|  |  |
| --- | --- |
| Genre | Web Option |
| Document Title | YW2EJOBX Model Implementation Example |
| Author | Raghunath Daita |
| Last Saved Date | December 27, 2012 |
|  |  |
| Revision | 1.00 |

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Last Revision By** | **Reason for Change** |
| 27/Dec/2012 | Raghunath\_Daita\_ DAIRA01 | Created Document |

NOTE 1: When changing a document, make sure to add a row in the table above, and to update last saved data and revision number in the top table.   
NOTE 2: Make sure to update the table of contents (right-click table of contents> update field > update entire field).

# Table of contents

[Table of contents 2](#_Toc344384780)

[Introduction to YW2EJOBX 3](#_Toc344384781)

[Sample Implementation of YW2EJOBX API 3](#_Toc344384782)

[Steps to implement YW2EJOBX API in a CA 2E model 4](#_Toc344384783)

[Known Problem with the YW2EJOBX API 13](#_Toc344384784)

# Introduction to YW2EJOBX

The YW2EJOBX API is shipped as part of Web Option and can be used by an application program to determine whether a specified job is a Web Option client job or not. This identification can be used to carry out user specific actions based on whether a Job is a Web Option job or a 5250 job, such as providing a specific menu system based on the type of the Job.

The API is documented in detail in the following location

Y2WEB/QTXTSRC (YW2EJOBX)

# Sample Implementation of YW2EJOBX API

A detailed implementation method for the YW2EJOBX API has not been included within the standard CA 2E manuals. To overcome this, a sample implementation of the YW2EJOBX API in a CA 2E model has been outlined in this document.

## Steps to implement YW2EJOBX API in a CA 2E model

1. Create a STR file, say “Web Option Session Data” with the following fields

**Field Name                                     Field Type       Length**

1. Web Session Identifier TXT                     16
2. Web VT Handle                           TXT                       25
3. Web Job Name                            VNM                    10
4. Web Job User                                VNM                   10
5. Web Job Number                        TXT                       6
6. Web Timeout Value                    NBR                      10, 0
7. Web Start Time                              TXT                       6
8. Web Last Transaction                    TXT                       6
9. Web End Time                                TXT                       6
10. Web Session Index                         NBR                      5, 0
11. Web Panel ID                                  TXT                       128
12. Web Job Status                               STS                       1
13. Web URL                                          TXT                      25
14. Web Script Flag                               STS                    1
15. Web Script Name                           TXT                      10
16. Web Script Label                           TXT                     10
17. Web Server Job Number             TXT                     6
18. Web Server User Profile              VNM                 10
19. Web Auto Disconnect                 STS                     1
20. Web Job Stage                               STS                     1
21. Web Signon Attempts                  NBR                    5, 0
22. Web JIT Only                                  STS                     1
23. Web Value List Lib                       VNM                 10
24. Web Value List File                      VNM                 10
25. Web Prompt Message File          VNM                 10

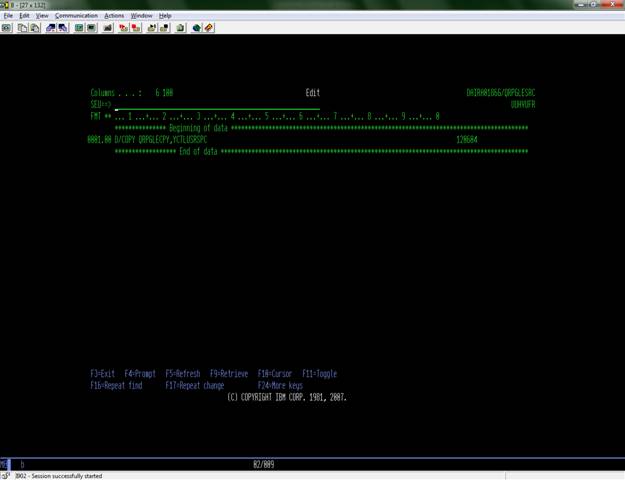
Define two separate fields as follows

1. Web Option Library VNM 10
2. Web Option Job Status STS 10

(Define a VAL condition “ACTIVE” for “Web Option Job Status” field, with a value of “\*ACTIVE”.)

