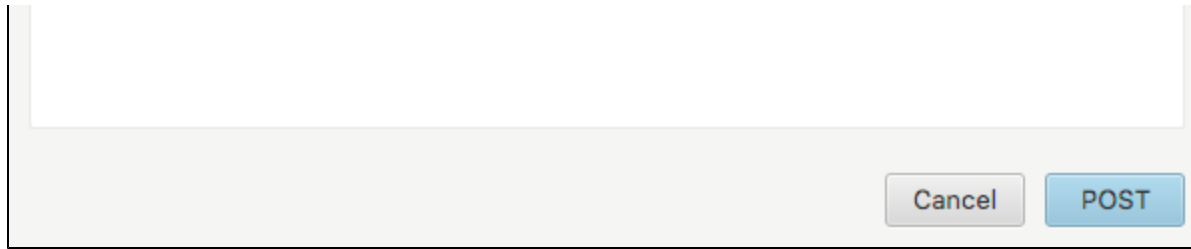
User name

User password

Trace

POST GET PUT DELETE

```
<customDay retainSysIds="true">
  <adjustment>None</adjustment>
  <adjustmentAmount>1</adjustmentAmount>
  <adjustmentType>Day</adjustmentType>
  <category>Day</category>
  <comments />
  <ctype>Relative Repeating Date</ctype>
  <date />
  <dateList />
  <day>1</day>
  <dayofweek>Sun</dayofweek>
  <holiday>>false</holiday>
  <month>Sep</month>
  <name>My special dayJos</name>
  <nthAmount>8</nthAmount>
  <nthType>Business DayJJJ</nthType>
  <period>>false</period>
  <relfreq>1</relfreq>
  <sysId>dae77560c0a80165006d8051753b806c</sysId>
</customDay>
```



This enables you to test XML calls from outside of UAC.

It contains four Tabs representing the different RESTful calls. It provides a resizable window and edit area to accommodate for changes.

All user input is saved over sessions except for the Trace function and password values.

