# Service Now Telstra POC Implementation

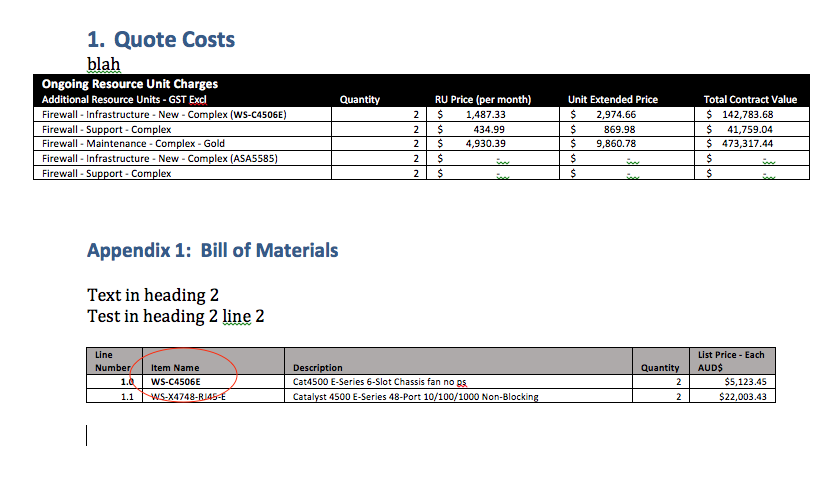
**Suggest 2 stages of implementation**

1. Service now automation using get\_data.exe in test mode. This returns specific responses based on filename, heading text provided
2. Expanded .NET get\_data.exe that actually downloads file from test sharepoint server

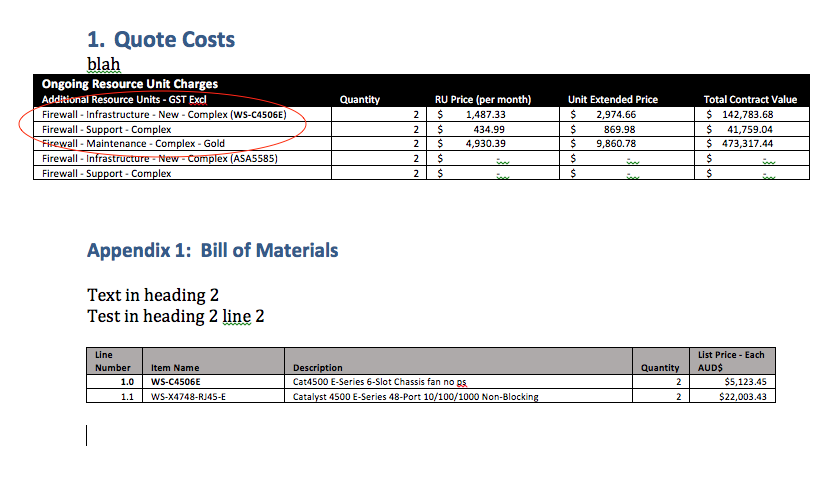
## Overview

*Manual process is currently*

1. Open file with name “SOW Filename” (SOW = Statement of Work) .docx file from a Sharepoint Site
2. Find Asset ids from Bill of Materials in SOW File : asset ids are in brackets e.g. (WS-C456OE). See example figure
3. Below. Asset IDS are always 1.0, or 2.0 in the Bill of Materials Table.



1. Once one or more Asset IDs have been found, we need to retrieve “all components for the asset”.
2. Components are listed in a separate section of the document “Quote Costs”



1. In the example above, for Asset ID, WS-C456OE, there are 2 components
   * Firewall - Support - Complex
   * Firewall - Maintenance - Complex - Gold
2. The objective is to match the Asset ID to these components and generate a user output.
3. This is repeated for all asset ids found in the Bill of Materials Table.
4. Errors must be raised if
   * No asset IDs are found
   * No information for an Asset ID is present in Quote Costs

