

stonebranch

Xpress Conversion Tool

© 2018 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	14
1.4 Audit Trail	17
1.5 Options	18
1.6 Command Line Interface	25
1.7 SAP	27
1.8 TWS for Distributed Data	31
1.9 Generate Sample Data	35
1.10 Keyword Mapping	37
1.11 Log	42
1.12 Job Control Language	43
1.13 Support	45

Xpress Conversion Tool

The Xpress Conversion Tool documentation contains the following pages:

- [Overview](#)
- [How to Use the Xpress Conversion Tool](#)
- [Program Output](#)
- [Audit Trail](#)
- [Options](#)
- [Command Line Interface](#)
- [SAP](#)
- [TWS for Distributed Data](#)
- [Generate Sample Data](#)
- [Keyword Mapping](#)
- [Log](#)
- [Job Control Language](#)
- [Support](#)

Overview

- Overview
 - Supported Data Formats
 - Features
 - Static Defaults
 - License
 - Linux

Overview

This document is a step-by-step guide on how to use the Xpress Conversion Tool (XCT). 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

- Autosys reports
- Comma-separated-value (.csv)
- Control-EM XML
- Cron file / directory
- JCL library
- JobTrac
- SAP calendars
- SAP Job definitions
- TWS for distributed
- TWS for z/OS
- Windows Scheduled tasks
- Windows XML

Features

- Exit codes translation
- Agent variables
- Multiple files from multiple schedulers
- Selective export
- Command scanning for variables
- Original definitions are archived as Notes
- User preferences
- Create workflows by manual selection

Static Defaults

The program generates static defaults with the same sys_id for:

	Name	Sys_id
defaultAgent	Localhost	aaf237d072c64ab49dfe1fcfd799e77
defaultCalendar	Conversion Calendar	bbf237d072c64ab49dfe1fcfd799e77
defaultCredential	Conversion Credential	ccf237d072c64ab49dfe1fcfd799e77

This allows for hooking up tasks to the same defaults over multiple exports and multiple sessions.

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
<pre>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)</pre>	<pre>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)</pre>

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

Terminal Input:

```
sudo apt-get install openjfx
```

OpenSuse

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.

How to Use the Xpress Conversion Tool

- Overview
- User Interface
 - Global Options
 - Specific Options
- Keyboard shortcuts
- Create Workflows Manually
 - Create Workflows from Tasks
 - Create Workflows from Triggers
- Basic Workflow
- Sub-Selections
 - Export (sub) Selection
 - Delete (sub) Selection

Overview

The program uses different kind of files 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.



Note

Reading files twice in the same session is not allowed.

It is recommended to set the applicable options before reading a file. For instance, a Business Service can be added for every task found when the option is checked. The next file can use a different Business Service, the same or none.

The program links all related objects and creates appropriate UAC objects for export to UAC. These can be exported in 2 ways:

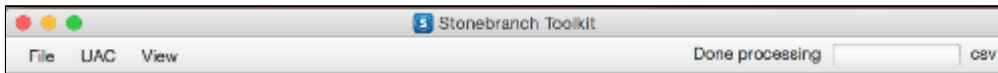
- Export to UAC using the RESTful API (preferred)
- Export to XML files

The second method requires the user to load the UAC database manually.

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.

▼ 8 Tasks											
<input type="checkbox"/>	<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.

▼ Log				
Search Log		log search argument		
datetime	S	message	reason	info
2016-10-21 17:03:21	I	Toolkit 6.3		
2016-10-21 17:03:21	I	Running on Java 1.8.0_45		
2016-10-21 17:21:20	I	jobfile.txt	Reading file	
2016-10-21 17:21:20	I	jobfile.txt	Valid file type found: insert_job:	
2016-10-21 17:21:20	C	67,644 Lines in inputfile		
2016-10-21 17:21:20	C	52,882 Lines used		
2016-10-21 17:21:20	C	14,762 Lines discarded (empty)		

4 Right pane displaying the options.

Global Options

Please set all options before reading any input file.

