

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

FOR INFORMATION PURPOSES ONLY

Terms of this presentation

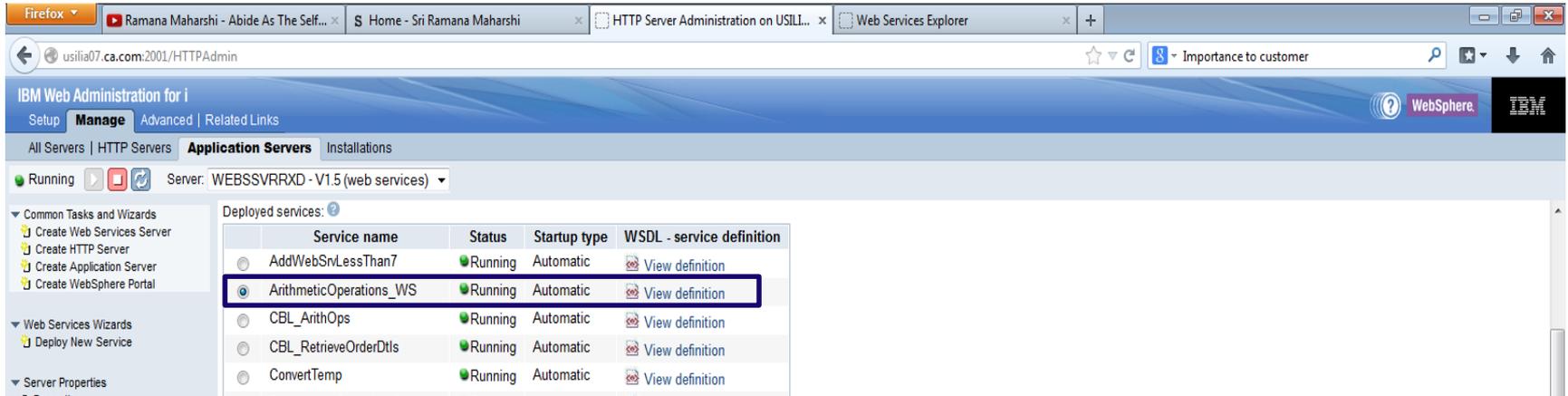
This presentation was based on current information and resource allocations as of April 2013 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations 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.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. CA confidential and proprietary. No unauthorized copying or distribution permitted.

Brief Overview of the Problem

- Consider a basic Arithmetic Operations Web Service, which has four operations
 - addition, subtraction, multiplication and division



The screenshot shows the IBM Web Administration console for a WebSphere environment. The browser address bar indicates the URL is `usilia07.ca.com:2001/HTTPAdmin`. The console displays the 'Application Servers' section for the server 'WEBSSVRRXD - V1.5 (web services)'. A table of 'Deployed services' is visible, with the 'ArithmeticOperations_WS' service highlighted. The table columns are 'Service name', 'Status', 'Startup type', and 'WSDL - service definition'. The 'ArithmeticOperations_WS' service is in a 'Running' status with an 'Automatic' startup type.

Service name	Status	Startup type	WSDL - service definition
AddWebSrvLessThan7	Running	Automatic	View definition
ArithmeticOperations_WS	Running	Automatic	View definition
CBL_ArithOps	Running	Automatic	View definition
CBL_RetrieveOrderDtls	Running	Automatic	View definition
ConvertTemp	Running	Automatic	View definition

Brief Overview of the Problem

- Invocation of Web Service using i5/OS Web services test client

The screenshot displays the i5/OS Web Services Test Client interface. The browser address bar shows the URL: `usilia07.ca.com:2006/wsexplorer/wsexplorer.jsp?wsdl=http://usilia07.ca.com:10072/web/services/ArithmeticOperations_WS?wsdl`. The interface is titled "i5/OS Web Services Test Client".

The "WSDL Binding Details" section shows the following text: "Shown below are the details for this SOAP <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints."

The "Operations" section contains a table with the following data:

Name	Documentation
uvhcxfr_XML	--
uvhbxf	--
uvhaxfr_XML	--
uvq9xfr_XML	--
uvq9xfr	--
uvhaxfr	--
uvhcxfr	--
uvhbxf_XML	--

The "Endpoints" section shows a single endpoint: `http://usilia07.ca.com:10072/web/services/ArithmeticOperations_WS.ArithmeticOperations_WSHttpSoap11Endpoint/`.

