How CMVC exploits SCCS

Document Number TR 29.3317

Angel Rivera

CMVC Customer Support
IBM Software Solutions
Research Triangle Park, North Carolina, USA

DISCLAIMER:

This technical report is not an official publication from the CMVC group. The author is solely responsible for its contents.

ABSTRACT

This technical report provides examples that illustrate the details on how CMVC interacts with SCCS for handling files: create, checkout/checkin, link, lock, unlock, delete, destroy, etc.

The objective is that this background information may help CMVC administrators to perform workaround procedures to fix certain situations in which the CMVC database and the SCCS versioning system are out of synch.

ITIRC KEYWORDS

- CMVC
- Versioning system

ABOUT THE AUTHOR

ANGEL RIVERA

Mr. Rivera is an Advisory Software Engineer and team lead for the CMVC Direct Customer Support team. He joined IBM in 1989 and since then has worked in the development and support of library systems.

Mr. Rivera has an M.S. in Electrical Engineering from The University of Texas at Austin, and B.S. in Electronic Systems Engineering from the Instituto Tecnológico y de Estudios Superiores de Monterrey, México.

CONTENTS

FIGURES

INTRODUCTION

This technical report provides examples that illustrate the details on how CMVC interacts with SCCS for handling files: create, checkout/checkin, link, lock, unlock, delete, destroy, etc.

The objective is that this background information may help CMVC administrators to perform workaround procedures to fix certain situations in which the CMVC database and the SCCS versioning system are out of synch.

The chapters in this TR are organized as follows:

- Chapter "Relationship between sourceId and SCCS file in the vc tree" on page 5 describes the relationship between certain attributes of the database related to CMVC files and CMVC versions and the related SCCS file in the $HOME/vc tree.
- Chapter "Handling a text file that is not common to other releases" on page 11 describes the actions in CMVC and SCCS when a text file that is not common to other releases is created, checked out and checked in.
- Chapter "Handling a text file that is common with another release" on page 25 describes the actions in CMVC and SCCS when a text file that is common to other releases is linked, checked out and checked in, and what happens when the link is broken.
- Chapter "Handling a binary file that is not common to other releases" on page 55 describes the actions in CMVC and SCCS when a binary file that is not common to other releases is created, checked out and checked in.
- Chapter "Handling a binary file that is common with another release" on page 69 describes the actions in CMVC and SCCS when a binary file that is common to other releases is linked, checked out and checked in, and what happens when the link is broken.

This technical report describes additional material that was originally mentioned in the following technical report:

```
CMVC frequently asked questions:
version control and database synchronization issues
TR 29.2297
```

The DBMS that is used in the examples in this document is DB2 Universal Database (UDB) 5.2. However, the SQL clauses are the same for other DBMSs.

DISCLAIMERS

To avoid misunderstandings with the purpose of this technical report and to better understand its scope, the following disclaimers are in order:

- This technical report is not an official publication from the CMVC group. The author is solely responsible for its contents.
- This technical report was prepared when working with CMVC 2.3.1.4 in AIX 4.2.1 using DB2 Universal Database (UDB) 5.2. Therefore, if you have a different version of the mentioned software, then you may expect some differences in the information or in the procedures described in this technical report.
- This technical report covers information that the author has gathered thru the years while working with the CMVC technical support team.
- It is the intention of this technical report to provide recommendations and guidelines that can be helpful to CMVC administrator. In some cases, the procedures will not be exhaustive, and will just show the overall sequence that has worked before, which might be different in your case.
- Real values that were used in our setup will be used in this technical report. Thus, you will need to customize the commands that you issue to reflect the values that are meaningful to your setup.
- It is assumed that the reader has knowledge of CMVC, the appropriate database management system (DBMS) and the appropriate operating system.

  This technical report is not a substitute to the information provided by CMVC and the appropriate DBMS and operating system. Please refer to the appropriate documentation provided with the corresponding software.

HOW TO GET THE MOST UP TO DATE VERSION OF THIS TECHNICAL REPORT.

The most up to date version of this technical report can be obtained from the IBM CMVC ftp site at URL:

`ftp://ftp.software.ibm.com/ps/products/cmvc/doc/tr/trcmfvc2.txt`

For the list of available technical reports, see the file:
ftp://ftp.software.ibm.com/ps/products/cmvc/doc/tr/README.index.txt

ACKNOWLEDGEMENTS

Many of the questions and answers that are compiled in this technical report were obtained from the CMVC forum in the IBMPC conferencing disk and from the CMVC6000 forum in the IBMUNIX conferencing disk. We want to thank the main participants in these electronic forums for their support!

We want to thank in particular the following co-workers:

- Lee Perlov, Websphere/VisualAge TeamConnection Services, IBM RTP, North Carolina, USA.
- Edna Wong Kyu, OEM Lab in IBM RTP, North Carolina, USA.
- Keith Purcell, OEM Lab in IBM RTP, North Carolina, USA.

RELATIONSHIP BETWEEN SOURCEID AND SCCS FILE IN THE VC TREE

This chapter describes the relationship between certain attributes of the database related to CMVC files and CMVC versions and the related SCCS file in the $HOME/vc tree.

For more details on the tables and views used in the CMVC database, see the "CMVC User's Reference", chapters 6 (Views) and 7 (Tables).

SEQUENCE TABLE: LASTSERIAL ATTRIBUTE FOR 'SOURCE'

The lastSerial attribute for the record named "source" in the Sequence table indicates what is the latest id number given to the sourceId attribute for the Files table.

Only when a new file is created in CMVC, this lastSerial value is read, then it is increased by 1 and the resulting value will be the actual sourceId used for the new file.

The rest of the CMVC File actions will not cause the modification of this attribute.

To find out the value for the lastSerial attribute for the record "source" in the Sequence table, you can issue the following command:

```
db2 "select * from Sequence where name='source'"
```

FILES TABLE (FILEVIEW VIEW): SOURCEID ATTRIBUTE

Every file stored in CMVC has associated in the File table or in the FileView view of the database, an attribute called "sourceId". The value for this attribute is an integer (up to 6 digits) that indicate the precise location of the SCCS file that contains the change deltas; this SCCS file is located inside a hierarchical structure under the $HOME/vc directory, which is called the "vc tree".

To find out all the values for the sourceId attributes in the Files table, you can issue the following command:

```
db2 "select sourceId,nuVersionId,id,baseName from Files"
```

The structure for the SCCS files is different between the 2 main different file types in CMVC, because each type is handled differently (text files use forward deltas and binary files use backward deltas):

- Text files are described in “Relationship between sourceId and vc tree for a text file” on page 7.
- Binary files are described in “Relationship between sourceId and vc tree for a binary file” on page 8.

VERSIONS TABLE: SOURCEID ATTRIBUTE

Every CMVC File action that modifies the version number (SID) of a file, such as the creation of the file (version 1.1) and the checkin of a file (such as 1.2), will create a new record in the Versions table that is associated with the sourceId for the file from the Files table.

To find out all the values for the sourceId attributes in the Version table, you can issue the following command:

```
db2 "select sourceId,id,previousId,SID from Versions"
```

ALL THE INFORMATION THAT SCCS USES IS IN THE VC TREE

CMVC maintains the information about the files and versions in several tables and views, and the detailed discussion of the relationship is outside the scope of this technical report, because the user should use CMVC commands to handle the files, and should not manually change the records in the database.

SCCS maintains the information about the files and their changes entirely in the vc tree directory by means of the s.XX and p.XX files. That is, there is no special database that SCCS uses under the covers. To solve some CMVC synchronization problems, it might be necessary to manually manipulate the SCCS files; it is strongly recommended to make a backup of these files (by using a name that SCCS will not use, to avoid confusion) before attempting to manipulate them.

USING THE VCPATH TOOL TO FIND OUT THE SCCS FILE

The CMVC server tool "vcPath" can be used to find out the name of the SCCS file/directory and where is located. For example:

```
vcPath text.txt rel1
```

The output should look like this:

```
The file, text.txt, associated with release,
rel1, maps to the file, s.45, in the
directory, $HOME/vc/0/1/2/3, of the CMVC family's account
```

RELATIONSHIP BETWEEN SOURCEID AND VC TREE FOR A TEXT FILE

Finding the SCCS file for a text file given a sourceId value

Let's assume that a text file created in CMVC is named "file1" and belongs to the release "rel1". The following SQL statement can be used to find out the sourceId for this file:

```
db2 "select sourceId from FileView where baseName='file1' and \
     releaseName='rel1' "
```

In this example, let's assume that the value is "12345". The first step is to prefix the number with enough zeros to make the number to be a 6 digit number; in this case "012345".

Given the sourceId of "012345" the 4 most significant digits ("0123") will indicate the hierarchy of subdirectories inside the $HOME/vc tree:

```
$HOME/vc/0/1/2/3/
```

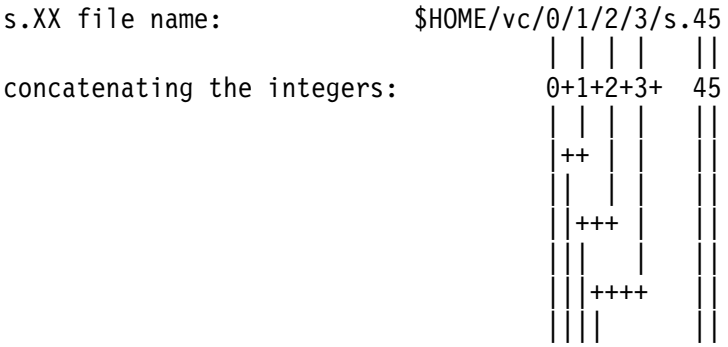
Then, the 2 least significant digits ("45") indicate the SCCS file that has the actual deltas for the CMVC file. The name for this SCCS file for a text CMVC file has the format "s.XX" where the XX is a placeholder for the last 2 significant digits from the sourceId attribute, in this case, the SCCS file name is "s.45".

