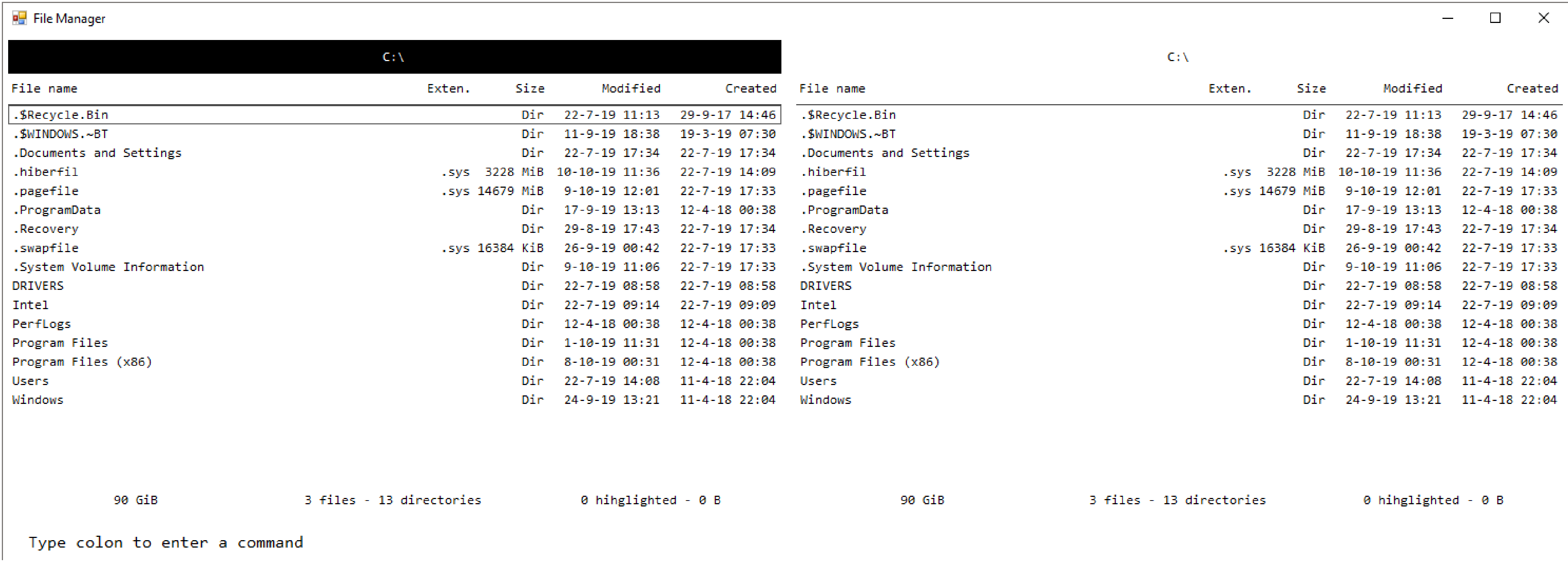
File Manager

David Burian



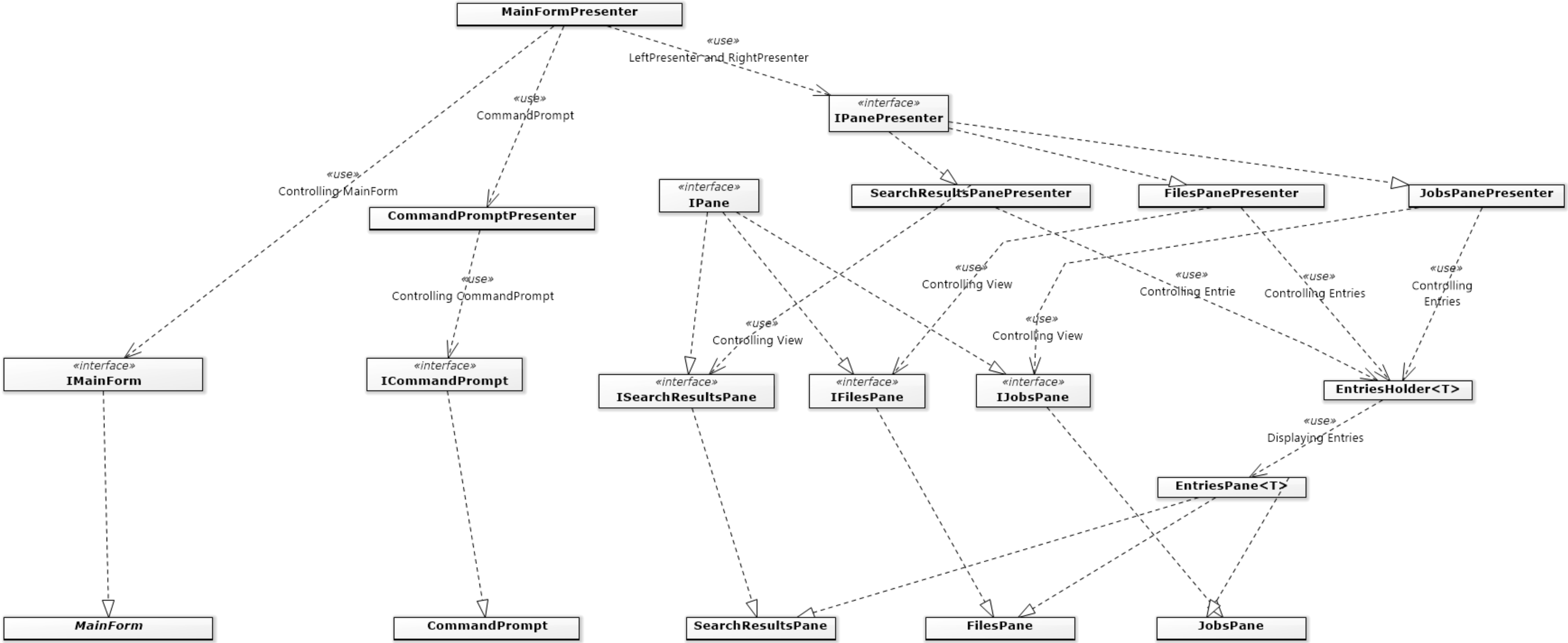
```
                          C:\                                                                        C:\
File name                       Exten.   Size      Modified         Created     File name                       Exten.   Size      Modified         Created
.$Recycle.Bin                             Dir   22-7-19 11:13   29-9-17 14:46   .$Recycle.Bin                             Dir   22-7-19 11:13   29-9-17 14:46
.$WINDOWS.~BT                             Dir   11-9-19 18:38   19-3-19 07:30   .$WINDOWS.~BT                             Dir   11-9-19 18:38   19-3-19 07:30
.Documents and Settings                   Dir   22-7-19 17:34   22-7-19 17:34   .Documents and Settings                   Dir   22-7-19 17:34   22-7-19 17:34
.hiberfil                         .sys  3228 MiB 10-10-19 11:36 22-7-19 14:09   .hiberfil                         .sys  3228 MiB 10-10-19 11:36 22-7-19 14:09
.pagefile                         .sys 14679 MiB  9-10-19 12:01 22-7-19 17:33   .pagefile                         .sys 14679 MiB  9-10-19 12:01 22-7-19 17:33
.ProgramData                              Dir   17-9-19 13:13   12-4-18 00:38   .ProgramData                              Dir   17-9-19 13:13   12-4-18 00:38
.Recovery                                 Dir   29-8-19 17:43   22-7-19 17:34   .Recovery                                 Dir   29-8-19 17:43   22-7-19 17:34
.swapfile                         .sys 16384 KiB 26-9-19 00:42  22-7-19 17:33   .swapfile                         .sys 16384 KiB 26-9-19 00:42  22-7-19 17:33
.System Volume Information                Dir    9-10-19 11:06  22-7-19 17:33   .System Volume Information                Dir    9-10-19 11:06  22-7-19 17:33
DRIVERS                                   Dir   22-7-19 08:58   22-7-19 08:58   DRIVERS                                   Dir   22-7-19 08:58   22-7-19 08:58
Intel                                     Dir   22-7-19 09:14   22-7-19 09:09   Intel                                     Dir   22-7-19 09:14   22-7-19 09:09
PerfLogs                                  Dir   12-4-18 00:38   12-4-18 00:38   PerfLogs                                  Dir   12-4-18 00:38   12-4-18 00:38
Program Files                             Dir    1-10-19 11:31  12-4-18 00:38   Program Files                             Dir    1-10-19 11:31  12-4-18 00:38
Program Files (x86)                       Dir    8-10-19 00:31  12-4-18 00:38   Program Files (x86)                       Dir    8-10-19 00:31  12-4-18 00:38
Users                                     Dir   22-7-19 14:08   11-4-18 22:04   Users                                     Dir   22-7-19 14:08   11-4-18 22:04
Windows                                   Dir   24-9-19 13:21   11-4-18 22:04   Windows                                   Dir   24-9-19 13:21   11-4-18 22:04



      90 GiB         3 files - 13 directories        0 hihglighted - 0 B          90 GiB          3 files - 13 directories        0 hihglighted - 0 B

 Type colon to enter a command
```