1. Zoom into the EDIT FUNCTIONS screen of this STR file.
2. Create an EXCUSRSRC of type RP4, say “CPY Web Opt CTLUSRSPC\_P”. Include a copybook as shown below

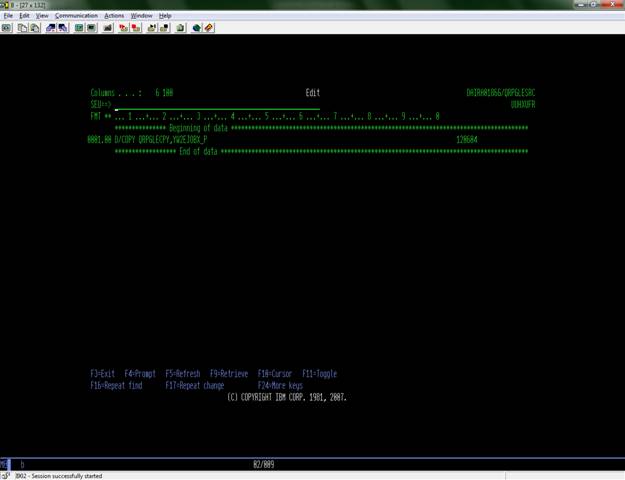
D/COPY QRPGLECPY,YCTLUSRSPC



This includes the Web Option Control User Space copybook on subsequent EXCUSRSRC inclusion, into the final function.

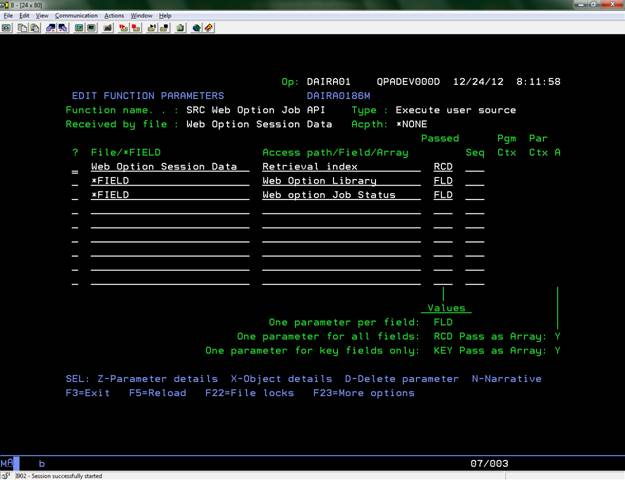
1. Create another EXCUSRSRC of type RP4, say “CPY Web Opt Job API\_P”. Include a copybook as shown below.

D/COPY QRPGLECPY,YW2EJOBX\_P



This includes the YW2EJOBX\_P copybook on subsequent EXCUSRSRC inclusion, into the final function. This copybook contains the YW2EJOBX call interface and the necessary parameter list.

1. Create another EXCUSRSRC, say, “SRC Web Option Job API”, such that it has three parameters.



Web Option Library, Web Option Job Status  are “I”nput parameters and all the fields in Web Option Session Data are “O”utput, except Web Job Name, Web Job User and Web Job Number, as shown below

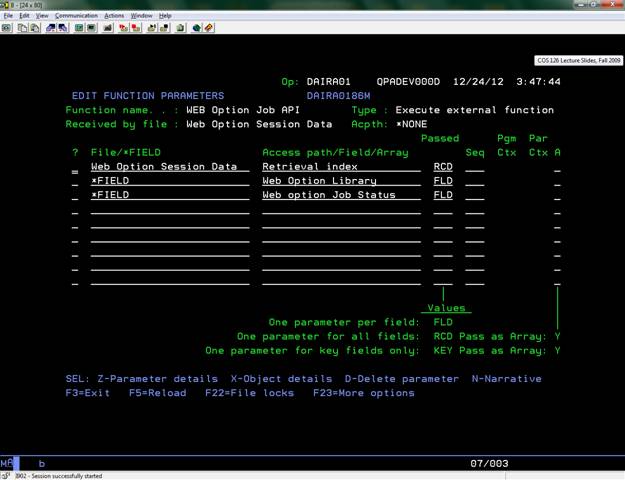


1. Within the source section of this EXCUSRSRC, code needs to be included which takes in a Job Name, Job User and Job Number and calls the “CheckWebOptionJob” procedure and subsequently displays the result of whether the job is an r8.5/r8.6 Web Option related job. A sample code which can be included in this copybook has been attached below. Note that within this source, the fields “JobName”, “JobUser” and “JobNumber” fields are being initialized with P1+corresponding implementation name for “Web Job Name”, “Web Job User” and “Web Job Number” respectively. Also, the field

“W2ELib” = P2 + corresponding Implementation name of Web Option Library and the field “JobStatus” = P3 + corresponding Implementation name of “Web option Job Status”.



1. Create an EXCEXTFUN, say, “WEB Option Job API”, such that it has three parameters as shown below.



“Web Option Library” and “Web Option Job Status” are “I”nput parameters and all the fields in Web Option Session Data are “O”utput, except Web Job Name, Web Job User and Web Job Number. Web Job Name, Web Job User and Web Job Number need to be defined with usage as “B”oth.