Add Notes

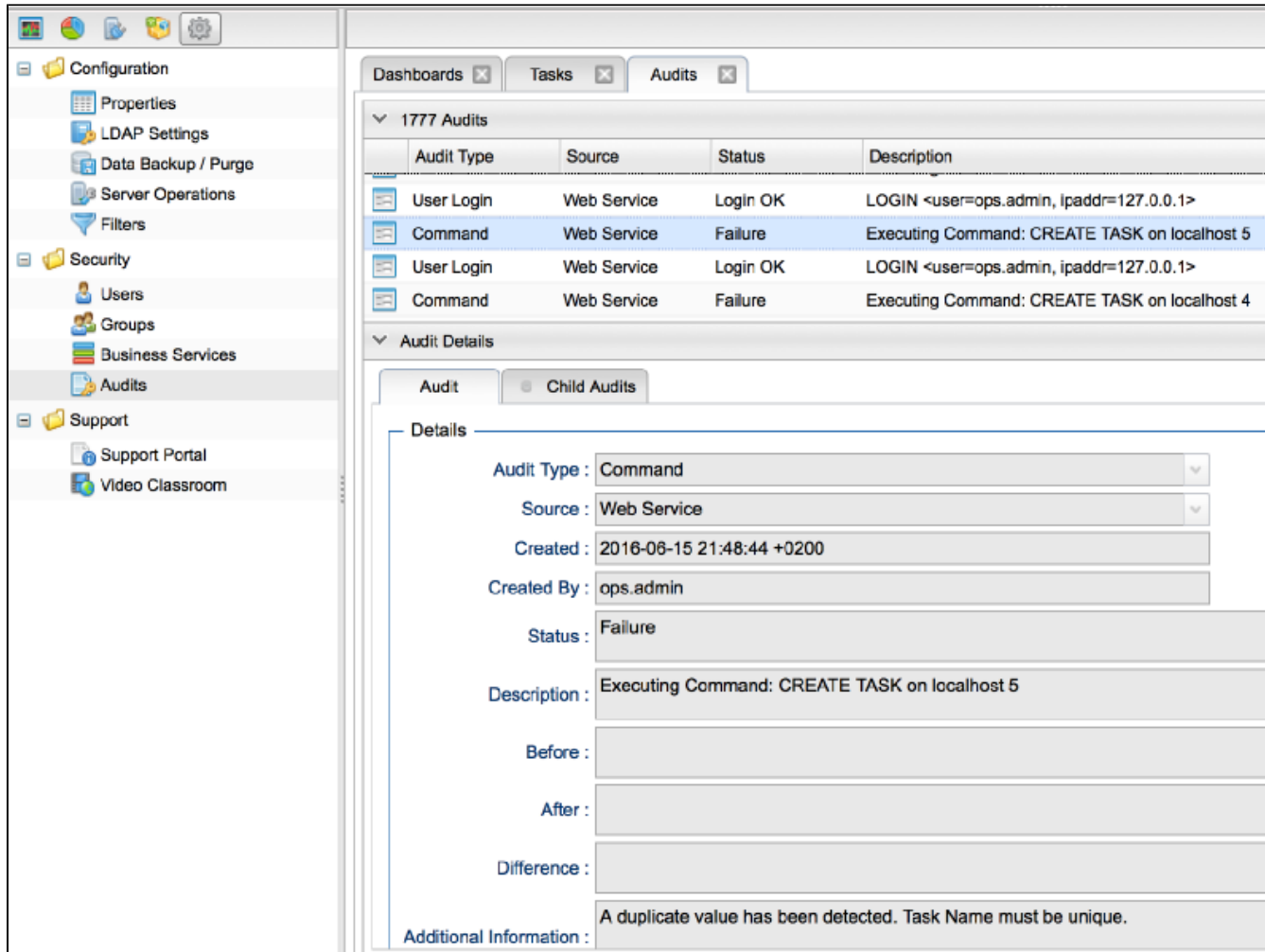
The request will prompt you for a directory containing text files.

These files should be named exactly like the tasks in UAC.

Only then the parser will create a Note and attach it to the task.

Audit Trail

One of the advantages of using the API is its audit trail. Every Web Service action is recorded and can be viewed from the UAC dialog.



The screenshot displays the 'Audits' section of the Xpress Conversion Tool. The left sidebar shows a navigation menu with categories: Configuration (Properties, LDAP Settings, Data Backup / Purge, Server Operations, Filters), Security (Users, Groups, Business Services, Audits), and Support (Support Portal, Video Classroom). The main window has tabs for Dashboards, Tasks, and Audits. Under the Audits tab, there is a list of 1777 audits. The table below shows a sample of these audits:

Audit Type	Source	Status	Description
User Login	Web Service	Login OK	LOGIN <user=ops.admin, ipaddr=127.0.0.1>
Command	Web Service	Failure	Executing Command: CREATE TASK on localhost 5
User Login	Web Service	Login OK	LOGIN <user=ops.admin, ipaddr=127.0.0.1>
Command	Web Service	Failure	Executing Command: CREATE TASK on localhost 4

Below the table, the 'Audit Details' section is expanded, showing the details for the selected audit:

- Audit Type:** Command
- Source:** Web Service
- Created:** 2016-06-15 21:48:44 +0200
- Created By:** ops.admin
- Status:** Failure
- Description:** Executing Command: CREATE TASK on localhost 5
- Before:**
- After:**
- Difference:**
- Additional Information:** A duplicate value has been detected. Task Name must be unique.

Select an audit record and browse the details for reasons of failure when loading the UAC database through the API.

Command Line Interface

- [Overview](#)
- [Parameters](#)
 - [List of Parameters](#)
- [Command String Examples](#)
 - [SAP Example - Output to XML File](#)
 - [SAP Example - Output to Database](#)

Overview

The Command Line Interface provides automation for conversions. You can schedule a conversion task by executing a command string containing parameters/arguments.

