MCL-Technologies

Sample "Book Catalog" with Web Service Calls – Project Notes

Contents

Overview

The purpose of this document is to provide an explanation for certain screens/related options concerning this application sample and to offer a few tips on *MCL-Designer V4* application development.

Project Description

The “Book Catalog…” application displays a list of books on a screen. The user is able to view that list and select an entry to check for more detailed information about the selected book.
The book list and further information is provided by a web service so the user can also check web service connectivity at any time.

This sample illustrates:

- How to call a web service for specific information.
- How to parse/store the retrieved information into a data file and variables.
- How to use an “Advanced List Box” to display the parsed information.

The Web Service

The web service being called by the “Book Catalog” application is a RESTful service developed by *MCL-Technologies,* solely for demonstration purposes.
Enter the following URL in your Internet browser to access it:

http://spw1.mcl4e.com:8080/bookservicecatalog/

The service provides 4 endpoints, each one returning specific information:

- **ECHO** – used to check the service’s availability, it echoes the string you define in the endpoint’s parameter.
  Ex: http://spw1.mcl4e.com:8080/bookservicecatalog/echo?name=william
  In this case, the REST service will return “william”

- **BOOK LIST –** returns a list of books in (XML or JSON format).
  Ex: http://spw1.mcl4e.com:8080/bookservicecatalog/books?mediaType=xml
  Ex: http://spw1.mcl4e.com:8080/bookservicecatalog/books?mediaType=json

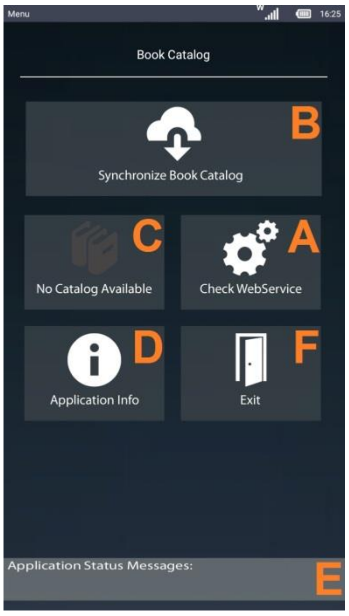
- **BOOK DETAILS –** returns detailed information based on a book’s ISBN (XML or JSON format).
  Ex: http://spw1.mcl4e.com:8080/bookservicecatalog/books/isbn/978-0439708180?mediaType=xml
  In this case, the REST service returns information in XML format of the book with the defined ISBN - title, author, price, and description.
  Ex: http://spw1.mcl4e.com:8080/bookservicecatalog/books/isbn/978-0439708180?mediaType=json
  In this case, the REST service returns information in JSON format of the book with the defined ISBN - title, author, price, and description.

Application Workflow

S:Menu

USER Perspective – First, the operator should check service availability by clicking the “Check WebService” button. Once he views the “Server Reached” status message, the operator must get the catalog information by pressing the “Synchronize Book Catalog” button. As soon as the status message confirms a successful synchronization (“Catalog Data Received”) the operator will be able to click the corresponding button to view the book catalog.

This screen includes several controls with different purposes – the call for a specific REST service, the parsing and display of information that is returned by the REST service and the redirection of application workflow:



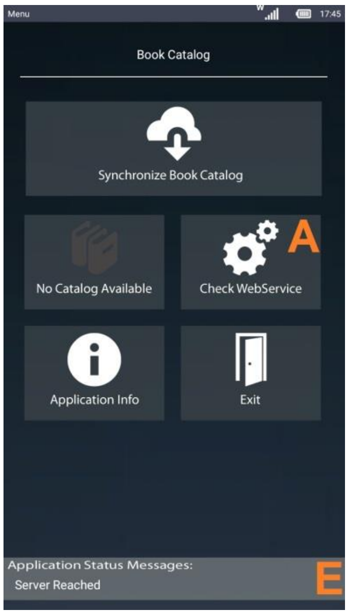
Button_Check_Service **(A)** [“Check WebService”]– This “Display Image” control calls the global procedure in charge of calling the *Echo Service* (“WebService_Call_Check_Service”).

The “Variable Assign” and “Timer” processes create/manage the status messages viewed by the operator on the screen’s footer **(E)**.

