i-Page Server
COM Library

WiPath Communications Ltd

Table of Contents

C++

File: IPageServer.idl

Library and Constants

```
[
uuid(9CA17DCD-E648-4009-941B-B9F88AB8F427),
  version(4.0),
  helpstring("i-Page Server 4.0 Library")

]
library IPageServer
{

  importlib("stdole2.tlb");
  importlib("stdvcl40.dll");

  interface IIPageServerCom;
  dispinterface IIPageServerComEvents;
  coclass IPageServerCom;
  interface IAccount;
  coclass Account;
  interface ICarrier;
  coclass Carrier;
  interface IContact;
  coclass Contact;
  interface IGroup;
  coclass Group;
  interface ITemplate;
  coclass Template;
  interface ISchedule;
  coclass Schedule;
  interface IReport;
  coclass Report;
  interface IMsgSettings;
  coclass MsgSettings;


  [
    uuid(24F36497-01D9-4E6F-BF16-B51A0410DBE6)
  ]
  enum TSysObjectType
  {
    sotContact = 0,
    sotGroup = 1,
    sotTemplate = 2,
    sotSchedule = 3,
    sotReport = 4,
    sotFolder = 5
  };
```

Interfaces

IIPageServerCom

```
[
  uuid(05FC6A4E-1D8E-440A-87DC-BB82B6B2F6D8),
  helpstring("Dispatch interface for IPageServerCom Object"),
  dual,
  oleautomation
]
interface IIPageServerCom: IDispatch
{
  [id(0x000000C9)]
  HRESULT _stdcall ConnectToServer([in] BSTR accountId, [in] BSTR
    ipAddress);
  [id(0x000000CA)]
  HRESULT _stdcall Start([out, retval] VARIANT_BOOL* Value);
  [id(0x000000CB)]
  HRESULT _stdcall Stop([out, retval] VARIANT_BOOL* Value);
  [id(0x000000CC)]
  HRESULT _stdcall Terminate(void);
  [id(0x000000CD)]
  HRESULT _stdcall Login([in] BSTR userName, [in] BSTR password, [in] BSTR
    ipAddress, [in] long ecoding);
  [id(0x000000CE)]
  HRESULT _stdcall Logout([in] BSTR accountId, [in] BSTR ipAddress);
  [id(0x000000CF)]
  HRESULT _stdcall ClientReady([in] BSTR accountId, [in] BSTR accountIP,
    [in] VARIANT_BOOL isReady);
  [id(0x000000D0)]
  HRESULT _stdcall SendMsg([in] BSTR accountId, [in] BSTR accountIP, [in]
    IStrings* msgData);
  [id(0x000000D1)]
  HRESULT _stdcall RunReport([in] long reportId, [out] BSTR* resultMsg,
    [out, retval] VARIANT_BOOL* success);
  [id(0x000000D2)]
  HRESULT _stdcall NewMsgObject([in] enum TSysObjectType objType, [in] BSTR
    accountId, [out, retval]  IDispatch** Value);
  [id(0x000000D3)]
  HRESULT _stdcall FindMsgObjects([in] enum TSysObjectType objType, [in]
    long searchType, [in] BSTR searchStr, [out, retval] IStrings** Value);
  [id(0x000000D4)]
  HRESULT _stdcall AssignMsgObjects([in] enum TSysObjectType objType, [in]
    BSTR accountId, [in] IStrings* objIds);
  [id(0x000000D5)]
  HRESULT _stdcall DeleteMsgObject([in] long objId, [in] enum TSysObjectType
    objType, [in] BSTR accountId, [out, retval] VARIANT_BOOL* Value);
  [propget, id(0x000000D6)]
  HRESULT _stdcall MsgObjectTypeName([in] enum TSysObjectType objType, [out,
    retval] BSTR* Value);
  [propget, id(0x000000D7)]
  HRESULT _stdcall IsActive([out, retval] VARIANT_BOOL* Value);
  [propget, id(0x000000D8)]
  HRESULT _stdcall Account([in] BSTR accountId, [out, retval] IAccount**
    Value);
  [propget, id(0x000000D9)]
  HRESULT _stdcall Carriers([out, retval] IStrings** Value);
  [propget, id(0x000000DA)]
  HRESULT _stdcall Carrier([in] long carrierId, [out, retval] ICarrier**
    Value);
  [propget, id(0x000000DB)]
  HRESULT _stdcall Contact([in] long contactId, [out, retval] IContact**
    Value);
  [propget, id(0x000000DC)]
  HRESULT _stdcall Group([in] long groupId, [out, retval] IGroup** Value);
  [propget, id(0x000000DD)]
  HRESULT _stdcall Template([in] long templateId, [out, retval] ITemplate**
    Value);
  [propget, id(0x000000DE)]
  HRESULT _stdcall Schedule([in] long scheduleId, [out, retval] ISchedule**
    Value);
  [propget, id(0x000000DF)]
  HRESULT _stdcall Report([in] long reportId, [out, retval] IReport**
    Value);
  [id(0x000000E0)]
  HRESULT _stdcall LoginWeb([in] BSTR userName, [in] BSTR password, [in]
    BSTR ipAddress, [in] BSTR browser, [out] BSTR* errorMsg, [out, retval]
    VARIANT_BOOL* success);
  [id(0x000000E1)]
  HRESULT _stdcall LogoutWeb([in] BSTR account, [in] BSTR ipAddress);
  [propget, id(0x000000E2)]
  HRESULT _stdcall WebTimeout([out, retval] long* Value);
  [id(0x000000E3)]
  HRESULT _stdcall ReportOnMsgs([in] BSTR accountId, [in] long msgShowType,
    [in] long noOfMsgs, [in] DATE fromDate, [in] DATE toDate, [out, retval]
    IStrings** result);
  [id(0x000000E4)]
  HRESULT _stdcall ReturnMsgResultWeb([in] BSTR accountId, [in] BSTR
    ipAddress, [out, retval] IStrings** result);
};
```