Write file XML to one directory

Add a sample trigger to workflow tasks

Add a sample trigger to non-workflow tasks

Allow duplicate task names

add workflow suffix LDATE_\${TRDATE}

Attach to task BusinessServiceName

File XML DateTime format yyyy-MM-dd HH:mm:ss e.g. sys_created_on

UAC controller http://localhost:8080/opwise

UAC administrator ops.admin

UAC administrator password

Toolkit License

Global Options

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

Specific Options

Specific options are valid only for a specific input format, such as Cron or .csv.

Keyboard shortcuts

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

Shortcut	Panel
0	Tasks
1	Agents
2	Resources

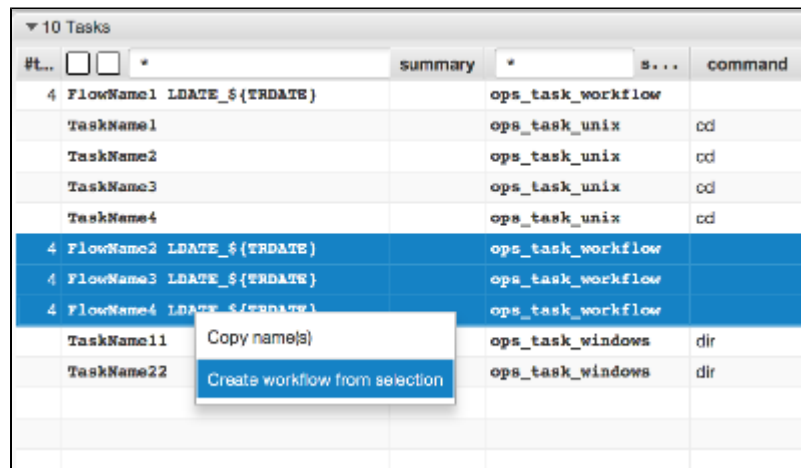
3	Credentials
4	Triggers
5	BusinessServices
6	Calendars
7	CustomDays
8	Variables
l	Log
o	options

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.



The screenshot shows a table titled "10 Tasks" with columns: #L., checkboxes, summary, *, s..., and command. The table contains several rows, with three rows selected (highlighted in blue). A context menu is open over the selected rows, showing options: "Copy name(s)" and "Create workflow from selection".

#L..	<input type="checkbox"/>	<input type="checkbox"/>	*	summary	*	s...	command
4			FlowName1	LEDATE_{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	LEDATE_{TRDATE}	ops_task_workflow		
4			FlowName3	LEDATE_{TRDATE}	ops_task_workflow		
4			FlowName4	LEDATE_{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.



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.

Basic Workflow

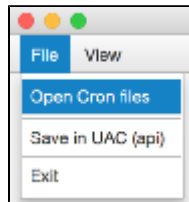
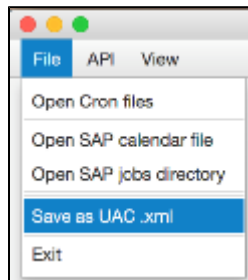
The toolkit minimizes the necessary steps to import data into UAC. Use the RESTful API to load directly into UAC or write XML files to file and import them manually.

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

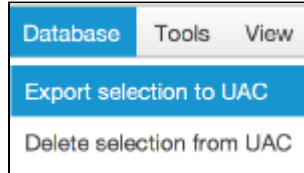
(When using the RESTful API, provide the connection parameters.)

UAC controller	<input type="text" value="http://localhost:8080/opswise/"/>
UAC administrator	<input type="text" value="ops.admin"/>
UAC administrator password	<input type="password"/>

The administrator must be a member of the administrator group.

Step 2 Open a file.**Step 3** Export the data.

OR



The results of the export will be visible in the Log Pane.

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.



