Name: Knowles, Dylan
NSID: dlk077
Project: WinMerge
Change Number: 1
Date: 04/03/2012
Group Number: Danger Twins

User Specified Time Differences

1. Change Request:

The Options -> Compare dialog allows the user to "Ignore time differences less than 3 seconds". Create a way for the user to specify times other than 3 seconds.

2. Concept Location:

Table 1. The list of all the classes visited during concept location.

#	File name	Tool used	Located?	Comments
1	Merge.rc	"Find in Files" (Ctrl-F)	No	[PROPAGATING] Tried using "Find in Files" to search for "Ignore time differences less than 3 seconds." I couldn't find it because the string had a menu-shortcut character (&) in it. Eventually, I started looking for fragments of this string (e.g., "Ignore time differences", "less than 3 seconds", etc.) and eventually found the file. It contained a constant (IDC_IGNORE_SMALLTIMEDIFF) which helped locate the concept.
2	PropCompareFolder.cpp	"Find in Files" (Ctrl-F)	Somewhat	[CHANGED] Found the method "WriteOptions" inside the class "PropCompareFolder" that tells the "option manager" to ignore small time differences. Passes only a Boolean; I'll have to change this to pass a value.
3	OptionsPanel.h	"Go to Header File"	No	[PROPAGATING] Went here because the "GetOptionManager" method used in PropCompareFolder got this method from this interface. Led me to OptionsPanel.c
4	OptionsPanel.c	N/A	No	[PROPAGATING] Went to OptionsPanel.c and found

				“COptionsMgr”, the class that actually manages options. Found the “SetOption” method. Realized I should be looking for the option being used, not the method setting it.
5	Dirdoc.cpp, hexmergeview.cpp, mergedoc.cpp	“Find all references” of OPT_IGNORE_SMALL_FILETIME	Yes	Found the places where the option exerts its impact.

3. Impact Analysis:

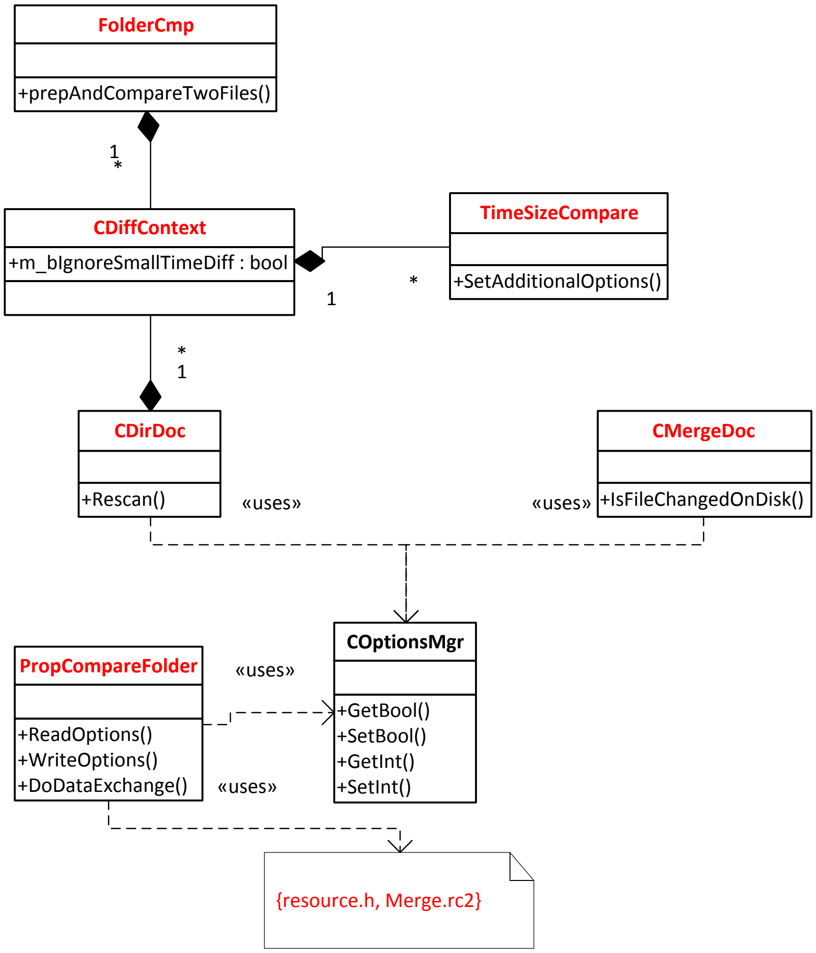
Table 2. The list of all the classes visited during impact analysis.

