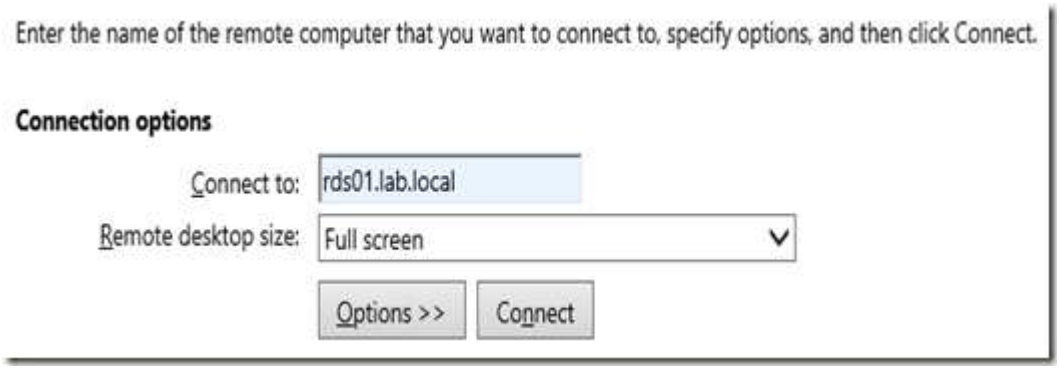
Customize RD Web Access, a drop down server list

As you might know, RD Web Access provides two different ways to allow users to connect. The tab “RemoteApp and Desktops” tab contains the Remote Apps and Desktops that are authorized to user. The tab “Connect to a remote PC” allows users to specify the destination remote client, server of farm by providing the DNS or hostname.



In some cases you might want to pre-define the hostname users have to enter. In this blog post I’ll guide you through the process of configuring a drop down list containing destinations we want users to be able to select.

STEP 1. We’ll be editing the desktops.aspx which is located in C:\Windows\Web\RDWeb\Pages\en-US\Desktops.aspx (may differ based on the language of the Server OS). Be sure to create a backup of that file first.

STEP 2. Locate the definition of the function **function GetParam(sParam, bReqd, vDefault)** and add the following function specified below that function definition. We’ll use this function to retrieve selected value of the dropdown box. We can’t use the existing GetParams function as this returns the number of the select item in de dropdown box. (Uses by for example the Remote desktop size dropdown box).

```
function GetDestination(sParam, bReqd, vDefault)
{
    var obj = document.getElementById(sParam);
    if(obj != null)
    {
        switch(obj.tagName)
        {
            case "SELECT":
                return obj.options[obj.selectedIndex].value;
                break;
            default:
                break;
        }
    }
    else
    {
        if ((bReqd) && ((vDefault == "") || (vDefault == null) || (obj == null)))
        {
            var L_ErrMsgInvalid_Text = "%ParameterName% is not a valid or available
parameter name.";  // {Placeholder="%ParameterName%"}
            var errMsgInvalid = sParam;
            errMsgInvalid = errMsgInvalid.replace("%ParameterName%", sParam);
            var retval = TSMsgBox(errMsgInvalid, vbInformation, L_sTitle_Text);
            return null;
        }
        else
        {
            return vDefault;
        }
    }
}
```

STEP 3. Replace the following code

```
<input name="MachineName" maxlength="255" id="MachineName" class="textInputField"
type="text"
                                    onfocus="updateConnectButtonState(this);"
onblur="updateConnectButtonState(this);"
```

```
                                        onkeyup="onConnectToKeyUp(this);"
onpropertychange="onConnectToPropertyChange(this);"/> With the code: <select
id="MachineName" style="width: 270px" name="MachineName">
                                            <option value="rds01.lab.local"
selected="selected">rds01.lab.local</option>
                                            <option
value="rds02.lab.local">rds02.lab.local</option>
                                            <option
value="rds03.lab.local">rds03.lab.local</option>
                                        </select>
```

STEP 4. To make sure the connect button is always available find the following string and remove the part **disabled="disabled"***<button type="button" id="ButtonConnect" name="ButtonConnect"* ***disabled="disabled"***

STEP 5. Replace the following piece of code

```
var RDPstr = "full address:s:" + GetParam("MachineName", true, "") + "\n";
```

With the code:

```
var RDPstr = "full address:s:" + GetDestination("MachineName", true, "") + "\n";
```

STEP 6. The end result should look like something below:

Upon clicking Connect a RDP session to the selected destination is launched.