Therefore, the full path name for the SCCS file that has the delta changes for the CMVC file that has the value "012345" for the attribute "sourceId" in FileView, is:

```
$HOME/vc/0/1/2/3/s.45
```

Finding the sourceId value given the full path name of a SCCS file

If you have the full pathName of the s.XX file in the CMVC vc tree, such as $HOME/vc/0/1/2/3/s.45, then you obtain the sourceId by concatenating all the integers that appear in the vc subdirectories and the file name, from left to right:



```
                                    ||||++++++++
                                    ||||||
CMVC sourceId in FileView:          012345
```

Once you have the sourceId, you can issue the following command to find out the file name and its release:

```
db2 "select releaseName,baseName from FileView where sourceId=012345"
```

Also, you could not specify the leading zeros:

```
db2 "select releaseName,baseName from FileView where sourceId=12345"
```

In both cases, the output should look like this:

```
RELEASENAME           BASENAME
--------------------- ------------------------------------
rel1                  file1
```

RELATIONSHIP BETWEEN SOURCEID AND VC TREE FOR A BINARY FILE

Finding the SCCS directory for a binary file given a sourceId value

Let's assume that a binary file created in CMVC is named "file2" and belongs to the release "rel1". The following SQL statement can be used to find out the sourceId for this file:

```
db2 "select sourceId from FileView where baseName='file2' and \
     releaseName='rel1' "
```

In this example, let's assume that the value is "6789". The first step is to prefix the number with enough zeros to make the number to be a 6 digit number; in this case "006789".

Given the sourceId of "006789" the 4 most significant digits ("0067") will indicate the hierarchy of subdirectories inside the $HOME/vc tree:

```
$HOME/vc/0/0/6/7/
```

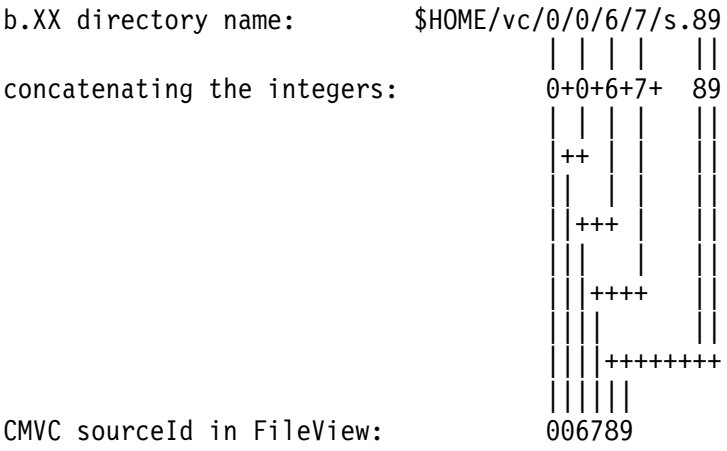
Then, the 2 least significant digits ("89") indicate the SCCS directory that has the actual deltas for the CMVC file. The name for this SCCS directory for a binary CMVC file has the format "b.XX" where the XX is a placeholder for the last 2 significant digits from the sourceId attribute, in this case, the SCCS directory name is "b.89".

Therefore, the full path name for the SCCS directory that has the delta changes for the CMVC file that has the value "006789" for the attribute "sourceId" in FileView, is:

```
$HOME/vc/0/0/6/7/b.89/
```

Finding the sourceId value given the full path name of a SCCS directory

If you have the full pathName of the b.XX directory in the CMVC vc tree, such as $HOME/vc/0/0/6/7/b.89, then you obtain the sourceId by concatenating all the integers that appear in the vc subdirectories and the file name, from left to right:



Once you have the sourceId, you can issue the following command to find out the file name and its release:

```
db2 "select releaseName,baseName from FileView where sourceId=006789"
```

Also, you could not specify the leading zeros:

```
db2 "select releaseName,baseName from FileView where sourceId=6789"
```

In both cases, the output should look like this:

```
RELEASENAME            BASENAME
---------------------- ------------------------------------
rel1                   file2
```

HANDLING A TEXT FILE THAT IS NOT COMMON TO OTHER RELEASES

This chapter describes the actions in CMVC and SCCS when a text file that is not common to other releases is:

- Created: see “Creating a text file” on page 12.
- Checked out: see “Checking out a text file” on page 15.
- Checked in: see “Checking in a text file” on page 17.
- Locked: see “Locking a text file” on page 20.
- Unlocked: see “Unlocking a text file” on page 21.

These actions are taken in a CMVC family that already has existing objects.

ESTABLISHING A BASELINE

Before creating a new file, it is necessary to establish a baseline of the appropriate tables in the CMVC database.

- We need to find out the value from the Sequence table inside the CMVC database, by issuing the following command:

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The result will look like this:

  ```
  NAME      LASTSERIAL
  --------  ----------
  source            11
  ```

  The record for "source" has a value of lastSerial of 11 and this is the value used in the attribute "sourceId" of the latest file created in CMVC.

  Furthermore, this means that the next source id will be the current value plus 1, which in this case is 12; that is, if a new file is created in CMVC, then it will have a sourceId value of 12.

- The last entry for the Files table is found by doing:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files order by sourceId"
  ```

  The last entry is:

  ```
  SOURCEID  NUVERSIONID    ID   BASENAME
        11          129   130   superHello.C
  ```

Notice the following:

- The sourceId value is 11, which comes from the Sequence table.
- The nuVersionId is 129, which comes from the Versions table.
- The id is 130, which is the unique identifier for the File table.

- The last entry for the Versions table is found by doing:

  ```
  db2 "select sourceId,id,previousId,SID from Versions order by sourceId"
  ```

  The last entry is:

  ```
  SOURCEID     ID   PREVIOUSID   SID
        11    129            0   1.1
  ```

  Notice the following:

  - The sourceId value is 11, which comes from the Sequence table.
  - The id is 129, which is the unique identifier for the Versions table.

    This id is cross-referenced with the nuVersionId attribute in the Files table.
  - The previousId is 0, which refers to an id in the Versions table that represents the previous id.
  - The SID or version number, such as 1.1.

CREATING A TEXT FILE

Using the CMVC file command to create the text file

Let's assume that the file "text.txt" is created and has only the following line:

```
This is a text file.
```

Then this file is created in the CMVC family as a text file:

```
File -relative /home/cmpc3db2/work -create text.txt -release rel1 \
     -component dev -defect 1 -remarks "Creating text file"   \
     -verbose
```

The informational message is:

```
File text.txt was created successfully.
```

Displaying information about the CMVC file

The following command can be used to verify the creation of this file:

```
Report -view FileView -where "baseName in ('text.txt')"
```

The summary of the result is as follows:

```
nuPathName   releaseName   compName  nuVersion VersionSID lastUpdate
------------ ------------- --------- --------- ---------- ----------
text.txt     rel1          dev       1.1                  2000/04/18
```

The details of the information inside CMVC about this file can be obtained by issuing:

```
File  -long -view text.txt -release rel1 | more
```

The output is:

```
baseName        text.txt
releaseName     rel1
compName        dev
versionSID
addDate
dropDate
lastUpdate      2000/04/18 14:39:22
pathName        text.txt
nuVersionSID    1.1
nuAddDate       2000/04/18 14:39:22
nuDropDate
nuPathName      text.txt
userLogin
file mode       0644

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- ----------
    1.1         cmpc3db2        1               create   creating file

versions:
     user     date       SID
     -------- ---------- -----------------------------------------------
     cmpc3db2 2000/04/18 1.1
Creating text file.
```

Verifying the attributes

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

```
NAME      LASTSERIAL
source            12
```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        12          152       153  text.txt
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
  ```

Information about the new SCCS file

Given a sourceId of "12" (in reality "000012"), we can determine that there should be a new SCCS delta file with the following full path:

```
ls -al $HOME/vc/0/0/0/0/s.12
```

The file permissions for this file are (showing relevant data only):

```
-r--r--r--  1  cmpc3db2 db2iadm2  /home/cmpc3db2/vc/0/0/0/0/s.12
```

It is important to notice that the file is fully owned by the CMVC family and that it has 444 file permissions.

The contents of s.12 is:

```
^Ah11283
^As 00001/00000/00000
^Ad D 1.1 00/04/18 14:50:51 cmpc3db2 1 0
^Ac date and time created 00/04/18 14:39:23 by cmpc3db2
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AE 1
```

The important data is in the 3rd row: the item after the capital D is the version of the file 1.1

Notice that the year information is shown with only 2 digits. The operating system needs to have the proper Year 2000 related patches for SCCS in order to SCCS to handle correctly the Year 2000. In this case, SCCS interprets all the 00 year as being really 2000 and 99 as being 1999.

CHECKING OUT A TEXT FILE

Using the CMVC file command to check out a text file

Issue the following CMVC File command to checkout the text.txt file:

```
File -checkout text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.1
new delta 1.2
1 line
File text.txt was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view text.txt -release rel1 | more
  ```

  The only change when viewing the information inside CMVC for this file that is checked out, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.2       2000/04/18 14:49:35 rel1            text.txt
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  text.txt    rel1            2000/04/18 1.2          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        12          152       153  text.txt
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.1 1.2 cmpc3db2 00/04/18 14:59:35
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  - 1.1 indicates the current version.
  - 1.2 indicates that is going to be the next version.
- Notice that the s.12 file is not touched at all.