IAccount

```
  [
    uuid(49DCE0D5-FDD6-4424-B77D-6AC68A3FD7D6),
    helpstring("Dispatch interface for Account Object"),
    dual,
    oleautomation
  ]
  interface IAccount: IDispatch
  {
    [propget, id(0x000000C9)]
    HRESULT _stdcall AccountId([out, retval] BSTR* Value);
    [propget, id(0x000000CA)]
    HRESULT _stdcall AccountRights([out, retval] long* Value);
    [propget, id(0x000000CB)]
    HRESULT _stdcall PagerId([out, retval] BSTR* Value);
    [propput, id(0x000000CB)]
    HRESULT _stdcall PagerId([in] BSTR Value);
    [propget, id(0x000000CC)]
    HRESULT _stdcall MobileNo([out, retval] BSTR* Value);
    [propput, id(0x000000CC)]
    HRESULT _stdcall MobileNo([in] BSTR Value);
    [propget, id(0x000000CD)]
    HRESULT _stdcall Email([out, retval] BSTR* Value);
    [propput, id(0x000000CD)]
    HRESULT _stdcall Email([in] BSTR Value);
    [propget, id(0x000000CE)]
    HRESULT _stdcall DisplayName([out, retval] BSTR* Value);
    [propput, id(0x000000CE)]
    HRESULT _stdcall DisplayName([in] BSTR Value);
    [id(0x000000D4)]
    HRESULT _stdcall SetAccount([in] BSTR accountId);
    [id(0x000000D5)]
    HRESULT _stdcall Save(void);
    [id(0x000000D6)]
    HRESULT _stdcall ChangePassword([in] BSTR oldPassword, [in] BSTR
      newPassword, [in] BSTR confirmPassword, [in] long encoding, [out] BSTR*
      errorMsg, [out, retval] VARIANT_BOOL* success);
    [id(0x000000D7)]
    HRESULT _stdcall AssignSystemObject([in] long objId, [in] enum
      TSysObjectType objType);
    [id(0x000000D8)]
    HRESULT _stdcall RemoveSystemObject([in] long objId, [in] enum
      TSysObjectType objType, [out] BSTR* errorMsg, [out, retval]
      VARIANT_BOOL* result);
    [propget, id(0x000000D9)]
    HRESULT _stdcall MsgObjects([in] enum TSysObjectType objType, [out,
      retval] IStrings** Value);
    [id(0x000000CF)]
    HRESULT _stdcall DeleteAllMessages(void);
    [id(0x000000D0)]
    HRESULT _stdcall DeleteOldMessages([in] long olderThanDays);
    [id(0x000000D1)]
    HRESULT _stdcall DeleteSelectedMessage([in] long msgId);
    [propget, id(0x000000D2)]
    HRESULT _stdcall MsgSettings([out, retval] IMsgSettings** Value);
    [id(0x000000D3)]
    HRESULT _stdcall GetMsgObjects([in] enum TSysObjectType objType, [in] long
      searchType, [in] BSTR searchStr, [out, retval] IStrings** result);
  };
```

