

	PRODUCT TECHNICAL DOCUMENT	
--	----------------------------	--

MODEL NAME	DOCUMENT No.

TITLE					
CallMonitor3 Programming Specifications					
	<table border="1"> <tr> <td style="text-align: center;">FINAL PAGE</td> <td></td> </tr> <tr> <td style="text-align: center;">TOTAL SHEETS</td> <td></td> </tr> </table>	FINAL PAGE		TOTAL SHEETS	
FINAL PAGE					
TOTAL SHEETS					
© Panasonic System Networks Co., Ltd. 2000					

COPY TO	
---------	--

1. Outline .....	3
2. Screen Views .....	3
2.1. Main Screen .....	3
2.2. Tree View .....	4
2.3. Line Information / Call Information .....	5
3. Class Organization .....	6
3.1. Function Blocks .....	6
3.2. List of Classes .....	7
3.3. List of Functions .....	8
3.3.1. CCallMonitorDoc(CDocument) .....	8
3.3.2. CMainFrame(CFrameWnd) .....	8
3.3.3. CCoView(CTreeView) .....	9
3.3.4. CExtView(CTreeView) .....	10
3.3.5. CCallData .....	11
3.3.6. CD_Base .....	14
3.3.7. CD_LineAddressCaps (CD_Base) .....	17
3.3.8. CD_LineAddressStatus (CD_Base) .....	17
3.3.9. CD_LineAppInfo (CD_Base) .....	18
3.3.10. CD_LineCallStatus (CD_Base) .....	18
3.3.11. CD_LineCallTreatmentEntry (CD_Base) .....	19
3.3.12. CD_LineDevCaps (CD_Base) .....	20
3.3.13. CD_LineDevStatus (CD_Base) .....	21
3.3.14. CD_LineDialParams (CD_Base) .....	21
3.3.15. CD_LineForward (CD_Base) .....	21
3.3.16. CD_LineTermCaps (CD_Base) .....	22
4. Program flowchart .....	23
4.1. Starting CallMonitor .....	23
4.2. Closing CallMonitor .....	24
4.3 While monitoring .....	25

# Monitoring extension and CO lines

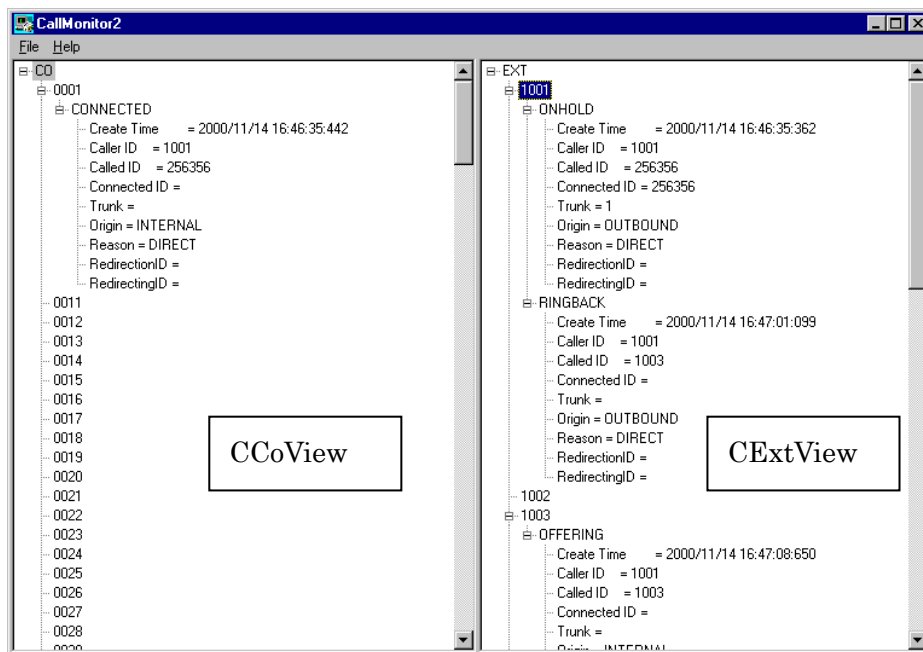
## 1. Outline

This application monitors the status of external and internal calls handled through the PBX. The call and line information can be saved in a log file.