Hlavní funkce programu

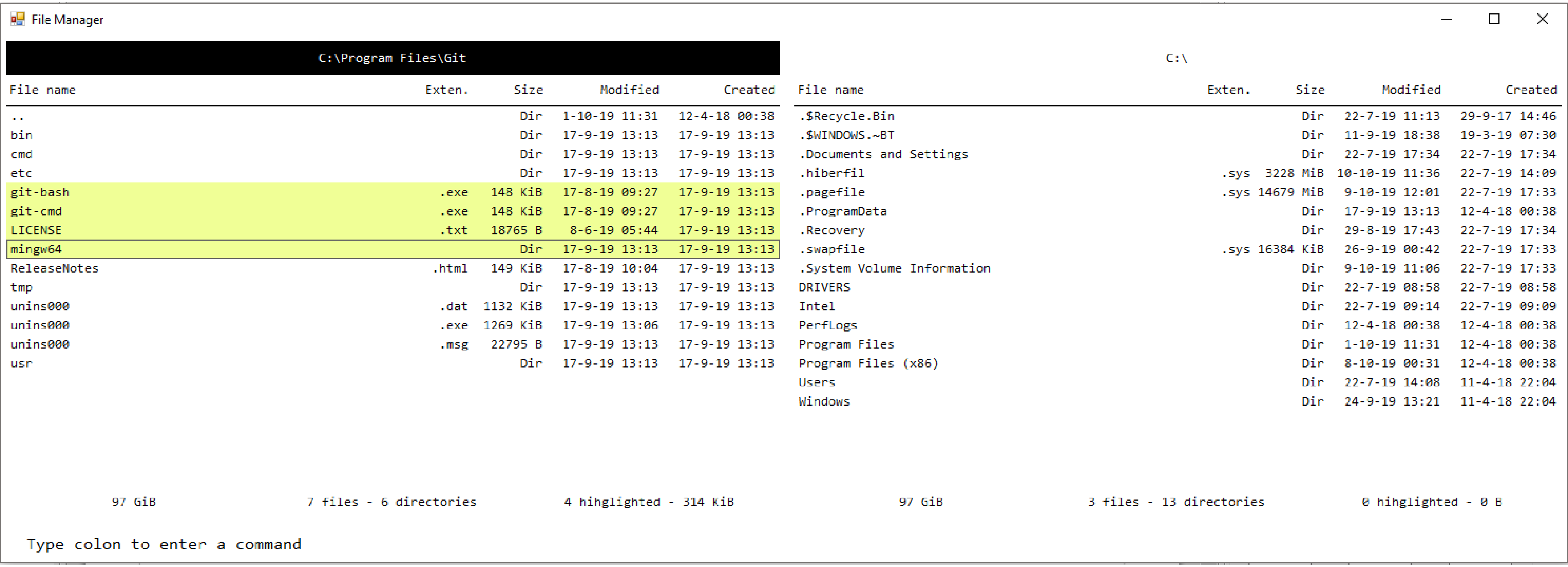
1. *Zobrazovat obsah složek spolu s dalšími užitečnými informacemi*
2. *Reagovat na klávesový vstup*
3. *Měnit vzhled aplikace*
4. *Zpracovávat příkazy od uživatele*
5. *Asynchronně vykonávat operace*
6. *Sledovat status operací napříč více vláken*
7. *Rušit operace napříč více vláken*

Základní architektura programu

- FileManager
  - Vykreslení a ovládání grafických prvků
  - Zpracovávání uživatelem vyvolaných událostí
- MultithreadedFileSystemOperations
  - Vykonává a řídí operace nad FS
  - Poskytuje API na synchronizaci


MainFormPresenter
«use»
LeftPresenter and RightPresenter
«interface»
IPanePresenter
«use»
CommandPrompt
«use»
Controlling MainForm
CommandPromptPresenter
«interface»
IPane
SearchResultsPanePresenter
FilesPanePresenter
JobsPanePresenter
«use»
Controlling CommandPrompt
«use»
Controlling View
«use»
Controlling Entrie
«use»
Controlling Entries
«use»
Controlling Entries
«use»
Controlling View
«use»
Controlling View
«interface»
IMainForm
«interface»
ICommandPrompt
«interface»
ISearchResultsPane
«interface»
IFilesPane
«interface»
IJobsPane
EntriesHolder<T>
«use»
Displaying Entries
EntriesPane<T>
MainForm
CommandPrompt
SearchResultsPane
FilesPane
JobsPane

Zobrazovat obsah složek spolu s dalšími užitečnými informacemi

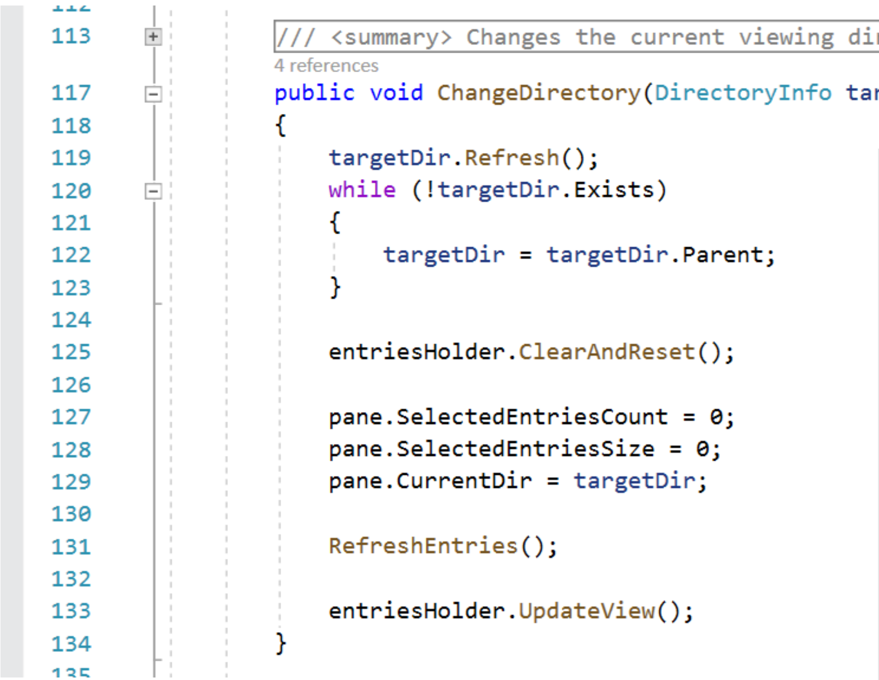


```
                        C:\Program Files\Git
File name                          Exten.   Size      Modified        Created
..                                           Dir   1-10-19 11:31   12-4-18 00:38
bin                                          Dir   17-9-19 13:13   17-9-19 13:13
cmd                                          Dir   17-9-19 13:13   17-9-19 13:13
etc                                          Dir   17-9-19 13:13   17-9-19 13:13
git-bash                             .exe  148 KiB 17-8-19 09:27   17-9-19 13:13
git-cmd                              .exe  148 KiB 17-8-19 09:27   17-9-19 13:13
LICENSE                              .txt  18765 B  8-6-19 05:44   17-9-19 13:13
mingw64                                      Dir   17-9-19 13:13   17-9-19 13:13
ReleaseNotes                        .html  149 KiB 17-8-19 10:04   17-9-19 13:13
tmp                                          Dir   17-9-19 13:13   17-9-19 13:13
unins000                             .dat 1132 KiB 17-9-19 13:13   17-9-19 13:13
unins000                             .exe 1269 KiB 17-9-19 13:06   17-9-19 13:13
unins000                             .msg  22795 B 17-9-19 13:13   17-9-19 13:13
usr                                          Dir   17-9-19 13:13   17-9-19 13:13

        97 GiB            7 files - 6 directories        4 hihglighted - 314 KiB
```