Global Procedure

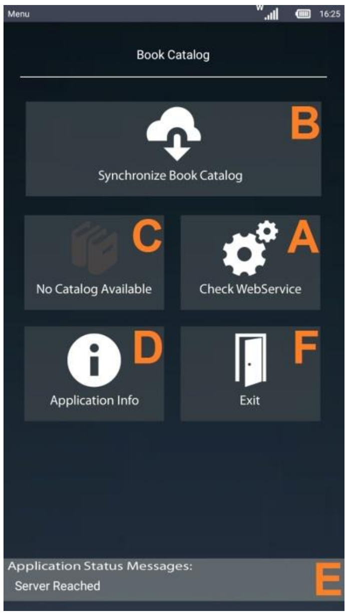
The included routine (“R:Check_Service”) starts with a “Variable Assign” process to assign the intended web service’s endpoint (http://spw1.mcl4e.com:8080/bookservice/echo) to the “P_Url” variable which is used in the “Web Service Call & Parse” process.

The routine is used to call the defined REST service but also to redirect the application’s workflow.
A successful outcome assigns the value “OK” to the variable “P_Check_Status” (the global procedure’s “param out”) and returns the execution flow to “Button_Check_Service **(A)**” (using a “Go to: <End of Routine>” process).
A *Comm* error or *HTTP* error leads to the corresponding label (“Set Label” process) – each label includes a “Message Box” with the appropriate error message, a “Variable Assign” process to assign the “KO” value to the “P_Check_Status” variable (the global procedure’s “param out”) and a “Go to: <End of Routine>” process to return to the “Actions” tab of *Display Image* **(A)**.



The “Test & Branch” process uses the “G_status” variable (the “param out” of the “Call Global Procedure“ process) to check if the web service call was successful (leading to the “WEBSERVICE_OK” label) or not (executing label “WEBSERVICE_KO”).
The “WEBSERVICE_OK” label uses the “Variable Assign” process to assign the appropriate text to the “G_status_msg” variable which is displayed on the screen’s footer **(E)** and a “Go to: <End of Action>” process to conclude the execution flow.
The “WEBSERVICE_KO” label also assigns a text to the “G_status_msg” variable to be displayed on the screen’s footer **(E)**.

Button_Sync_Catalog **(B)** ["Synchronize Book Catalog"]– This *Display Image* control calls the global procedure in charge of getting a list of books in XML format ("WebService_Call_Catalog_Data").



Global Procedure

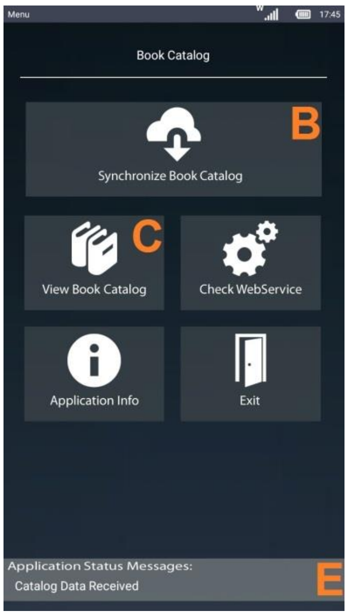
Includes a routine ("R:BookCatalogWS") that starts with a "File Delete Record" process prior to the web service request to ensure that only the current web service response is stored in the intended data file ("BookList.DAT").

The intended web service endpoint is assigned to the "P_Url" variable (http://&1A/bookservice/bookListXML/) which is, then, used within the "Web Service Call & Parse" process.

The "Web Service Call & Parse" process parses the response into the fields of the defined data file ("BookList.DAT") and redirects the workflow – if the request is successful, the workflow exits the global procedure and returns to "Button_Sync_Catalog **(B)**" (via a "Go to: <End of Routine>" process) **OR** in case of error, the application continues into the intended label (created with a "Set Label" process) and, once the "P_Check_Status" variable (the global procedure's "param out") is assigned the corresponding value, the "Go to: <End of Routine>" process is used to return to "Button_Sync_Catalog **(B)**".

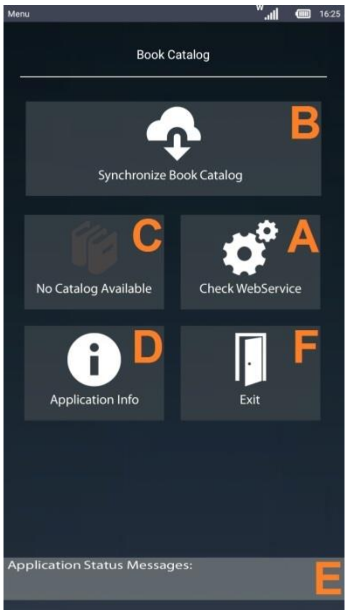
The value contained in the variable "G_status" (the "param out" of the "Call Global Procedure" process) is used in the "Test & Branch" process.