Parameters

When the conversion toolkit is started with parameters, the program runs in CLI mode.

The minimum requirement for running in CLI mode is at least one cliType parameter.

The following is a very basic example:

```
java -jar /PathTo/toolkit.jar --cliType=CRON --cliJobs=/PathTo/YourJobs
```

List of Parameters

Parameter	Required	Type	Example	Default
cliCalendarFreeDays	No	File	--cliCalendarFreeDays=/PathTo/YourHolidayCalendars.txt	
cliCalendarWorkDays	No	File	--cliCalendarWorkDays=/PathTo/YourFactoryCalendars.txt	
cliJobs	No	File Directory	--cliJobs=/PathTo/YourJobs	
cliModifyExisting	No	Boolean	- cliModifyExisting=true	
cliOutputDir	No	Directory	--cliOutputDir=/PathTo/YourOutputDirectory	
cliOutputDirPerType	No	Boolean	--cliOutputDirPerType=true	false
cliPassword	No	String	--cliPassword=YourPassword	
cliReadEncoding	No	String	--cliReadEncoding=UTF-8 --cliReadEncoding=UTF-16 --cliReadEncoding=IBM865	ISO-8859-1

cliSapAbapStepDivider	No	Boolean	--cliSapAbapStepDivider	False
cliSapConnectionVar	No	String	--cliSapConnectionVar=MySapConnectionName	\${SAP_connection}
cliSapCredentialVar	No	String	--cliSapCredentialVar=MySapCredentialName	\${SAP_credential}
cliSapCommandOptions	No	String	--cliSapCommandOptions="-dest DESTVAL -client 001"	-dest DESTVAL -client 001
cliSapJobnameToUpper	No	Boolean	--cliSapJobnameToUpper	False
cliTrace	No	String	--cliTrace=ON	OFF
cliType	Yes	String	--cliType=SAP	
cliUrl	No	URL	--cliUrl= http://Your/localhost:8080/opswise/	http://localhost:8080/opswise/
cliUser	No	String	--cliUser=YourUserName	

Command String Examples

When cliOutputDir is defined, the program writes one XML file per object to this output directory. When cliOutputDir is omitted, the program tries to use the Web Service to load the objects in the UAC database.

All messages will be written to STDOUT.

SAP Example - Output to XML File

```
java -jar /PathTo/toolkit.jar
--cliType=SAP
--cliCalendarWorkDays=YourFactoryCalendars.txt
--cliCalendarFreeDays=YourHolidayCalendars.txt
--cliJobs=YourJobs
--cliOutputDir=YourOutputDirectory
```

SAP Example - Output to Database

```
java -jar /PathTo/toolkit.jar
--cliType=SAP
--cliCalendarWorkDays=YourFactoryCalendars.txt
--cliCalendarFreeDays=YourHolidayCalendars.txt
```


SAP

- [SAP](#)
- [SAP Calendars](#)
- [Custom Days](#)
- [SAP Job Definitions](#)
 - [Example](#)
 - [EVENTPARM](#)
- [Exporting SAP Definitions to UAC](#)

SAP

The conversion program supports:

- SAP Factory and Freeday calendars
- SAP Job definitions

These reports can be generated through a Stonebranch Utility.

Both formats can be read through the File Menu Option. Since there can be multiple calendar files, please read all factory calendars first and all freeday calendars second.

Third, 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 Calendars

A calendar file can contain many calendar IDs, each describing different years. All unique calendar IDs are stored with a prefix describing its function. A working day (flagged 1 in the file) will be set accordingly in the created calendar, hence different workday definitions.

Custom Days

SAP custom day records describe each day of the year, flagged as 1 or 0. They can occur for different calendar IDs describing different years. Often, it is not necessary to convert all of them.

- Records older than 730 days are discarded, leaving only the last one or two years valid.
- Records newer than 1500 days are discarded, leaving the first 4 or 5 upcoming years valid.

The log will display which records 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.