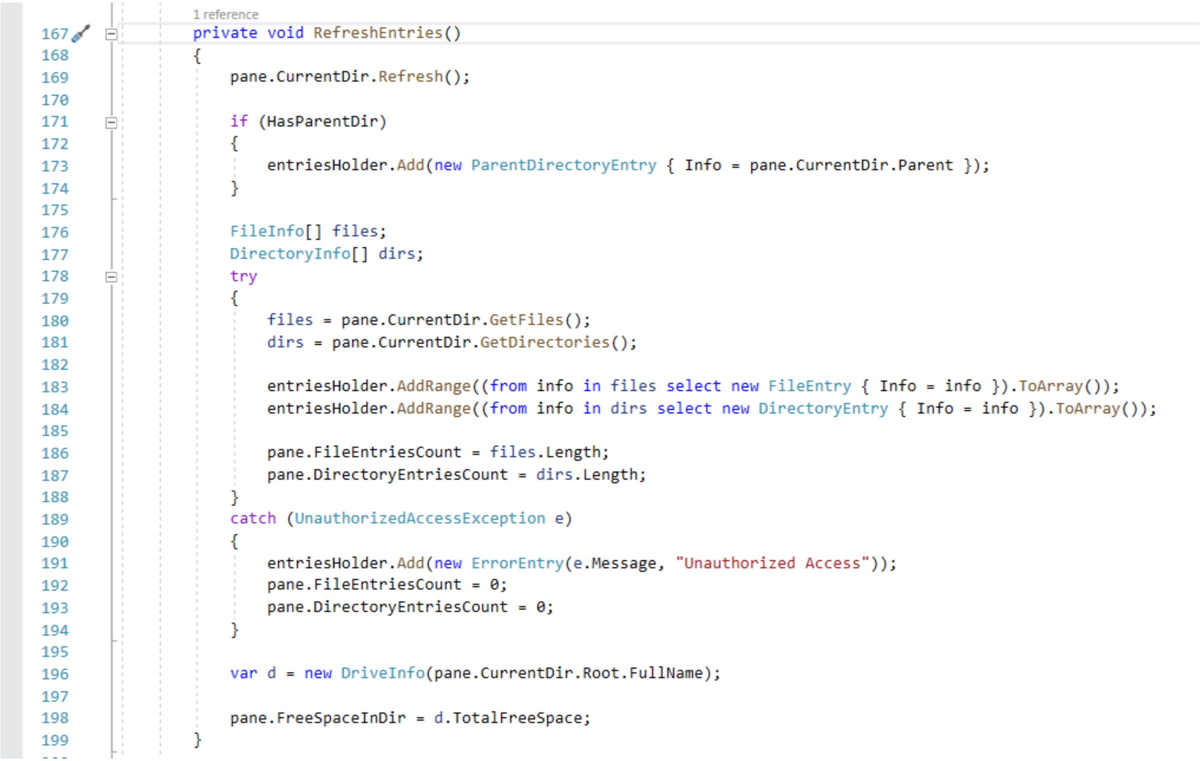
```
                                C:\
File name                          Exten.   Size      Modified        Created
.$Recycle.Bin                                Dir   22-7-19 11:13   29-9-17 14:46
.$WINDOWS.~BT                                Dir   11-9-19 18:38   19-3-19 07:30
.Documents and Settings                      Dir   22-7-19 17:34   22-7-19 17:34
.hiberfil                            .sys 3228 MiB 10-10-19 11:36  22-7-19 14:09
.pagefile                            .sys 14679 MiB 9-10-19 12:01  22-7-19 17:33
.ProgramData                                 Dir   17-9-19 13:13   12-4-18 00:38
.Recovery                                    Dir   29-8-19 17:43   22-7-19 17:34
.swapfile                            .sys 16384 KiB 26-9-19 00:42  22-7-19 17:33
.System Volume Information                   Dir   9-10-19 11:06   22-7-19 17:33
DRIVERS                                      Dir   22-7-19 08:58   22-7-19 08:58
Intel                                        Dir   22-7-19 09:14   22-7-19 09:09
PerfLogs                                     Dir   12-4-18 00:38   12-4-18 00:38
Program Files                                Dir   1-10-19 11:31   12-4-18 00:38
Program Files (x86)                          Dir   8-10-19 00:31   12-4-18 00:38
Users                                        Dir   22-7-19 14:08   11-4-18 22:04
Windows                                      Dir   24-9-19 13:21   11-4-18 22:04

        97 GiB            3 files - 13 directories        0 hihglighted - 0 B
```

```
Type colon to enter a command
```

Zobrazovat obsah složek spolu s dalšími užitečnými informacemi

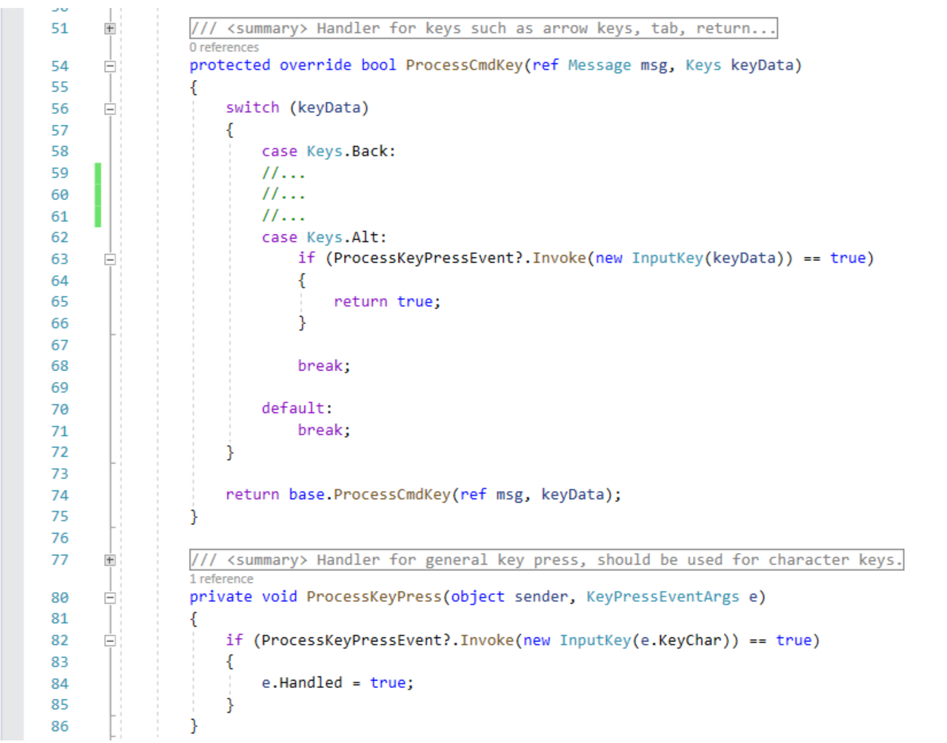


```
/// <summary> Changes the current viewing directory to targetDir.
4 references
public void ChangeDirectory(DirectoryInfo targetDir)
{
    targetDir.Refresh();
    while (!targetDir.Exists)
    {
        targetDir = targetDir.Parent;
    }

    entriesHolder.ClearAndReset();

    pane.SelectedEntriesCount = 0;
    pane.SelectedEntriesSize = 0;
    pane.CurrentDir = targetDir;

    RefreshEntries();

    entriesHolder.UpdateView();
}
```



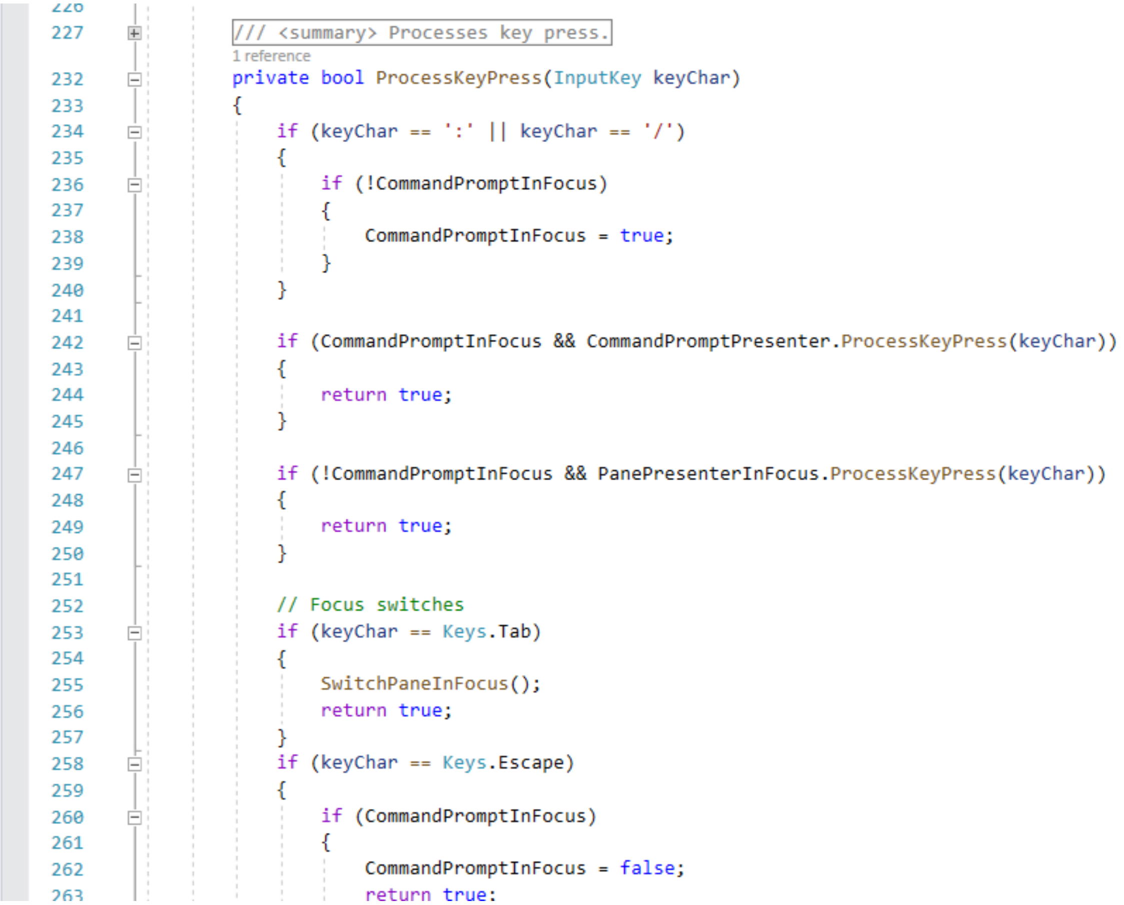
```
1 reference
private void RefreshEntries()
{
    pane.CurrentDir.Refresh();

    if (HasParentDir)
    {
        entriesHolder.Add(new ParentDirectoryEntry { Info = pane.CurrentDir.Parent });
    }

    FileInfo[] files;
    DirectoryInfo[] dirs;
    try
    {
        files = pane.CurrentDir.GetFiles();
        dirs = pane.CurrentDir.GetDirectories();

        entriesHolder.AddRange((from info in files select new FileEntry { Info = info }).ToArray());
        entriesHolder.AddRange((from info in dirs select new DirectoryEntry { Info = info }).ToArray());

        pane.FileEntriesCount = files.Length;
        pane.DirectoryEntriesCount = dirs.Length;
    }
    catch (UnauthorizedAccessException e)
    {
        entriesHolder.Add(new ErrorEntry(e.Message, "Unauthorized Access"));
        pane.FileEntriesCount = 0;
        pane.DirectoryEntriesCount = 0;
    }

    var d = new DriveInfo(pane.CurrentDir.Root.FullName);

    pane.FreeSpaceInDir = d.TotalFreeSpace;
}
```

