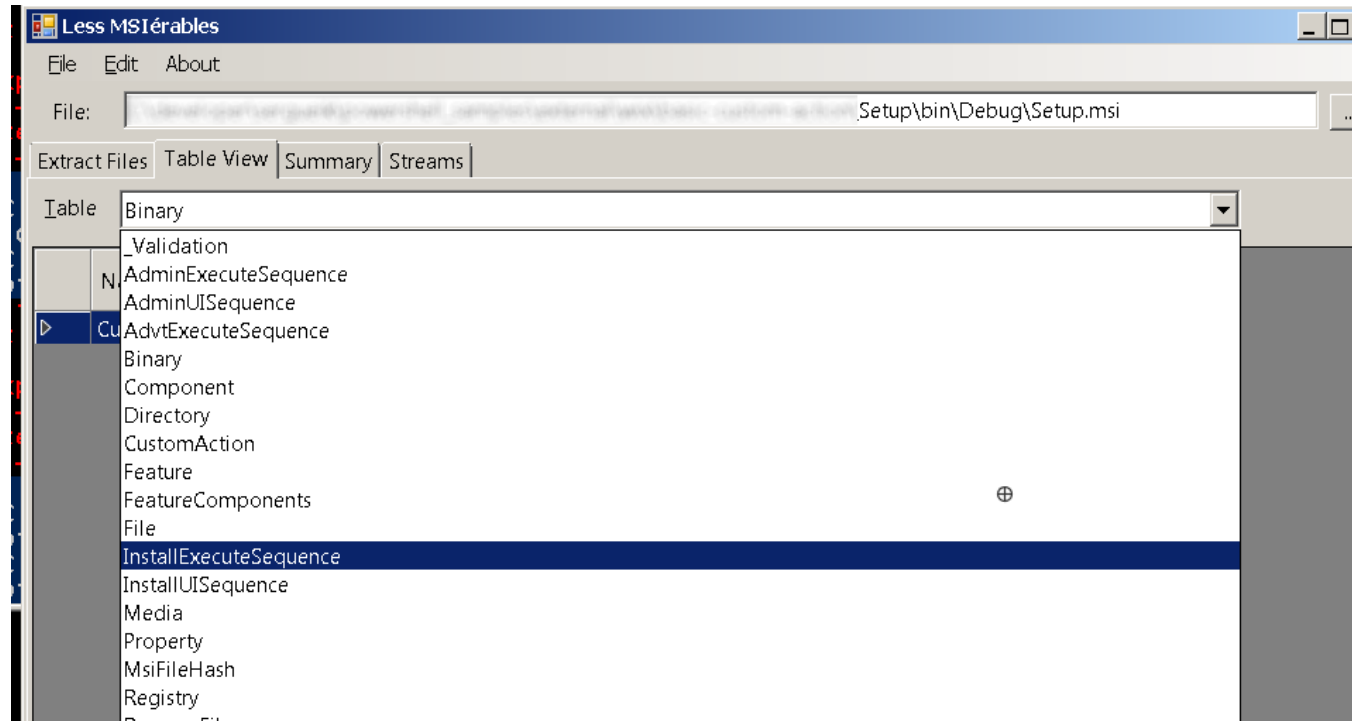
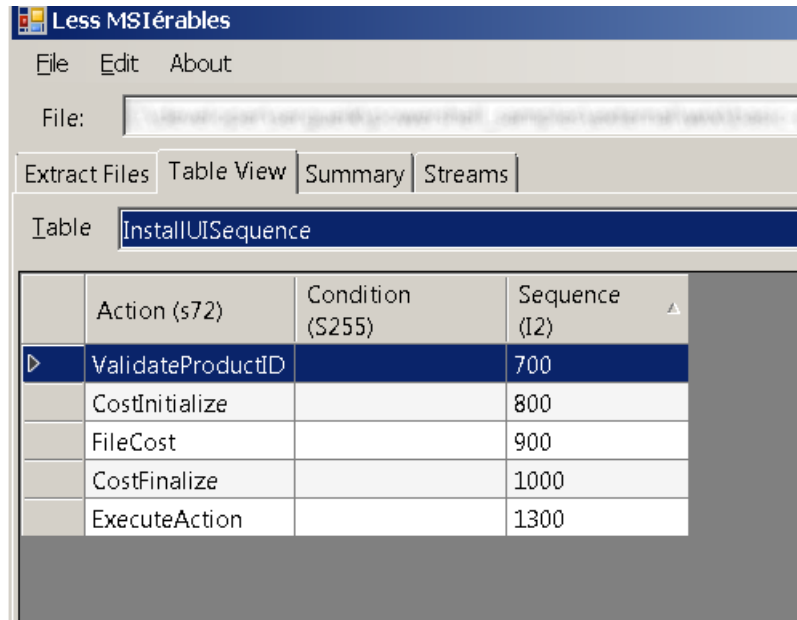