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

The screenshot shows the Web Services Explorer interface in a Firefox browser. The main window displays the configuration for a WSDL operation. The left pane shows a tree view of the WSDL structure, including the service, bindings, and operations. The right pane, titled 'Actions', contains a form for configuring the operation. The 'Endpoints' section has a dropdown menu with the selected endpoint: `http://usilia07.ca.com:10072/web/services/ArithmeticOperations_WS/ArithmeticOperations_WSHttpSoap11Endpoint/`. Below this, the 'Arguments' section is expanded to show a list of parameters. Each parameter is represented by a blue bar with a checkbox on the left and a 'Content' label on the right. The parameters are: `args0 nil?`, `WP0001 nil? string`, `WP0002 nil? string`, `WP0003 nil? string`, and `pORTN nil? string`. Each parameter has a 'Values' input field below it. At the bottom of the 'Arguments' section are 'Go' and 'Reset' buttons. The status bar at the bottom of the browser shows a JavaScript error: `javascript:checkMinOccursAndRemoveSelectedRows('tableID::inputparameters:0:0:1:0:0:1:0', 0)`.

As can be seen above, the parameter names are not very user-friendly!

Necessity of the solution

- 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

- 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.

Implementation of PoC

- Following Command Level Changes have been introduced
 - YCRTWS – New command parameter added

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
Create Web Service Instance (YCRTWS)

Type choices, press Enter.

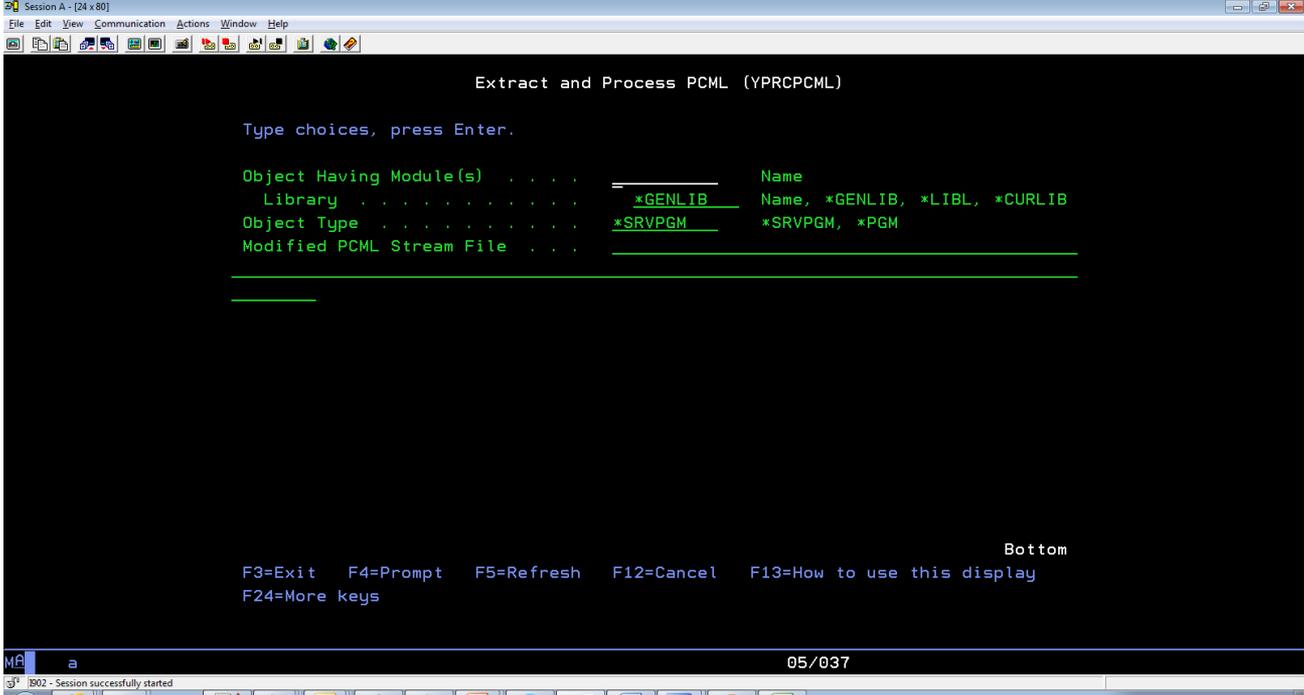
Update model? . . . . . *ADD      *ADD, *NO, *UPDINSSTS
Install to server? . . . . . *NO     *YES, *NO
2E WS model file . . . . . _____
2E WS model function . . . . . _____
Machine . . . . . *CURRENT  Name, *CURRENT
Web Services Server . . . . . _____  Character value
Web Service . . . . . _____
Program object . . . . . _____  Name
Library name . . . . . _____  Name
User profile . . . . . *USRPRF  Name, *USRPRF, *SRVID
Runtime library list . . . . . *NOCHG  Character value, *NOCHG
+ for more values
Meaningful names? . . . . . *MDLVAL *MDLVAL, *NO, *YES