## 2. Screen Views

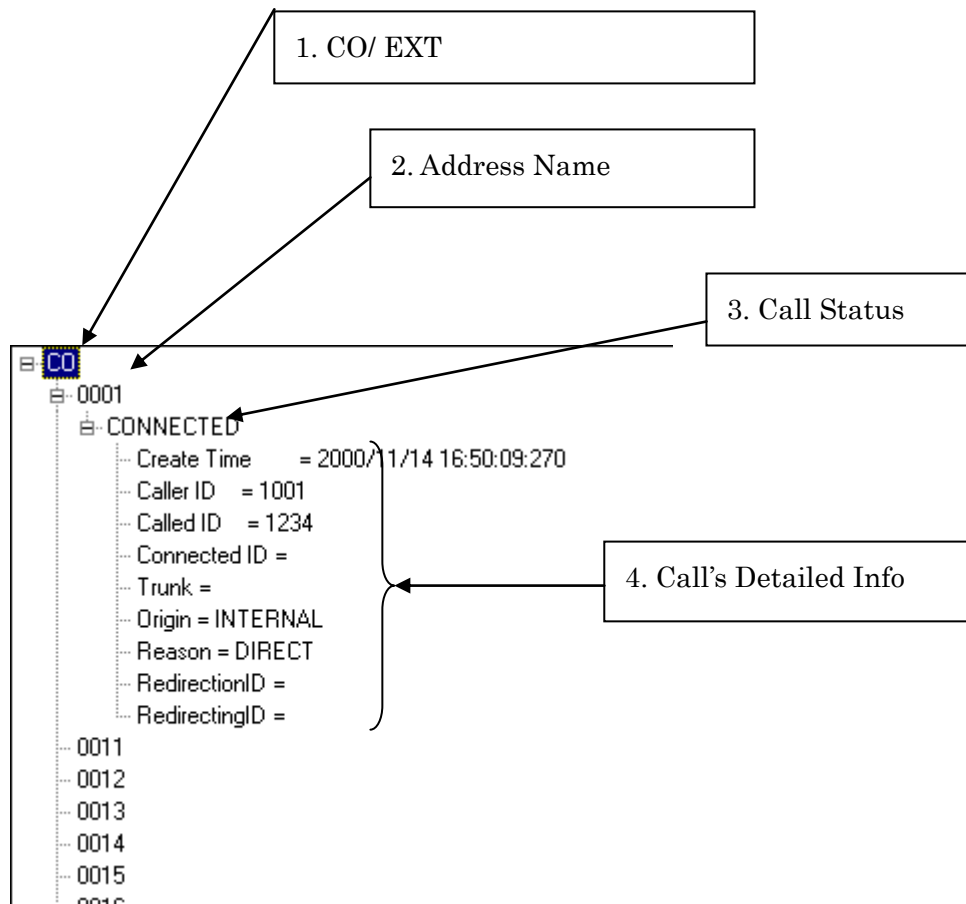
By running the exe file, TAPI is initialized and then the Main screen is displayed.

### 2.1. Main Screen



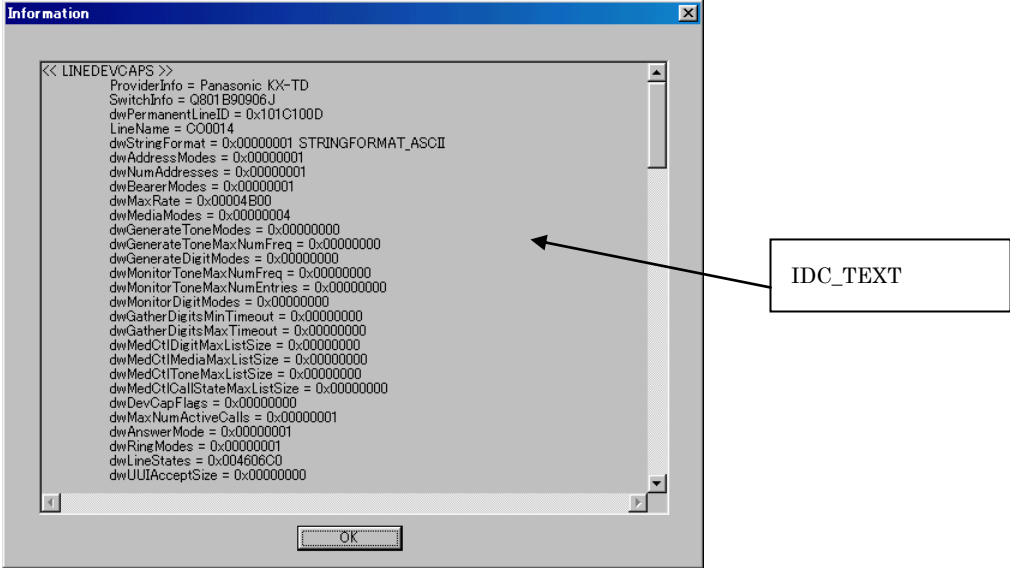
Initial screen

## 2.2. Tree View



- |                       |   |
|-----------------------|---|
| 1. CO/EXT             | CO = CO line / EXT = Extension line   |
| 2. Address name       | Address name of the CO/EXT line   |
| 3. Call Status        | IDLE, OFFERING, CONNECTED, etc.   |
| 4. Call Detailed Info | Created time, Caller ID, Called ID, Connected ID, Trunk, Origin, Reason, RedirectionID, RedirectingID |