#	Class name	Tool used	Impacted?	Comments
1	CMergeDoc	“Find all references” of OPT_IGNORE_SMALL_FILETIME	Yes	Looks as though I need to change the IsFileChangedOnDisk method. This is the only place that the timeout is required. The simplest fix would be to change the OPT_IGNORE_SMALL_FILETIME option to be a time value rather than a Boolean.
2	CHexMergeView	“Find all references” of OPT_IGNORE_SMALL_FILETIME	Yes	The same logic as above applies to this class. There is only one location that needs to be changed to make the class accommodate the desired behaviour
3	CDirDoc	“Find all references” of OPT_IGNORE_SMALL_FILETIME	Yes	Will need to be changed to pass a value to CDiffContext. Realized here that I needed to find CDiffContext.
4	CDiffContext	“Go to declaration”	Yes	Need to change a

				Boolean to a time (int) value.
5	FolderCmp	"Find all references" of the field in CDiffContext that records the value of OPT_IGNORE_SMALL_FILETIME	Yes	Need to change a Boolean to a time value.
6	PropCompareFolder	"Find all references" of the field in CDiffContext that records the value of OPT_IGNORE_SMALL_FILETIME	Yes	Need to change a Boolean to a time value.
7	TimeSizeCompare	Several uses of "find all references" led me to this class. I searched for it after noticing that it was being passed the Boolean value that I was removing in FolderCmp.	Yes	Need to change a Boolean to a time value.
8.	Resource.h	"Find all references" of the constant IDC_IGNORE_SMALLTIMEDIFF after looking at PropCompareFolder more closely.	Yes	Need to change the identifier to be more readable. This is a minor change. The identifier will need to refer to a text field rather than a check box as is currently done. This will require a change to Merge.rc (as we discovered doing change request #2).
9.	Merge.rc2		Yes	Need to modify this to change our gui appropriately.

I don't think Visual Studio requires any more tools to find target code. However, after this experience I can say that it could use better highlighting capabilities – particularly a highlighted scrollbar like Eclipse has – as this would make finding references to a variable in a file much easier.

4. Learning process:


FolderCmp
+prepAndCompareTwoFiles()
1
*
CDiffContext
+m_bIgnoreSmallTimeDiff : bool
TimeSizeCompare
+SetAdditionalOptions()
1
*
*
1
CDirDoc
+Rescan()
«uses»
CMergeDoc
«uses»
+IsFileChangedOnDisk()
PropCompareFolder
+ReadOptions()
+WriteOptions()
+DoDataExchange()
«uses»
COptionsMgr
+GetBool()
+SetBool()
+GetInt()
+SetInt()
«uses»
{resource.h, Merge.rc2}

5. Description of the implementation:

This is a pseudo-implementation; we could not get the project to compile. Additions/changes that we feel need to be made are in green. For readability, the constant "OPT_IGNORE_SMALL_FILETIME" has been refactored to "OPT_TIME_TOLERANCE." Comments (sadly, also in green) have been left in where appropriate to help the reader understand what the code is doing.

<MergeDoc.cpp>

The code used to load the time difference tolerance can be replaced by a call to the options manager, like so:

```
CMergeDoc::FileChange CMergeDoc::IsFileChangedOnDisk(LPCTSTR szPath, DiffFileInfo &dfi,
       bool bSave, int nBuffer)
{
       DiffFileInfo *fileInfo = NULL;
       bool bFileChanged = false;
       int tolerance = GetOptionsMgr()->GetInt(OPT_TIME_TOLERANCE);

       if (bSave)
              fileInfo = m_pSaveFileInfo[nBuffer];
       else
              fileInfo = m_pRescanFileInfo[nBuffer];

       …

       __int64 timeDiff = dfi.mtime - fileInfo->mtime;
       timeDiff = _abs64(timeDiff);
       if ((timeDiff > tolerance) || (dfi.size != fileInfo->size))
       {
              bFileChanged = true;
       }

       …
}
```

<DirDoc.cpp>

The code that sets the field "m_bIgnoreSmallTimeDiff" in "m_pCtxt" can be replaced by a call to the options manager and an integer field:

```
void CDirDoc::Rescan()
{
       …

       m_pCtxt->m_nTolerance = GetOptionsMgr()->GetInt(OPT_TIME_TOLERANCE);
       m_pCtxt->m_bStopAfterFirstDiff = GetOptionsMgr()->
       GetBool(OPT_CMP_STOP_AFTER_FIRST);
       m_pCtxt->m_nQuickCompareLimit = GetOptionsMgr()->GetInt(OPT_CMP_QUICK_LIMIT);
       m_pCtxt->m_bPluginsEnabled = GetOptionsMgr()->GetBool(OPT_PLUGINS_ENABLED);
       m_pCtxt->m_bWalkUniques = GetOptionsMgr()->GetBool(OPT_CMP_WALK_UNIQUE_DIRS);
       m_pCtxt->m_pCompareStats = m_pCompareStats;
       m_pCtxt->m_bRecursive = !!m_bRecursive;

       …
}
```

<DiffContext.h, DiffContext.cpp>
Here, we simply need to replace the Boolean field "m_bIgnoreSmallTimeDiff" with the integer field "m_nTolerance" in both files. This change, as can be seen in the code snippets above and below, cascades out to a few other files.