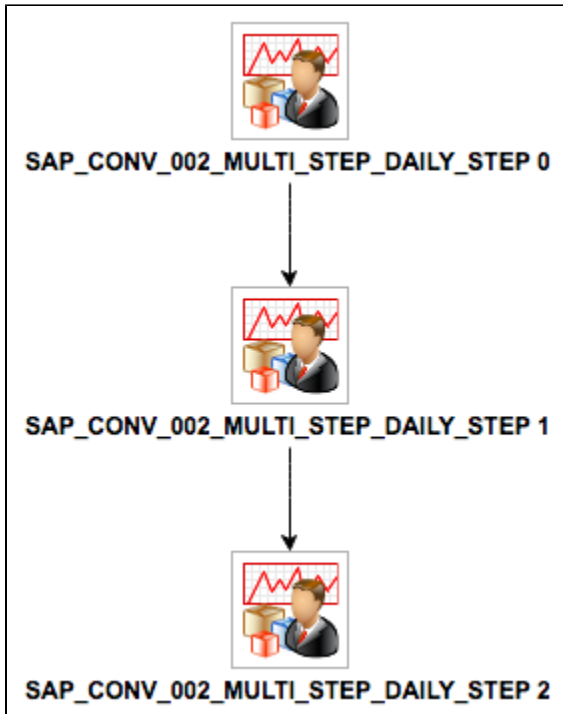
Every ABAP step in the file then is converted into a separate task and automatically receives a dependency to its predecessor ABAP step.

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.

Exporting SAP Definitions to UAC

The API currently does not support scripts or credentials. It is therefore possible only to load scripts and credentials by generating XML files and loading them manually.

This means you have to load UAC in either of two ways:

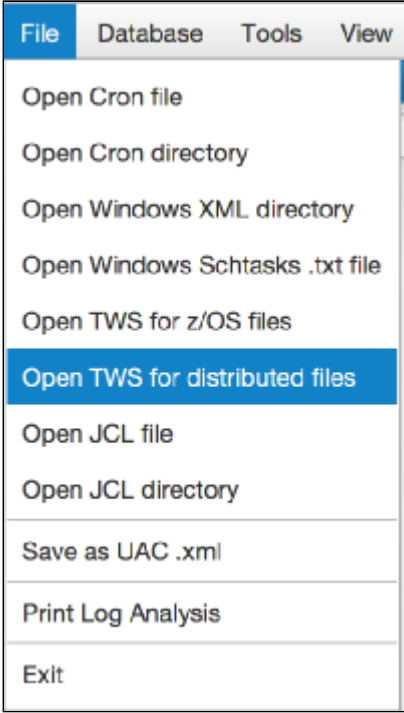
1	<ol style="list-style-type: none"> 1. Save as UAC .xml. 2. Import all XML files.
2	<ol style="list-style-type: none"> 1. Save as UAC .xml. 2. Load only scripts and credentials. 3. Use UAC API Post.