An "OK" value within the variable leads to the "WEBSERVICE_OK" label – As a result, a success message is displayed on the footer **(E)** and button **(C)** is activated (**ex:** "No Catalog Available" becomes "View Book Catalog"). This activation is executed with the use of two "Display Image" controls, one on top of the other and the "Set State" processes which enable/disable and hide/show the intended "Display Image" control after a successful synchronization.
Any other value continues the workflow to the "WEBSERVICE_KO" label – The variable being assigned an error message, by the "Variable Assign" process ("Set G_status_msg" variable), is displayed on the screen's footer **(E)**.



Button_Display_Catalog_Disabled ["No Catalog Available"] **(C)** **+**
Button_Display_Catalog_Enabled ["View Book Catalog"] **(C)**

These *Display Image* controls are overlapped. Prior to the book catalog synchronization, "Button_Display_Catalog_Disabled" is the visible control. It contains a "Message Box" process that is triggered if the user presses this button before synchronizing the book catalog **(B)**.
Once the synchronization is executed with success, the *Display Image* underneath ("Button_Display_Catalog_Enabled") becomes visible while the previously visible control is hidden (via the "Set State" processes included in the "Button_Sync_Catalog" **(B)** control. The newly activated "Display Image" contains a "Variable Assign" process to ensure that the "G_status_msg" variable is empty again –removing unnecessary messages from the screen's footer **(E)** – and a "Go Sub" process to call the intended screen - *S: Catalog*. See *S:Catalog*.



Button_App_Info (D) [“Application Info”] – Includes a “GoSub” process that calls the *S:Application_Info* screen. See *S:Application_Info*

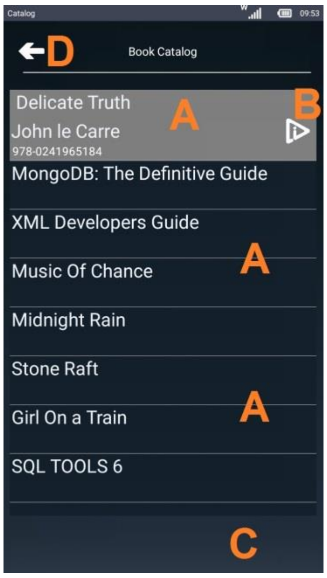
Footer_Banner (E) [“Application …”] + **Display_Text_App_Message (E)**
The screen’s footer includes a “Display Image” (visual purpose) and a “Display Text” that shows the content of the “G_status_msg” variable.

Button_Exit (F) [“Exit”] - includes a “Go to: <Exit>” process to close the application.

S: Catalog

USER Perspective – The operator views a list of book titles and if he presses a book title, the row expands to display the selected book’s author and ISBN. He will also be able to get more information on the book if he presses ▷.

This screen includes 2 *Display Text* controls (the one on the header displays the word “Catalog” and the other on the footer contains a variable **(C)**), a *Display Image* **(D)** with an arrow to return to the previous screen (*S: Menu*) and an *Advanced List Box* **(A)**.



Book_List **(A)** – The *Advanced List Box* displays the information returned by the REST service that was called with the “WebService_Call_Book_Details” global procedure and stored in the “Booklist.DAT” data file.

The “Variable Assign” process removes any previous text from the screen’s footer (shown in “Display_Text_App_Message (**C**)”) by emptying the “G_status_msg” variable.

The style of this control is customized so it displays the data file fields that were defined in the control’s “General” tab (title, author, ISBN) in a specific order and includes a button that calls a web service that returns more details on the selected book.

- “Button Info” ▷
  This button which is contained within the *Advanced List Box* calls the “WebService_Call_Book_Details” global procedure with the corresponding *Branch* process.