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

a 05/037
IB02 - Session successfully started
```

Implementation of PoC

- YPRCPCML – New command has been created



The screenshot shows a terminal window titled "Session A - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main content is a command-line interface for "Extract and Process PCML (YPRCPCML)". It prompts the user to "Type choices, press Enter." and displays a table of options:

Object Having Module(s)	Name
Library	*GENLIB, *LIBL, *CURLIB
Object Type	*SRVPGM, *PGM
Modified PCML Stream File	

At the bottom of the terminal, there are function key instructions: "Bottom", "F3=Exit", "F4=Prompt", "F5=Refresh", "F12=Cancel", "F13=How to use this display", and "F24=More keys". The status bar at the bottom shows "05/037" and "0902 - Session successfully started".

- Takes in a *SRVPGM/*PGM object having *MODULES and creates an improved PCML file having “meaningful” names.

Implementation of PoC

- New Model Values have been created
 - YPCMTYP – PCML Naming Type - *SYS, *MDL
 - YPCMLOG – PCML Logging - *JOBLOG, *FILE, *NONE
 - YPCMDIR – Location to store parsed PCML

Implementation of PoC

- The model object name of any TOP-LEVEL (e.g. defined on EDIT FUNCTION PARAMMETERS panel) parameter field converted into a more informative format in the final “meaningful” name
 - p_<seqnbr(singledigit)>_<entity_type>_<entity_name>_<how_passed>_<MIAP>
 - E.g. Field Customer passed as the 2nd parameter, might translate into p_2_FLD_Customer_FLD
 - E.g. *ARRAYS file Customer-Array passed as RCD as the 3rd parameter, might translate into p_3_ARR_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.

Implementation of PoC

- After the changes, the operations on a Web Service come up as shown below

15/08 Web Services Test Client

Navigator

- WSDL Main
- http://usilia07.ca.com:10072/web/services/Demo_Rtv_Ord_Dtls_MngNam?wsdl
- Demo_Rtv_Ord_Dtls_MngNam
- Demo_Rtv_Ord_Dtls_MngNamSoap11Binding

Actions

WSDL Binding Details

Shown below are the details for this SOAP <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.

Operations

Name	Documentation
rtvorderdetails miap m	--
rtvorderdetails miap m XML	--

Endpoints [Add](#) [Remove](#)

Endpoints
<input type="checkbox"/> http://usilia07.ca.com:10072/web/services/Demo_Rtv_Ord_Dtls_MngNam.Demo_Rtv_Ord_Dtls_MngNamHttpSoap11Endpoint/

Go Reset

- As can be seen above, the operation names do have meaningful names.

Implementation of PoC

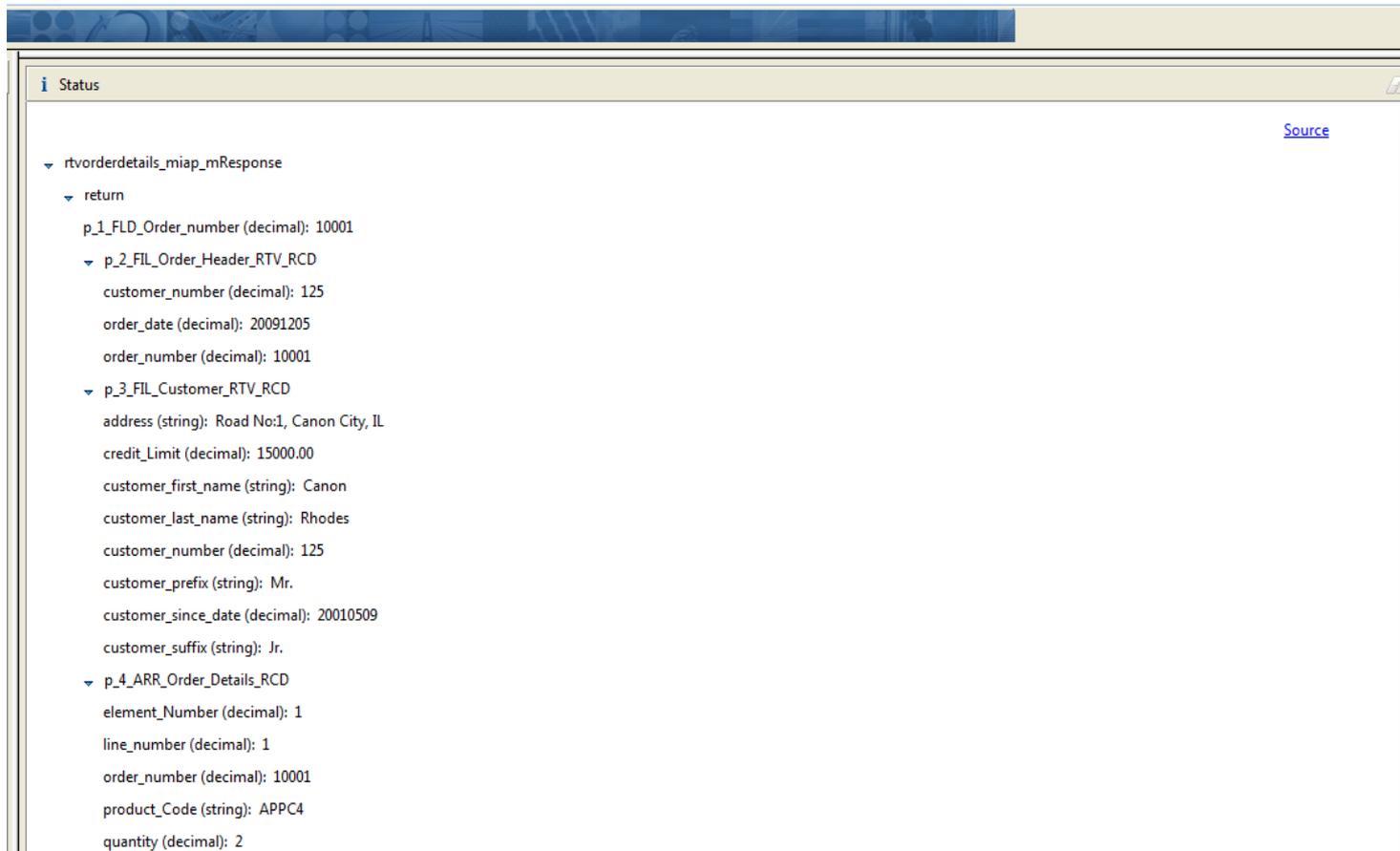
- After the changes, the parameters for an operation come up as shown below.

The screenshot displays the '15/OS Web Services Test Client' interface. On the left, the 'Navigator' pane shows a tree view of the service structure, including 'WSDL Main', 'http://usilia07.ca.com:10072/web/services/Demo_Rtv_Ord_Dtls_MngNam?wsdl', 'Demo_Rtv_Ord_Dtls_MngNam', 'Demo_Rtv_Ord_Dtls_MngNamSoap11Binding', 'rtvorderdetails_miap_m', and 'rtvorderdetails_miap_m_XML'. The 'rtvorderdetails_miap_m' operation is selected. The main 'Actions' pane shows the configuration for this operation. It includes a dropdown for 'args0 nil?' with 'Add' and 'Remove' links. Below this, there are several parameter rows, each with a 'Content' field and a 'Values' field. The parameters are: 'p 1 FLD Order number nil? decimal Add Remove', 'p 2 FIL Order Header RTV RCD nil? Add Remove', 'p 3 FIL Customer RTV RCD nil? Add Remove', and 'p 4 ARR Order Details RCD nil? Add Remove'. The 'return Code nil? string Add Remove' parameter is also visible. At the bottom of the 'Actions' pane, there are 'Go' and 'Reset' buttons. The 'Status' pane at the very bottom is currently empty.

- As can be seen above, the parameter names do have meaningful names.

Implementation of PoC

- After the changes, the result of an operation invocation comes up as shown below.



The screenshot shows a web application interface with a status bar at the top. Below the status bar, there is a tree view of a JSON response. The root node is 'rtvorderdetails_miap_mResponse', which contains a 'return' object. This object has four main sections: 'p_1_FLD_Order_number', 'p_2_FIL_Order_Header_RTV_RCD', 'p_3_FIL_Customer_RTV_RCD', and 'p_4_ARR_Order_Details_RCD'. Each section contains various fields with their values and data types.