CHECKING IN A TEXT FILE

Using the CMVC file command to check in a text file

Use vi to add a 2nd line to the checked out file:

```
This is a text file.
Adding line 2.
```

Issue the following CMVC File command to checkin the text.txt file:

```
File -checkin  text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "First checkin, adding line 2"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.2
1 inserted
0 deleted
1 unchanged
File text.txt was checked in successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view text.txt -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel1
lastUpdate      2000/04/18 15:05:31
pathName        text.txt
nuVersionSID    1.2
nuAddDate       2000/04/18 14:39:22

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- ----------
    1.1         cmpc3db2        1               create   creating file
    1.2         cmpc3db2        1               delta    creating file

versions:
     user     date       SID
     -------- ---------- ------------------------------------------
     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highligths are:

- Sequence: no change.
- Files: the nuVersionId now points to the new record in Versions.
- Versions: a new record was added for SID 1.2.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

```
db2 "select sourceId,nuVersionId,id,baseName from Files \
    where baseName='text.txt'"
```

  In this case, the result is:

```
SOURCEID  NUVERSIONID        ID  BASENAME
      12          154       153  text.txt
```

- The main data for the Versions records associated with this sourceId is found as follows:

```
db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
```

  In this case, the result is:

```
SOURCEID     ID    PREVIOUSID    SID
      12    152             0    1.1
      12    154           152    1.2
```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.2, the same number mentioned in the p.XX file.

- The s.12 file was modified as follows:

```
^Ah16066
^As 00001/00000/00001
^Ad D 1.2 00/04/18 15:05:32 cmpc3db2 2 1
^Ae
^As 00001/00000/00000
^Ad D 1.1 00/04/18 14:39:23 cmpc3db2 1 0
^Ac date and time created 00/04/18 14:39:23 by cmpc3db2
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AI 2
Adding line 2.
^AE 2
^AE 1
```

  Notice that the header has now an entry for the Delta 1.2.

LOCKING A TEXT FILE

Using the CMVC file command to lock a text file

Issue the following CMVC File command to lock the text.txt file:

```
File -lock text.txt -release rel1 -verbose
```

The informational message after a successful checkout looks like:

```
1.2
new delta 1.3
2 lines
File text.txt was locked successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view text.txt -release rel1 | more
  ```

  The only change when viewing the information inside CMVC for this file that is locked, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3       2000/04/18 15:13:15 rel1            text.txt
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  text.txt    rel1            2000/04/18 1.3          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        12          154       153  text.txt
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 groupid /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.2 1.3 cmpc3db2 00/04/18 15:13:15
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a lock of the file:

  – 1.2 indicates the current version.

  – 1.3 indicates that is going to be the next version.

- Notice that the s.12 file is not touched at all.

UNLOCKING A TEXT FILE

Using the CMVC file command to unlock a text file

Issue the following CMVC File command to unlock the text.txt file:

```
File -unlock text.txt -release rel1 -verbose
```

The informational message after a successful checkin looks like:

```
The unlock action for file text.txt
was completed successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view text.txt -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
locks: none.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

```
db2 "select sourceId,nuVersionId,id,baseName from Files \
    where baseName='text.txt'"
```

  In this case, the result is:

```
SOURCEID  NUVERSIONID        ID  BASENAME
      12          154       153  text.txt
```

- The main data for the Versions records associated with this sourceId is found as follows:

```
db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
```

  In this case, the result is:

```
SOURCEID     ID    PREVIOUSID    SID
      12    152             0    1.1
      12    154           152    1.2
```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.
- The s.12 file was not modified.
- The version 1.3 (mentioned in the p.12 before it was deleted) was not created. That is, the current version remains 1.2.

HANDLING A TEXT FILE THAT IS COMMON WITH ANOTHER RELEASE

This chapter describes the actions in CMVC and SCCS when a text file that is common with another release is:

- Linked: see “Linking a text file.”
- Checked out (common): see “Checking out a text file (common)” on page 27.
- Checked in (common): see “Checking in a text file (common)” on page 29.
- Checked out (common link to be broken): see “Checking out a text file (common link to be broken)” on page 33.
- Checked in (breaking the common link): see “Checking in a text file (breaking the common link)” on page 35.
- Checked out (common link broken: rel2): see “Checking out a text file whose common link was broken (rel2)” on page 40.
- Checked in (common link broken: rel2): see “Checking in a text file (from rel2)” on page 43.
- Checked out (common link broken: rel1): see “Checking out a text file whose common link was broken (rel1)” on page 47.
- Checked in (common link broken: rel1): see “Checking in a text file (from rel1)” on page 48.

These actions are taken in a CMVC family that already has existing objects.

The following diagram may help to understand the versions of the file that is common between 2 releases and then the link is broken and then the non-common files are further processed:

```
1.3  common version between rel1 and rel2
 |
 +---> 1.4 --> 1.5 versions exclusive for rel2 (common link is broken)
 |
 +---> 1.3.1.1     version exclusive for rel1 (common link is broken)
```

LINKING A TEXT FILE

Using the CMVC file command to unlock a text file

Issue the following CMVC File command to link the text.txt file from rel1 to rel2:

```
File -link text.txt -release rel1 -to rel2 -verbose -defect 1
```

The informational message after a successful link looks like:

```
File text.txt was linked successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view text.txt -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
common files:
    releaseName     SID         pathName
    --------------- ----------- -----------------
    rel2            1.2         text.txt
```

Notice that if the File -view command specified the release rel2:

```
File  -long -view text.txt -release rel1 | more
```

Then the information about the file will show the section about common files reflecting the release rel1:

```
common files:
    releaseName     SID         pathName
    --------------- ----------- -----------------
    rel1            1.2         text.txt
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new record for sourceId.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

The value in this case is:

```
NAME      LASTSERIAL
source            12
```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          154       153        147 text.txt
        12          154       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- No existing files were modified in the vc tree. That is, the s.12 file was not modified.
- No new files were created in the vc tree.

CHECKING OUT A TEXT FILE (COMMON)

Using the CMVC file command to check out a text file

Issue the following CMVC File command to checkout the text.txt file which is common to rel1 and rel2:

```
File -checkout text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.2
new delta 1.3
2 lines
A common file is associated with release rel2.
File text.txt was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view text.txt -release rel1 | more
    or
  File  -long -view text.txt -release rel2 | more
  ```

  The only change when viewing the information inside CMVC for this common file that is checked out, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3       2000/04/18 15:26:38 rel1            text.txt
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  text.txt    rel1            2000/04/18 1.3          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

```
db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
    where baseName='text.txt'"
```

In this case, the result is:

```
SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
      12          154       153        147 text.txt
      12          154       156        148 text.txt
```

The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.2 1.3 cmpc3db2 00/04/18 15:26:38
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.2 indicates the current version.

  – 1.3 indicates that is going to be the next version.

- Notice that the s.12 file is not touched at all.

CHECKING IN A TEXT FILE (COMMON)

Using the CMVC file command to check in a text file

Use vi to add a 3rd line to the checked out file:

```
This is a text file.
Adding line 2.
Adding line 3.
```

Issue the following CMVC File command to checkin the common text.txt file:

```
File -checkin  text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Second checkin, adding line 3"   \
     -common rel2
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.3
1 inserted
0 deleted
2 unchanged
File text.txt was checked in successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view text.txt -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel1
lastUpdate      2000/04/18 15:31:19
pathName        text.txt
nuVersionSID    1.3
nuAddDate       2000/04/18 14:39:22

locks: none.

common files:
    releaseName     SID         pathName
    --------------- ----------- ------------
    rel2            1.3         text.txt

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        1               create   creating text
    1.2         cmpc3db2        1               delta    creating text
    1.3         cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- ----------------------------------------------

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new values for nuVersionId for both records.
- Versions: new record for SID 1.3.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          157       153        147 text.txt
        12          157       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
        12    157           154    1.3
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.3, the same number mentioned in the p.XX file.

- The s.12 file was modified as follows:

  ```
  ^Ah20853
  ^As 00001/00000/00002
  ^Ad D 1.3 00/04/18 15:31:19 cmpc3db2 3 2
  ^Ae
  ^As 00001/00000/00001
  ^Ad D 1.2 00/04/18 15:05:32 cmpc3db2 2 1
  ^Ae
  ^As 00001/00000/00000
  ^Ad D 1.1 00/04/18 14:39:23 cmpc3db2 1 0
  ^Ac date and time created 00/04/18 14:39:23 by cmpc3db2
  ```

```
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AI 2
Adding line 2.
^AI 3
Adding line 3.
^AE 3
^AE 2
^AE 1
```

Notice that the header has now an entry for the Delta 1.3.

CHECKING OUT A TEXT FILE (COMMON LINK TO BE BROKEN)

Using the CMVC file command to check out a text file

Issue the following CMVC File command to checkout the text.txt file which is common to rel1 and rel2, but which link will be broken during the checkin:

```
File -checkout text.txt -release rel2 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.3
new delta 1.4
3 lines
A common file is associated with release rel1.
File text.txt was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view text.txt -release rel1 | more
    or
  File  -long -view text.txt -release rel2 | more
  ```

  The only change when viewing the information inside CMVC for this common file that is checked out, is the following:

```
locks:
    userLogin    newSID    checkOutDate        releaseName     fileNuPath
    ------------ --------- ------------------- --------------- ----------
    cmpc3db2     1.4       2000/04/18 15:36:03 rel2            text.txt
```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  text.txt    rel2            2000/04/18 1.4          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          157       153        147 text.txt
        12          157       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

```
SOURCEID     ID    PREVIOUSID    SID
      12    152             0    1.1
      12    154           152    1.2
      12    157           154    1.3
```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.3 1.4 cmpc3db2 00/04/18 15:36:03
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.3 indicates the current version.