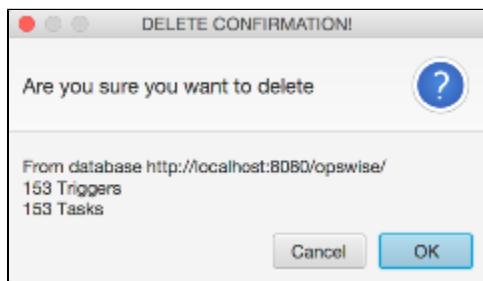
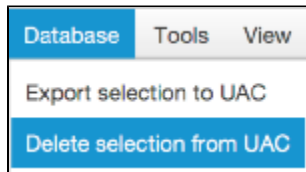
Note

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.



Program Output

- Program Output
- Output Directories
- Import Generated XML

Program Output

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

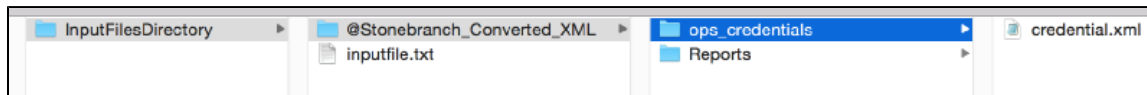
- Using the UAC RESTful API.
- Writing to file XML.

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:

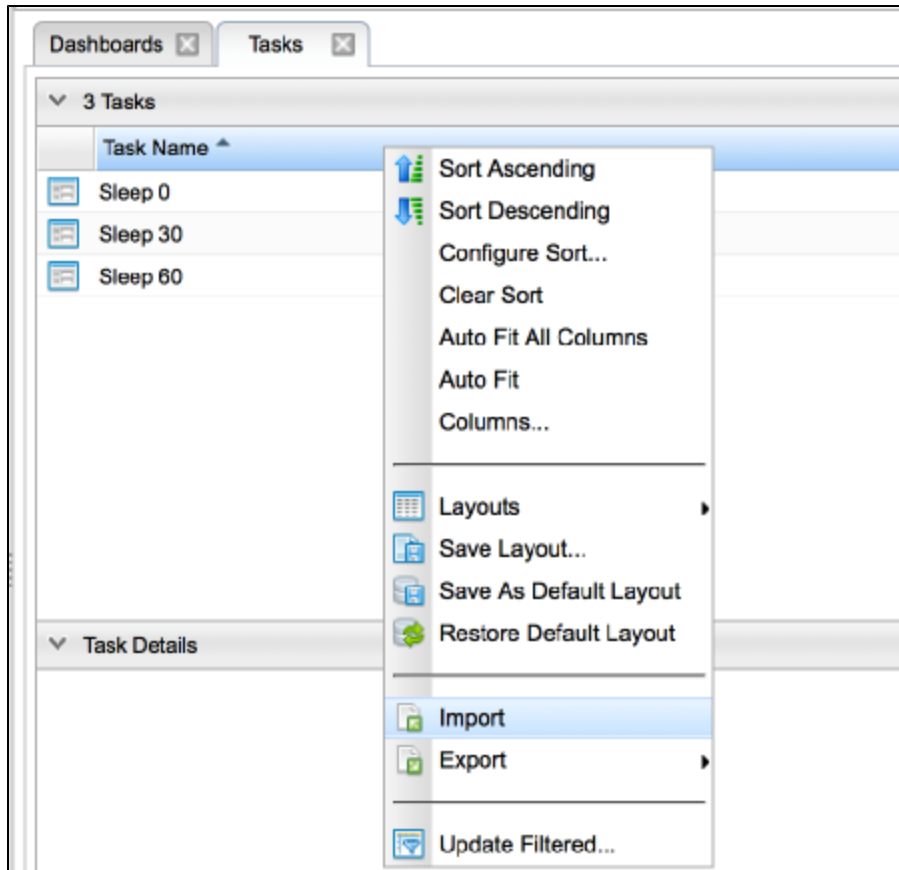
- In one directory.
- To multiple typed directories.

See Global Options for selecting this checkbox.