Reagovat na klávesový vstup



```
/// <summary> Handler for keys such as arrow keys, tab, return...
0 references
protected override bool ProcessCmdKey(ref Message msg, Keys keyData)
{
    switch (keyData)
    {
        case Keys.Back:
        //...
        //...
        //...
        case Keys.Alt:
            if (ProcessKeyPressEvent?.Invoke(new InputKey(keyData)) == true)
            {
                return true;
            }

            break;

        default:
            break;
    }

    return base.ProcessCmdKey(ref msg, keyData);
}

/// <summary> Handler for general key press, should be used for character keys.
1 reference
private void ProcessKeyPress(object sender, KeyPressEventArgs e)
{
    if (ProcessKeyPressEvent?.Invoke(new InputKey(e.KeyChar)) == true)
    {
        e.Handled = true;
    }
}
```

MainForm

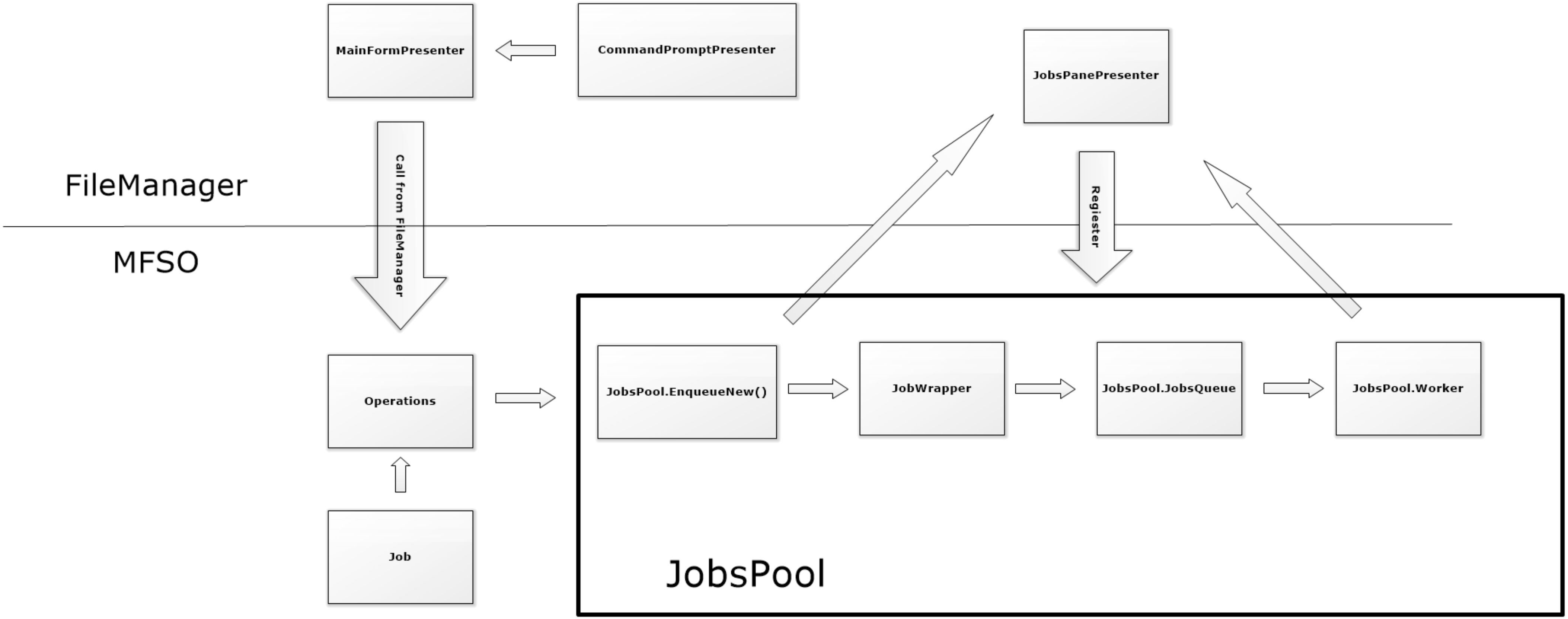


```
/// <summary> Processes key press.
1 reference
private bool ProcessKeyPress(InputKey keyChar)
{
    if (keyChar == ':' || keyChar == '/')
    {
        if (!CommandPromptInFocus)
        {
            CommandPromptInFocus = true;
        }
    }

    if (CommandPromptInFocus && CommandPromptPresenter.ProcessKeyPress(keyChar))
    {
        return true;
    }

    if (!CommandPromptInFocus && PanePresenterInFocus.ProcessKeyPress(keyChar))
    {
        return true;
    }

    // Focus switches
    if (keyChar == Keys.Tab)
    {
        SwitchPaneInFocus();
        return true;
    }
    if (keyChar == Keys.Escape)
    {
        if (CommandPromptInFocus)
        {
            CommandPromptInFocus = false;
            return true;
```

MainFormPresenter

MFSO

5. *Asynchronně vykonávat operace*
6. *Sledovat status operací napříč více vláken*
7. *Rušit operace napříč více vláken*