## Architecture

The proposed architecture is

* Service Now Orchestration/Automation module control logic in javascript (?)
* MID Server running compiled .NET program to download and extract text from SOW Files.
* .NET program – runs standalone to download file based on url arg ument, and extract raw text based on the “Heading” e.g. get\_data.exe https://blah.com/folder/file.doc “Bill of Materials”
* retrieves text content from document section titled “Bill of Materials”

## So Far

We have

1. Test sharepoint site (login details below) with some test SOW files
2. Specified test scenarios (see below)
3. Written parsing logic in javascript which
   1. Parses text to extract asset ids (in brackets) from a paragraph of text
   2. For an asset ID, retrieves the components
4. Written a .NET processor (get\_data.exe) that
   1. Downloads a file from test sharepoint and extracts text from document when given a specific heading text
   2. Has a test mode to return various responses in xml for testing

## Tasks

* Implement Service Now automation logic
* Allow user to select from pre-populated list of SOW (on test sharepoint + test scenarios)
* Execute automation : most javascript logic complete
* Call MID Server which runs the .NET compiled get\_data.exe with the right parameters
* Presents results & basic error scenarios

**Stage 1**

## Pre-requisites

1. Sample / test .js & .cs (.NET) code on private github.com
2. Setup MID server
3. install .NET binary get\_data.exe (see .cs .net code, binary to be provided)
   1. get\_data.exe <testscnario\_filename> <heading> -test
4. prepopulate user interface to allow user to select from following “SOW filenames”,
   1. testdata: TestDoc\_SimpleBOM\_SimpleQuote\_singleAsset\_withtable.docx, YYY
   2. testscenarios: testsingleassetid1, Testsingleassetid2, errorheadingnotfound, ErrorBOMHasAssetButNotInQuote, ErrorBOMNoAssets

## Approach

1. Implement POC workflow
   1. User selects from pre-coded drop down list of SOW Filenames (above)
   2. Run Service Now Automation
      1. Includes executing MID Server get\_data.exe process in test mode
   3. Retrieve returned strings
   4. Implement pattern matching (see *javascript\_servicenow\_automation.js*)

## Example test workflows

Test mode for file download

### User selects SOW Filename “*testsingleassetid1”*

ServiceNow automation

1. User selects testsingleassetid1
2. Use MID Server get\_data.exe process to get Paragraph under “Bill of Materials” (call get\_data.exe “testsingleassetid1” “Bill of Materials” -test) – store in variable: *sow\_bill\_of\_materials*
3. Use MID Server get\_data.exe process to get Paragraph under “Quote” (call get\_data.exe “testsingleassetid1” “Quote” -test ) store in variable:*detail\_bom\_info*
4. Call find\_sow\_ids(sow\_bill\_of\_materials) to match all SOW asset IDs retrieved from paragraph text in Bill of Materials of document, result from test data will be [WS-C4999-E] store in say SOW\_ITEM\_IDS
5. Call lookup\_sow\_id\_description(SOW\_ITEM\_IDS, *detail\_bom\_info*) to retrieve component descriptions held in the paragraph retrieved from Quote section. Result from test data will be array with asset ID and description text ([WS-C4999-E , “Firewall-Support line 1, Firewall-Support xline2”])
6. Display result in table to user.

## User selects SOW Filename “*ErrorBOMHasAssetButNotInQuote”*

ServiceNow automation

1. User selects ErrorBOMHasAssetButNotInQuote
2. Use MID Server get\_data.exe process to get Paragraph under “Bill of Materials” (call get\_data.exe “ErrorBOMHasAssetButNotInQuote” “Bill of Materials” -test) – store in variable:
3. Use MID Server get\_data.exe process to get Paragraph under “Quote” (get\_data.exe “ErrorBOMHasAssetButNotInQuote” “Quote” -test) store in variable:*detail\_bom\_info*
4. Call find\_sow\_ids(sow\_bill\_of\_materials) to match all SOW asset IDs retrieved from paragraph text in Bill of Materials of document, result from test data will be [WS-C4999-E] store in say SOW\_ITEM\_IDS
5. Call lookup\_sow\_id\_description(SOW\_ITEM\_IDS, *detail\_bom\_info*) to retrieve component descriptions held in the paragraph retrieved from Quote section. Result from test data will be empty array as no asset information is listed in the Quote text ([WS-C4999-E , “”])
6. Display result in table to user with error as description is empty.