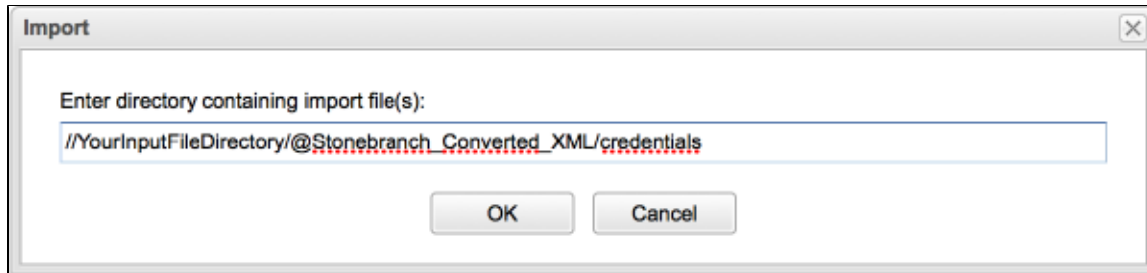
Import Generated XML

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.



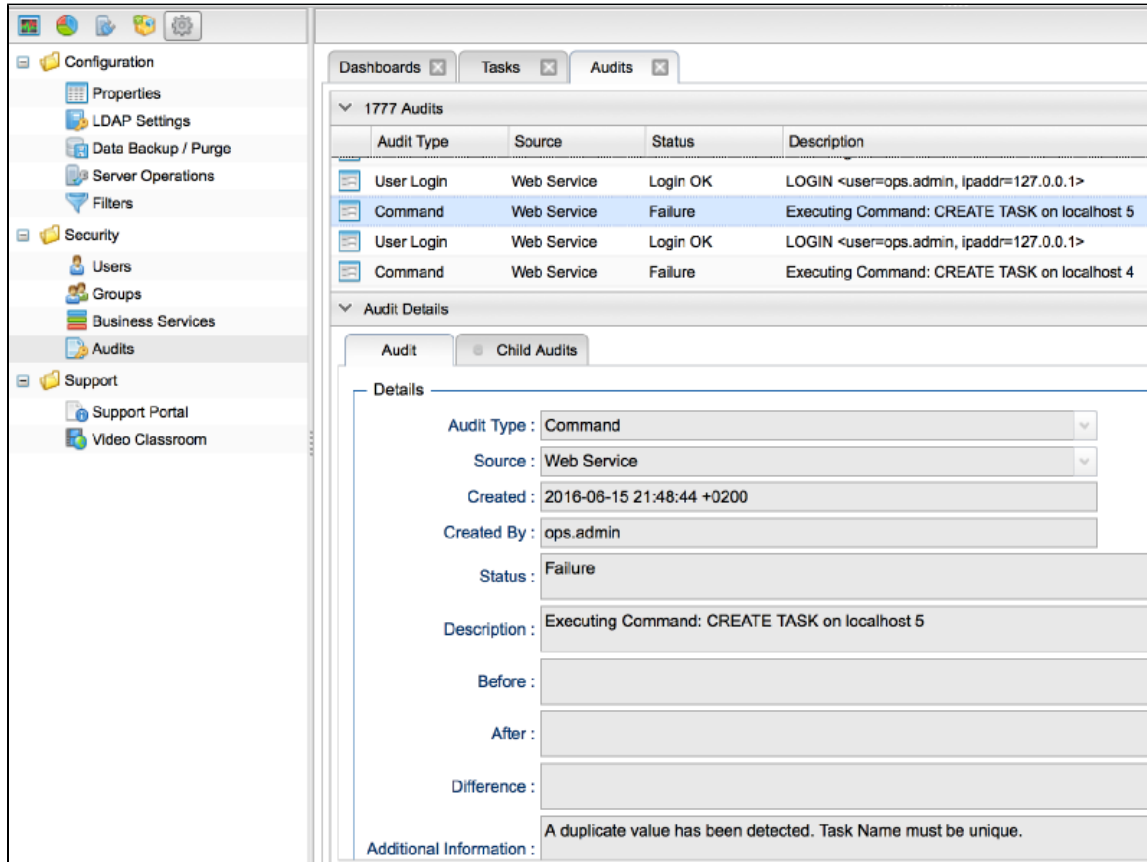
Paste the path string in the text field and then press SMLR!OK.