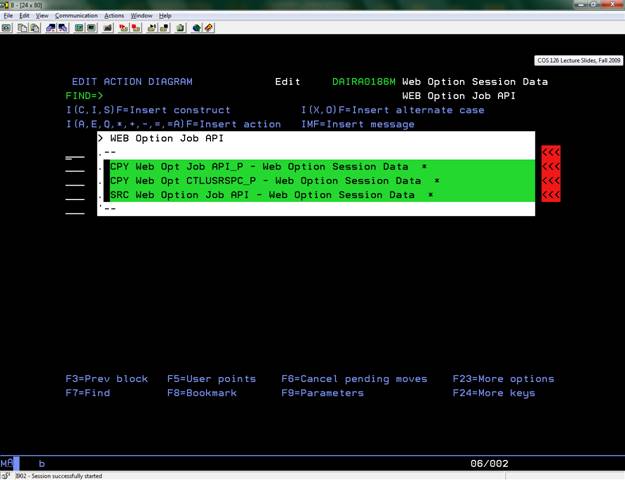
1. Within the AD of this function. Include calls to the following functions.

FIL - “Web Option Session Data”, FUN – “CPY Web Opt Job API\_P”

FIL – “Web Option Session Data”, FUN – “CPY Web Opt CTLUSRSPC\_P”

FIL – “Web Option Session Data”, FUN – “SRC Web Option Job API”

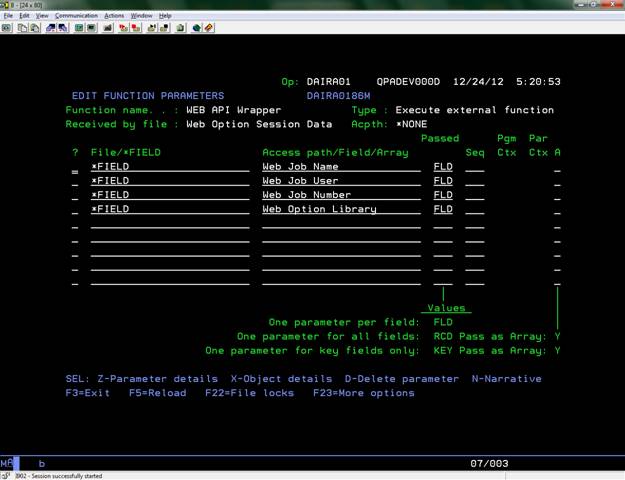
As shown below



Refer to the FIL – “Web Option Session Data” Function – “WEB Option Job API” for the parameter list to be used when including the function “SRC Web Option Job API” in this EEF.

At this stage, we have the stage set to call the YW2EJOBX API. All that we need now is to have a Wrapper in place to call this EEF, with an easy call parameter interface.

