

[temp.txt](#)

```
# — Config


---


$ScriptDir = Split-Path -Parent $MyInvocation.MyCommand.Path
$HostsFile = Join-Path $ScriptDir "hosts.csv"
$SitesFile = Join-Path $ScriptDir "sites.csv"
$OutputFile = "C:\inetpub\wwwroot\dns\index.html" # adjust if needed

# — Colour palette


---


$Palette =
@("#3b82f6", "#10b981", "#f59e0b", "#8b5cf6", "#ef4444", "#ec4899", "#06b6d4",
"#84cc16")

# — Load sites.csv


---


$SiteRows = Import-Csv $SitesFile
$SiteMap = @{} # prefix -> site name
$SiteColors = @{} # site name -> colour
$SiteOrder = [System.Collections.Generic.List[string]]::new()
$ColorIdx = 0

foreach ($row in $SiteRows) {
    $prefix = $row.prefix.Trim()
    $site = $row.site.Trim()
    $SiteMap[$prefix] = $site
    if (-not $SiteColors.ContainsKey($site)) {
        $SiteColors[$site] = $Palette[$ColorIdx % $Palette.Count]
        $SiteOrder.Add($site)
        $ColorIdx++
    }
}

# — DNS lookup + site matching


---


$HostRows = Import-Csv $HostsFile
$Results = [System.Collections.Generic.List[hashtable]]::new()
$SiteCounts = @{}
$ErrorCount = 0
$Total = 0

foreach ($h in $HostRows) {
    $hostname = $h.hostname.Trim()
    $name = $h.name.Trim()

    try {
        $dns = Resolve-DnsName -Name $hostname -Type A -ErrorAction
Stop
        $ip = ($dns | Where-Object Type -eq 'A' | Select-Object -First
1).IPAddress
```

```
    $parts = $ip -split '\.'
    $prefix = "$($parts[0]). $($parts[1])"
    $site = if ($SiteMap.ContainsKey($prefix)) {
$SiteMap[$prefix] } else { "Unknown Location" }
    $color = if ($SiteColors.ContainsKey($site)) {
$SiteColors[$site] } else { "#6b7280" }
    $SiteCounts[$site] = ($SiteCounts[$site] ?? 0) + 1
} catch {
    $ip = $null
    $site = "DNS Lookup Failed"
    $color = "#6b7280"
    $ErrorCount++
}

    $Results.Add(@{ Hostname=$hostname; Name=$name; IP=$ip; Site=$site;
Color=$color })
    $Total++
}

$LastUpdated = Get-Date -Format "yyyy-MM-dd HH:mm:ss"

# — Build summary cards


---


$SiteCards = ""
foreach ($site in $SiteOrder) {
    $count = $SiteCounts[$site] ?? 0
    if ($count -eq 0) { continue }
    $col = $SiteColors[$site]
    $SiteCards += @"

    <div class="card" style="border-color:${col}33">
        <div class="card-label">Hosts at site</div>
        <div class="card-value" style="color:$col">$count</div>
        <div class="card-sub" title="$site">$site</div>
    </div>
"@
}

if ($ErrorCount -gt 0) {
    $SiteCards += @"

    <div class="card" style="border-color:#ef444433">
        <div class="card-label">DNS Failures</div>
        <div class="card-value" style="color:#ef4444">$ErrorCount</div>
        <div class="card-sub">could not resolve</div>
    </div>
"@
}

# — Build table rows


---


```

```

$TableRows = ""
foreach ($r in $Results) {
    $ipHtml = if ($r.IP) {
        "<span class=`"ip`">${$r.IP}</span>"
    } else {
        "<span class=`"ip failed`">unresolved</span>"
    }
    $col = $r.Color
    $TableRows += @"

        <tr>
            <td><span class="hostname">${$r.Hostname}</span></td>
            <td><span class="friendly">${$r.Name}</span></td>
            <td>$ipHtml</td>
            <td><span class="badge" style="background:${col}22;
color:${col}; border:1px solid ${col}55">${$r.Site}</span></td>
        </tr>
"@
}

# — Write HTML

$OutDir = Split-Path -Parent $OutputFile
if (-not (Test-Path $OutDir)) { New-Item -ItemType Directory -Path
$OutDir | Out-Null }

@"
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>DNS Location Lookup</title>
<style>
    *, *::before, *::after { box-sizing: border-box; margin: 0; padding:
0; }
    body {
        font-family: system-ui, -apple-system, 'Segoe UI', sans-serif;
        background: #0f172a; color: #e2e8f0;
        min-height: 100vh; padding: 2rem 1rem;
    }
    .container { max-width: 960px; margin: 0 auto; }
    header {
        display: flex; align-items: center; justify-content: space-between;
        flex-wrap: wrap; gap: 1rem; margin-bottom: 2rem;
    }
    .header-left h1 { font-size: 1.6rem; font-weight: 700; color:
#f1f5f9; }
    .header-left p { font-size: 0.8rem; color: #64748b; margin-top:
0.2rem; }
    .summary { display: flex; flex-wrap: wrap; gap: 0.75rem; margin-

