

# TR Contact Finder ActiveX Control

Version 1.2-03 Dec 2009

**(v1.0.0.45)**

How to invoke Functions, Properties and Events from Java Script

Table of Contents

**Introduction 3**

**1. Initialize AX**  **4**

**2. Count PAB Folders 5**

**3. Count PAB Contacts 6**

**4. Search PAB Contacts 7**

**5. Count NK2 Contacts 8**

**6. Search NK2 Contacts 9**

**7. Logging 11**

**8. Error Codes 11**

**Introduction**

Before going to invoke any function or property of AX we have to first setup AX in the html body within OBJECT tag as follow:

<object

id="TRContactFinder"

classid="clsid:51F5C78A-40AA-4d0d-BEDE-143DE3061919"

codebase="TRContactFinderCab.cab#version=1,0,0,45">

</object>

1. Id represents the identifier of our AX object which will be used in java script code to call AX functions.
2. Classid is the unique identifier of AX and will never change.
3. Codebase have name of cab file with specific version and placed where the html file reside.

Now we are able to invoke any function or property of AX. There is only one restriction that we should invoke InitCtrl() method before any other function or property.

Here are the detail and java script examples for all functions, properties and events exposed by the TR Contact Finder ActiveX Control.

Complete source and examples are included in default.htm file. All Error codes and their meanings are listed in section 8 of this document.

1. **Initialize AX:-**

For initializing AX to invoke functions, properties and events, we need to first call InitCtrl()method.

Methods:

InitCtrl()

Properties:

SubFolderLevel

Event Handlers:

OnError(errorCode, error)

To initialize AX this is the first function which should be called before any other AX function.

Example:

Add Java Script function for body onload event and call first AX function InitCtrl() which will initialize AX. On onload event of body formLoad function will be called in java script.

<script type="text/javascript">

var trContactFinder = null;

var trContactFinderInit = false;

function formLoad()

{

trContactFinder = document.getElementById("TRContactFinder");

if (trContactFinder == null||trContactFinder.object == null)

{

return;

}

if (trContactFinder.InitCtrl() > 0)

{

trContactFinderInit = true;

trContactFinder.SubFolderLevel = 3; //maximum can be 4

}

}

</script>

<body onload="formLoad();">

1. **Count PAB Folders :-**

For PAB folder counting following AX functions, properties and events are invloved:

Methods:

StartPABFolderCounting ()

AbortPABFolderCounting()

Event Handlers:

OnFinishPABFolderCounting(result)

OnError(errorCode, error)

To count PAB folders, StartPABFolderCounting function of AX called.

To abort PAB folder counting, AbortPABFolderCounting function of AX called.

To get result handle OnFinishPABFolderCounting

To get error if any handle OnError

Example:

<script type="text/javascript">

function getPABFolderCount()

{

if (trContactFinderInit)

{

trContactFinder.StartPABFolderCounting();

}

}

function abortPABFolderCounting()

{

if (trContactFinderInit)

{

trContactFinder.AbortPABFolderCounting();

}

}

</script>

*Note: Script for event handlers should mention the* ***FOR*** *attribute and functions name syntax is as follow:*

<script type="text/javascript" **FOR**="TRContactFinder">

function TRContactFinder::OnFinishPABFolderCounting(result)

{

alert(result);

}

function TRContactFinder::OnError(errorCode,error)

{

alert(error);

}

</script>

1. **Count PAB Contacts:-**

For PAB contacts counting following AX functions, properties and events are invloved:

Methods:

StartPABContactCounting ()

AbortPABContactCounting()

Event Handlers:

OnFinishPABContactCounting (result)

OnError(errorCode, error)

To count PAB contacts, StartPABContactCounting function of AX called.

To abort PAB contacts counting, AbortPABContactCounting function of AX called.