  – 1.4 indicates that is going to be the next version.

- Notice that the s.12 file is not touched at all.

CHECKING IN A TEXT FILE (BREAKING THE COMMON LINK)

Using the CMVC file command to check in a text file

Use vi to add a 4th line to the checked out file:

```
This is a text file.
Adding line 2.
Adding line 3.
Adding line 4.
```

Issue the following CMVC File command to checkin the text.txt file and do not specify neither the -common nor the -force flag (this is for illustration purposes):

```
File -checkin  text.txt -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, adding line 4"
```

The error message looks like:

```
0010-050 The file, text.txt, that you are checking in
         is common with a file associated with release rel1.
         You must specify the force option if you want to break the
         common link or you must specify the common releases to maintain
         file commonality.
0010-261 File text.txt associated with release
         rel2 cannot be checked in.
```

Issue the following CMVC File command to checkin the text.txt file and use the -force flag to break the common link with release rel1.

```
File -checkin  text.txt -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, adding line 4"    \
     -force
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.4
1 inserted
0 deleted
3 unchanged
File text.txt was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view text.txt -release rel1 | more
```

Notice that the basic information for this file remains the same at version 1.3 (that is, version 1.3 is still common to rel1 and rel2, but version 1.4 is only applicable to rel2, not to rel1):

```
releaseName     rel1
lastUpdate      2000/04/18 15:31:19
pathName        text.txt
nuVersionSID    1.3
nuAddDate       2000/04/18 14:39:23

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        1               create   creating text
    1.2         cmpc3db2        1               delta    creating text
    1.3         cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- ----------------------------------------------

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
common files: none.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view text.txt -release rel2 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel1
lastUpdate      2000/04/18 15:40:33
pathName        text.txt
nuVersionSID    1.4
nuAddDate       2000/04/18 14:39:22

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.2         cmpc3db2        1               link     creating text
    1.3         cmpc3db2        1               delta    creating text
    1.4         cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- -------------------------------------------

     cmpc3db2 2000/04/18 1.4
Third checkin, adding line 4

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new nuVersionId value for the file in rel2 (relId=148).
- Versions: new record for SID 1.4 whose previousId is SID 1.3.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          157       153        147 text.txt
        12          158       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
        12    157           154    1.3
        12    158           157    1.4
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.4, the same number mentioned in the p.XX file. This new version is only for rel2; rel1 still sees version 1.3 as the latest version.

- The s.12 file was modified as follows:
-

```
^Ah25645
^As 00001/00000/00003
^Ad D 1.4 00/04/18 15:40:34 cmpc3db2 4 3
^Ae
^As 00001/00000/00002
^Ad D 1.3 00/04/18 15:31:19 cmpc3db2 3 2
^Ae
^As 00001/00000/00001
^Ad D 1.2 00/04/18 15:05:32 cmpc3db2 2 1
^Ae
^As 00001/00000/00000
^Ad D 1.1 00/04/18 14:39:23 cmpc3db2 1 0
^Ac date and time created 00/04/18 14:39:23 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AI 2
Adding line 2.
^AI 3
Adding line 3.
^AI 4
Adding line 4.
^AE 4
^AE 3
^AE 2
^AE 1
```

  Notice that the header has now an entry for the Delta 1.4.

CHECKING OUT A TEXT FILE WHOSE COMMON LINK WAS BROKEN (REL2)

Using the CMVC file command to check out a text file

Issue the following CMVC File command to checkout the text.txt file which was common between rel1 and rel2, but which link was broken during the latest checkin:

```
File -checkout text.txt -release rel2 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.4
new delta 1.5
4 lines
File text.txt was checked out successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue the CMVC command to display the information about the file from the release rel1.

```
File  -long -view text.txt -release rel1 | more
```

There are no new changes, because the file is no longer common. The current version is 1.3.

Displaying information about the CMVC file (from release rel2)

- Let's issue the CMVC command to display the information about the file from the release rel2.

  ```
  File  -long -view text.txt -release rel2 | more
  ```

  Notice that only the lock section has changed:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.5       2000/04/19 09:34:29 rel2            text.txt
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  text.txt    rel2            2000/04/19 1.5          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          157       153        147 text.txt
        12          158       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
        12    157           154    1.3
        12    158           157    1.4
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.4 1.5 cmpc3db2 00/04/19 09:34:29
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.4 indicates the current version.

  – 1.5 indicates that is going to be the next version.

- Notice that the s.12 file is not touched at all.

CHECKING IN A TEXT FILE (FROM REL2)

Using the CMVC file command to check in a text file

Use vi to add a 5th line to the checked out file:

```
This is a text file.
Adding line 2.
Adding line 3.
Adding line 4.
Adding line 5.
```

Issue the following CMVC File command to checkin the text.txt file:

```
File -checkin  text.txt -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Fourth checkin, adding line 5"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.5
1 inserted
0 deleted
4 unchanged
File text.txt was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view text.txt -release rel1 | more
```

The basic information for this file remains the same at version 1.3 (that is, version 1.3 is still common to rel1 and rel2, but version 1.5 is only applicable to rel2, not to rel1):

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view text.txt -release rel2 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel2
lastUpdate      2000/04/19 09:39:00
pathName        text.txt
nuVersionSID    1.5
nuAddDate       2000/04/18 15:19:37

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.2         cmpc3db2        1               link     creating text
    1.3         cmpc3db2        1               delta    creating text
    1.4         cmpc3db2        1               delta    creating text
    1.5         cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- ------------------------------------------

     cmpc3db2 2000/04/19 1.5
Fourth checkin, adding line 5

     cmpc3db2 2000/04/18 1.4
Third checkin, adding line 4

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new nuVersionId of 159 to point to the new SID 1.5.
- Versions: new version with id 159 for SID 1.5.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          157       153        147 text.txt
        12          159       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
        12    157           154    1.3
        12    158           157    1.4
        12    159           158    1.5
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.4, the same number mentioned in the p.XX file. This new version is only for rel2; rel1 still sees version 1.3 as the latest version.

- The s.12 file was modified as follows:

-

```
^Ah30454
^As 00001/00000/00004
^Ad D 1.5 00/04/19 09:39:00 cmpc3db2 5 4
^Ae
^As 00001/00000/00003
^Ad D 1.4 00/04/18 15:40:34 cmpc3db2 4 3
^Ae
^As 00001/00000/00002
^Ad D 1.3 00/04/18 15:31:19 cmpc3db2 3 2
^Ae
^As 00001/00000/00001
^Ad D 1.2 00/04/18 15:05:32 cmpc3db2 2 1
^Ae
^As 00001/00000/00000
^Ad D 1.1 00/04/18 14:39:23 cmpc3db2 1 0
^Ac date and time created 00/04/18 14:39:23 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AI 2
Adding line 2.
^AI 3
Adding line 3.
^AI 4
Adding line 4.
^AI 5
Adding line 5.
^AE 5
^AE 4
^AE 3
^AE 2
^AE 1
```

Notice that the header has now an entry for the Delta 1.5.

CHECKING OUT A TEXT FILE WHOSE COMMON LINK WAS BROKEN (REL1)

Using the CMVC file command to check out a text file

Issue the following CMVC File command to checkout the text.txt file which was common between rel1 and rel2, but which link was broken:

```
File -checkout text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.3
new delta 1.3.1.1
3 lines
File text.txt was checked out successfully.
```

Displaying information about the CMVC file (from release rel2)

Let's issue the CMVC command to display the information about the file from the release rel2.

```
File  -long -view text.txt -release rel2 | more
```

There are no new changes, because the file is no longer common. The version is shown to be 1.5.

Displaying information about the CMVC file (from release rel1)

- Let's issue the CMVC command to display the information about the file from the release rel1.

  ```
  File  -long -view text.txt -release rel1 | more
  ```

  Notice that only the lock section has changed:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3.1.1   2000/04/19 09:46:25 rel1            text.txt
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
text.txt    rel1            2000/04/19 1.3.1.1      cmpc3db2
```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 is created in the same location as the s.12 file, in $HOME/vc/0/0/0/0. This file indicates that there is a lock for file s.12.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/p.12

- The contents of the p.12 file is:

  ```
  1.3 1.3.1.1 cmpc3db2 00/04/19 09:46:25
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.3 indicates the current version.

  – 1.3.1.1 indicates that is going to be the next version.

- Notice that the s.12 file is not touched at all.

