More Meaningful Names for Web Service parameter fields in WSDL

Raghunath Daita
Senior Software Engineer
Raghunath Daita

CA Technologies, Senior Software Engineer

CA 2E has evolved, as of r8.5, to enable users to expose business logic built using CA 2E to the outside world in the form of Web Services. However, due to existing limitations in the generation of names for components in CA 2E, the names of parameters on the resulting Web Service interface are not very user friendly and easily identifiable. During this session, we will show you the problem in brief and importantly walk you through a PoC that has been carried out to enable us to work with names from the model definitions rather than generated names on the Web Service interface and thereby simplify Web Service client development.
Agenda

- Brief Overview of the Problem
- Necessity of the solution
- Customer Response to the Problem
- Identified Process flow as part of PoC
- Implementation of PoC
- Demo
- Q&A
This presentation was based on current information and resource allocations as of April 2013 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA’s sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA’s general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation “as is” without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.
Consider a basic Arithmetic Operations Web Service, which has four operations
- addition, subtraction, multiplication and division
Invocation of Web Service using i5/OS Web services test client

As can be seen above, the operation names are not very user-friendly.
Brief Overview of the Problem

- Parameters for an operation of the Web Service

As can be seen above, the parameter names are not very user-friendly.
Problem is not very prominent for simple Web Services

Problem becomes very pronounced when used in practical scenarios with large number of parameters, say access paths/arrays with large number of fields

Necessitates a solution for overcoming the problem and having meaningful names on the Web Service interface, instead of the generated names from the model
Customer Response to the problem

- Customer requirements raised around this problem in the past
  - DAR Requests
    - 18180410-1 IMPROVE WSDL GENERATION
    - 18182359-1 WEB SERVICE NAMES
- Idea posted by Mathew Morris on the Idea Wall
- Most voted Idea on the Idea Wall.
- Mathew Morris from ‘desynit.com’ also mentioned about this problem in a blog. However, it looks like this blog no longer exists.
Disclaimer: - “More Meaningful Names for Web Service parameter fields in WSDL” is not a current feature of CA 2E. It is currently on our product backlog.

During the course of this presentation, we are only attempting to demonstrate a PoC that we have carried out to generate meaningful names from model object definitions on web services instead of generated names.
As part of PoC, we attempted to implement the following solution.

- Extract the generated PCML from the *SRVPGM/*PGM object
- Examine all the bound *MODULE objects containing PCML data
- Parse the PCML and cross-reference the field/function names against the model to find "meaningful" model object names
- Create a PCML file in the IFS with the updated information
- Provide this updated PCML file at the time WS Deployment instead of the PCML embedded in the *MODULE object(s)

This creates a WSDL/Web Service interface with "meaningful" model object names instead of 2E generated names.
Following Command Level Changes have been introduced

- YCRTWS – New command parameter added
YPRCPCML – New command has been created

- Takes in a *SRVPGM/*PGM object having *MODULES and creates an improved PCML file having “meaningful” names.
New Model Values have been created

- YPCMTYP – PCML Naming Type - *SYS, *MDL
- YPCMDIR – Location to store parsed PCML
The model object name of any TOP-LEVEL (e.g. defined on EDIT FUNCTION PARMAMETERS panel) parameter field converted into a more informative format in the final “meaningful” name

- \( p_{\langle \text{seqnbr(singledigit)} \rangle}_{\langle \text{entity_type} \rangle}_{\langle \text{entity_name} \rangle}_{\langle \text{how_passed} \rangle}_{\langle \text{MIAP} \rangle} \)

- E.g. Field Customer passed as the 2nd parameter, might translate into \( p_{2}_FLD_{\text{Customer}}_{FLD} \)

- E.g. *ARRAYS file Customer-Array passed as RCD as the 3rd parameter, might translate into \( p_{3}_ARR_{\text{Customer-Array}}_{RCD} \) etc.

Also this ensures that the order of TOP-LEVEL parameters in the 2E function matches that with the order on PCML/WSDL.

Any unsupported characters in the PCML are replaced by an underscore (“_”) in modified PCML.
After the changes, the operations on a Web Service come up as shown below.

As can be seen above, the operation names do have meaningful names.
After the changes, the parameters for an operation come up as shown below.

As can be seen above, the parameter names do have meaningful names.
After the changes, the result of an operation invocation comes up as shown below.

As can be seen above, the parameter names do have meaningful names.
Demo
Thank You

Note:- Please fill out the evaluation forms at the end of the session.