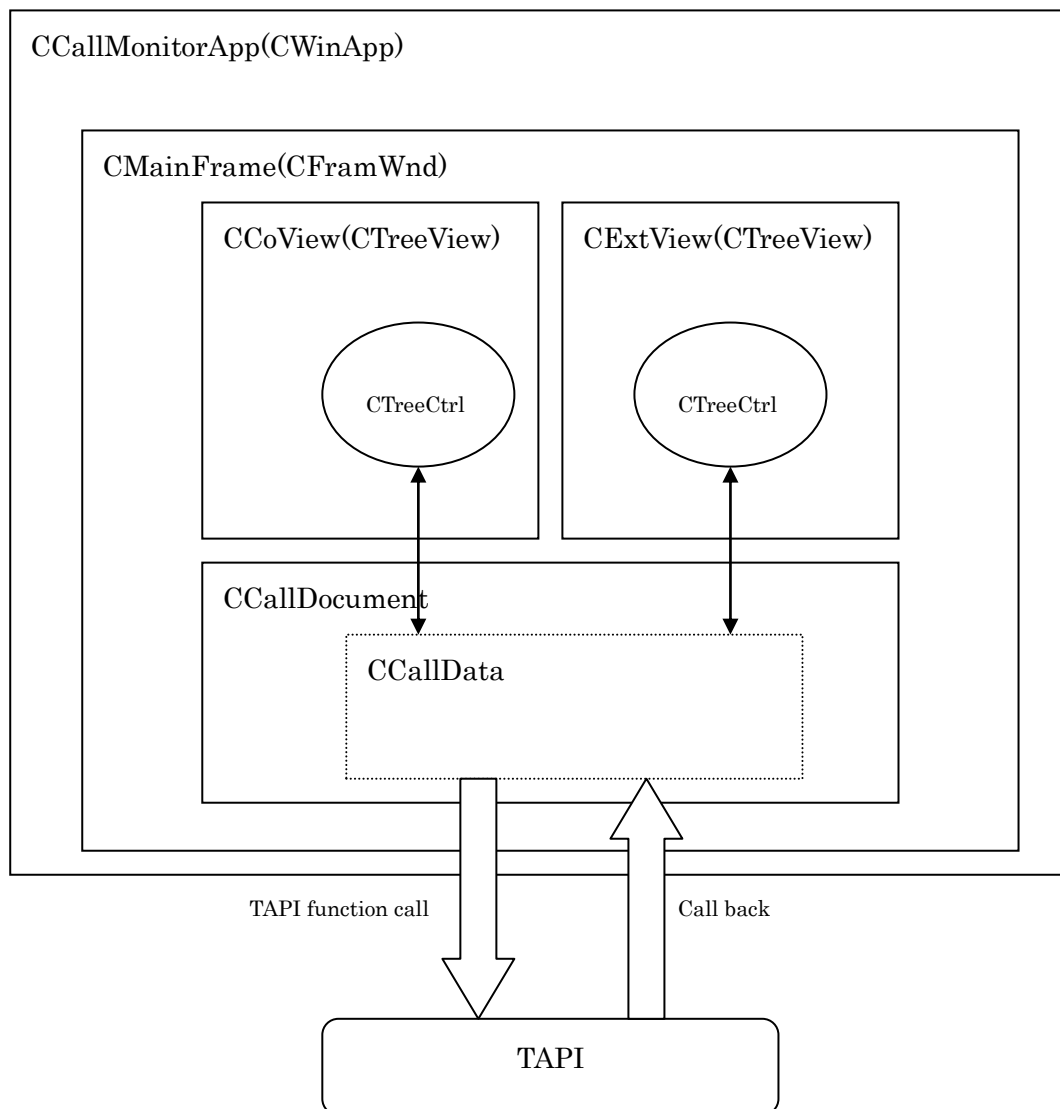
2.3. Line Information / Call Information



ID IDD\_INFORM  
Class CInformDlg

### 3. Class Organization

#### 3.1. Function Blocks



CallMonitor3 offers CCoView, CExtView and CEtcView on the Main Screen. The views are referring to CCallDocument. CCallData on the CCallDocument communicates with TAPI, gets the line status & call status information and reflects such information given by TAPI to the Tree controls.