To get result handle OnFinishPABContactCounting

To get error if any handle OnError

Example:

<script type="text/javascript">

function getPABContactCount()

{

if (trContactFinderInit)

{

trContactFinder.StartPABContactCounting();

}

}

function abortPABContactCounting()

{

if (trContactFinderInit)

{

trContactFinder.AbortPABContactCounting();

}

}

</script>

*Note: Script for event handlers should mention the* ***FOR*** *attribute and functions name syntax is as follow:*

<script type="text/javascript" **FOR**="TRContactFinder">

function TRContactFinder::OnFinishPABContactCounting(result)

{

alert(result);

}

function TRContactFinder::OnError(errorCode,error)

{

alert(error);

}

</script>

1. **Search PAB Contacts:-**

For Searching PAB contacts following AX functions, properties and events are invloved:

Methods:

StartPABContactSearch(searchBy, startWith, pageNo)

AbortPABSearch()

Properties:

FetchUnique

PABPageSize

Event Handlers:

OnFinishPABContactsSearch(result)

OnError(errorCode, error)

To search PAB contacts, StartPABContactSearch function of AX called.

To abort PAB contact search, AbortPABSearch function of AX called.

To search unique contacts, set FetchUnique property of AX to 1 or 0

To search specific number of contacts per page, set PABPageSize property of AX

To get result handle OnFinishPABContactsSearch

To get error if any handle OnError

Example:

StartPABContactSearch is an asynchronous method and it does not return data. As soon as AX finish seacrhing or abort it fire event and send result and we handle this event in java script.

<script type="text/javascript">

function getPABContacts()

{

if (trContactFinder == null)

return;

if (trContactFinder.object == null)

return;

trContactFinder.FetchUnique = 1

trContactFinder.PABPageSize = 100;

var searchBy = "Name"; // or with Email

var startWith = "A";

var pageNo = 1;

trContactFinder.StartPABContactSearch(searchBy, startWith, pageNo);

}

function abortPABSearch()

{

if (trContactFinderInit)

{

trContactFinder.AbortPABSearch();

}

}

</script>

*Note: Script for event handlers should mention the* ***FOR*** *attribute and functions name syntax is as follow:*

<script type="text/javascript" **FOR**="TRContactFinder">

function TRContactFinder::OnFinishPABContactsSearch(result)

{

var jsonObj = $.parseJSON(result);

if(jsonObj != null)

{

var contacts = jsonObj.contacts;

if (contacts.length > 0)

{

for(var i=0;i<contacts.length;i++)

{

var contact = contacts[i]; //traverse all contacts here

}

}

}

}

function TRContactFinder::OnError(errorCode,error)

{

alert(error);

}

</script>

1. **Count NK2 Contacts :-**

For NK2 contact counting following AX functions, properties and events are invloved:

Methods:

StartNK2ContactCounting ()

AbortNK2ContactCounting()

Properties:

LookupGAL

Event Handlers:

OnFinishNK2ContactCounting(result)

OnError(errorCode, error)

To count PAB folders, StartNK2ContactCounting function of AX called.

To abort PAB folder counting, AbortNK2ContactCounting function of AX called.

To lookup GAL for exchange contacts which are not found in NK2 file, set LookupGAL value to 1 otherwise 0.

To get result handle OnFinishNK2ContactCounting

To get error if any handle OnError

Example:

<script type="text/javascript">

function getNK2ContactCount ()

{

if (trContactFinderInit)

{

trContactFinder.LookupGAL = 1; //or 0

trContactFinder.StartNK2ContactCounting();

}

}

function abortPABFolderCounting()

{

if (trContactFinderInit)

{

trContactFinder. AbortNK2ContactCounting();

}

}

</script>

*Note: Script for event handlers should mention the* ***FOR*** *attribute and functions name syntax is as follow:*

<script type="text/javascript" **FOR**="TRContactFinder">

function TRContactFinder::OnFinishNK2ContactCounting(result)

{

var jsonObj = $.parseJSON(result);

if(jsonObj != null)

{

var contacts = jsonObj.contacts;

if (contacts.length > 0)

{

for(var i=0;i<contacts.length;i++)