```

```
bottom: 2rem; }
.card {
  background: #1e293b; border: 1px solid #334155;
  border-radius: 0.75rem; padding: 0.9rem 1.25rem; flex: 1 1 140px;
}
.card-label { font-size: 0.7rem; text-transform: uppercase; letter-
spacing: 0.08em; color: #64748b; margin-bottom: 0.3rem; }
.card-value { font-size: 1.8rem; font-weight: 700; line-height: 1; }
.card-sub { font-size: 0.75rem; color: #94a3b8; margin-top:
0.25rem; white-space: nowrap; overflow: hidden; text-overflow:
ellipsis; }
.table-wrap { background: #1e293b; border: 1px solid #334155; border-
radius: 0.75rem; overflow: hidden; }
table { width: 100%; border-collapse: collapse; }
thead { background: #0f172a; }
th {
  text-align: left; padding: 0.75rem 1rem; font-size: 0.7rem;
  text-transform: uppercase; letter-spacing: 0.07em; color: #64748b;
font-weight: 600;
}
td { padding: 0.85rem 1rem; border-top: 1px solid #1e293b; font-size:
0.9rem; vertical-align: middle; }
tr:nth-child(even) td { background: #172033; }
tr:hover td { background: #1d2d45; }
.hostname { font-weight: 600; color: #f1f5f9; font-family: monospace;
font-size: 0.95rem; }
.friendly { color: #94a3b8; }
.ip { font-family: monospace; color: #7dd3fc; }
.ip.failed { color: #f87171; font-style: italic; }
.badge { display: inline-block; padding: 0.25rem 0.7rem; border-
radius: 9999px; font-size: 0.75rem; font-weight: 600; white-space:
nowrap; }
.hint {
  margin-top: 1.5rem; background: #1e293b; border: 1px solid #334155;
  border-radius: 0.75rem; padding: 1rem 1.25rem; font-size: 0.8rem;
  color: #64748b; display: flex; gap: 2rem; flex-wrap: wrap;
}
.hint strong { color: #94a3b8; display: block; margin-bottom:
0.25rem; }
.hint code {
  background: #0f172a; border: 1px solid #334155;
  padding: 0.1rem 0.4rem; border-radius: 0.25rem; font-family:
monospace; color: #7dd3fc;
}
@media (max-width: 600px) { th:nth-child(3), td:nth-child(3) {
display: none; } }
</style>
</head>
<body>
<div class="container">
  <header>
```

```

<div class="header-left">
  <h1>DNS Location Lookup</h1>
  <p>Last updated: $LastUpdated</p>
</div>
</header>
<div class="summary">
  <div class="card">
    <div class="card-label">Total Hosts</div>
    <div class="card-value" style="color:#f1f5f9">$Total</div>
    <div class="card-sub">across all sites</div>
  </div>
  $SiteCards
</div>
<div class="table-wrap">
  <table>
    <thead>
      <tr>
        <th>Hostname</th>
        <th>Friendly Name</th>
        <th>IP Address</th>
        <th>Location</th>
      </tr>
    </thead>
    <tbody>
      $TableRows
    </tbody>
  </table>
</div>
<div class="hint">
  <div>
    <strong>Add / remove hosts</strong>
    Edit <code>hosts.csv</code> &mdash; columns:
<code>hostname</code>, <code>name</code>
  </div>
  <div>
    <strong>Add / change sites</strong>
    Edit <code>sites.csv</code> &mdash; columns: <code>prefix</code>
(first two octets), <code>site</code>
  </div>
  <div>
    <strong>Auto-refresh</strong>
    Page regenerated by Task Scheduler &mdash; last run shown above
  </div>
</div>
</div>
</body>
</html>
"@ | Set-Content -Path $OutputFile -Encoding UTF8

Write-Host "Written to $OutputFile at $LastUpdated"

```

From:

<https://jdlewis.net/> - **John D. Lewis**

Permanent link:

<https://jdlewis.net/doku.php?id=temp&rev=1772216159>

Last update: **2026/02/27 18:15**