### 3.2. List of Classes

	Class Name	Descriptions
1	CCallMonitorApp	Application class
2	CAboutDlg	Version information dialog class
3	CDialNo	Dial dialog class
4	CInformDlg	Line & Call information dialog class
5	CMainFrame	Main frame
6	CCoView	CO line view
7	CExtView	Extension line view
8	CCallMonitorDoc	Document
9	CCallData	Call handling
10	CD_Base	Structure data base class
11	CD_LineAddressCaps	LINEADDRESSCAPS structure class
12	CD_LineAddressStatus	LINEADDRESSSTATUS structure class
13	CD_LineCallInfo	LINECALLINFO structure class
14	CD_LineCallStatus	LINECALLSTATUS structure class
15	CD_LineDevCaps	LINEDEVCAPS structure class
16	CD_LineDevStatus	LINEDEVSTATUS structure class
17	CD_LineAppInfo	LINEAPPINFO structure class
18	CD_LineDialParam	LINEDIALPARAM structure class
19	CD_LineForward	LINEFORWARD structure class
20	CD_LineTermCaps	LINETERMCAPS structure class
21	CD_LineCallTreatmentEntry	LINECALLTREATMENTENTRY structure class

### 3.3. List of Functions

#### 3.3.1. CCallMonitorDoc(CDocument)

	Attributes	Function Name	MFC	Descriptions
1	pub	virtual BOOL OnNewDocument()	Y	A part of "Create new" command in the File menu
2	pub	virtual void Serialize(CArchive& ar)	Y	Read and write an object from and to the archive
3	pub	virtual void DeleteContents()	Y	Deleting document data (not deleting CDocument object)
4	pub	virtual void OnCloseDocument()	Y	Part of "Exit" command in the File menu when a document is closed

#### 3.3.2. CMainFrame(CFrameWnd)

	Attributes	Function Name	MFC	Descriptions
1	pro	BOOL OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext)	Y	Creates the client window
2	pub	BOOL PreCreateWindow(CREATESTRUCT& cs)	Y	Called just before the window is created
3	pro	void OnContextMenu(CWnd* pWnd, CPoint point)	Y	Called at right-click action on the window



### 3.3.3. CCoView(CTreeView)

	Attributes	Function Name	MFC	Descriptions
1	pub	void ContextMenu(CWnd* pWnd, CPoint point)		Right-click menu processing
2	pub	void Clear()		Initializing variables of Views
3	pub	virtual void OnDraw(CDC* pDC)	Y	Drawing a document image
4	pub	Virtual BOOL PreCreateWindow(CREATESTRUCT& cs)	Y	Called just before the window is created
5	pro	virtual void OnInitialUpdate()	Y	Called at initializing tree views
6	pro	void OnDestroy()	Y	Discard tree views
7	pro	void OnRButtonDown(UINT nFlags, CPoint point)	Y	Called at right-click action
8	pro	void OnLineInformation()		Called at getting line information
9	pro	void OnCallInformation()		Called at getting call information

### 3.3.4. CExtView(CTreeView)

	Attributes	Function Name	MFC	Descriptions
1	pub	void ContextMenu(CWnd* pWnd, CPoint point)		Right-click menu processing
2	pub	void Clear()		Initializing variables of Views
3	pub	virtual void OnDraw(CDC* pDC)	Y	Drawing a document image
4	pub	Virtual BOOL PreCreateWindow(CREATESTRUCT& cs)	Y	Called just before the window is created
5	pro	virtual void OnInitialUpdate()	Y	Called before the View is initially displayed after the Document is bound with the view
6	pro	void OnDestroy()	Y	Called to notify CWnd that the view is disccated
7	pro	void OnRButtonDown(UINT nFlags, CPoint point)	Y	Called at right-click action
8	pro	void OnLineInformation()		Called at getting line information
9	pro	void OnCallInformation()		Called at getting call information
10	pro	void OnMakecall()		Called when making a call (MakeCall)
11	pro	void OnAnswer()		Called when answering a call (Answer)
12	pro	void OnDrop()		Called when dropping a call (Drop)

### 3.3.5. CCallData

	Attributes	Function Name	MFC	Descriptions
1	pub	CCallData()		Constructor
2	pub	virtual ~CCallData()		Destructor
3	pro	static void CALLBACK Callback(DWORD hDevice,DWORD dwMsg,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		TAPI message callback
4	pro	void OnLineReply(DWORD hDevice,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		Processing LINE_REPLY message
5	pro	void OnLineAppnewcall(DWORD hDevice,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		Processing LINE_APPNEWCALL message
6	pro	void OnLineCallstate(DWORD hDevice,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		Processing LINE_CALLSTATE message
7	pro	void OnLineCallinfo(DWORD hDevice,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		Processing LINE_CALLINFO message
8	pro	void OnLineLinedevstate(DWORD hDevice,DWORD dwCBIInst,DWORD dwParam1, DWORD dwParam2,DWORD dwParam3)		Processing LINE_LINEDEVSTATE message
9	pro	CD_LineDevCaps* GetDevCaps(DWORD dwDeviceID)		Getting LINEDEVSTATE structure info
10	pro	CD_LineDevStatus* GetLineDevStatus(HLINE hLine)		Getting LINEDEVSTATUS structure info