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

Audit Trail

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 shows the Xpress Conversion Tool interface. On the left is a navigation sidebar 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. The Audits tab is active, showing a list of 1777 audits. The table below is a summary of the audit records:

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 for the selected record. It shows the following details:

- 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: (empty)
- After: (empty)
- Difference: (empty)
- 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.

Options

- Conversion Toolkit Options
- Global Options
 - Sample Trigger
 - Business Services
 - Workflow Suffix
 - Date Time Format
 - Duplicate Task Names
- Cron Options
 - Cron File Type
 - Cron Task Names
- .csv Options
 - Usage
 - Variables
 - Expected Input Template

Conversion Toolkit Options

The Conversion program provides two types of options:

- SMLRI:Global
- SMLRI:Schedular-specific

Global Options

Sample Trigger

A sample trigger, daily 00:00, can be provided for each workflow without triggers.

Add sample trigger to workflows

Business Services

Attach a business service to every task regardless which format is being parsed. For example, the name of the business service can be changed per file to add different business groups while converting.

▼ Global Options

Please set all options before reading any input file.

Attach to task

BusinessServiceName

add workflow suffix

LDATE_\${TRDATE}

Workflow Suffix

Add a suffix to every workflow to include the trigger date in the workflow name. This enables the user to distinguish between multiple instances of the same workflow.

Date Time Format

This option applies only to the file XML and the log. The API does not use this option.

XML file DateTime format

▼ yyyy-MM-dd HH:mm:ss ▼

Duplicate Task Names

Some input formats can contain duplicate task names; for example, when a task is used in multiple workflows (jobstreams). For these situations a "duplicate policy" can be applied.

Duplicate task names

Do Not Allow
 Allow, Rename

Do Not Allow	Store only unique names. Duplicates get rejected and only the first found occurrence is used.
Allow, Rename	Store every duplicate with a dup# suffix.

Both options should generate a collection of unique names. Consider the following sample input:

1	taskname	taskdescription	command	user	agent
2	job4	text4	mkdir \${dir}	user1	hostB
3	job1	text1	ls	user1	hostA
4	job2	text2	ls -a	user1	hostA
5	job3	text3	ls -la	user1	hostB
6	job4	text4	mkdir	user1	hostB
7	job1	text1	ls	user1	hostA
8	job2	text2	ls -a	user1	hostA
9	job3	text3	ls -la	user1	hostB
10	job1	text1	ls	user1	hostC
11	job1	text1	ls	user1	hostC
12	job1	text1	ls	user1	hostC
13	job1	text1	ls	user1	hostC
14	job5	text5	rm	user5	hostA

This input file will produce different lists according to the duplicate policy.

Do Not Allow	Allow, Rename
job4	job4
job1	job1
job2	job2
job3	job3
job5	job4_dup1
	job1_dup1
	job2_dup1
	job3_dup1
	job1_dup2
	job1_dup3
	job1_dup4
	job1_dup5
	job5

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 ▼ Create environment variables

Column: variable value ▼


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="background-color: #ffffcc; padding: 5px; border: 1px solid #ccc;">  Note Note! The column DropDown-boxes are not filled yet! </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.

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	
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
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:

