

ISXEVE - HOWTO: Directional Scanner Window Interaction

```
function main()
{
    variable index:directionalscannerresult ScanResults
    variable iterator ScanResultsIterator
    variable int Counter = 1

    if (!$EVEWindow[directionalScannerWindow](exists))
    {
        EVE:Execute[OpenDirectionalScanner]
        do
        {
            waitframe
        }
        while !$EVEWindow[directionalScannerWindow](exists) && ${Counter:Inc} < 1000
        wait 15
        if (!$EVEWindow[directionalScannerWindow](exists))
        {
            echo "[DScanWindow] ERROR - Unable to Open and/or Access the DirectionalScannerWidow."
            return
        }
        Counter:Set[1]
    }

    echo "[DScanWindow] Range set to ${EVEWindow[directionalScannerWindow].Range}"
    echo "[DScanWindow] Angle set to ${EVEWindow[directionalScannerWindow].Angle}"

    ;;;;;;;;;;;;;;
    ;; SCAN
    echo "[DScanWindow] Scanning..."
    EVEWindow[directionalScannerWindow].Button[1]:Press
    do
    {
        waitframe
    }
    while ${EVEWindow[directionalScannerWindow].IsScanning} && ${Counter:Inc} < 1000
    if (${EVEWindow[directionalScannerWindow].IsScanning})
    {
        echo "[DScanWindow] ERROR - IsScanning TRUE too long"
        return
    }
    else
        Counter:Set[1]
    ;;;;;;;;;;;;;;

    EVEWindow[directionalScannerWindow]:GetScanResults[ScanResults]
    ScanResults:GetIterator[ScanResultsIterator]
    echo "[DScanWindow] Scan Finished - ${ScanResults.Used} entries:"

    if ${ScanResultsIterator:First(exists)}
    do
    {
        echo "[DScanWindow] ${Counter}. ${ScanResultsIterator.Value.Name} (ID:
${ScanResultsIterator.Value.ID})"
        echo "[DScanWindow] ${Counter}. -- Type: ${ScanResultsIterator.Value.Type}
(${ScanResultsIterator.Value.TypeID})"
        echo "[DScanWindow] ${Counter}. -- Group: ${ScanResultsIterator.Value.Group}
(${ScanResultsIterator.Value.GroupID})"
        if (${ScanResultsIterator.Value.ToEntity.ID} > 0)
        {
            ;; If ToEntity returns a value, then you have access to "Distance" as well as ALL
other members/methods of the 'entity' datatype
            echo "[DScanWindow] ${Counter}. -- Distance:
${ScanResultsIterator.Value.ToEntity.Distance}"
        }
    }
}
```

```
else
{
    ;; If ToEntity does not return a value, then the only information available for this
scan result entry is ID, Name, Group, and Type
    echo "[DScanWindow] ${Counter}. -- Distance: Unknown"
}
echo "==="
Counter:Inc
}
while ${ScanResultsIterator:Next(exists)}
}
```