ICarrier

```
  [
    uuid(25FC3D06-4448-4B22-9357-36450F6952EC),
    helpstring("Dispatch interface for Carrier Object"),
    dual,
    oleautomation
  ]
  interface ICarrier: IDispatch
  {
    [propget, id(0x000000C9)]
    HRESULT _stdcall CarrierId([out, retval] long* Value);
    [propget, id(0x000000CA)]
    HRESULT _stdcall CarrierName([out, retval] BSTR* Value);
    [propget, id(0x000000CB)]
    HRESULT _stdcall CarrierType([out, retval] long* Value);
    [propget, id(0x000000CC)]
    HRESULT _stdcall CarrierTypeStr([out, retval] BSTR* Value);
    [propget, id(0x000000CD)]
    HRESULT _stdcall AddressType([out, retval] long* Value);
    [propget, id(0x000000CE)]
    HRESULT _stdcall AddressTypeStr([out, retval] BSTR* Value);
    [propget, id(0x000000CF)]
    HRESULT _stdcall MaxMsgLength([out, retval] long* Value);
    [propget, id(0x000000D0)]
    HRESULT _stdcall CarrierInfo([out, retval] IStrings** Value);
    [id(0x000000D1)]
    HRESULT _stdcall SetCarrier([in] long carrierId);
  };
```

IContact

```
  [
    uuid(82C8C40F-AF9F-4FA4-8366-F1C31BCC287C),
    helpstring("Dispatch interface for Contact Object"),
    dual,
    oleautomation
  ]
  interface IContact: IDispatch
  {
    [propget, id(0x000000C9)]
    HRESULT _stdcall ContactId([out, retval] long* Value);
  [propget, id(0x000000CA)]
  HRESULT _stdcall ContactName([out, retval] BSTR* Value);
  [propput, id(0x000000CA)]
  HRESULT _stdcall ContactName([in] BSTR Value);
  [propget, id(0x000000CB)]
  HRESULT _stdcall AccountOwner([out, retval] BSTR* Value);
  [propput, id(0x000000CB)]
  HRESULT _stdcall AccountOwner([in] BSTR Value);
  [propget, id(0x000000CC)]
  HRESULT _stdcall IsSystemObject([out, retval] VARIANT_BOOL* Value);
  [propget, id(0x000000CD)]
  HRESULT _stdcall Notes([out, retval] BSTR* Value);
  [propput, id(0x000000CD)]
  HRESULT _stdcall Notes([in] BSTR Value);
  [propget, id(0x000000CE)]
  HRESULT _stdcall ContactAddress([out, retval] BSTR* Value);
  [propput, id(0x000000CE)]
  HRESULT _stdcall ContactAddress([in] BSTR Value);
  [propget, id(0x000000CF)]
  HRESULT _stdcall CarrierId([out, retval] long* Value);
  [propput, id(0x000000CF)]
  HRESULT _stdcall CarrierId([in] long Value);
  [propget, id(0x000000D0)]
  HRESULT _stdcall CarrierName([out, retval] BSTR* Value);
  [propget, id(0x000000D1)]
  HRESULT _stdcall CarrierType([out, retval] BSTR* Value);
  [propget, id(0x000000D2)]
  HRESULT _stdcall AddressType([out, retval] long* Value);
  [propget, id(0x000000D3)]
  HRESULT _stdcall AddressTypeStr([out, retval] BSTR* Value);
  [id(0x000000D4)]
  HRESULT _stdcall SetContact([in] long contactId);
  [id(0x000000D5)]
  HRESULT _stdcall Save(void);
};
```