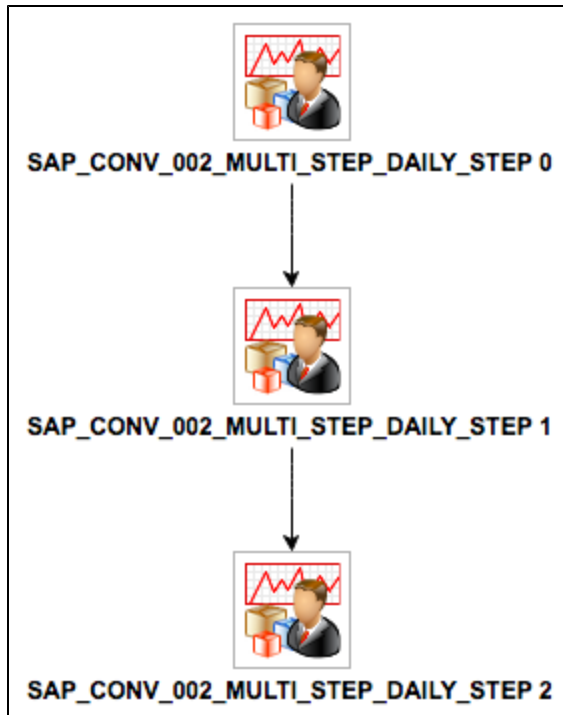
```
/* Job Header statement. */
JOBNAME = "SAP_CONV_002_MULTI_STEP_DAILY"

/* 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:

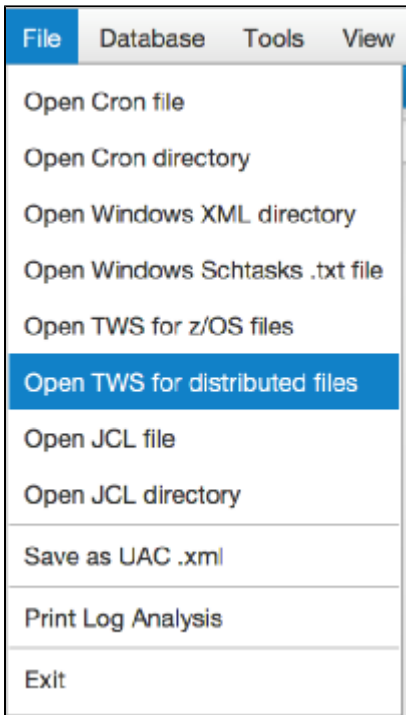
1	<ol style="list-style-type: none"> 1. Save as UAC .xml. 2. Import all XML files.
2	<ol style="list-style-type: none"> 1. Save as UAC .xml. 2. Load only scripts and credentials. 3. Use UAC API Post.

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.

Enter your input files

Variables	/Users/jsb/Documents/testdata/TWSd/sb/20180601/vartable.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 \${ }

Name of SAP credential variable \${ }

SAP command options

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

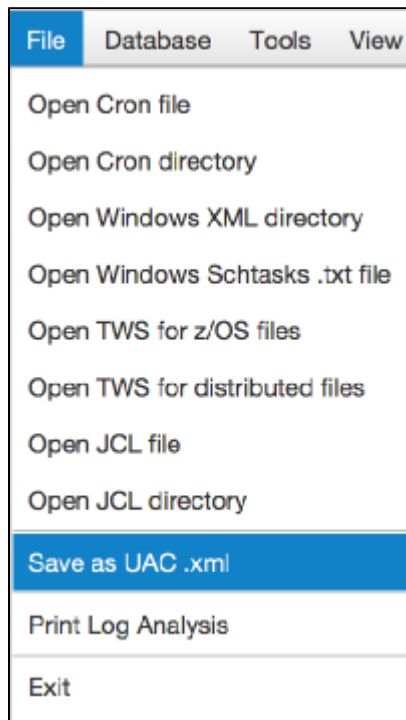
The name of SAP credential variable can be set for each transition.

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 Save as XML.

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

Output XML files to one directory	<input checked="" type="checkbox"/>
Use User-defined output directory	<input type="checkbox"/>

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

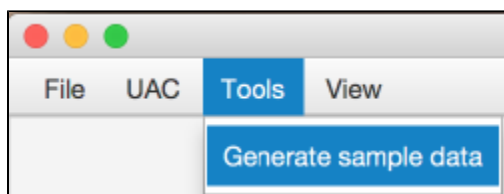
Optionally, you can request to write all XML files to one directory instead of separate directories. This enables you to import the files in UAC at one time.

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