CHECKING IN A TEXT FILE (FROM REL1)

Using the CMVC file command to check in a text file

Use vi to modify the 3rd line to the checked out file:

```
This is a text file.
Adding line 2.
Modifying line 3.
```

Issue the following CMVC File command to checkin the text.txt file:

```
File -checkin  text.txt -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, modifying line 3"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.3.1.1
1 inserted
1 deleted
2 unchanged
File text.txt was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view text.txt -release rel1 | more
```

Notice that the new version is 1.3.1.1, which shows a branching with respect to release rel2 (version 1.5):

```
releaseName     rel1
lastUpdate      2000/04/24 09:00:14
pathName        text.txt
nuVersionSID    1.3.1.1
nuAddDate       2000/04/18 14:39:22

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        1               create   creating text
    1.2         cmpc3db2        1               delta    creating text
    1.3         cmpc3db2        1               delta    creating text
    1.3.1.1     cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- ----------------------------------------------

     cmpc3db2 2000/04/24 1.3.1.1
Third checkin, modifying line 3

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view text.txt -release rel2 | more
```

Notice that there are no new changes and that the version is still 1.5.

```
releaseName     rel2
lastUpdate      2000/04/19 09:39:00
pathName        text.txt
nuVersionSID    1.5
nuAddDate       2000/04/18 15:19:37

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.2         cmpc3db2        1               link     creating text
    1.3         cmpc3db2        1               delta    creating text
    1.4         cmpc3db2        1               delta    creating text
    1.5         cmpc3db2        1               delta    creating text

versions:
     user     date       SID
     -------- ---------- ------------------------------------------

     cmpc3db2 2000/04/19 1.5
Fourth checkin, adding line 5

     cmpc3db2 2000/04/18 1.4
Third checkin, adding line 4

     cmpc3db2 2000/04/18 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/18 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/18 1.1
Creating text file.
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new nuVersionId=160 for text.txt for release rel1.
- Versions: new record with id=160 for SID 1.3.1.1 and which has as previousId the id=157 (SID 1.3).

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            12
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,releaseId,baseName from Files \
      where baseName='text.txt'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID BASENAME
        12          160       153        147 text.txt
        12          159       156        148 text.txt
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=12"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        12    152             0    1.1
        12    154           152    1.2
        12    157           154    1.3
        12    158           157    1.4
        12    159           158    1.5
        12    160           157    1.3.1.1
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.12 was removed from the same location as the s.12 file, in $HOME/vc/0/0/0/0.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.3.1.1, the same number mentioned in the p.XX file. This new version is only for rel1; rel2 still sees 1.5 as the latest version.

- The s.12 file was modified as follows:

-

  ```
  ^Ah36154
  ^As 00001/00001/00002
  ^Ad D 1.3.1.1 00/04/24 09:00:15 cmpc3db2 6 3
  ^Ae
  ^As 00001/00000/00004
  ```

```
^Ad D 1.5 00/04/19 09:39:00 cmpc3db2 5 4
^Ae
^As 00001/00000/00003
^Ad D 1.4 00/04/18 15:40:34 cmpc3db2 4 3
^Ae
^As 00001/00000/00002
^Ad D 1.3 00/04/18 15:31:19 cmpc3db2 3 2
^Ae
^As 00001/00000/00001
^Ad D 1.2 00/04/18 15:05:32 cmpc3db2 2 1
^Ae
^As 00001/00000/00000
^Ad D 1.1 00/04/18 14:39:23 cmpc3db2 1 0
^Ac date and time created 00/04/18 14:39:23 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
This is a text file.
^AI 2
Adding line 2.
^AI 3
Adding line 3.
^AI 6
Modifying line 3.
^AE 6
^AI 4
Adding line 4.
^AI 5
Adding line 5.
^AE 5
^AE 4
^AE 3
^AE 2
^AE 1
```

Notice that the header has now an entry for the Delta 1.3.1.1.

HANDLING A BINARY FILE THAT IS NOT COMMON TO OTHER RELEASES

This chapter describes the actions in CMVC and SCCS when a binary file that is not common to other releases is:

- Created: see “Creating a binary file” on page 56.
- Checked out: see “Checking out a binary file” on page 59.
- Checked in: see “Checking in a binary file” on page 61.
- Locked: see “Locking a binary file” on page 64.
- Unlocked: see “Unlocking a binary file” on page 66.

These actions are taken in a CMVC family that already has existing objects.

ESTABLISHING A BASELINE

Before creating a new file, it is necessary to establish a baseline of the appropriate tables in the CMVC database.

- We need to find out the value from the Sequence table inside the CMVC database, by issuing the following command:

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The result will look like this:

  ```
  NAME      LASTSERIAL
  --------  ----------
  source            12
  ```

  The record for "source" has a value of lastSerial of 12 and this is the value used in the attribute "sourceId" of the latest file created in CMVC.

  Furthermore, this means that the next source id will be the current value plus 1, which in this case is 13; that is, if a new file is created in CMVC, then it will have a sourceId value of 13.

- The last entry for the Files table is found by doing:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files order by sourceId"
  ```

  The last entry is:

  ```
  SOURCEID  NUVERSIONID    ID   BASENAME
        12          159   156   text.txt
  ```

Notice the following:

- The sourceId value is 12, which comes from the Sequence table.
- The nuVersionId is 159, which comes from the Versions table.
- The id is 156, which is the unique identifier for the File table.

- The last entry for the Versions table is found by doing:

  ```
  db2 "select sourceId,id,previousId,SID from Versions order by sourceId"
  ```

  The last entry is:

  ```
  SOURCEID     ID   PREVIOUSID   SID
        12    160          157   1.3.1.1
  ```

  Notice the following:

  - The sourceId value is 12, which comes from the Sequence table.
  - The id is 160, which is the unique identifier for the Versions table.

    This id is cross-referenced with the nuVersionId attribute in the Files table.
  - The previousId is 157, which refers to an id in the Versions table that represents the previous id.
  - The SID or version number, such as 1.3.1.1.

CREATING A BINARY FILE

Using the CMVC file command to create the binary file

Let's assume that the file "binary.exe" is created and has only the following line:

```
This is a binary file.
```

This binary file has a text string in order to better follow the handling of the corresponding SCCS file.

Then this file is created in the CMVC family as a binary file:

```
File -relative /home/cmpc3db2/work -create binary.exe -release rel1 \
     -component dev -defect 1 -remarks "Creating binary file"   \
     -verbose -binary
```

The informational message is:

```
File binary.exe was created successfully.
```

Displaying information about the CMVC file

The following command can be used to verify the creation of this file:

```
Report -view FileView -where "compName in ('dev')"
```

The summary of the result is as follows:

```
nuPathName   releaseName   compName  nuVersion VersionSID lastUpdate
------------ ------------- --------- --------- ---------- ----------
binary.exe   rel1          dev       1.1                  2000/04/24
```

The details of the information inside CMVC about this file can be obtained by issuing:

```
File  -long -view binary.exe -release rel1 | more
```

The output is:

```
baseName        binary.exe
releaseName     rel1
compName        dev
versionSID
addDate
dropDate
lastUpdate      2000/04/24 09:16:25
pathName        binary.exe
nuVersionSID    1.1
nuAddDate       2000/04/24 09:16:25
nuDropDate
nuPathName      binary.exe
userLogin
file mode       0644

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- ----------
    1.1         cmpc3db2        1               create   creating file

versions:
     user     date       SID
     -------- ---------- ---------------------------------------------
     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

Verifying the attributes

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        13          162       163  binary.exe
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
  ```

Information about the new SCCS file

Given a sourceId of "13" (in reality "000013"), we can determine that there should be a new SCCS delta file with the following full path:

```
ls -al $HOME/vc/0/0/0/0/b.13
```

The file permissions for this directory are (showing relevant data only):

```
drwxr-xr-x  cmpc3db2 db2iadm2  /home/cmpc3db2/vc/0/0/0/0/b.13
```

The b.13 directory contains the following files:

```
-rw-r--r--  1  cmpc3db2 db2iadm2  1.1
-r--r--r--  1  cmpc3db2 db2iadm2  s.binary
```

It is important to notice that the files are fully owned by the CMVC family.

The contents of b.13/1.1 is:

```
This is a binary file.
```

The contents of b.13/s.binary is:

```
^Ah08344
^As 00000/00000/00000
^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
^Ac date and time created 00/04/24 09:16:25 by cmpc3db2
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
^AE 1
```

The important data is in the 3rd row: the item after the capital D is the version of the file 1.1

Notice that the year information is shown with only 2 digits. The operating system needs to have the proper Year 2000 related patches for SCCS in order to SCCS to handle correctly the Year 2000. In this case, SCCS interprets all the 00 year as being really 2000 and 99 as being 1999.

CHECKING OUT A BINARY FILE

Using the CMVC file command to check out a binary file

Issue the following CMVC File command to checkout the binary.exe file:

```
File -checkout binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.1
new delta 1.2
0 lines
   Please ignore the above line count information,
   because binary.exe is a binary file
   and the line count returned by SCCS for a binary
   file has no meaning.
File binary.exe was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view binary.exe -release rel1 | more
  ```

The only change when viewing the information inside CMVC for this file that is checked out, is the following:

```
locks:
    userLogin    newSID    checkOutDate        releaseName     fileNuPath
    ------------ --------- ------------------- --------------- ----------
    cmpc3db2     1.2       2000/04/24 09:24:50 rel1            binary.exe
```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  binary.exe  rel1            2000/04/24 1.2          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        13          162       163  binary.exe
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The lastSerial value for the source record in the Sequence table is not modified. This means that a checkout does not alter the sourceId of a file.
- The file p.binary is created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13. This file indicates that there is a lock for file s.binary.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary
- The contents of the p.binary file is:

  ```
  1.1 1.2 cmpc3db2 00/04/24 09:24:50
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.1 indicates the current version.

  – 1.2 indicates that is going to be the next version.
- Notice that the b.13/s.binary file is not touched at all.

CHECKING IN A BINARY FILE

Using the CMVC file command to check in a binary file

Use vi to add a 2nd line to the checked out file:

```
This is a binary file.
Adding line 2.
```

Issue the following CMVC File command to checkin the file:

```
File -checkin  binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "First checkin, adding line 2"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.2
0 inserted
0 deleted
0 unchanged
   Please ignore the above line count information,
   because binary.exe is a binary file
   and the line count returned by SCCS for a binary
   file has no meaning.
File binary.exe was checked in successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view binary.exe -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel1
lastUpdate      2000/04/24 09:29:16
pathName        binary.exe
nuVersionSID    1.2
nuAddDate       2000/04/24 09:16:25

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- ----------
    1.1         cmpc3db2        1               create   creating file
    1.2         cmpc3db2        1               delta    creating file

versions:
     user     date       SID
     -------- ---------- ----------------------------------------------
     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: the nuVersionId=162 points now to a new record in Versions.
- Versions: a new record, id=162, was added for SID 1.2.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        13          164       163  binary.exe
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.2, the same number mentioned in the p.binary file.