11	pro	CD_LineAddressCaps* GetAddressCaps(DWORD dwDeviceID,DWORD dwAddressID)		Getting LINEADDRESSCAPS structure info
12	pro	CD_LineAddressStatus* GetAddressStatus(HLINE hLine,DWORD dwAddressID)		Getting LINEADDRESSSTATUS structure info
13	pro	CD_LineCallInfo* GetCallInfo(HCALL hCall)		Getting LINECALLINFO structure info
14	pro	CD_LineCallStatus* GetCallStatus(HCALL hCall)		Getting LINECALLSTATUS structure info
15	pro	void SetInfoData(HCALL hCall,CTreeCtrl* pTree,HTREEITEM hTree)		Setting call info to Tree controls
16	pro	void ViewLineInformation(CD_LineDevCaps * pDevCaps)		Displaying line Information dialog
17	pro	void ViewCallInformation(HCALL hCall)		Displaying Call Information dialog
18	pro	void WriteLogString( LPCTSTR pText)		Writing log strings
19	pro	void WriteLogStateData(CTreeCtrl* pTree,HTREEITEM hTree,LPCTSTR strMes)		Writing LINESTATE of a call to the log
20	pro	void WriteLogInfoData(HCALL hCall,CTreeCtrl* pTree,HTREEITEM hTree,LPCTSTR strMes)		Writing LINEINFO of a call to the log
21	pro	CString GetTimeString()		Getting Time string for the log
22	pro	HTREEITEM CreateCall(HCALL hCall,CD_LineDevCaps* pDevCaps,CTreeCtrl* pTree)		Creating a call in a tree
23	pub	BOOL Init()		TAPI initialize processing
24	pub	void Close()		TAPI close processing
25	pub	void LineInformationCO(HTREEITEM hTree)		Displaying CO line information
26	pub	void LineInformationExt(HTREEITEM hTree)		Displaying extension line information

27	pub	void CallInformationCO(HTREEITEM hTree)	Displaying call information at a CO line
28	pub	void CallInformationExt(HTREEITEM hTree)	Displaying call information at an extension line
29	pub	int GetCODataSize()	Getting data size (number of rows) of a CO line tree
30	pub	CD_LineDevCaps* GetCOData(int no)	Getting CO line information
31	pub	int GetExtDataSize()	Getting data size (number of rows) of a CO line tree
32	pub	CD_LineDevCaps* GetExtData(int no)	Getting Extension line information
33	pub	void TreeExtSet(CTreeCtrl* pTree)	Setting an EXT tree to CCallData
34	pub	void TreeCOSet(CTreeCtrl* pTree)	Setting a CO tree to CCallData
35	pub	void TreeExpCOData()	Expanding a CO tree
36	pub	void TreeExpExtData()	Expanding an EXT tree
37	pub	void ExtDestroy()	Discarding ExtView
38	pub	void CODestroy()	Discarding COView
39	pub	int MakeCall(HTREEITEM hTree, LPCSTR pDialNo)	MakeCall processing
40	pub	BOOL CanAnswer(HTREEITEM hTree)	Judging if it is able to Answer a call
41	pub	BOOL CanDrop(HTREEITEM hTree)	Judging if it is able to Drop a call
42	pub	int Answer(HTREEITEM hTree)	Answering a call
43	pub	int Drop(HTREEITEM hTree)	Dropping a call
44	pub	void SetLogFileName(LPCSTR filename)	Setting the log file name

### 3.3.6. CD\_Base

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_Base()		Constructor
2	pub	virtual ~CD_Base()		Destructor
3	pro	CString GetStringData(DWORD d_offset,DWORD d_size)		Getting string data (OFFSET)
4	pro	CString GetStringData(LPTSTR p_str,DWORD d_size)		Getting string data (Pointer)
5	pro	LPTSTR GetBinaryData(DWORD d_offset,DWORD d_size)		Getting binary data
6	pro	CString GetIndentString(int nIndent)		Getting indent strings
7	pro	CString GetBinaryString(LPCTSTR pData,DWORD d_size,int nIndent=0)		Getting binary strings
8	pro	CString GetStringFormat(DWORD dwStringFormat)		Getting strings set to dwStringFormat
9	pro	CString GetAddressMode(DWORD dwAddressModes)		Getting strings set to dwAddressModes
10	pro	CString GetBearerMode(DWORD dwBearerModes)		Getting strings set to dwBearerModes
11	pro	CString GetMediaMode(DWORD dwMediaModes)		Getting strings set to dwMediaModes
12	pro	CString GetGenerateToneMode(DWORD dwGenerateToneModes)		Getting strings set to dwGenerateToneModes
13	pro	CString GetGenerateDigitMode(DWORD dwGenerateDigitModes)		Getting strings set to dwGenerateDigitModes
14	pro	CString GetDigitMode(DWORD dwDigitModes)		Getting strings set to dwDigitModes
15	pro	CString GetDevCapFlag(DWORD dwDevCapFlags)		Getting strings set to dwDevCapFlags
16	pro	CString GetAnswerMode(DWORD dwAnswerMode)		Getting strings set to dwAnswerMode
17	pro	CString GetDevStatusFlag(DWORD dwDevStatusFlags)		Getting strings set to dwDevStatusFlags
18	pro	CString GetLineFeature(DWORD dwLineFeatures)		Getting strings set to dwLineFeatures
19	pro	CString GetRoamMode(DWORD dwRoamMode)		Getting strings set to dwRoamMode