TWS for Distributed Data

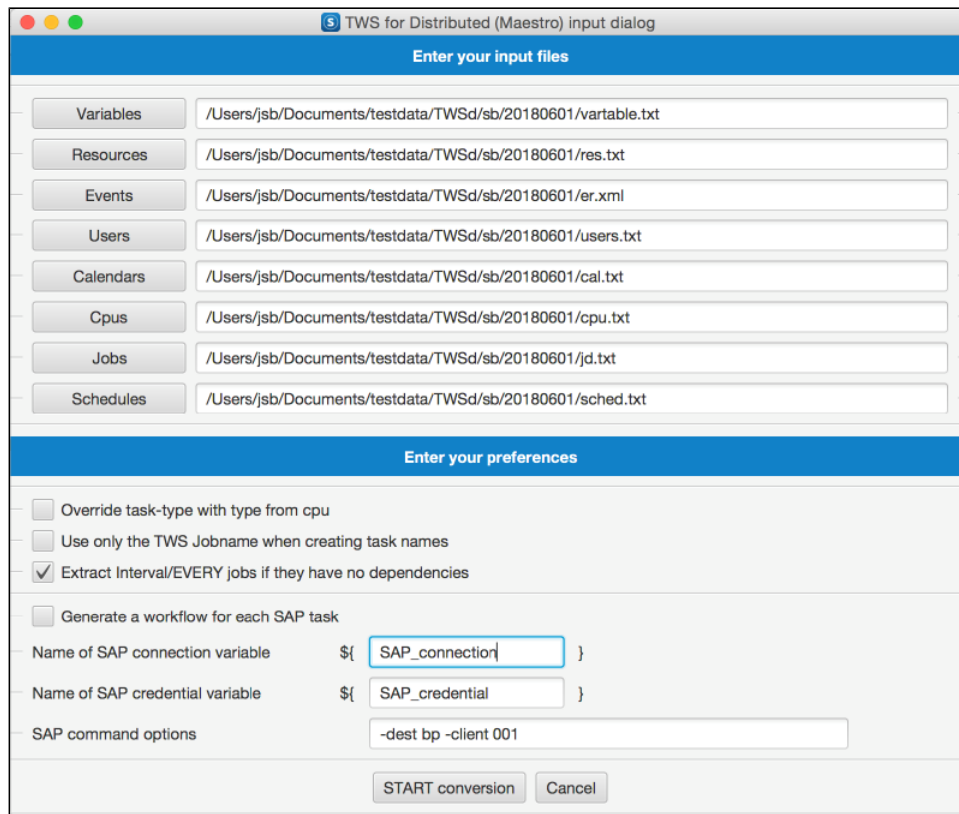
- [TWS for Distributed Data](#)

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.



The screenshot shows a dialog box titled "TWS for Distributed (Maestro) input dialog". It is divided into two main sections: "Enter your input files" and "Enter your preferences".

Enter your input files:

- Variables: /Users/jsb/Documents/testdata/TWSd/sb/20180601/variable.txt
- Resources: /Users/jsb/Documents/testdata/TWSd/sb/20180601/res.txt
- Events: /Users/jsb/Documents/testdata/TWSd/sb/20180601/er.xml
- Users: /Users/jsb/Documents/testdata/TWSd/sb/20180601/users.txt
- Calendars: /Users/jsb/Documents/testdata/TWSd/sb/20180601/cal.txt
- Cpus: /Users/jsb/Documents/testdata/TWSd/sb/20180601/cpu.txt
- Jobs: /Users/jsb/Documents/testdata/TWSd/sb/20180601/jd.txt
- Schedules: /Users/jsb/Documents/testdata/TWSd/sb/20180601/sched.txt

Enter your preferences:

- Override task-type with type from cpu
- Use only the TWS Jobname when creating task names
- Extract Interval/EVERY jobs if they have no dependencies
- Generate a workflow for each SAP task
- Name of SAP connection variable: \${ SAP_connection }
- Name of SAP credential variable: \${ SAP_credential }
- SAP command options: -dest bp -client 001

Buttons: START conversion, Cancel

Step 3

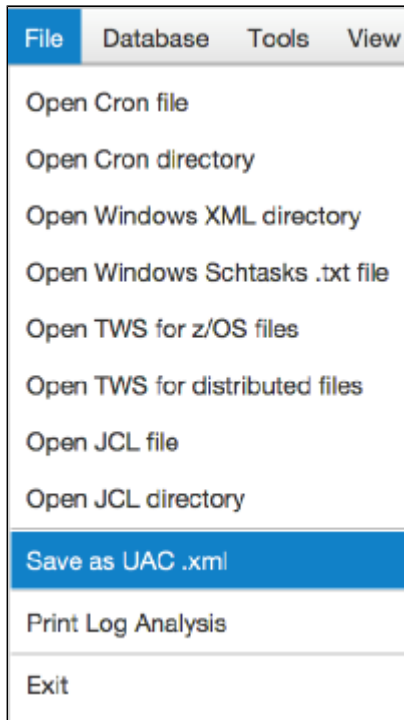
Set the TWS-d input file locations.