- The file $HOME/vc/0/0/0/0/b.13/1.1 was deleted.
- The b.13 directory contains the following files:

  ```
  -rw-r--r--  1  cmpc3db2 db2iadm2  1.2
  -rw-r--r--  1  cmpc3db2 db2iadm2  1.2d
  -r--r--r--  1  cmpc3db2 db2iadm2  s.binary
  ```

  It is important to notice that the files are fully owned by the CMVC family.
- The file 1.2 contains now the most up-to-date version of the file:

  ```
  This is a binary file.
  Adding line 2.
  ```
- The file 1.2d contains the backward delta to generate the 1.1 version from 1.2.

  The contents of the delta file is in binary format and it is not legible.
- The s.binary file was modified as follows:

  ```
  ^Ah11611
  ^As 00000/00000/00000
  ^Ad D 1.2 00/04/24 09:29:16 cmpc3db2 2 1
  ^Ae
  ^As 00000/00000/00000
  ^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
  ^Ac date and time created 00/04/24 09:16:25 by family
  ^Ae
  ^Au
  ^AU
  ^Af j
  ^At
  ^AT
  ^AI 1
  ^AE 1
  ```

  Notice that the header has now an entry for the Delta 1.2.

LOCKING A BINARY FILE

Using the CMVC file command to lock a binary file

Issue the following CMVC File command to lock the file:

```
File -lock binary.exe -release rel1 -verbose
```

The informational message after a successful checkout looks like:

```
1.2
new delta 1.3
0 lines
File binary.exe was locked successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view binary.exe -release rel1 | more
  ```

  The only change when viewing the information inside CMVC for this file that is locked, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3       2000/04/24 09:37:47 rel1            binary.exe
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  binary.exe  rel1            2000/04/24 1.3          cmpc3db2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary is created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13 This file indicates that there is a lock for file b.13/s.binary

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary

- The contents of the p.binary file is:

  ```
  1.2 1.3 cmpc3db2 00/04/24 09:37:47
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a lock of the file:

  – 1.2 indicates the current version.

  – 1.3 indicates that is going to be the next version.

- Notice that the b.13/s.binary file is not touched at all.

UNLOCKING A BINARY FILE

Using the CMVC file command to unlock a binary file

Issue the following CMVC File command to unlock the file:

```
File -unlock binary.exe -release rel1 -verbose
```

The informational message after a successful checkin looks like:

```
The unlock action for file binary.exe
was completed successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view binary.exe -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
locks: none.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

```
NAME      LASTSERIAL
source            13
```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  BASENAME
        13          164       163  binary.exe
  ```

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13
- The s.binary file was not modified.
- The version 1.3 (mentioned in p.binary before it was deleted) was not created. That is, the current version remains 1.2.

HANDLING A BINARY FILE THAT IS COMMON WITH ANOTHER RELEASE

This chapter describes the actions in CMVC and SCCS when a binary file that is common with another release is:

- Linked: see “Linking a binary file.”
- Checked out (common): see “Checking out a binary file (common)” on page 71.
- Checked in (common): see “Checking in a binary file (common)” on page 73.
- Checked out (common link to be broken): see “Checking out a binary file (common link to be broken)” on page 77.
- Checked in (breaking the common link): see “Checking in a binary file (breaking the common link)” on page 79.
- Checked out (common link broken: rel2): see “Checking out a binary file whose common link was broken (rel2)” on page 85.
- Checked in (common link broken: rel2): see “Checking in a binary file (from rel2)” on page 87.
- Checked out (common link broken: rel1): see “Checking out a binary file whose common link was broken (rel1)” on page 92.
- Checked in (common link broken: rel1): see “Checking in a binary file (from rel1)” on page 94.

These actions are taken in a CMVC family that already has existing objects.

The following diagram may help to understand the versions of the file that is common between 2 releases and then the link is broken and then the non-common files are further processed:

```
1.3  common version between rel1 and rel2
 |
 +---> 1.4 --> 1.5 versions exclusive for rel2 (common link is broken)
 |
 +---> 1.3.1.1     version exclusive for rel1 (common link is broken)
```

LINKING A BINARY FILE

Using the CMVC file command to unlock a binary file

Issue the following CMVC File command to link the binary.exe file from rel1 to rel2:

```
File -link binary.exe -release rel1 -to rel2 -verbose -defect 1
```

The informational message after a successful link looks like:

```
File binary.exe was linked successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view binary.exe -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
common files:
    releaseName     SID         pathName
    --------------- ----------- -----------------
    rel2            1.2         binary.exe
```

Notice that if the File -view command specified the release rel2:

```
File  -long -view binary.exe -release rel2 | more
```

Then the information about the file will show the section about
common files reflecting the release rel1:

```
common files:
    releaseName     SID         pathName
    --------------- ----------- -----------------
    rel1            1.2         binary.exe
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: new record for file binary.exe for release rel2 (releaseId=148).
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

The value in this case is:

```
NAME      LASTSERIAL
source            13
```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          164       163        147  binary.exe
        13          164       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- No existing files were modified in the vc tree. That is, the b.13 directory was not modified.
- No new files were created in the vc tree.

CHECKING OUT A BINARY FILE (COMMON)

Using the CMVC file command to check out a binary file

Issue the following CMVC File command to checkout the file which is common to rel1 and rel2:

```
File -checkout binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.2
new delta 1.3
0 lines
(Usual message about being a binary file)
A common file is associated with release rel2.
File binary.exe was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view binary.exe -release rel1 | more
    or
  File  -long -view binary.exe -release rel2 | more
  ```

  The only change when viewing the information inside CMVC for this common file that is checked out, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3       2000/04/24 13:42:10 rel1            binary.exe
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  binary.exe  rel1            2000/04/24 1.3          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          164       163        147  binary.exe
        13          164       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary is created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13. This file indicates that there is a lock for file s.binary.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary

- The contents of the p.binary file is:

  ```
  1.2 1.3 cmpc3db2 00/04/24 13:42:10
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.2 indicates the current version.

  – 1.3 indicates that is going to be the next version.

- Notice that the s.binary file is not touched at all.

CHECKING IN A BINARY FILE (COMMON)

Using the CMVC file command to check in a binary file

Use vi to add a 3rd line to the checked out file:

```
This is a binary file.
Adding line 2.
Adding line 3.
```

Issue the following CMVC File command to checkin the common binary.exe file:

```
File -checkin  binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Second checkin, adding line 3"   \
     -common rel2
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.3
0 inserted
0 deleted
0 unchanged
(Usual message about being a binary file)
File binary.exe was checked in successfully.
```

Displaying information about the CMVC file

Let's issue again the CMVC command to display the information about the file:

```
File  -long -view binary.exe -release rel1 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel1
lastUpdate      2000/04/24 13:46:30
pathName        binary.exe
nuVersionSID    1.3
nuAddDate       2000/04/24 09:16:25

locks: none.

common files:
    releaseName     SID         pathName
    --------------- ----------- ------------
    rel2            1.3         binary.exe

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        2               create   creating binary
    1.2         cmpc3db2        2               delta    creating binary
    1.3         cmpc3db2        2               delta    creating binary

versions:
     user     date       SID
     -------- ---------- ---------------------------------------------

     cmpc3db2 2000/04/24 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: for both releases, the new nuVersionId is 166 (for SID 1.3).
- Versions: a new record with id=166 for SID 1.3.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          166       163        147  binary.exe
        13          166       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.3, the same number mentioned in the p.binary file.

- The file $HOME/vc/0/0/0/0/b.13/1.2 was deleted.

- The b.13 directory contains the following files:

  ```
  -rw-r--r--  1  cmpc3db2 db2iadm2  1.2d
  -rw-r--r--  1  cmpc3db2 db2iadm2  1.3
  -rw-r--r--  1  cmpc3db2 db2iadm2  1.3d
  -r--r--r--  1  cmpc3db2 db2iadm2  s.binary
  ```

  It is important to notice that the files are fully owned by the CMVC family.

- The file 1.3 contains now the most up-to-date version of the file:

  ```
  This is a binary file.
  Adding line 2.
  Adding line 3.
  ```

- The file 1.3d contains the backward delta to generate the 1.2 version from 1.3.

  The file 1.2d contains the backward delta to generate the 1.1 version from 1.2.

  The contents of the delta file is in binary format and it is not legible.

- The s.binary file was modified as follows:

  ```
  ^Ah14887
  ^As 00000/00000/00000
  ^Ad D 1.3 00/04/24 13:46:32 cmpc3db2 3 2
  ^Ae
  ^As 00000/00000/00000
  ^Ad D 1.2 00/04/24 09:29:16 cmpc3db2 2 1
  ^Ae
  ^As 00000/00000/00000
  ^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
  ^Ac date and time created 00/04/24 09:16:25 by family
  ^Ae
  ^Au
  ^AU
  ^Af j
  ^At
  ^AT
  ^AI 1
  ^AE 1
  ```

  Notice that the header has now an entry for the Delta 1.3.