20	pro	CString GetTerminalMode(DWORD dwTerminalMode)		Getting strings set to dwTerminalMode
21	pro	CString GetCallInfoStates(DWORD dwCallInfoStates)		Getting strings set to dwCallInfoStates
22	pro	CString GetCallStates(DWORD dwCallStates)		Getting strings set to dwCallStates
23	pro	CString GetLineDevState(DWORD dwLineDevState)		Getting strings set to dwLineDevState
24	pro	CString GetCallPartyID(DWORD dwCallPartyID)		Getting strings set to dwCallPartyID
25	pro	CString GetDialToneMode(DWORD dwDialToneModes)		Getting strings set to dwDialToneModes
26	pro	CString GetBusyMode(DWORD dwBusyModes)		Getting strings set to dwBusyModes
27	pro	CString GetSpecialInfo(DWORD dwSpecialInfo)		Getting strings set to dwSpecialInfo
28	pro	CString GetDisconnectMode(DWORD dwDisconnectModes)		Getting strings set to dwDisconnectModes
29	pro	CString GetCallFeature(DWORD dwCallFeatures)		Getting strings set to dwCallFeatures
30	pro	CString GetCallFeature2(DWORD dwCallFeature2s)		Getting strings set to dwCallFeature2s
31	pro	CString GetTransferMode(DWORD dwTransferModes)		Getting strings set to dwTransferModes
32	pro	CString GetParkMode(DWORD dwParkModes)		Getting strings set to dwParkModes

33	pro	CString GetForwardMode(DWORD dwForwardModes)		Getting strings set to dwForwardModes
34	pro	CString GetCallCompletionCond(DWORD dwCallCompletionCond)		Getting strings set to dwCallCompletionCond
35	pro	CString GetCallCompletionMode(DWORD dwCallCompletionMode)		Getting strings set to dwCallCompletionMode
36	pro	CString GetAddressFeature(DWORD dwAddressFeature)		Getting strings set to dwAddressFeature
37	pro	CString GetAddressSharing(DWORD dwAddressSharing)		Getting strings set to dwAddressSharing
38	pro	CString GetAddressState(DWORD dwAddressStates)		Getting strings set to dwAddressStates
39	pro	CString GetAddrCapFlag(DWORD dwAddrCapFlags)		Getting strings set to dwAddrCapFlags
40	pro	CString GetRemoveFromConfCap(DWORD dwRemoveFromConfCaps)		Getting strings set to dwRemoveFromConfCaps
41	pro	CString GetLineConnectedMode(DWORD dwLineConnectedMode)		Getting strings set to dwLineConnectedMode
42	pro	CString GetLineOfferingMode(DWORD dwLineOfferingMode)		Getting strings set to dwLineOfferingMode
43	pro	CString GetOrigin(DWORD dwOrigin)		Getting strings set to dwOrigin
44	pro	CString GetReason(DWORD dwReason)		Getting strings set to dwReason
45	pro	CString GetCallParamFlag(DWORD dwCallParamFlag)		Getting strings set to dwCallParamFlag
46	pro	CString GetCallPrivilege(DWORD dwCallPrivilege)		Getting strings set to dwCallPrivilege
47	pub	static CString GetCallState(DWORD dwCallState)		Getting strings set to dwCallState
48	pub	static CString GetCallStateMode(DWORD dwCallState, DWORD dwCallSateMode)		Getting strings set to dwCallSateMode of dwCallState
49	pub	static CString GetCallInfoState(DWORD dwCallInfoState)		Getting strings set to dwCallInfoState



### 3.3.7. CD\_LineAddressCaps (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineAddressCaps ()		Constructor
2	pub	virtual ~ CD_LineAddressCaps ()		Destructor
3	pub	CD_LineAddressCaps(LPLINEADDRESSCAPS plineAddressCaps,LPSTR pTop=NULL)		Constructor using LINEADDRESSSCAPS
3	pub	void SetData(LPLINEADDRESSSCAPS plineAddressCaps,LPSTR pTop=NULL)		Setting LINEADDRESSSCAPS to CD_LineAddressCaps
4	pub	CString GetAddress()		Getting address name
5	pub	CString GetAddressP2()		Getting address name (4-digit)
6	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
7	pro	void ClearData()		Initializing data