MainFormPresenter

CommandPromptPresenter

JobsPanePresenter

FileManager

Call from FileManager

Regiester

MFSO

Operations

Job

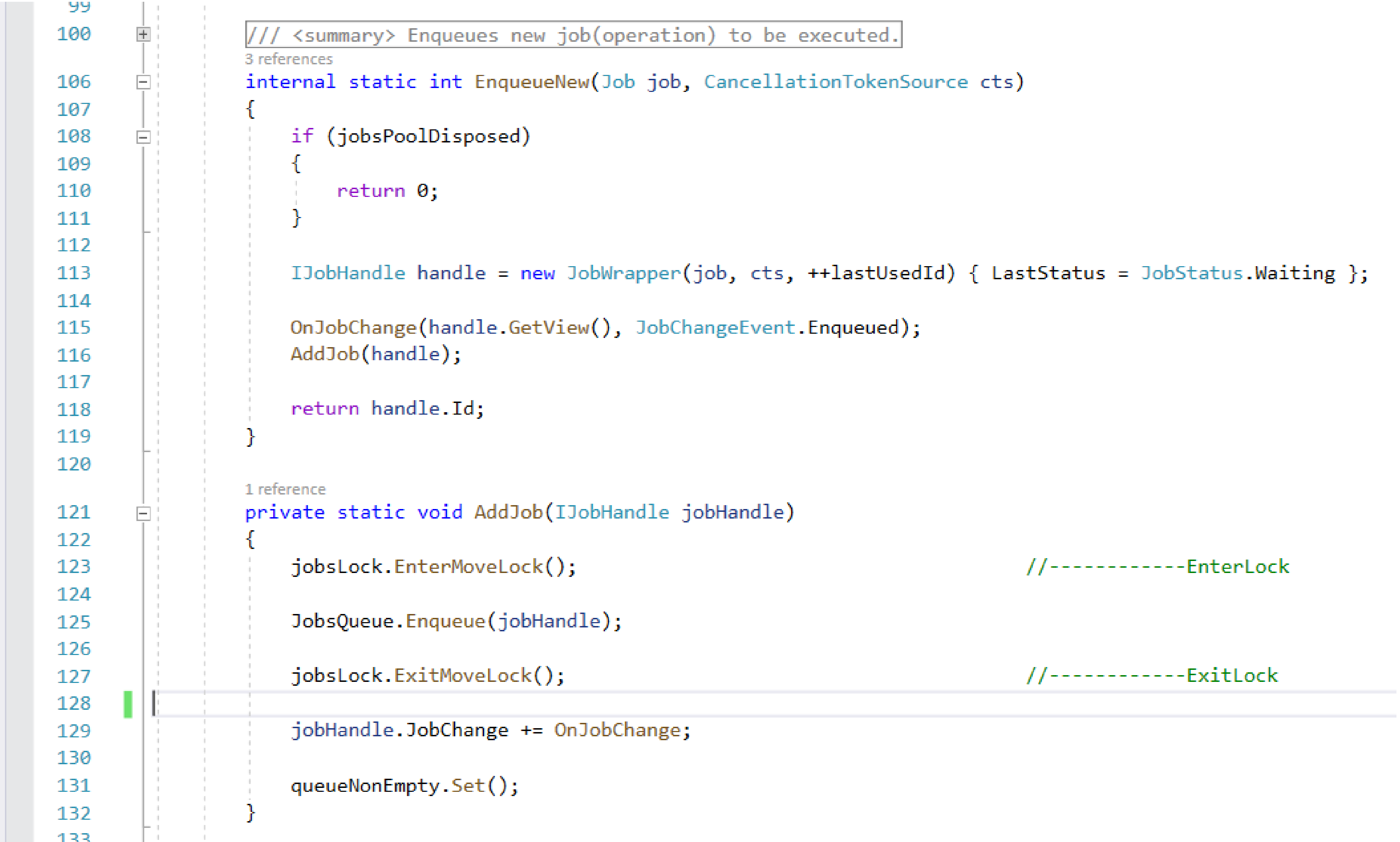
JobsPool.EnqueueNew()

JobWrapper

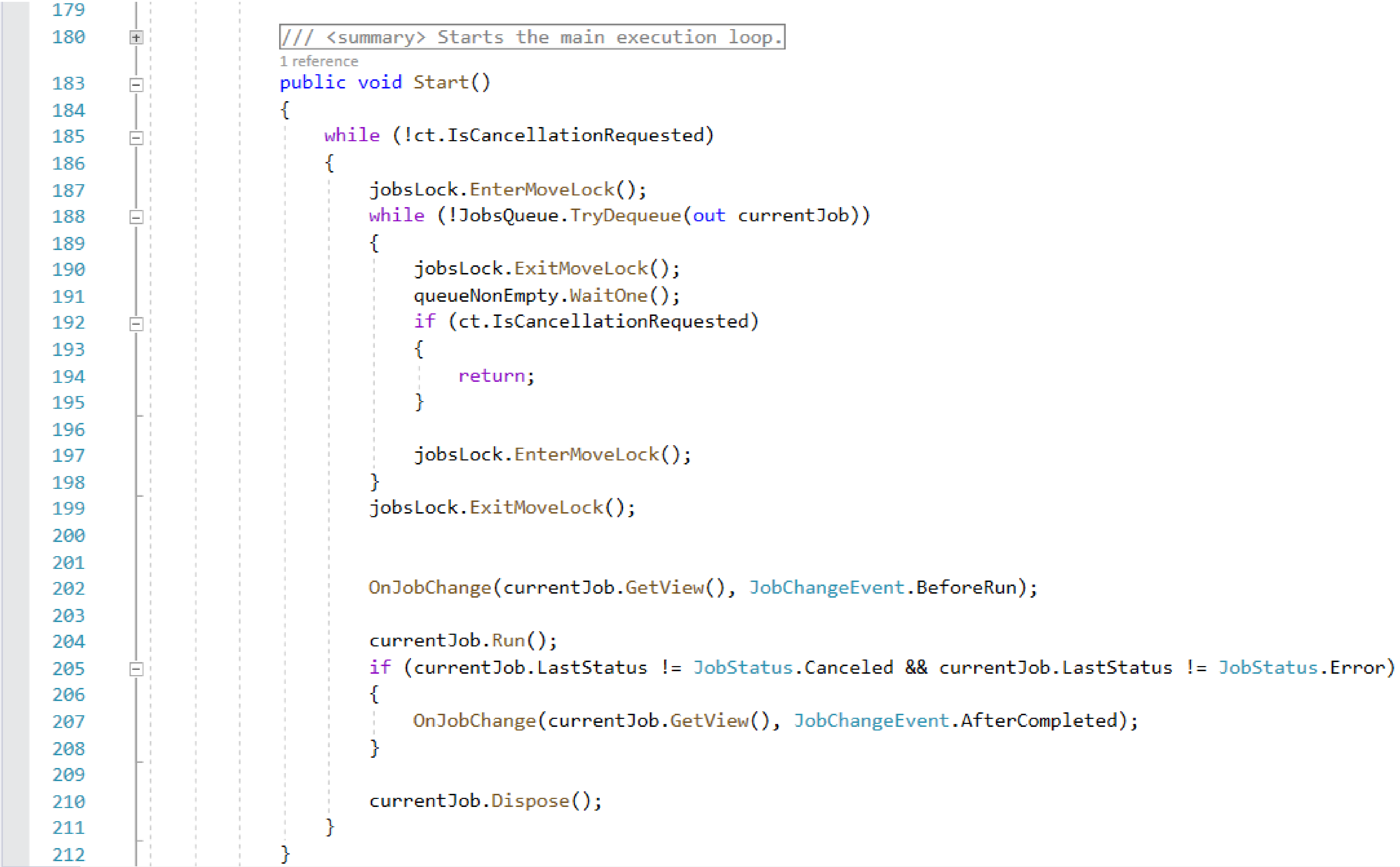
JobsPool.JobsQueue

JobsPool.Worker

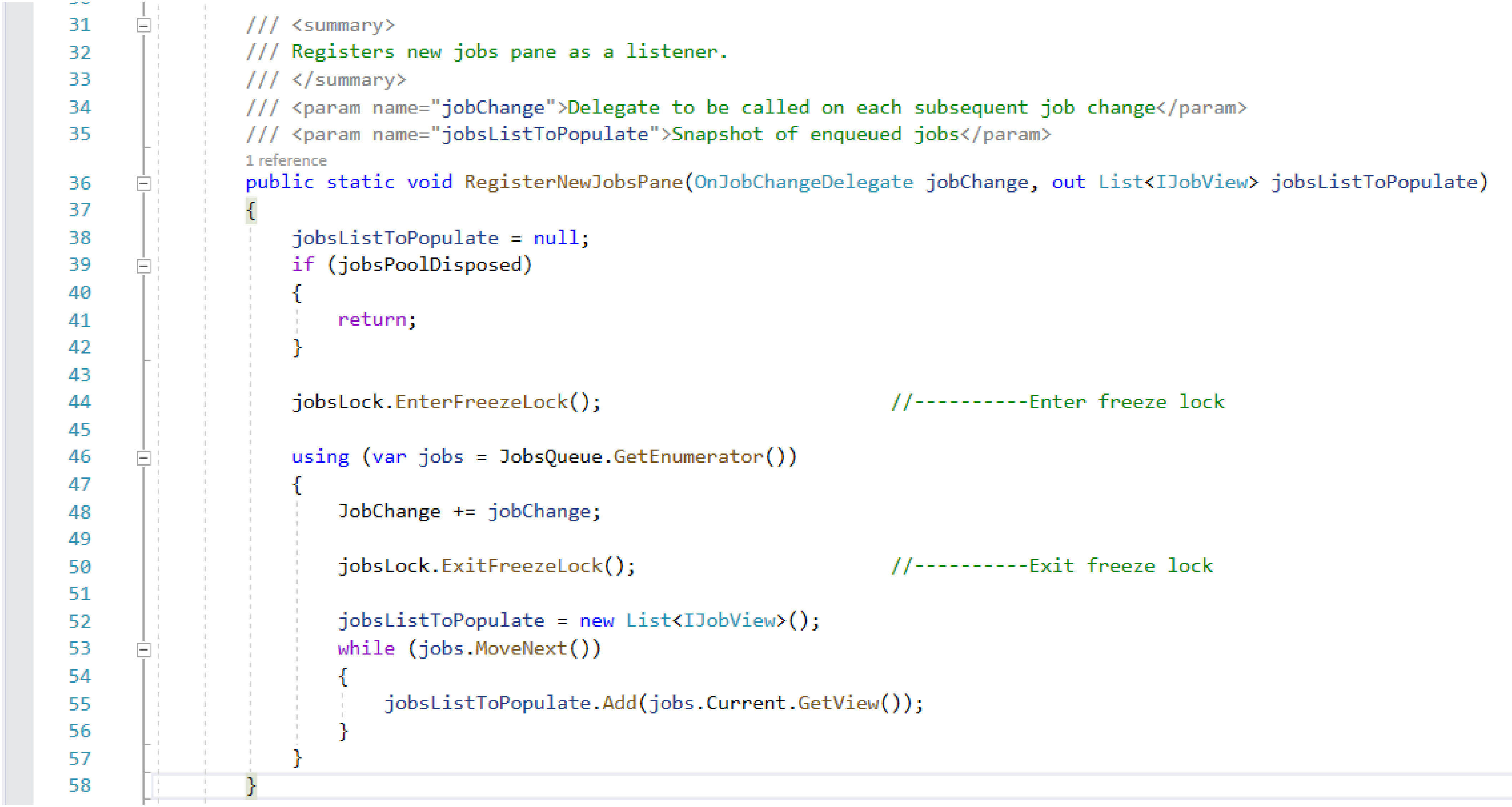
JobsPool



```
/// <summary> Enqueues new job(operation) to be executed.
internal static int EnqueueNew(Job job, CancellationTokenSource cts)
{
    if (jobsPoolDisposed)
    {
        return 0;
    }

    IJobHandle handle = new JobWrapper(job, cts, ++lastUsedId) { LastStatus = JobStatus.Waiting };

    OnJobChange(handle.GetView(), JobChangeEvent.Enqueued);
    AddJob(handle);

    return handle.Id;
}

private static void AddJob(IJobHandle jobHandle)
{
    jobsLock.EnterMoveLock();                                           //------------EnterLock

    JobsQueue.Enqueue(jobHandle);

    jobsLock.ExitMoveLock();                                            //------------ExitLock

    jobHandle.JobChange += OnJobChange;

    queueNonEmpty.Set();
}
```