CHECKING OUT A BINARY FILE (COMMON LINK TO BE BROKEN)

Using the CMVC file command to check out a binary file

Issue the following CMVC File command to checkout the binary.exe file which is common to rel1 and rel2, but which link will be broken during the checkin:

```
File -checkout binary.exe -release rel2 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.3
new delta 1.4
0 lines
(Usual message about being a binary file)
A common file is associated with release rel1.
File binary.exe was checked out successfully.
```

Displaying information about the CMVC file

- Let's issue again the CMVC command to display the information about the file:

  ```
  File  -long -view binary.exe -release rel1 | more
    or
  File  -long -view binary.exe -release rel2 | more
  ```

  The only change when viewing the information inside CMVC for this common file that is checked out, is the following:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.4       2000/04/24 13:52:51 rel2            binary.exe
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  binary.exe  rel2            2000/04/24 1.4          cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

```
db2 "select sourceId,nuVersionId,id,baseName from Files \
    where baseName='binary.exe'"
```

  In this case, the result is:

```
SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
      13          166       163        147  binary.exe
      13          166       165        148  binary.exe
```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

```
db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
```

  In this case, the result is:

```
SOURCEID     ID    PREVIOUSID    SID
      13    162             0    1.1
      13    164           162    1.2
      13    166           164    1.3
```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13. This file indicates that there is a lock for file s.binary.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary

- The contents of the p.binary file is:

```
1.3 1.4 cmpc3db2 00/04/24 13:52:51
```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.3 indicates the current version.

  – 1.4 indicates that is going to be the next version.

- Notice that the s.binary file is not touched at all.

CHECKING IN A BINARY FILE (BREAKING THE COMMON LINK)

Using the CMVC file command to check in a binary file

Use vi to add a 4th line to the checked out file:

```
This is a binary file.
Adding line 2.
Adding line 3.
Adding line 4.
```

Issue the following CMVC File command to checkin the binary.exe file and do not specify neither the -common nor the -force flag (this is for illustration purposes):

```
File -checkin binary.exe -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, adding line 4"
```

The error message looks like:

```
0010-050 The file, binary.exe, that you are checking in
         is common with a file associated with release rel1.
         You must specify the force option if you want to break the
         common link or you must specify the common releases to maintain
         file commonality.
0010-261 File binary.exe associated with release
         rel2 cannot be checked in.
```

Issue the following CMVC File command to checkin the binary.exe file and use the -force flag to break the common link with release rel1.

```
File -checkin binary.exe -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, adding line 4"    \
     -force
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.4
0 inserted
0 deleted
0 unchanged
(Usual message about being a binary file)
File binary.exe was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view binary.exe -release rel1 | more
```

Notice that the basic information for this file remains the same at version 1.3 (that is, version 1.3 is still common to rel1 and rel2, but version 1.4 is only applicable to rel2, not to rel1):

```
releaseName     rel1
lastUpdate      2000/04/24 13:46:30
pathName        binary.exe
nuVersionSID    1.3
nuAddDate       2000/04/24 09:16:25

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        2               create   creating binary
    1.2         cmpc3db2        2               delta    creating binary
    1.3         cmpc3db2        2               delta    creating binary

versions:
     user     date       SID
     -------- ---------- ---------------------------------------------

     cmpc3db2 2000/04/24 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
common files: none.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view binary.exe -release rel2 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel2
lastUpdate      2000/04/24 13:56:25
pathName        binary.exe
nuVersionSID    1.4
nuAddDate       2000/04/24 13:37:43

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.2         cmpc3db2        2               link     creating binary
    1.3         cmpc3db2        2               delta    creating binary
    1.4         cmpc3db2        2               delta    creating binary

versions:
     user     date       SID
     -------- ---------- ------------------------------------------

     cmpc3db2 2000/04/24 1.4
Third checkin, adding line 4

     cmpc3db2 2000/04/24 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: the file in release rel2 now has the nuVersionId=167 for SID 1.4.
- Versions: new record id=167 for SID 1.4 which is based on previousId=166 (SID 1.3).

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          166       163        147  binary.exe
        13          167       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
        13    167           166    1.4
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13.

  As mentioned in the informational message when the file was checked in, the new version

of the file is 1.4, the same number mentioned in the p.binary file. This new version is only for rel2; rel1 still sees version 1.3 as the latest version.

- The file $HOME/vc/0/0/0/0/b.13/1.3 was deleted.
- The b.13 directory contains the following files:

```
-rw-r--r--  1  cmpc3db2 db2iadm2  1.2d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.3d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.4
-rw-r--r--  1  cmpc3db2 db2iadm2  1.4d
-r--r--r--  1  cmpc3db2 db2iadm2  s.binary
```

  It is important to notice that the files are fully owned by the CMVC family.

- The file 1.4 contains now the most up-to-date version of the file:

```
This is a binary file.
Adding line 2.
Adding line 3.
Adding line 4.
```

-
- The file 1.4d contains the backward delta to generate the 1.3 version from 1.4.

  The file 1.3d contains the backward delta to generate the 1.2 version from 1.3.

  The file 1.2d contains the backward delta to generate the 1.1 version from 1.2.

  The contents of the delta file is in binary format and it is not legible.

- The s.binary file was modified as follows:

```
^Ah18166
^As 00000/00000/00000
^Ad D 1.4 00/04/24 13:56:25 cmpc3db2 4 3
^Ae
^As 00000/00000/00000
^Ad D 1.3 00/04/24 13:46:32 cmpc3db2 3 2
^Ae
^As 00000/00000/00000
^Ad D 1.2 00/04/24 09:29:16 cmpc3db2 2 1
^Ae
^As 00000/00000/00000
^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
^Ac date and time created 00/04/24 09:16:25 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
^AE 1
```

Notice that the header has now an entry for the Delta 1.4.

CHECKING OUT A BINARY FILE WHOSE COMMON LINK WAS BROKEN (REL2)

Using the CMVC file command to check out a binary file

Issue the following CMVC File command to checkout the binary.exe file which was common between rel1 and rel2, but which link was broken during the latest checkin:

```
File -checkout binary.exe -release rel2 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.4
new delta 1.5
0 lines
(Usual message about being a binary file)
File binary.exe was checked out successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue the CMVC command to display the information about the file from the release rel1.

```
File  -long -view binary.exe -release rel1 | more
```

There are no new changes, because the file is no longer common. The current version is 1.3.

Displaying information about the CMVC file (from release rel2)

- Let's issue the CMVC command to display the information about the file from the release rel2.

  ```
  File  -long -view binary.exe -release rel2 | more
  ```

  Notice that only the lock section has changed:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.5       2000/04/24 14:03:42 rel2            binary.exe
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
binary.exe  rel2            2000/04/24 1.5          cmpc3db2
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          166       163        147  binary.exe
        13          167       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
        13    167           166    1.4
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary is created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13. This file indicates that there is a lock for file s.binary.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary

- The contents of the p.binary file is:

  ```
  1.4 1.5 cmpc3db2 00/04/24 14:03:42
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.4 indicates the current version.

  – 1.5 indicates that is going to be the next version.

- Notice that the s.binary file is not touched at all.

CHECKING IN A BINARY FILE (FROM REL2)

Using the CMVC file command to check in a binary file

Use vi to add a 5th line to the checked out file:

```
This is a binary file.
Adding line 2.
Adding line 3.
Adding line 4.
Adding line 5.
```

Issue the following CMVC File command to checkin the file:

```
File -checkin binary.exe -release rel2 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Fourth checkin, adding line 5"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.5
0 inserted
0 deleted
0 unchanged
(Usual message about being a binary file)
File binary.exe was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view binary.exe -release rel1 | more
```

Notice that the basic information for this file remains the same at version 1.3 (that is, version 1.3 is still common to rel1 and rel2, but version 1.5 is only applicable to rel2, not to rel1).

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view binary.exe-release rel2 | more
```

The changes when viewing the information inside CMVC for this file that was checked in are the following:

```
releaseName     rel2
lastUpdate      2000/04/24 14:07:01
pathName        binary.exe
nuVersionSID    1.5
nuAddDate       2000/04/24 13:37:43

locks: none.

common files: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.2         cmpc3db2        2               link     creating binary
    1.3         cmpc3db2        2               delta    creating binary
    1.4         cmpc3db2        2               delta    creating binary
    1.5         cmpc3db2        2               delta    creating binary

versions:
     user     date       SID
     -------- ---------- ------------------------------------------

     cmpc3db2 2000/04/24 1.5
Fourth checkin, adding line 5

     cmpc3db2 2000/04/24 1.4
Third checkin, adding line 4

     cmpc3db2 2000/04/24 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: for binary.exe in release rel2, the nuVersionId is now 168 for SID 1.5.
- Versions: a new record id=168 was created for SID 1.5, which is based on previousId=167 for SID 1.4.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          166       163        147  binary.exe
        13          168       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
        13    167           166    1.4
        13    168           167    1.5
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.5, the same number mentioned in the p.binary file. This new version is only for rel2; rel1 still sees version 1.3 as the latest version.

- The file $HOME/vc/0/0/0/0/b.13/1.4 was deleted.

- The b.13 directory contains the following files:

```
-rw-r--r--  1  cmpc3db2 db2iadm2  1.2d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.3d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.4d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.5
-rw-r--r--  1  cmpc3db2 db2iadm2  1.5d
-r--r--r--  1  cmpc3db2 db2iadm2  s.binary
```

  It is important to notice that the files are fully owned by the CMVC family.

- The file 1.5 contains now the most up-to-date version of the file:

```
This is a binary file.
Adding line 2.
Adding line 3.
Adding line 4.
Adding line 5.
```

- The file 1.5d contains the backward delta to generate the 1.4 version from 1.5.

  The file 1.4d contains the backward delta to generate the 1.3 version from 1.4.

  The file 1.3d contains the backward delta to generate the 1.2 version from 1.3.

  The file 1.2d contains the backward delta to generate the 1.1 version from 1.2.

  The contents of the delta file is in binary format and it is not legible.

- The s.binary file was modified as follows:

```
^Ah21454
^As 00000/00000/00000
^Ad D 1.5 00/04/24 14:07:01 cmpc3db2 5 4
^Ae
^As 00000/00000/00000
^Ad D 1.4 00/04/24 13:56:25 cmpc3db2 4 3
^Ae
^As 00000/00000/00000
^Ad D 1.3 00/04/24 13:46:32 cmpc3db2 3 2
^Ae
^As 00000/00000/00000
```

```
^Ad D 1.2 00/04/24 09:29:16 cmpc3db2 2 1
^Ae
^As 00000/00000/00000
^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
^Ac date and time created 00/04/24 09:16:25 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
^AE 1
```

Notice that the header has now an entry for the Delta 1.5.

CHECKING OUT A BINARY FILE WHOSE COMMON LINK WAS BROKEN (REL1)

Using the CMVC file command to check out a binary file

Issue the following CMVC File command to checkout the binary.exe file which was common between rel1 and rel2, but which link was broken:

```
File -checkout binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose
```

The informational message after a successful checkout looks like:

```
1.3
new delta 1.3.1.1
0 lines
(Usual message about being a binary file)
File binary.exe was checked out successfully.
```

Displaying information about the CMVC file (from release rel2)

Let's issue the CMVC command to display the information about the file from the release rel2.