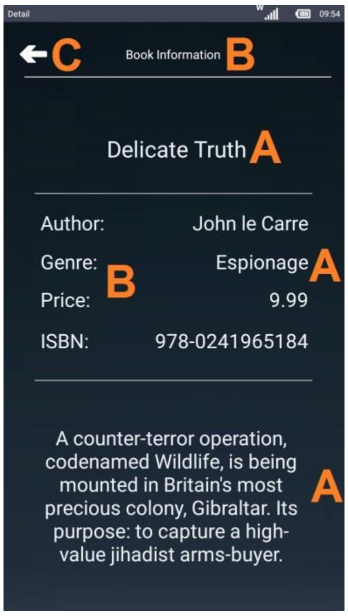
  Global Procedure
  The included routine (“R:BookDetailsWS”) opens with a “Variable Assign” process to attribute the intended web service’s URL (http://&1A/bookservice/getBook?isbn=&2A) to the variable “P_Url” which is used in the “Web Service Call & Parse” process.
  The variable “&2A” associated to the URL contains a book’s ISBN.
  The “Web Service Call & Parse” process parses the response into several variables and redirects the workflow according to the request’s success or failure.
  The “Variable Assign” process assigns the appropriate value to the “P_Check_Status” variable (a “param out” of the global procedure).

Once the application flow is returned to the “Actions” tab of the *Advanced List Box*, a “Test & Branch” process is executed to verify the value of the “G_status” variable (one of the “Call Global Procedure’s” output parameters).
Depending on the variable’s value, the application is redirected to one of the labels (created with the “Set Label” process):
“WEBSERVICE_KO” label - uses a “Variable Assign” process to attribute an error message to “G_status_msg” to be displayed on the screen’s footer **(C)**.
“WEBSERVICE_OK” label – The “G_status_msg” variable is emptied by the “Variable Assign” process and the “GoSub” process leads the application into screen *S:Detail*.

S: Detail

USER Perspective – The operator can view the details of the book he selected in the previous screen (*S:Catalog*).

This screen includes 3 *Display Line* controls for visual effect, a *Display Image***(C)** with an arrow to return to the previous screen (*S: Catalog*) and several *Display Text* **(A; B)** controls. Some display static text **(B)** and others show the values included in variables **(A)** (output parameter variables defined in the global procedure “WebService_Call_Book_Details”).



Delicate Truth

Author:	John le Carre
Genre:	Espionage
Price:	9.99
ISBN:	978-0241965184

A counter-terror operation, codenamed Wildlife, is being mounted in Britain's most precious colony, Gibraltar. Its purpose: to capture a high-value jihadist arms-buyer.

Display Text **(A)** – Each control displays the value of the defined variable. These variables were used within the global procedure “WebService_Call_Book_Details”, specifically the “Web Service Call & Parse” process, to store the REST service’s parsed response.

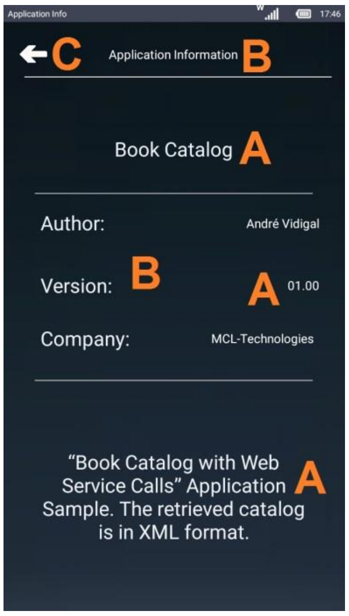
Display Text **(B)** – Each control displays the text that was entered in the corresponding *properties* window.

Display Image **(C)** – Used to return to the previous page (*S:Catalog*), it MUST have a “Go to: <Exit> ” process because the current screen – *S: Detail* - was called with a “GoSub” process (by the “Button Info” contained in the *Advanced List Box*).

S:Application Info

USER Perspective – The operator views information on the application itself (**ex:** author, version, etc.).

This screen contains 3 *Display Line* controls (visual purpose), a *Display Image*(C) with an arrow to return to the previous screen (*S: Menu*) and several *Display Text* (A; B) controls. Some display static text and others show the values included in system variables (variables containing specific information that is retrieved from the system, not written by the user).



Display Text controls(A) – Each control displays the value of the defined variable. These variables were used within the global procedure “WebService_Call_Book_Details”, specifically the “Web Service Call & Parse” process, to store the REST service’s parsed response.

Display Text controls(B) – Each control displays the text that was entered in the corresponding *properties* window.

Button_Back (C) – This *Display Image* control is used to return to the previous page (*S:Menu*). It MUST have a “Go to: <Exit> ” process because the current screen – *S: Application_Info* - was called with a “GoSub” process (in the “Button_App_Info” control).