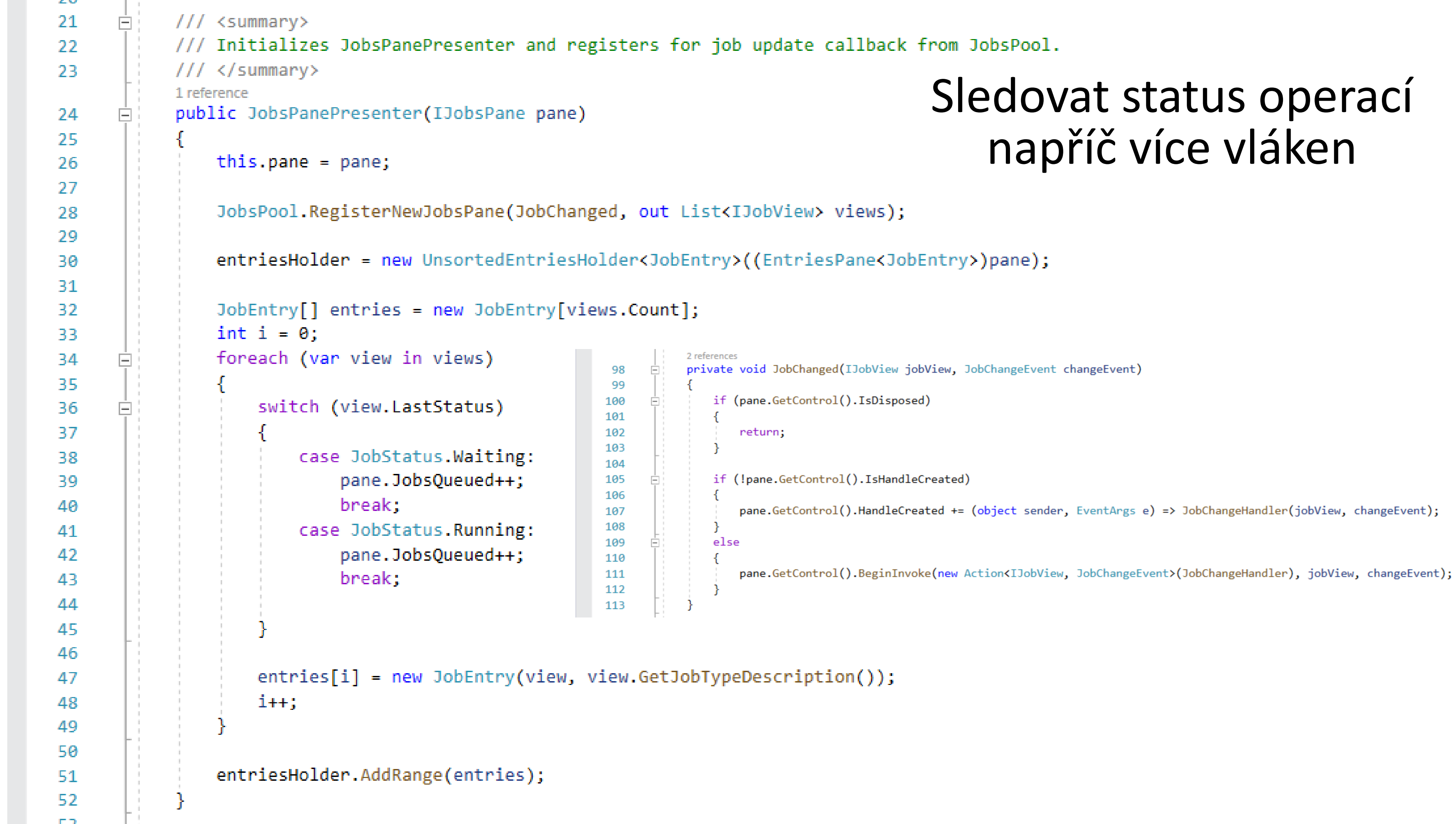


```
/// <summary> Starts the main execution loop.
public void Start()
{
    while (!ct.IsCancellationRequested)
    {
        jobsLock.EnterMoveLock();
        while (!JobsQueue.TryDequeue(out currentJob))
        {
            jobsLock.ExitMoveLock();
            queueNonEmpty.WaitOne();
            if (ct.IsCancellationRequested)
            {
                return;
            }

            jobsLock.EnterMoveLock();
        }
        jobsLock.ExitMoveLock();


        OnJobChange(currentJob.GetView(), JobChangeEvent.BeforeRun);

        currentJob.Run();
        if (currentJob.LastStatus != JobStatus.Canceled && currentJob.LastStatus != JobStatus.Error)
        {
            OnJobChange(currentJob.GetView(), JobChangeEvent.AfterCompleted);
        }

        currentJob.Dispose();
    }
}
```



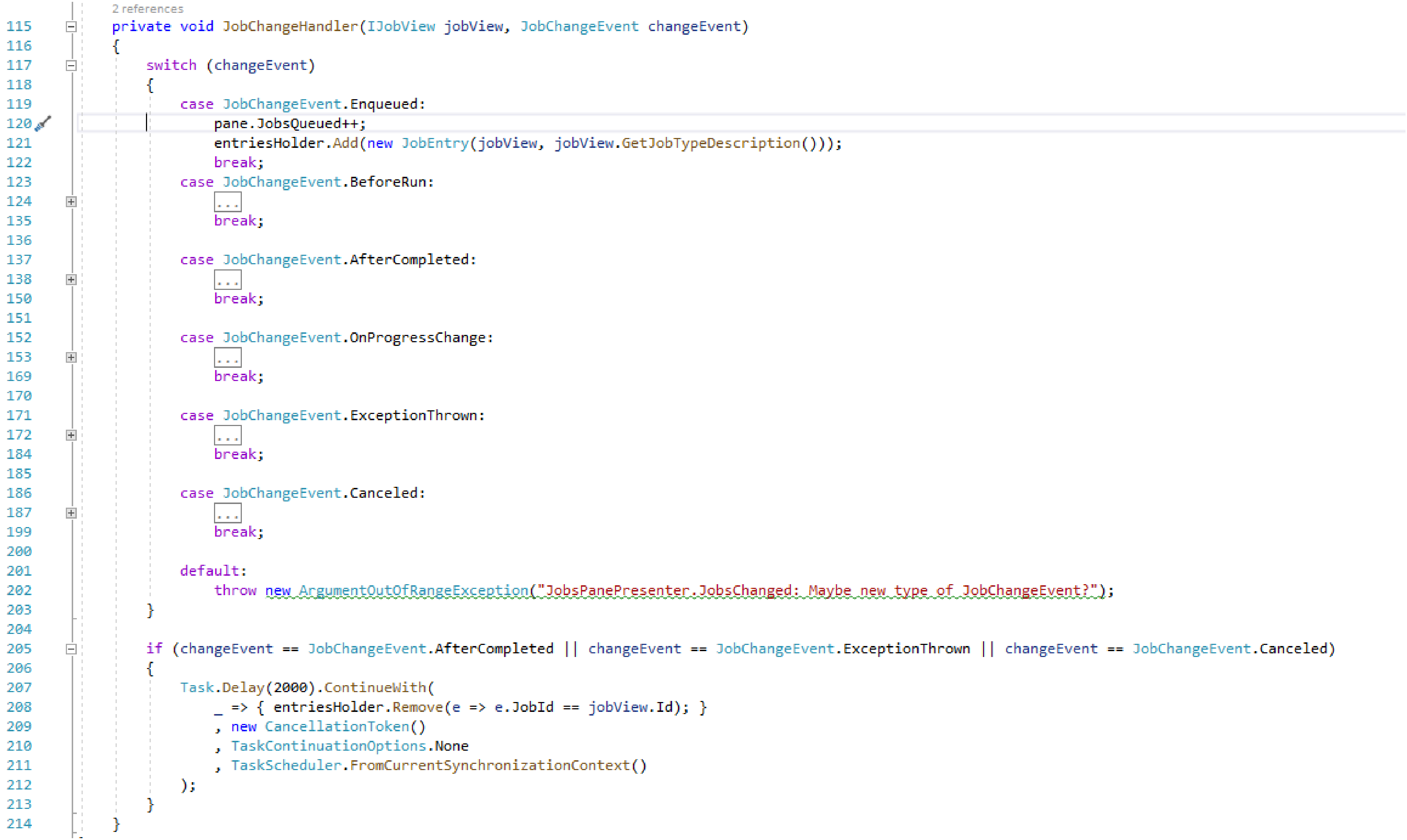
```
/// <summary>
/// Registers new jobs pane as a listener.
/// </summary>
/// <param name="jobChange">Delegate to be called on each subsequent job change</param>
/// <param name="jobsListToPopulate">Snapshot of enqueued jobs</param>
1 reference
public static void RegisterNewJobsPane(OnJobChangeDelegate jobChange, out List<IJobView> jobsListToPopulate)
{
    jobsListToPopulate = null;
    if (jobsPoolDisposed)
    {
        return;
    }

    jobsLock.EnterFreezeLock();                            //----------Enter freeze lock

    using (var jobs = JobsQueue.GetEnumerator())
    {
        JobChange += jobChange;

        jobsLock.ExitFreezeLock();                         //----------Exit freeze lock

        jobsListToPopulate = new List<IJobView>();
        while (jobs.MoveNext())
        {
            jobsListToPopulate.Add(jobs.Current.GetView());
        }
    }
}
```