{

var contact = contacts[i]; //traverse all contacts here

}

}

}

}

function TRContactFinder::OnError(errorCode,error)

{

alert(error);

}

</script>

1. **Search NK2 Contacts:-**

For Searching NK2 contacts following AX functions, properties and events are invloved:

Methods:

StartNK2ContactSearch(searchBy, startWith, pageNo)

AbortNK2Search()

Properties:

FetchUnique

NK2PageSize

LookupGAL

Event Handlers:

OnFinishNK2ContactsSearch(result)

OnError(errorCode, error)

To search NK2 contacts, StartNK2ContactSearch function of AX called.

To abort NK2 contact search, AbortNK2Search function of AX called.

To search unique contacts, set FetchUnique property of AX to 1 or 0

To search specific number of contacts per page, set NK2PageSize property of AX

To lookup GAL for exchange contacts which are not found in NK2 file, set LookupGAL value to 1 otherwise 0.

To get result handle OnFinishPABContactsSearch

To get error if any handle OnError

Example:

StartNK2ContactSearch is an asynchronous method and it does not return data. As soon as AX finish seacrhing or abort it fire event and send result and we handle this event in java script.

<script type="text/javascript">

function getNK2Contacts()

{

if (trContactFinder == null)

return;

if (trContactFinder.object == null)

return;

trContactFinder.FetchUnique = 1

trContactFinder.NK2PageSize = 100;

var searchBy = "Name"; // or with Email

var startWith = "A";

var pageNo = 1;

trContactFinder.StartNK2ContactSearch(searchBy, startWith, pageNo);

}

function abortNK2Search()

{

if (trContactFinderInit)

{

trContactFinder.AbortNK2Search();

}

}

</script>

*Note: Script for event handlers should mention the* ***FOR*** *attribute and functions name syntax is as follow:*

<script type="text/javascript" **FOR**="TRContactFinder">

function TRContactFinder::OnFinishNK2ContactsSearch(result)

{

var jsonObj = $.parseJSON(result);

if(jsonObj != null)

{

var contacts = jsonObj.contacts;

if (contacts.length > 0)

{

for(var i=0;i<contacts.length;i++)

{

var contact = contacts[i]; //traverse all contacts here

}

}

}

}

function TRContactFinder::OnError(errorCode,error)

{

alert(error);

}

</script>

1. **Logging:-**

To enable AX keep log of all steps for debugging purpose set EnableLog property of AX to 1.

Properties:

EnableLog

<script type="text/javascript">

function onEnableLog(sender) //sender is checkbox

{

if (trContactFinderInit)

{

trContactFinder.EnableLog = sender.checked ? 1:0;

}

}

1. **Return Error Codes:**

Following is the list of defined error codes as per requirements in the AX.

|  |  |
| --- | --- |
| Code | Error Message |
| 101 | Outlook is not installed. |
| 102 | Outlook is not set as default mail client. |
| 103 | Access is denied. Cannot initialize TRContactFinder. |
| 104 | TRContactFinder does not support current OS. |
| 105 | TRContactFinder only support IE 6.\* and above. |
| 106 | TRContactFinder only support Outlook 2003 and Outlook 2007. |
| 107 | TRContactFinder Cannot find private folders. |
| 108 | TRContactFinder cannot find NK2 file. |
| 109 | TRContactFinder Cannot open private folders. |
| 110 | TRContactFinder is blocked by firewall. |
| 111 | TRContactFinder is not Initialized. |
| 112 | TRContactFinder does not support GetPABFolderCount method. Please call StartPABFolderCounting method |
| 113 | TRContactFinder does not support GetPABContactsCount method. Please call StartPABContactCounting method |
| 114 | TRContactFinder is in progress. To work with this function please abort the previous thread. |
| 115 | TRContactFinder NK2 Contact Reading in progress. To work with this function abort running thread |
| 116 | TRContactFinder does not support GetNK2ContactsCount method. Please call StartNK2ContactCounting |