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.

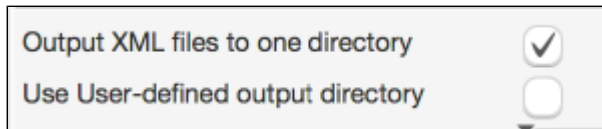
Step 4	<p>Set TWS-d preferences:</p> <ol style="list-style-type: none">1. Override task type with type from cpu. Normally, a task type is determined by the job definition. In some cases, the determined type can be overridden by the type of operating system it runs on. Use only on direction of Stonebranch 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	<p>Set SAP preferences.</p> <ol style="list-style-type: none">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	<p>Start conversion.</p> <p>After starting the conversion, all files will be processed in sequence. A progress bar at the top right corner informs you of the progress.</p>

Step 7

Save as XML.



Before writing the UAC XML, please consider the following options:



Output, by default, is written to the location where your input resides. Separate directories for all XML types and reports will be created.

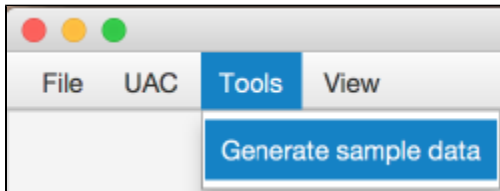
Optionally, you can request to write all XML files to one directory instead of separate directories. This enables you to import the files in UAC at one time.

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							
<input type="checkbox"/>	<input type="checkbox"/>	* <input type="text"/>	name	agent_var	* <input type="text"/>	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

- [Autosys](#)
- [Windows Scheduled Task](#)
- [Windows XML](#)
- [Cron](#)
- [TWS Distributed](#)
- [TWS z/OS](#)
- [Control-EM Distributed](#)
- [CA-7](#)

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.

watch_interval	
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.

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

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

Control-EM Distributed

Control-EM	UAC
JOB	<ul style="list-style-type: none"> • 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

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 <i>Requirements and Network Connections</i> 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 ?

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.

S	name	reason	info	remark
R				
R	GRAND TOTALS			
R	-----			
R	all tasks	502		
R	workflows	100		
R	timers	100		
R	windows tasks	102		
R	unix/linux	100		
R	z/OS tasks	100		
R	file Monitors	0		

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

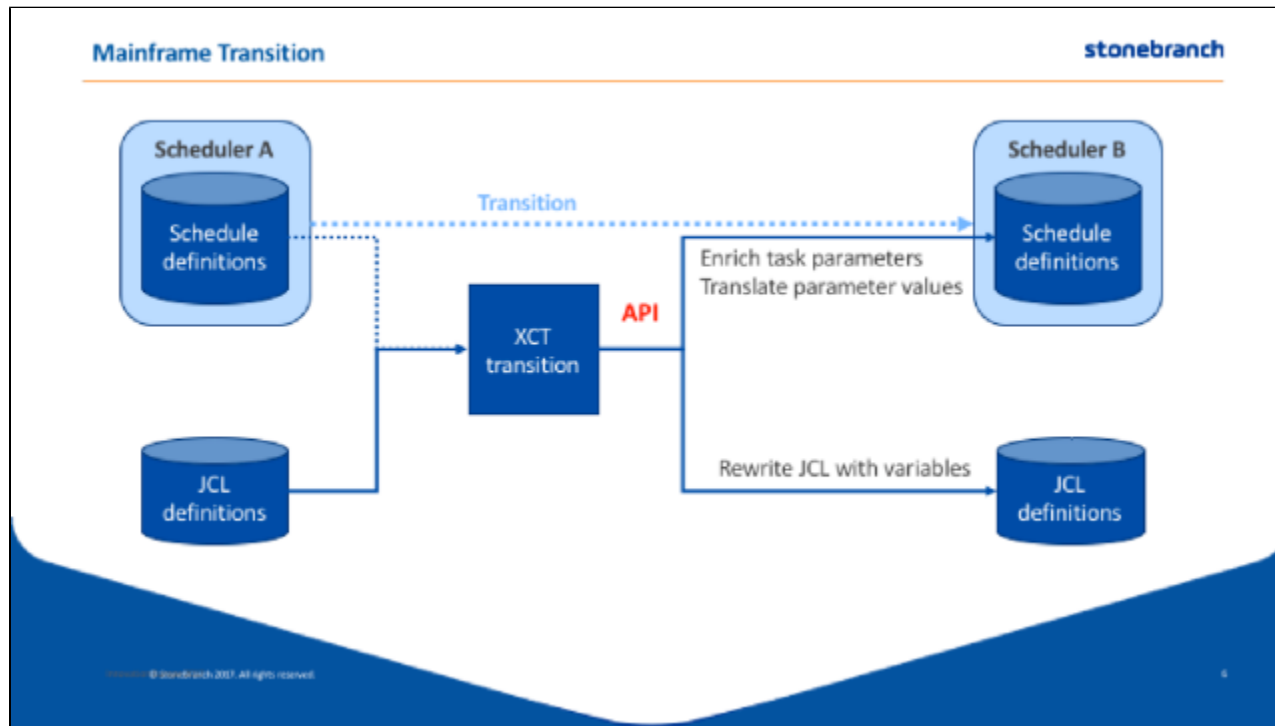
Mainframe scheduler transitions are, by nature, large and complex, but more importantly, there is always a second component that must be converted: JCL, Job Control Language.