IGroup

```
[
  uuid(F8ADE1C5-B2BB-4255-94A2-73C2B762CC68),
  helpstring("Dispatch interface for Group Object"),
  dual,
  oleautomation
]
interface IGroup: IDispatch
{
  [propget, id(0x000000C9)]
  HRESULT _stdcall GroupId([out, retval] long* Value);
  [propget, id(0x000000CA)]
```

```
    HRESULT _stdcall GroupName([out, retval] BSTR* Value);
    [propput, id(0x000000CA)]
    HRESULT _stdcall GroupName([in] BSTR Value);
    [propget, id(0x000000CB)]
    HRESULT _stdcall AccountOwner([out, retval] BSTR* Value);
    [propput, id(0x000000CB)]
    HRESULT _stdcall AccountOwner([in] BSTR Value);
    [propget, id(0x000000CC)]
    HRESULT _stdcall IsSystemObject([out, retval] VARIANT_BOOL* Value);
    [propget, id(0x000000CD)]
    HRESULT _stdcall Notes([out, retval] BSTR* Value);
    [propput, id(0x000000CD)]
    HRESULT _stdcall Notes([in] BSTR Value);
    [propget, id(0x000000CE)]
    HRESULT _stdcall GroupContacts([out, retval] IStrings** Value);
    [propget, id(0x000000CF)]
    HRESULT _stdcall GroupContact([in] long contactId, [out, retval]
      IContact** Value);
    [propget, id(0x000000D0)]
    HRESULT _stdcall GroupContactByName([in] BSTR contactName, [out, retval]
      IContact** Value);
    [id(0x000000D1)]
    HRESULT _stdcall SetGroupContacts([in] IStrings* contactIds);
    [id(0x000000D2)]
    HRESULT _stdcall SetGroup([in] long groupId);
    [id(0x000000D3)]
    HRESULT _stdcall Save(void);
  };
```

ITemplate

```
  [
    uuid(07BBC3EC-F61E-4A81-BC82-B90E6532A937),
    helpstring("Dispatch interface for Template Object"),
    dual,
    oleautomation
  ]
  interface ITemplate: IDispatch
  {
    [propget, id(0x000000C9)]
    HRESULT _stdcall TemplateId([out, retval] long* Value);
    [propget, id(0x000000CA)]
    HRESULT _stdcall TemplateName([out, retval] BSTR* Value);
    [propput, id(0x000000CA)]
    HRESULT _stdcall TemplateName([in] BSTR Value);
    [propget, id(0x000000CB)]
    HRESULT _stdcall AccountOwner([out, retval] BSTR* Value);
    [propput, id(0x000000CB)]
    HRESULT _stdcall AccountOwner([in] BSTR Value);
```

```
      [propget, id(0x000000CC)]
      HRESULT _stdcall IsSystemObject([out, retval] VARIANT_BOOL* Value);
      [propget, id(0x000000CD)]
      HRESULT _stdcall Notes([out, retval] BSTR* Value);
      [propput, id(0x000000CD)]
      HRESULT _stdcall Notes([in] BSTR Value);
      [propget, id(0x000000CE)]
      HRESULT _stdcall TemplateText([out, retval] BSTR* Value);
      [propput, id(0x000000CE)]
      HRESULT _stdcall TemplateText([in] BSTR Value);
      [id(0x000000CF)]
      HRESULT _stdcall SetTemplate([in] long templateId);
      [id(0x000000D0)]
      HRESULT _stdcall Save(void);
    };
```