# Test Scenarios supported by get\_data.cs (.NET) based on filename & heading

Calling get\_data.exe <Filename> <Heading> -d will result in following outputs being generated for testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Heading** | **Description** | **Xml returned** |
| *testsingleassetid1* | *Bill of Materials* | Returns single line of asset id (WS-C4999-E) entry from SOW | <response><paragraph>1.0 WS-C4999-E Cat4500 </paragraph></response> |
| *testsingleassetid1* | *Quote* | Multi component description based on on SOW asset id above | <response><paragraph></paragraph></response> |
| *Testsingleassetid2* | *Bill of Materials* | Returns multi line with multiple asset ids WS-C4999-E & WS-c4999-F | <response><paragraph></paragraph></response> |
| *Testsingleassetid2* | *Quote* | Multi component description for each SOW Asset ID above | <response><paragraph></paragraph></response> |
| *errorheadingnotfound* | *Bill of Materials* | Returns error that Heading cannot be found in document | <response><error></error></response> |
| *errorheadingnotfound* | *Quote* | Returns error that Heading cannot be found in document | <response><error></error></response> |
| *ErrorBOMHasAssetButNotInQuote* | *Bill of Materials* | Returns Bill Of Materials paragraph text with valid asset id, but asset id missing from Quote below | <response><paragraph> 1.0 WS-C4999-E Cat4500 </paragraph></response> |
| *ErrorBOMHasAssetButNotInQuote* | *Quote* | Returns Quote paragraph text, but there are no asset ids listed that is in the previous BOM | <response><paragraph>…< paragraph></response> |
| *ErrorBOMNoAssets* | *Bill of Materials* | Returns Bill Of Materials paragraph text, but there are no asset ids in the text | <response><paragraph>blah blah blah</paragraph></response> |
| *ErrorBOMNoAssets* | *Quote* | Returns valid text inc asset id and components but nothing will match from Bill of Materials | <response><paragraph>…</paragraph></response> |

# Detailed examples

Service Now js automation code (*javascript\_servicenow\_automation.js*)

Override user inputs, with hard coded variables sow\_bill\_of\_materials & detail\_bom\_info. Lines ~170.

## Two sow asset id listed in SOW table, and described in Quote paragraph text

// example with 2 asset ids

sow\_bill\_of\_materials = "1.0 WX-C9999-X Cat4500 E-Series 6-Slot Chassis fan no ps 2 $7,461.75 \n1.1 YY-X1111-RJ45-E Catalyst 4500 E-Series 48-Port10/100/1000 Non-Bl;ocking 2 $10,449.44\n2.0 WS-c4999-F Cat4500 E-Series 6-Slot Chassis fan no ps 2 $7,461.75\n2.1 YY-X1111-RJ45-E Cataqlyst 4500 E-Series 48-Port 10/100/1000 Non-Blcoking 2 $7,461.75";

detail\_bom\_info = "Pricing Table Notes: \n \n 1. A Contract Variation will be executed between the parties to add the incremental charge to the existing ‘TWS Service 3 – Data Centre to Data Centre’ Resource Unit\n 2. This solution will be delivered under the T&Cs of the existing DNV agreement between Qantas and Telstra. A contract variation will be required to add some new Resource Units (price points), otherwise the service model will be as per the existing agreement\n \n 22\n \n Once Off Charges\n \n \n Consultancy Services ~ GST Excl Units Unit Price Extended Price\n \n \n Proramme Support / Imlementation Da - 8 hours\n 5 1135.68 $5,678.40\n \n Ongoing Resource Unit Charges\n Additional Resource Units -GST Excl Quanity RU Price(per month) Unit Extended Price Total Contract Value\n Firewall-Infrastructure-New-Complex (WX-C9999-X) 2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support line 1\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Support xline2\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Infrastructure-New-Complex (ASA5585)\t2\t$ - $ - $ -\n Firewall-Infrastructure-New-Complex (WS-c4999-F)\t2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support YYYY\t2\t$ 324.44 $ 648.88 $ 31,146.24\n Firewall-Infrastructure-New-Blah (Cat4506) $ 470.97 $ 1,883.88 $ 90,426.24\n"