<TimeSizeCompare.h, TimeSizeCompare.cpp>
The method SetAdditionalOptions(bool) needs to take an integer (time) parameter instead of a bool. The method also needs to record this new integer parameter in an appropriate field (i.e., "m_diffTolerance", or something similar) that will be used to replace "m_ignoreSmallDiff." This new field will be used in the CompareFiles function like so:

```
        if (m_diffTolerance != 0)
        {
                // If option to ignore small timediffs (couple of seconds)
                // is set, decrease absolute difference by allowed diff
                nTimeDiff -= m_diffTolerance;
        }
        …
```

<FolderCmp.cpp>
The method "prepAndCompareTwoFiles" needs to be modified to use the new "m_nTolerance" field in the TimeSizeCompare class:

```
UINT FolderCmp::prepAndCompareTwoFiles(CDiffContext * pCtxt, DIFFITEM &di)
{
        …

        else if (nCompMethod == CMP_DATE || nCompMethod == CMP_DATE_SIZE || nCompMethod ==
        CMP_SIZE)
        {
                if (m_pTimeSizeCompare == NULL)
                        m_pTimeSizeCompare = new TimeSizeCompare();

                m_pTimeSizeCompare->SetAdditionalOptions(pCtxt->m_nTolerance);
                code = m_pTimeSizeCompare->CompareFiles(nCompMethod, di);

        }
        …

        return code;
}
```

<PropFolderCmp.cpp, resources.h, Merge.rc2>
The class in this file is a properties window; it simply needs to read, write, and display different values. To this end, ReadOptions will read in the recorded time tolerance instead of a Boolean value from file and WriteOptions will write the time tolerance entered by the user instead of a Boolean value. This change will also require us to change a Boolean field (m_bIgnoreSmallTimeDiff) to an integer field (m_nTolerance):

```
void PropCompareFolder::ReadOptions()
```

```
{
        m_compareMethod = GetOptionsMgr()->GetInt(OPT_CMP_METHOD);
        m_bStopAfterFirst = GetOptionsMgr()->GetBool(OPT_CMP_STOP_AFTER_FIRST);
        m_nTolerance = GetOptionsMgr()->GetInt(OPT_TIME_TOLERANCE);
        m_bIncludeUniqFolders = GetOptionsMgr()->GetBool(OPT_CMP_WALK_UNIQUE_DIRS);
        m_nQuickCompareLimit = GetOptionsMgr()->GetInt(OPT_CMP_QUICK_LIMIT) / Mega ;
}

…

void PropCompareFolder::WriteOptions()
{
        GetOptionsMgr()->SaveOption(OPT_CMP_METHOD, (int)m_compareMethod);
        GetOptionsMgr()->SaveOption(OPT_CMP_STOP_AFTER_FIRST, m_bStopAfterFirst == TRUE);
        GetOptionsMgr()->SaveOption(OPT_TIME_TOLERANCE, m_nTolerance);
        GetOptionsMgr()->SaveOption(OPT_CMP_WALK_UNIQUE_DIRS, m_bIncludeUniqFolders ==
TRUE);

        …
}
```

DoDataExchange will display this new time tolerance field in a text box rather than using a check box.

```
void PropCompareFolder::DoDataExchange(CDataExchange* pDX)
{
        …
        DDX_Check(pDX, IDC_COMPARE_STOPFIRST, m_bStopAfterFirst);
        DDX_Text(pDX, IDC_TIME_DIFF_TOLERANCE, m_nTolerance);
        DDX_Check(pDX, IDC_COMPARE_WALKSUBDIRS, m_bIncludeUniqFolders);
        …
}
```

This will require a modification to resources.h to add a new constant (IDC_TIME_DIFF_TOLERANCE; not shown here due to the simplicity) and a modification of Merge.rc2 to add the text box to the GUI. The change would be akin to the following:

```
IDD_PROPPAGE_COMPARE_FOLDER DIALOGEX 0, 0, 235, 205
STYLE DS_SETFONT | DS_FIXEDSYS | WS_POPUP | WS_CAPTION
CAPTION "Folder"
FONT 8, "MS Shell Dlg", 00, 0, 0x1
BEGIN
    …
    LTEXT           Ign&ore time differences less than (seconds):",IDC_STATIC,7,60,200,8
    EDITTEXT        IDC_TIME_DIFF_TOLERANCE,7,70,50,14,ES_AUTOHSCROLL,ES_NUMBER
    …
END
```

6. Sources:

(No additional sources)