```
rtvorderdetails_miap_mResponse
├── return
│   ├── p_1_FLD_Order_number (decimal): 10001
│   ├── p_2_FIL_Order_Header_RTV_RCD
│   │   ├── customer_number (decimal): 125
│   │   ├── order_date (decimal): 20091205
│   │   ├── order_number (decimal): 10001
│   │   └── p_3_FIL_Customer_RTV_RCD
│   │       ├── address (string): Road No:1, Canon City, IL
│   │       ├── credit_limit (decimal): 15000.00
│   │       ├── customer_first_name (string): Canon
│   │       ├── customer_last_name (string): Rhodes
│   │       ├── customer_number (decimal): 125
│   │       ├── customer_prefix (string): Mr.
│   │       ├── customer_since_date (decimal): 20010509
│   │       ├── customer_suffix (string): Jr.
│   │       └── p_4_ARR_Order_Details_RCD
│   │           ├── element_number (decimal): 1
│   │           ├── line_number (decimal): 1
│   │           ├── order_number (decimal): 10001
│   │           ├── product_code (string): APPC4
│   │           └── quantity (decimal): 2
```

- As can be seen above, the parameter names do have meaningful names.

Demo



Q&A



Thank You

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



2E/PLEX

plex2e.com

2013 WORLDWIDE DEVELOPER CONFERENCE