# MSI Database



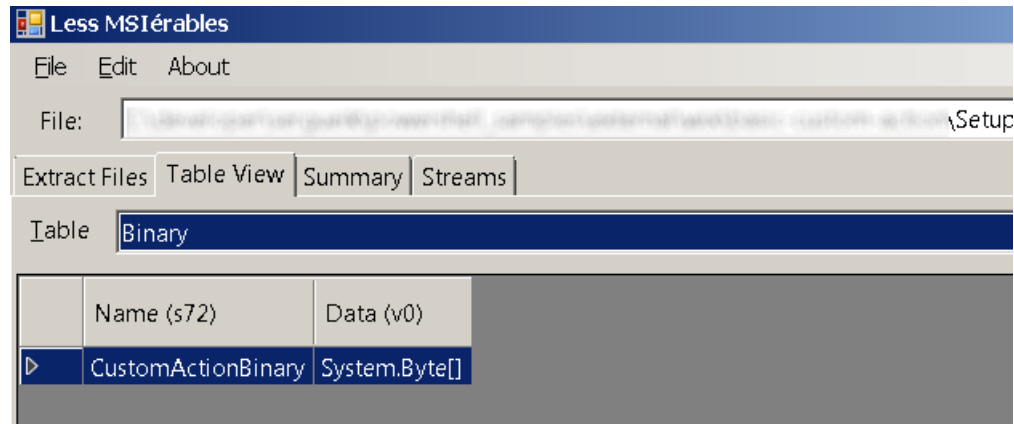
# MSI Tables



The screenshot shows a software application titled "Less MSI Tables". It has a menu bar with "File", "Edit", and "About". Below the menu is a "File:" label followed by a text field containing a file path. There are four tabs: "Extract Files", "Table View", "Summary", and "Streams". The "Table View" tab is active, and within it, a sub-tab labeled "Table" shows a table named "InstallUISequence". This table has four columns: an index column, "Action (s72)", "Condition (S255)", and "Sequence (I2)". The first row is selected and contains the values 1, "ValidateProductID", an empty condition, and 700. The subsequent rows are "CostInitialize" (800), "FileCost" (900), "CostFinalize" (1000), and "ExecuteAction" (1300).

	Action (s72)	Condition (S255)	Sequence (I2)
1	ValidateProductID		700
	CostInitialize		800
	FileCost		900
	CostFinalize		1000
	ExecuteAction		1300

# MSI Custom Action



# MSI Custom Code

```
<?xml version="1.0"?>
<product>
  <Binary Id="dll" SourceFile="$(var.TargetDir)CA.dll"/>
  <Id="GetDateTime" BinaryKey="dll" DllEntry="GetDateTime" Execute="immediate" Return="check"/>
  <Id="Display" Directory="SystemFolder"
    ExeCommand="echo START_TIME = &quot;[START_TIME]&quot;;" Execute="deferred" Impersonate="no"/>
  <InstallExecuteSequence>
    <Custom Action="GetDateTime" Before="Display"/>
    <Custom Action="Display" Before="InstallFinalize">NOT Installed</Custom>
  </InstallExecuteSequence>
</product>
```

```
using Microsoft.Deployment.WindowsInstaller;
using System;
```

```
[CustomAction]
public static ActionResult GetDateTime(Session session) {
    session["START_TIME"] = String.Format("{0:HH:mm}", DateTime.Now.Add(new TimeSpan(0, 0, 2, 0)));
    return ActionResult.Success;
}
```