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.

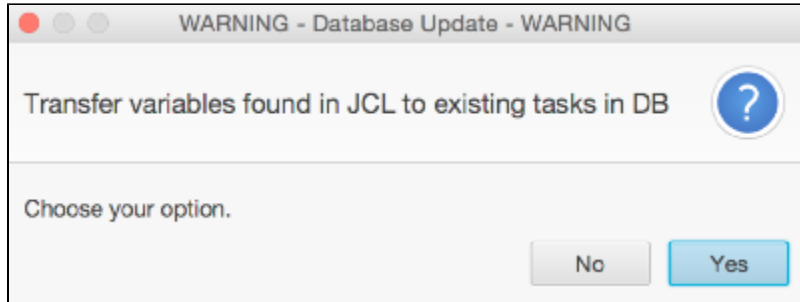


How to Perform a JCL Transition

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.

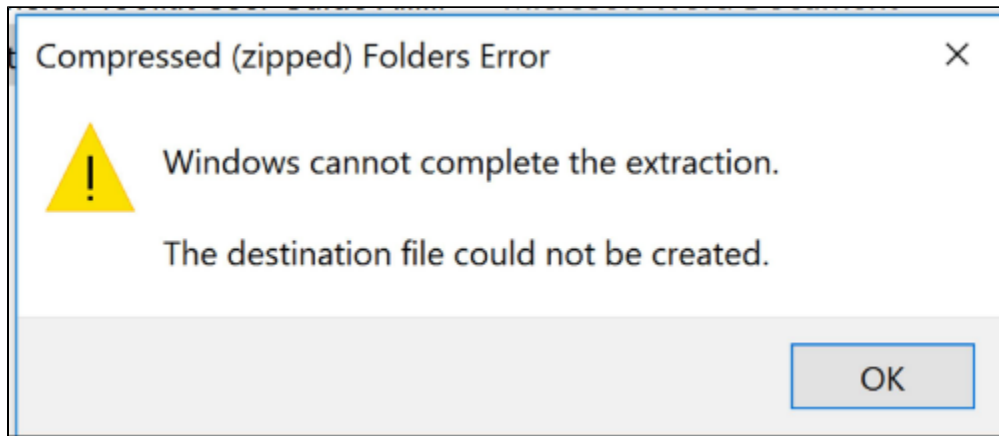
Scheduler-specific variables will be translated, as much as possible, and created as parameters on the z/OS task via the web API.

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.