### 3.3.8. CD\_LineAddressStatus (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineAddressStatus ()		Constructor
2	pub	virtual ~ CD_LineAddressStatus ()		Destructor
3	pub	CD_LineAddressStatus(LPLINEADDRESSSTATUS plineAddressStatus,LPSTR pTop=NULL)		Constructor using LINEADDRESSSTATUS
4	pub	void SetData(LPLINEADDRESSSTATUS plineAddressStatus,LPSTR pTop=NULL)		Setting LINEADDRESSSTATUS to CD_LineAddressStatus
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pub	CString GetExtName()		Getting extension name
7	pro	void ClearData()		Initializing data

### 3.3.9. CD\_LineAppInfo (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineAppInfo ()		Constructor
2	pub	virtual ~ CD_LineAppInfo ()		Destructor
3	pub	CD_LineAppInfo(LPLINEAPPINFO pLineAppInfo,LPSTR pTop=NULL)		Constructor using LINEAPPINFO
4		void SetData(LPLINEAPPINFO pLineAppInfo,LPSTR pTop=NULL)		Setting LINEAPPINFO to CD_LineAppInfo
5		CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

### 3.3.10. CD\_LineCallStatus (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineCallStatus ()		Constructor
2	pub	virtual ~ CD_LineCallStatus ()		Destructor
3	pub	CD_LineCallStatus(LPLINECALLSTAT US pLineCallStatus,LPSTR pTop=NULL)		Constructor using LINECALLSTATUS
4	pub	void SetData(LPLINECALLSTATUS pLineCallStatus,LPSTR pTop=NULL)		Setting LINECALLSTATUS to CD_LineCallStatus
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pub	BOOL CanAnswer()		Judging if it is able to Answer a call
7	pub	BOOL CanDrop()		Judging if it is able to Drop a call
8	pro	void ClearData()		Initializing data

### 3.3.11. CD\_LineCallTreatmentEntry (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineCallTreatmentEntry ()		Constructor
2	pub	virtual ~ CD_LineCallTreatmentEntry ()		Destructor
3	pub	CD_LineCallTreatmentEntry(LPLINECALLTREATMENTENTRY pLineCallTreatmentEntry, LPCTSTR pTop=NULL)		Constructor using LINECALLTREATMENTENTRY
4	pub	void SetData(LPLINECALLTREATMENTENTRY pLineCallTreatmentEntry, LPCTSTR pTop=NULL)		Setting LINECALLTREATMENTENTRY to CD_LineCallTreatmentEntry
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

### 3.3.12. CD\_LineDevCaps (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineDevCaps()		Constructor
2	pub	virtual ~CD_LineDevCaps()		Destructor
3	pub	CD_LineDevCaps(LPLINEDEVCAPS pLineDevCaps, LPSTR pTop=NULL)		Constructor using LINEDEVCAPS
4	pub	void SetData(LPLINEDEVCAPS pLineDevCaps, LPSTR pTop=NULL)		Setting LINEDEVCAPS to CD_LineDevCaps
5	pub	BOOL CheckProvider(LPCSTR lpStr)		Checking provider name
6	pub	void SetAddressCaps(CD_LineAddressCaps* pData)		Setting the pointer of CD_LineAddressCaps
7	pub	CD_LineAddressCaps* GetAddressCaps()		Setting the pointer of CD_LineAddressCaps
8	pub	BOOL IsCOLine()		Judging if it is a CO line
9	pub	BOOL IsExtLine()		Judging if it is an extension line
10	pub	CString GetLineName()		Getting line name
11	pub	void SetDeviceID(DWORD dwID)		Setting DeviceID
12	pub	DWORD GetDeviceID()		Getting DeviceID
13	pub	void SethLine(HLINE hLine)		Setting line handle
14	pub	HLINE GethLine()		Getting line handle
15	pub	void SethTree(HTREEITEM hTree)		Setting tree view items
16	pub	HTREEITEM GethTree()		Getting tree view items
17	pub	void SetExtName(LPCSTR name)		Setting extension name
18	pub	CString GetExtName()		Getting extension name
19	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
20	pro	void ClearData()		Initializing data

### 3.3.13. CD\_LineDevStatus (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineDevStatus ()		Constructor
2	pub	virtual ~ CD_LineDevStatus ()		Destructor
3	pub	CD_LineDevStatus(LPLINEDEVSTATUS S plineDevStatus,LPSTR pTop=NULL)		Constructor using LINEDEVSTATUS
4	pub	void SetData(LPLINEDEVSTATUS plineDevStatus,LPSTR pTop=NULL)		Setting LINEDEVSTATUS to CD_LineDevStatus
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