Sledovat status operací napříč více vláken

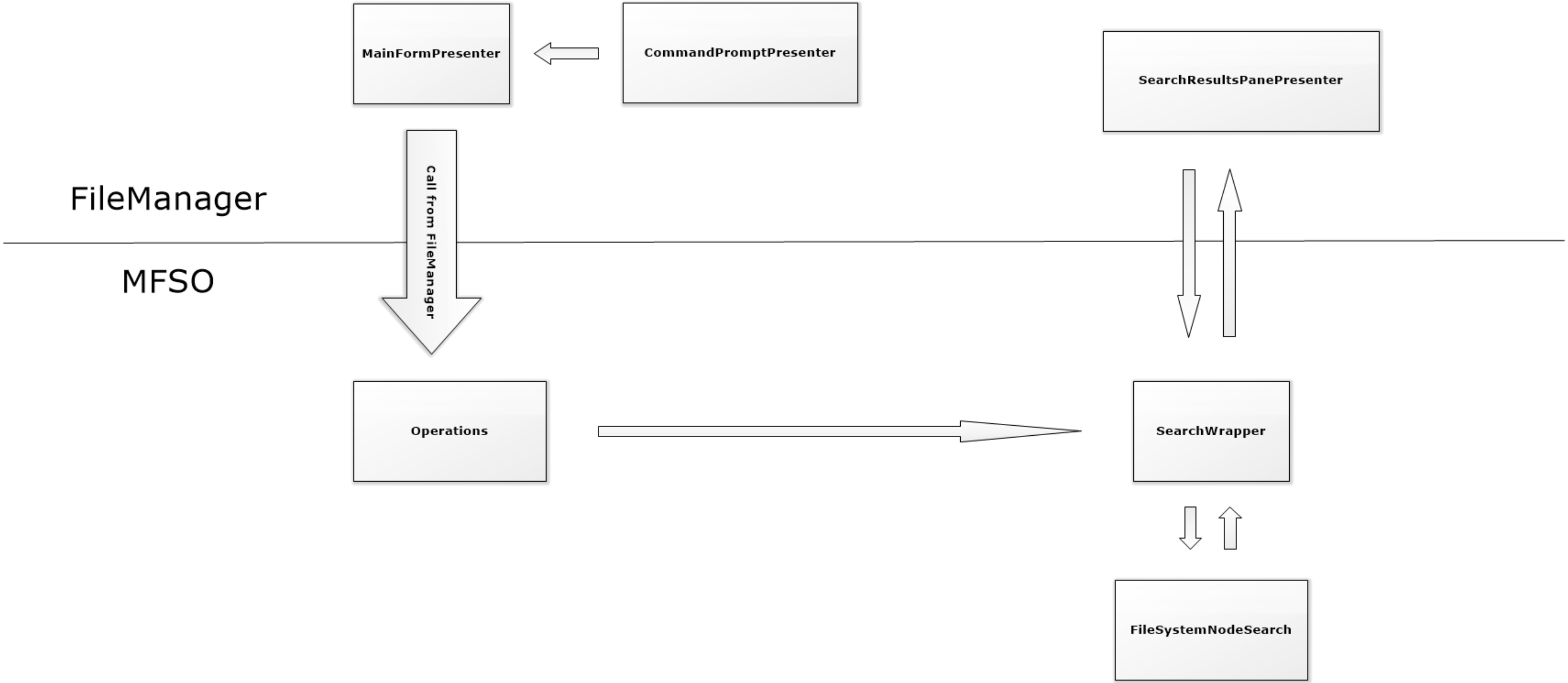


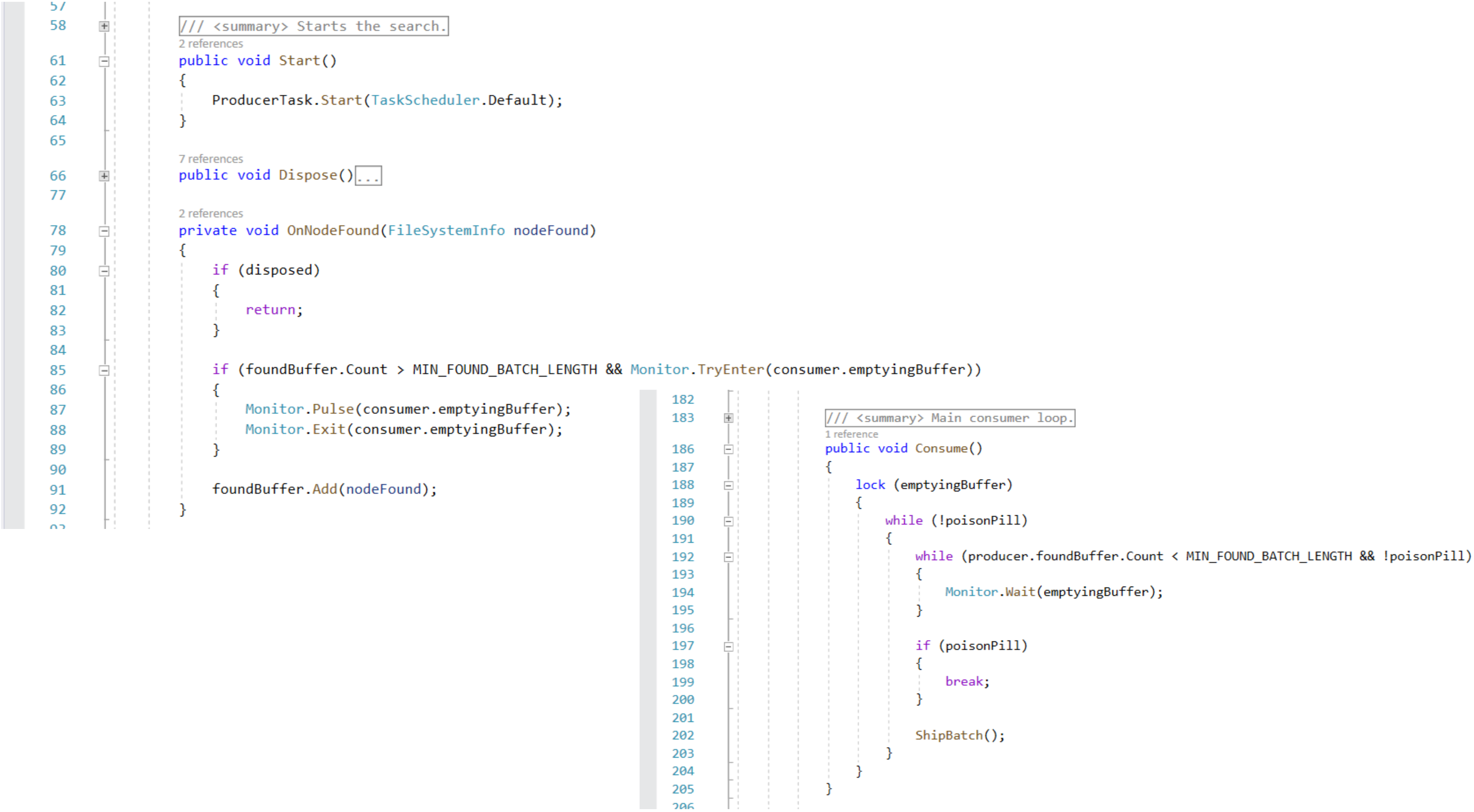
```
/// <summary>
/// Initializes JobsPanePresenter and registers for job update callback from JobsPool.
/// </summary>
1 reference
public JobsPanePresenter(IJobsPane pane)
{
    this.pane = pane;

    JobsPool.RegisterNewJobsPane(JobChanged, out List<IJobView> views);

    entriesHolder = new UnsortedEntriesHolder<JobEntry>((EntriesPane<JobEntry>)pane);

    JobEntry[] entries = new JobEntry[views.Count];
    int i = 0;
    foreach (var view in views)
    {
        switch (view.LastStatus)
        {
            case JobStatus.Waiting:
                pane.JobsQueued++;
                break;
            case JobStatus.Running:
                pane.JobsQueued++;
                break;

        }

        entries[i] = new JobEntry(view, view.GetJobTypeDescription());
        i++;
    }

    entriesHolder.AddRange(entries);
}
```

```
2 references
private void JobChanged(IJobView jobView, JobChangeEvent changeEvent)
{
    if (pane.GetControl().IsDisposed)
    {
        return;
    }

    if (!pane.GetControl().IsHandleCreated)
    {
        pane.GetControl().HandleCreated += (object sender, EventArgs e) => JobChangeHandler(jobView, changeEvent);
    }
    else
    {
        pane.GetControl().BeginInvoke(new Action<IJobView, JobChangeEvent>(JobChangeHandler), jobView, changeEvent);
    }
}
```