- Entity Mapping TWS Distributed
- Entity Mapping TWS z/OS

Autosys

First, set the Autosys options.

▼ CA Autosys Options

Generate a trigger rather than a calendar for the run calendar keyword

When the run_calendar keyword is encountered, the program can perform either of two actions:

- Create a calendar with the name of the run_calendar value.
- Create a trigger with the task name value and a suffix. (-TR#)

Choose trigger when the calendar name has some descriptive text, such as "every first of month."

Keyword	Results In
case 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

Entity Mapping 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

Entity Mapping 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

Log

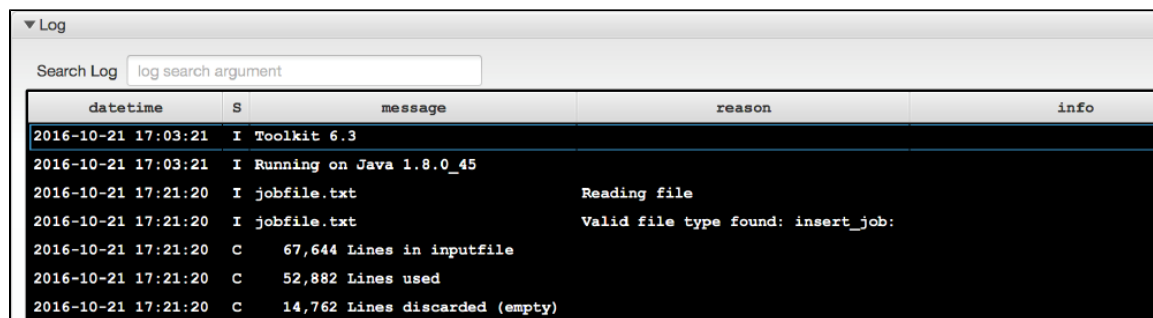
- Log

Log

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

- Date time stamp
- Message severity
- Message (usually the task name)
- Message reason
- Message info field
- Empty field where exception messages can occur

An additional a search field is provided, which searches through all rows and columns.



The screenshot shows a window titled "Log" with a search field containing "log search argument". Below the search field is a table with the following columns: datetime, S, message, reason, and info. The table contains the following rows:

datetime	S	message	reason	info
2016-10-21 17:03:21	I	Toolkit 6.3		