```
File  -long -view binary.exe -release rel2 | more
```

There are no new changes, because the file is no longer common. The version for rel 2 is shown to be 1.5.

Displaying information about the CMVC file (from release rel1)

- Let's issue the CMVC command to display the information about the file from the release rel1.

  ```
  File  -long -view binary.exe -release rel1 | more
  ```

  Notice that only the lock section has changed:

  ```
  locks:
      userLogin    newSID    checkOutDate        releaseName     fileNuPath
      ------------ --------- ------------------- --------------- ----------
      cmpc3db2     1.3.1.1   2000/04/24 14:17:29 rel1            binary.exe
  ```

- To verify that indeed the file is locked, issue the following CMVC command:

  ```
  Report -view FilesOutView | more
  ```

  The output should look like this (showing only the relevant fields):

  ```
  path        releaseName     checkOutD  newSID       userLogin
  ----------- --------------- ---------- ------------ ----------
  binary.exe  rel1            2000/04/24 1.3.1.1      cmpc3db2
  ```

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: no change.
- Versions: no change.

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          166       163        147  binary.exe
        13          168       165        148  binary.exe
  ```

The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
        13    167           166    1.4
        13    168           167    1.5
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary is created in the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13. This file indicates that there is a lock for file s.binary.

  The file characteristics are: -rw-r--r-- 1 cmpc3db2 db2iadm2 /home/cmpc3db2/vc/0/0/0/0/b.13/p.binary

- The contents of the p.binary file is:

  ```
  1.3 1.3.1.1 cmpc3db2 00/04/24 14:17:29
  ```

  The main items from this file are the first 2 items, which are mentioned in the informational messages when you do a checkout of the file:

  – 1.3 indicates the current version.

  – 1.3.1.1 indicates that is going to be the next version.

- Notice that the s.binary file is not touched at all.

CHECKING IN A BINARY FILE (FROM REL1)

Using the CMVC file command to check in a binary file

Use vi to modify the 3rd line to the checked out file:

```
This is a binary file.
Adding line 2.
Modifying line 3.
```

Issue the following CMVC File command to checkin the file:

```
File -checkin binary.exe -release rel1 -relative /home/cmpc3db2/work  \
     -verbose -defect 1 -remarks "Third checkin, modifying line 3"
```

The informational message after a successful checkin looks like:

```
There are no SCCS identification keywords in the file. (cm7)
1.3.1.1
0 inserted
0 deleted
0 unchanged
(Usual message about being a binary file)
File binary.exe was checked in successfully.
```

Displaying information about the CMVC file (from release rel1)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel1:

```
File  -long -view binary.exe -release rel1 | more
```

Notice that the new version is 1.3.1.1, which shows a branching with respect to release rel2 (version 1.5):

```
releaseName     rel1
lastUpdate      2000/04/24 14:23:30
pathName        binary.exe
nuVersionSID    1.3.1.1
nuAddDate       2000/04/24 09:16:25

locks: none.

changes:
    versionSID  userLogin       name            type     abstract
    ----------- --------------- --------------- -------- --------------
    1.1         cmpc3db2        2               create   creating binary
    1.2         cmpc3db2        2               delta    creating binary
    1.3         cmpc3db2        2               delta    creating binary
    1.3.1.1     cmpc3db2        2               delta    creating binary

versions:
     user     date       SID
     -------- ---------- ---------------------------------------------

     cmpc3db2 2000/04/24 1.3.1.1
Third checkin, modifying line 3

     cmpc3db2 2000/04/24 1.3
Second checkin, adding line 3

     cmpc3db2 2000/04/24 1.2
First checkin, adding line 2

     cmpc3db2 2000/04/24 1.1
Creating binary file.
```

To verify that the file is no longer locked, issue the following CMVC command:

```
Report -view FilesOutView | more
```

The output should look like this (showing only the relevant fields):

```
path        releaseName     checkOutD  newSID       userLogin
----------- --------------- ---------- ------------ ----------
```

Displaying information about the CMVC file (from release rel2)

Let's issue again the CMVC command to display the information about the file, from the point of view of the release rel2:

```
File  -long -view binary.exe -release rel2 | more
```

Notice that there are no new changes and that the version is still 1.5.

Verifying the attributes

The main highlights are:

- Sequence: no change.
- Files: file binary.exe for release rel2 has a nuVersionId=169 for SID 1.3.1.1.
- Versions: a new record id=169 was created for SID 1.3.1.1, which is based on previousId=166 (SID 1.3).

The details are shown below:

- We can check the Sequence table for the row "source":

  ```
  db2 "select * from Sequence where name='source'"
  ```

  The value in this case is:

  ```
  NAME      LASTSERIAL
  source            13
  ```

- The sourceId for this file in the Files table is found as follows:

  ```
  db2 "select sourceId,nuVersionId,id,baseName from Files \
      where baseName='binary.exe'"
  ```

  In this case, the result is:

  ```
  SOURCEID  NUVERSIONID        ID  RELEASEID  BASENAME
        13          169       163        147  binary.exe
        13          168       165        148  binary.exe
  ```

  The releaseId of 147 corresponds to rel1 and the releaseId of 148 corresponds to rel2.

- The main data for the Versions records associated with this sourceId is found as follows:

  ```
  db2 "select sourceId,id,previousId,SID from Versions where sourceId=13"
  ```

  In this case, the result is:

  ```
  SOURCEID     ID    PREVIOUSID    SID
        13    162             0    1.1
        13    164           162    1.2
        13    166           164    1.3
        13    167           166    1.4
        13    168           167    1.5
        13    169           166    1.3.1.1
  ```

Main actions in SCCS

The main actions in SCCS for our analysis are the following:

- The file p.binary was removed from the same location as the s.binary file, in $HOME/vc/0/0/0/0/b.13.

  As mentioned in the informational message when the file was checked in, the new version of the file is 1.3.1.1 in rel 1, the same number mentioned in the p.binary file. This new version is only for rel1; rel2 still sees 1.5 as the latest version.

- No files were deleted.

- The b.13 directory contains the following files:

```
-rw-r--r--  1  cmpc3db2 db2iadm2  1.2d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.3.1.1
-rw-r--r--  1  cmpc3db2 db2iadm2  1.3d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.4d
-rw-r--r--  1  cmpc3db2 db2iadm2  1.5
-rw-r--r--  1  cmpc3db2 db2iadm2  1.5d
-r--r--r--  1  cmpc3db2 db2iadm2  s.binary
```

  It is important to notice that the files are fully owned by the CMVC family.

- The file 1.3.1.1 was added and contains the latest version for rel1:

```
This is a binary file.
Adding line 2.
Modifying line 3.
```

  In contrast, file 1.5 contains the latest version for rel2.

```
This is a binary file.
Adding line 2.
Adding line 3.
Adding line 4.
Adding line 5.
```

- The file 1.5d contains the backward delta to generate the 1.4 version from 1.5.

  The file 1.4d contains the backward delta to generate the 1.3 version from 1.4.

  The file 1.3d contains the backward delta to generate the 1.2 version from 1.3.

  The file 1.2d contains the backward delta to generate the 1.1 version from 1.2.

  The contents of the delta file is in binary format and it is not legible.

  Note: I do not really know how the version 1.3 is generated from 1.3.1.1, because I could not find a delta from 1.3.1.1 to 1.3. One possibility is that 1.3 is generated from the other branch, by applying 1.4d.

- The s.binary file was modified as follows:

```
^Ah24931
^As 00000/00000/00000
^Ad D 1.3.1.1 00/04/24 14:23:30 cmpc3db2 6 3
^Ae
^As 00000/00000/00000
^Ad D 1.5 00/04/24 14:07:01 cmpc3db2 5 4
^Ae
^As 00000/00000/00000
^Ad D 1.4 00/04/24 13:56:25 cmpc3db2 4 3
^Ae
^As 00000/00000/00000
^Ad D 1.3 00/04/24 13:46:32 cmpc3db2 3 2
^Ae
^As 00000/00000/00000
^Ad D 1.2 00/04/24 09:29:16 cmpc3db2 2 1
^Ae
^As 00000/00000/00000
^Ad D 1.1 00/04/24 09:16:25 cmpc3db2 1 0
^Ac date and time created 00/04/24 09:16:25 by family
^Ae
^Au
^AU
^Af j
^At
^AT
^AI 1
^AE 1
```

Notice that the header has now an entry for the Delta 1.3.1.1.

COPYRIGHTS, TRADEMARKS AND SERVICE MARKS

The following terms used in this technical report, are trademarks or service marks of the indicated companies:

TRADEMARK, REGISTERED TRADEMARK OR SERVICE MARK	COMPANY
IBM, AIX, CMVC, DB2 Universal Database, UDB	IBM Corporation

END OF DOCUMENT