# js console output

$ cd ~/sn\_poc/servicenow

$ jsc javascript\_servicenow\_automation.js

--> <item><asset\_id>WX-C9999-X</asset\_id><components>WX-C9999-X,, Firewall-Support line 1, Firewall-Support xline2</components><item>

--> <item><asset\_id>WS-c4999-F</asset\_id><components>WS-c4999-F,, Firewall-Support YYYY</components><item>

$

## Single (1) sow asset id listed in SOW table, and described in Quote paragraph text

// single asset id with description text

sow\_bill\_of\_materials = "1.0 WS-c4999-F Cat4500 ";

detail\_bom\_info = "Firewall-Infrastructure-New-Complex (WX-C9999-X) 2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support line 1\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Support xline2\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Infrastructure-New-Complex (ASA5585)\t2\t$ - $ - $ -\n Firewall-Infrastructure-New-Complex (WS-c4999-F)\t2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support YYYY\t2\t$ 324.44 $ 648.88 $ 31,146.24\n Firewall-Infrastructure-New-Blah (Cat4506) $ 470.97 $ 1,883.88 $ 90,426.24\n"

# js console output

$ cd ~/sn\_poc/servicenow

$ jsc javascript\_servicenow\_automation.js

--> <item><asset\_id>WS-c4999-F</asset\_id><components>WS-c4999-F,, Firewall-Support YYYY</components><item>$

## Single (1) sow asset id listed in SOW table, but not described in quote

// single asset id with description text

sow\_bill\_of\_materials = "1.0 WS-c4999-F Cat4500 ";

// detail\_bom\_info = "Firewall-Infrastructure-New-Complex (WX-C9999-X) 2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support line 1\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Support xline2\t2\t$ 324.44 $ 648.88 $ 31,146.24\nFirewall-Infrastructure-New-Complex (ASA5585)\t2\t$ - $ - $ -\n Firewall-Infrastructure-New-Complex (WS-c4999-F)\t2 $ 1,837.33 $ 3,674.66 $ 176,383.68\n Firewall-Support YYYY\t2\t$ 324.44 $ 648.88 $ 31,146.24\n Firewall-Infrastructure-New-Blah (Cat4506) $ 470.97 $ 1,883.88 $ 90,426.24\n"

detail\_bom\_info = "blah\nblah\n1.0 XXXXXX\thello there after XXXX\tcell2\tcell3\n something else\n something else\n2.0 xxxxxxx nothing here\nblah\nblah\n3.0 YYYYYY\tthis is text after YYYYYY\tcell2.x\n";

# js console output

$ cd ~/sn\_poc/servicenow

$ jsc javascript\_servicenow\_automation.js

--> <item><asset\_id>WS-c4999-F</asset\_id><components>,</components><item>

Note in ServiceNow this should raise an error as components is empty.

# Test Data

Test sharepoint server

<https://sntestportal.portalfront.com>

username: [bforrest@kpmg.com.au](mailto:bforrest@kpmg.com.au)

password: mypassw0rd+

All documents in <https://sntestportal.portalfront.com/Shared%20Documents>

## Documents

|  |  |  |
| --- | --- | --- |
| Filename | BOM | Quote Detail |
| TestDoc\_SimpleBOM\_SimpleQuote\_singleAsset\_withtable.docx | In “Appendix 1: Bill of Materials”  WS-C4506E | In “1. Quote Costs”  Firewall - Infrastructure - New - Complex (WS-C4506E)  Firewall - Support - Complex  Firewall - Maintenance - Complex - Gold  Firewall - Infrastructure - New - Complex (ASA5585)  Firewall - Support - Complex |

# Developer Notes

.NET program requires PIA / XML SDK

for 2012 .NET: <https://github.com/OfficeDev/Open-XML-Package-Editor-Power-Tool-for-Visual-Studio>