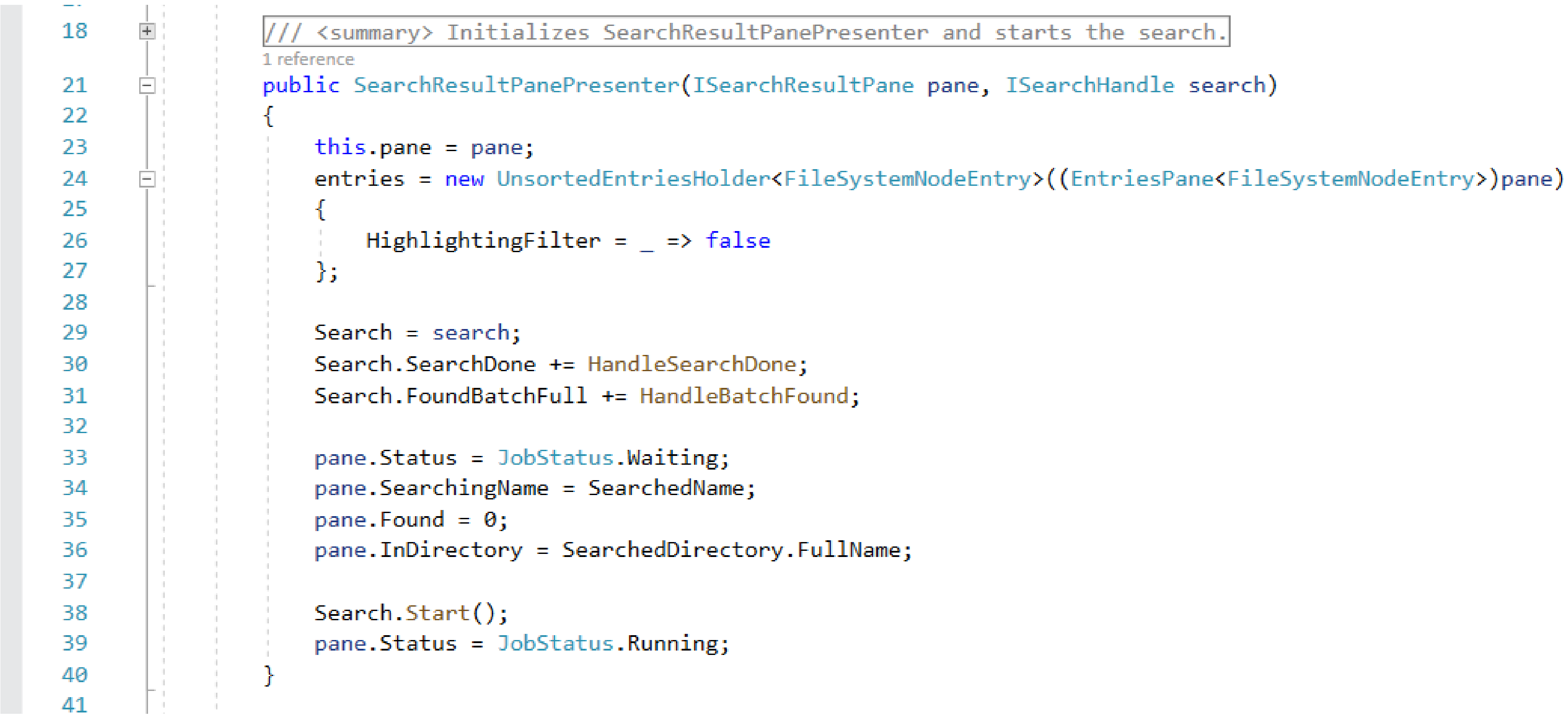
```
2 references
private void JobChangeHandler(IJobView jobView, JobChangeEvent changeEvent)
{
    switch (changeEvent)
    {
        case JobChangeEvent.Enqueued:
            pane.JobsQueued++;
            entriesHolder.Add(new JobEntry(jobView, jobView.GetJobTypeDescription()));
            break;
        case JobChangeEvent.BeforeRun:
            ...
            break;

        case JobChangeEvent.AfterCompleted:
            ...
            break;

        case JobChangeEvent.OnProgressChange:
            ...
            break;

        case JobChangeEvent.ExceptionThrown:
            ...
            break;

        case JobChangeEvent.Canceled:
            ...
            break;

        default:
            throw new ArgumentOutOfRangeException("JobsPanePresenter.JobsChanged: Maybe new type of JobChangeEvent?");
    }

    if (changeEvent == JobChangeEvent.AfterCompleted || changeEvent == JobChangeEvent.ExceptionThrown || changeEvent == JobChangeEvent.Canceled)
    {
        Task.Delay(2000).ContinueWith(
            _ => { entriesHolder.Remove(e => e.JobId == jobView.Id); }
            , new CancellationToken()
            , TaskContinuationOptions.None
            , TaskScheduler.FromCurrentSynchronizationContext()
        );
    }
}
```


FileManager
MFSO
MainFormPresenter
CommandPromptPresenter
SearchResultsPanePresenter
Call from FileManager
Operations
SearchWrapper
FileSystemNodeSearch



```
/// <summary> Starts the search.
2 references
public void Start()
{
    ProducerTask.Start(TaskScheduler.Default);
}

7 references
public void Dispose()...

2 references
private void OnNodeFound(FileSystemInfo nodeFound)
{
    if (disposed)
    {
        return;
    }

    if (foundBuffer.Count > MIN_FOUND_BATCH_LENGTH && Monitor.TryEnter(consumer.emptyingBuffer))
    {
        Monitor.Pulse(consumer.emptyingBuffer);
        Monitor.Exit(consumer.emptyingBuffer);
    }

    foundBuffer.Add(nodeFound);
}
```

```
/// <summary> Main consumer loop.
1 reference
public void Consume()
{
    lock (emptyingBuffer)
    {
        while (!poisonPill)
        {
            while (producer.foundBuffer.Count < MIN_FOUND_BATCH_LENGTH && !poisonPill)
            {
                Monitor.Wait(emptyingBuffer);
            }

            if (poisonPill)
            {
                break;
            }

            ShipBatch();
        }
    }
}
```



```
/// <summary> Initializes SearchResultPanePresenter and starts the search.
public SearchResultPanePresenter(ISearchResultPane pane, ISearchHandle search)
{
    this.pane = pane;
    entries = new UnsortedEntriesHolder<FileSystemNodeEntry>((EntriesPane<FileSystemNodeEntry>)pane)
    {
        HighlightingFilter = _ => false
    };

    Search = search;
    Search.SearchDone += HandleSearchDone;
    Search.FoundBatchFull += HandleBatchFound;

    pane.Status = JobStatus.Waiting;
    pane.SearchingName = SearchedName;
    pane.Found = 0;
    pane.InDirectory = SearchedDirectory.FullName;

    Search.Start();
    pane.Status = JobStatus.Running;
}
```