ISchedule

```
    [
      uuid(43AE1889-83E0-4418-B97F-FA021624F2CE),
      helpstring("Dispatch interface for Schedule Object"),
      dual,
      oleautomation
    ]
    interface ISchedule: IDispatch
    {
      [propget, id(0x000000C9)]
      HRESULT _stdcall ScheduleId([out, retval] long* Value);
      [propget, id(0x000000CA)]
      HRESULT _stdcall ScheduleName([out, retval] BSTR* Value);
      [propput, id(0x000000CA)]
      HRESULT _stdcall ScheduleName([in] BSTR Value);
      [propget, id(0x000000CB)]
      HRESULT _stdcall AccountOwner([out, retval] BSTR* Value);
      [propput, id(0x000000CB)]
      HRESULT _stdcall AccountOwner([in] BSTR Value);
      [propget, id(0x000000CC)]
      HRESULT _stdcall IsSystemObject([out, retval] VARIANT_BOOL* Value);
      [propget, id(0x000000CD)]
      HRESULT _stdcall Notes([out, retval] BSTR* Value);
      [propput, id(0x000000CD)]
      HRESULT _stdcall Notes([in] BSTR Value);
      [propget, id(0x000000CE)]
      HRESULT _stdcall ScheduleEnabled([out, retval] VARIANT_BOOL* Value);
      [propput, id(0x000000CE)]
      HRESULT _stdcall ScheduleEnabled([in] VARIANT_BOOL Value);
      [propget, id(0x000000CF)]
      HRESULT _stdcall ScheduleType([out, retval] long* Value);
      [propput, id(0x000000CF)]
```

```
HRESULT _stdcall ScheduleType([in] long Value);
[propget, id(0x000000D0)]
HRESULT _stdcall ScheduleInterval([out, retval] long* Value);
[propput, id(0x000000D0)]
HRESULT _stdcall ScheduleInterval([in] long Value);
[propget, id(0x000000D1)]
HRESULT _stdcall ScheduleEndType([out, retval] long* Value);
[propput, id(0x000000D1)]
HRESULT _stdcall ScheduleEndType([in] long Value);
[propget, id(0x000000D2)]
HRESULT _stdcall NoOfMessages([out, retval] long* Value);
[propput, id(0x000000D2)]
HRESULT _stdcall NoOfMessages([in] long Value);
[propget, id(0x000000D3)]
HRESULT _stdcall EndDate([out, retval] DATE* Value);
[propput, id(0x000000D3)]
HRESULT _stdcall EndDate([in] DATE Value);
[propget, id(0x000000D4)]
HRESULT _stdcall StartDate([out, retval] DATE* Value);
[propput, id(0x000000D4)]
HRESULT _stdcall StartDate([in] DATE Value);
[propget, id(0x000000D5)]
HRESULT _stdcall SentMessagesNo([out, retval] long* Value);
[propget, id(0x000000D6)]
HRESULT _stdcall LastMessageSent([out, retval] DATE* Value);
[propget, id(0x000000D7)]
HRESULT _stdcall MessagesPaused([out, retval] long* Value);
[propput, id(0x000000D7)]
HRESULT _stdcall MessagesPaused([in] long Value);
[propget, id(0x000000D8)]
HRESULT _stdcall MessagePriority([out, retval] long* Value);
[propput, id(0x000000D8)]
HRESULT _stdcall MessagePriority([in] long Value);
[propget, id(0x000000D9)]
HRESULT _stdcall MessageAgeLimit([out, retval] long* Value);
[propput, id(0x000000D9)]
HRESULT _stdcall MessageAgeLimit([in] long Value);
[propget, id(0x000000DA)]
HRESULT _stdcall EmailSubject([out, retval] BSTR* Value);
[propput, id(0x000000DA)]
HRESULT _stdcall EmailSubject([in] BSTR Value);
[propget, id(0x000000DB)]
HRESULT _stdcall MessageText([out, retval] BSTR* Value);
[propput, id(0x000000DB)]
HRESULT _stdcall MessageText([in] BSTR Value);
[propget, id(0x000000DC)]
HRESULT _stdcall ScheduleContacts([out, retval] IStrings** Value);
[propget, id(0x000000DD)]
HRESULT _stdcall ScheduleContactsStr([out, retval] BSTR* Value);
```

```
  [propget, id(0x000000DE)]
  HRESULT _stdcall ScheduleContact([in] long contactId, [out, retval]
    IContact** Value);
  [id(0x000000DF)]
  HRESULT _stdcall SetScheduleContacts([in] IStrings* contactIds);
  [id(0x000000E0)]
  HRESULT _stdcall SetScheduleContactsStr([in] BSTR contactIds);
  [id(0x000000E1)]
  HRESULT _stdcall SetSchedule([in] long scheduleId);
  [id(0x000000E2)]
  HRESULT _stdcall Save(void);
  [propget, id(0x000000E3)]
  HRESULT _stdcall AllowEmptyMsg([out, retval] VARIANT_BOOL* Value);
  [propput, id(0x000000E3)]
  HRESULT _stdcall AllowEmptyMsg([in] VARIANT_BOOL Value);
  [id(0x000000E4)]
  HRESULT _stdcall HasCorrectSettings([out] BSTR* errorMsg, [out, retval]
    VARIANT_BOOL* result);
  [id(0x000000E5)]
  HRESULT _stdcall Test([out] BSTR* errorMsg, [out, retval] IStrings**
    result);
  [propget, id(0x000000E6)]
  HRESULT _stdcall Criteria([out, retval] BSTR* Value);
};
```