### 3.3.14. CD\_LineDialParams (CD\_Base)

	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineDialParams ()		Constructor
2	pub	virtual ~ CD_LineDialParams ()		Destructor
3	pub	CD_LineDialParams(LPLINEDIALPAR AMS plineDialParams,LPSTR pTop=NULL)		Constructor using LINEDIALPARAMS
4	pub	void SetData(LPLINEDIALPARAMS plineDialParams,LPSTR pTop=NULL)		Setting LINEDIALPARAMS to CD_LineDialParams
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

### 3.3.15. CD\_LineForward (CD\_Base)

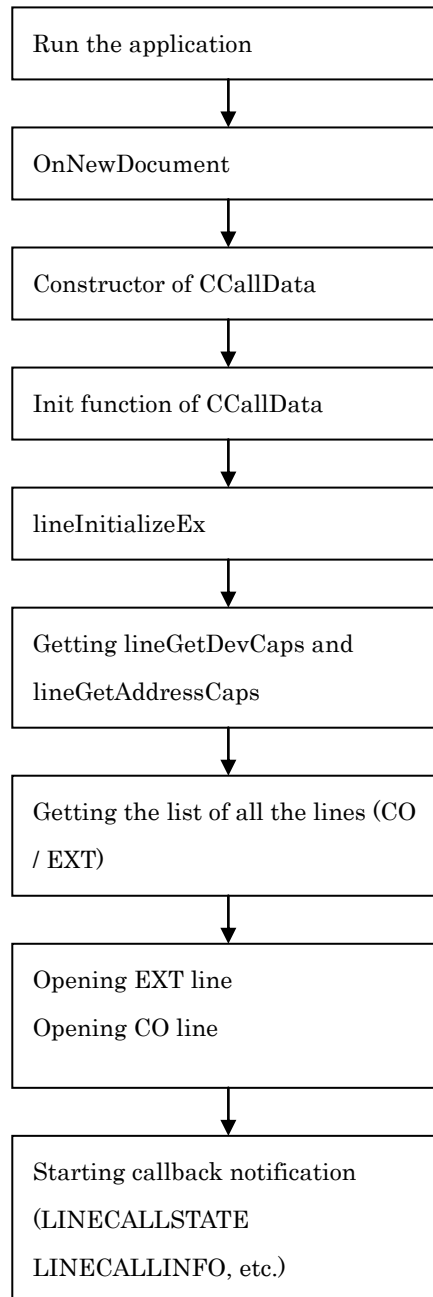
	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineDialParams ()		Constructor
2	pub	virtual ~ CD_LineDialParams ()		Destructor
3	pub	CD_LineForward(LPLINEFORWARD plineForward,LPSTR pTop=NULL)		Constructor using LINEFORWARD
4	pub	void SetData(LPLINEFORWARD plineForward,LPSTR pTop=NULL)		Setting LINEFORWARD to CD_LineForward
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

### 3.3.16. CD\_LineTermCaps (CD\_Base)

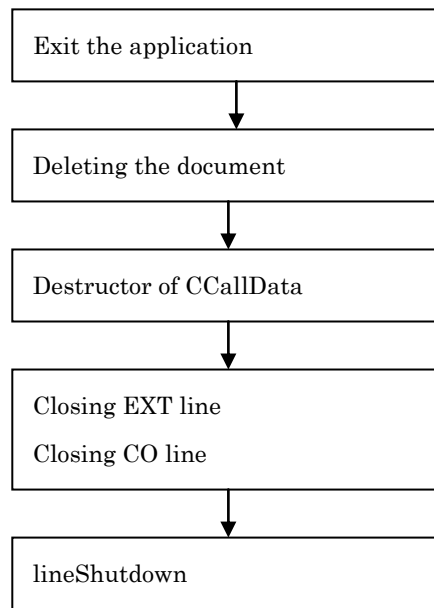
	Attributes	Function Name	MFC	Descriptions
1	pub	CD_LineTermCaps ()		Constructor
2	pub	virtual ~ CD_LineTermCaps ()		Destructor
3	pub	CD_LineTermCaps(LPLINETERMCAP S pLineTermCaps,LPSTR pTop=NULL)		Constructor using LINETERMCAPS
4	pub	void SetData(LPLINETERMCAPS pLineTermCaps,LPSTR pTop=NULL)		Setting LINETERMCAPS to CD_LineTermCaps
5	pub	CString GetInformation(int nIndent=0)		Getting strings of Information dialog
6	pro	void ClearData()		Initializing data

## 4. Program flowchart

### 4.1. Starting CallMonitor



## 4.2. Closing CallMonitor





### 4.3 While monitoring