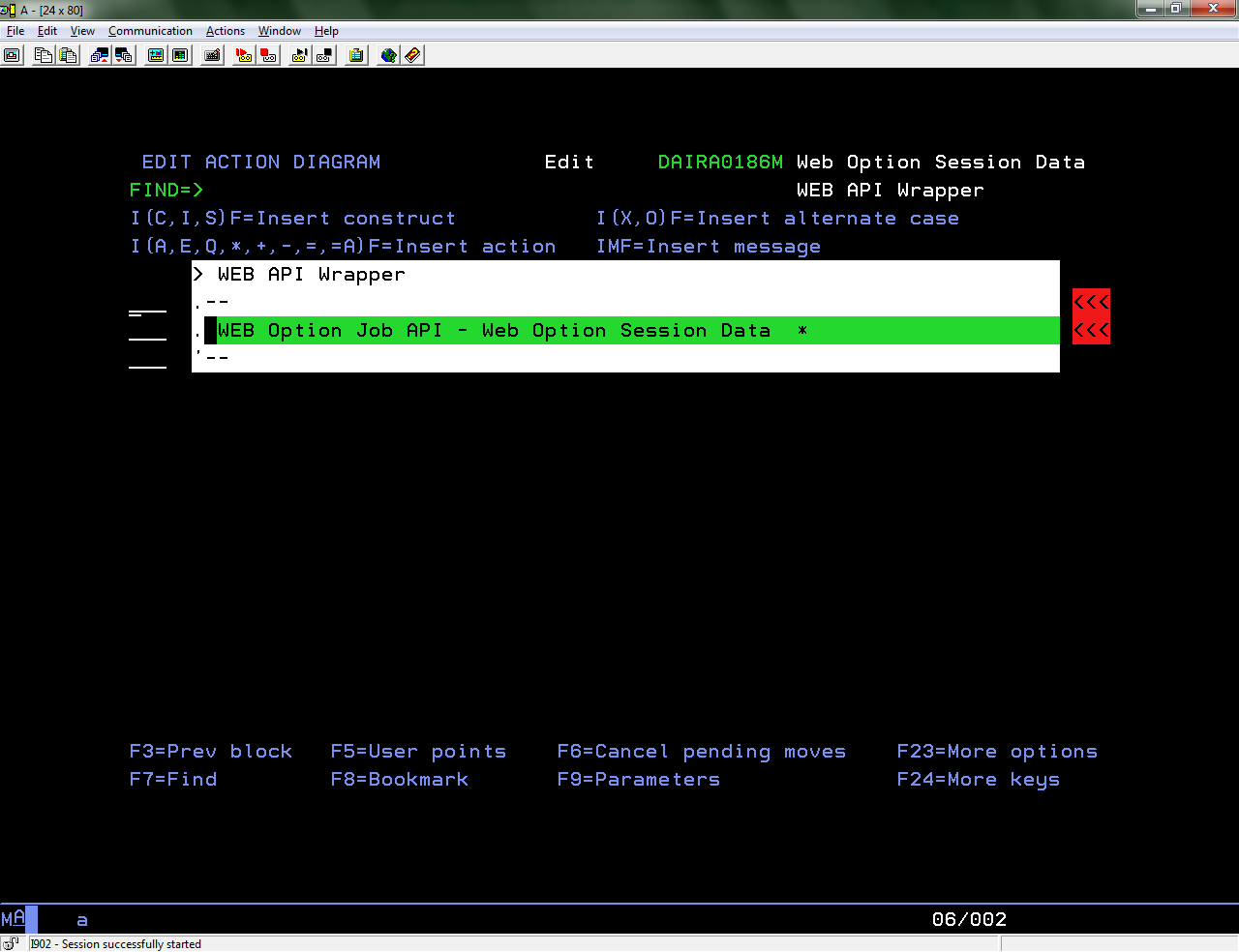
1. Create another EXCEXTFUN, say “WEB API Wrapper”, such that it has parameters as shown below.



“Web Job Name”, “Web Job User”, “Web Job Number” and “Web Option Library” are “I”nput parameters.

1. Within the AD of this function, include a call to the EEF function

“WEB Option Job API” as shown below.



Refer to the FIL – “Web Option Session Data” Function – “WEB API Wrapper” for the parameter list to be used when calling the function “WEB Option Job API” in this EEF.

1. Generate the two EEF’s “WEB Option Job API” and “WEB API Wrapper”.
2. Call the “Web API Wrapper” function with the details of a Web Option job.

An example is as follows

Web Job Name, e.g., QPADEV000J

Web Job User, e.g., DAIRA01

Web Job Number, e.g., 618314

Web Option Library, e.g., RXD86WEB

If the Job details provided are related to the Web Option version of the Web Option Library provided, then we a result gets displayed.

The output of the call to the “Web API Wrapper” function, based on the input provided, is as follows

* If the Job details provided are related to r8.5 and the Web Option library provided is related to r8.5, then “r8.5\_Web\_Option related job” is displayed.
* If the Job details provided are related to r8.6 and the Web Option library provided is related to r8.6, then “r8.6\_Web\_Option related job” is displayed.
* If the Job details provided are related to an earlier version of Web Option and a valid Web Option library name is passed, then “Web Option job” is displayed.
* If the Job details provided are not related to Web Option, then “5250 job” is displayed.

## Known Problem with the YW2EJOBX API

In all versions of YW2EJOBX API upto the current GA r8.6, the API was version incompatible i.e., when it is attempted to identify if a running job is a Web Option related job or not, the API does not identify the job correctly, when the release of Web Option associated with the job is different from that of the YW2EJOBX API.

As an example, when the r8.6 version of YW2EJOBX API is used and the job details of an r8.6 related Web Option Job and an r8.6 related Web Option library are provided as inputs, the job gets identified correctly. However, when the r8.6 version of YW2EJOBX API is used and the job details of an r8.5 related Web Option Job and an r8.5 related Web Option library are provided as inputs, the job does not get identified correctly.

Changes have been carried in YC2WEB242B to make the YW2EJOBX API as version independent. Using the modified YW2EJOBX API, it is possible to identify a Web Option related job, irrespective of the version of job related Web Option and YW2EJOBX API. In other words, after the changes in YC2WEB242B, when the modified YW2EJOBX API is used, it is possible to identify both r8.5 and r8.6 related Web Option jobs with the same modified YW2EJOBX, thereby making it version independent.