2016-10-21 17:03:21	I	Running on Java 1.8.0_45		
2016-10-21 17:21:20	I	jobfile.txt	Reading file	
2016-10-21 17:21:20	I	jobfile.txt	Valid file type found: insert_job:	
2016-10-21 17:21:20	C	67,644 Lines in inputfile		
2016-10-21 17:21:20	C	52,882 Lines used		
2016-10-21 17:21:20	C	14,762 Lines discarded (empty)		

[SMLRI:Enter] to search the log.

Examples:

- [] enter nothing for all messages, default.
- [SMLRI:blank-w-blank] – all warnings
- [SMLRI:blank-e-blank] – all errors
- [SMLRI:blank-r-blank] – all report lines

Right-click will show a context menu with copy options.

The log will be inserted into the file report when the user selects the file XML export option.

Job Control Language

- Overview
- How to Perform a JCL Transition
- Output

Overview

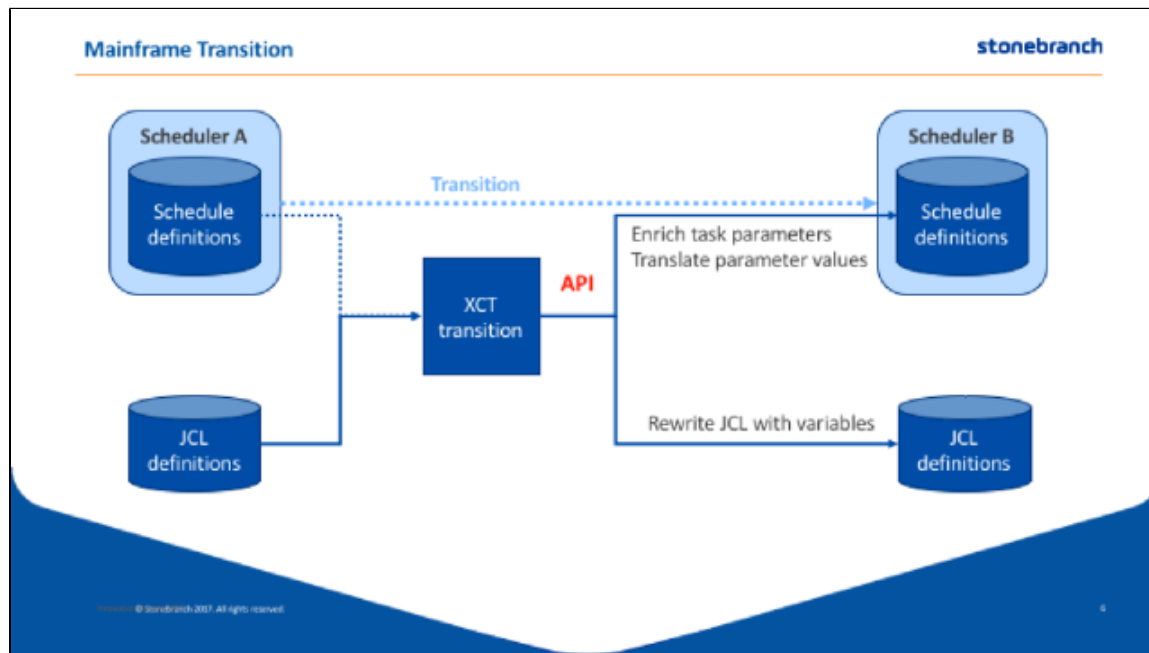
Mainframe scheduler transitions are, by nature, large and complex, but more importantly, there is always a second component that must be converted: JCL, Job Control Language.

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.

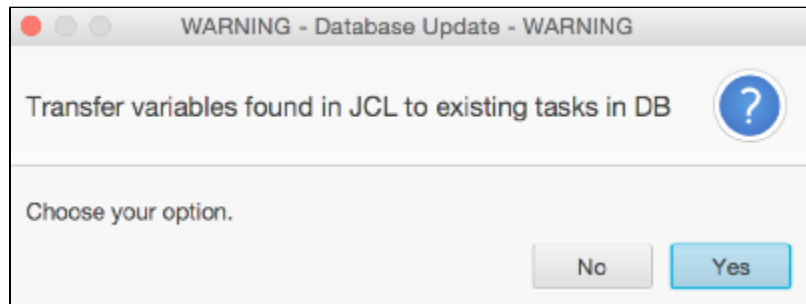


How to Perform a JCL Transition

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.

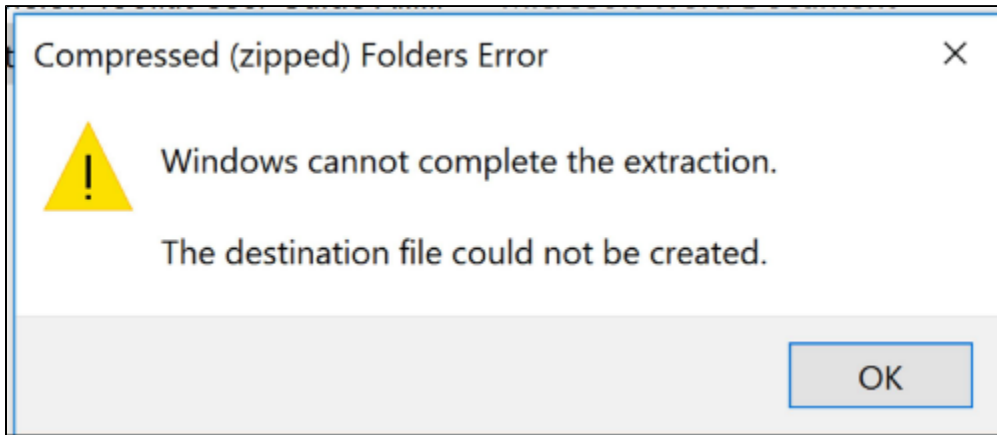
Scheduler-specific variables will be translated, as much as possible, and created as parameters on the z/OS task via the web API.

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.