IReport

```
[
  uuid(ADCB8D15-62B2-438E-BC26-D51B6D66AF7F),
  helpstring("Dispatch interface for Report Object"),
  dual,
  oleautomation
]
interface IReport: IDispatch
{
  [propget, id(0x000000C9)]
  HRESULT _stdcall ReportId([out, retval] long* Value);
  [propget, id(0x000000CA)]
  HRESULT _stdcall ReportName([out, retval] BSTR* Value);
  [propput, id(0x000000CA)]
  HRESULT _stdcall ReportName([in] BSTR Value);
  [propget, id(0x000000CB)]
  HRESULT _stdcall AccountOwner([out, retval] BSTR* Value);
  [propput, id(0x000000CB)]
  HRESULT _stdcall AccountOwner([in] BSTR Value);
  [propget, id(0x000000CC)]
  HRESULT _stdcall IsSystemObject([out, retval] VARIANT_BOOL* Value);
  [propget, id(0x000000CD)]
  HRESULT _stdcall Notes([out, retval] BSTR* Value);
```

```
[propput, id(0x000000CD)]
HRESULT _stdcall Notes([in] BSTR Value);
[propget, id(0x000000CE)]
HRESULT _stdcall DateRangeType([out, retval] long* Value);
[propput, id(0x000000CE)]
HRESULT _stdcall DateRangeType([in] long Value);
[propget, id(0x000000CF)]
HRESULT _stdcall DateFrom([out, retval] DATE* Value);
[propget, id(0x000000D0)]
HRESULT _stdcall DateTo([out, retval] DATE* Value);
[propget, id(0x000000D1)]
HRESULT _stdcall IsSorted([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D1)]
HRESULT _stdcall IsSorted([in] VARIANT_BOOL Value);
[propget, id(0x000000D2)]
HRESULT _stdcall SortAscending([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D2)]
HRESULT _stdcall SortAscending([in] VARIANT_BOOL Value);
[propget, id(0x000000D3)]
HRESULT _stdcall SortOnField([out, retval] long* Value);
[propput, id(0x000000D3)]
HRESULT _stdcall SortOnField([in] long Value);
[propget, id(0x000000D4)]
HRESULT _stdcall MsgStatuses([out, retval] long* Value);
[propput, id(0x000000D4)]
HRESULT _stdcall MsgStatuses([in] long Value);
[propget, id(0x000000D5)]
HRESULT _stdcall Priorities([out, retval] long* Value);
[propput, id(0x000000D5)]
HRESULT _stdcall Priorities([in] long Value);
[propget, id(0x000000D6)]
HRESULT _stdcall CarrierTypes([out, retval] long* Value);
[propput, id(0x000000D6)]
HRESULT _stdcall CarrierTypes([in] long Value);
[propget, id(0x000000D7)]
HRESULT _stdcall AddressTypes([out, retval] long* Value);
[propput, id(0x000000D7)]
HRESULT _stdcall AddressTypes([in] long Value);
[propget, id(0x000000D8)]
HRESULT _stdcall MsgCarriers([out, retval] BSTR* Value);
[propput, id(0x000000D8)]
HRESULT _stdcall MsgCarriers([in] BSTR Value);
[propget, id(0x000000D9)]
HRESULT _stdcall MsgAddresses([out, retval] BSTR* Value);
[propput, id(0x000000D9)]
HRESULT _stdcall MsgAddresses([in] BSTR Value);
[propget, id(0x000000DA)]
HRESULT _stdcall MsgAccounts([out, retval] BSTR* Value);
[propput, id(0x000000DA)]
```

```
HRESULT _stdcall MsgAccounts([in] BSTR Value);
[propget, id(0x000000DB)]
HRESULT _stdcall TextSearchType([out, retval] long* Value);
[propput, id(0x000000DB)]
HRESULT _stdcall TextSearchType([in] long Value);
[propget, id(0x000000DC)]
HRESULT _stdcall SearchText([out, retval] BSTR* Value);
[propput, id(0x000000DC)]
HRESULT _stdcall SearchText([in] BSTR Value);
[propget, id(0x000000DD)]
HRESULT _stdcall CaseSensitive([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000DD)]
HRESULT _stdcall CaseSensitive([in] VARIANT_BOOL Value);
[propget, id(0x000000DE)]
HRESULT _stdcall PartialMatch([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000DE)]
HRESULT _stdcall PartialMatch([in] VARIANT_BOOL Value);
[propget, id(0x000000DF)]
HRESULT _stdcall ReportColour([out, retval] long* Value);
[propput, id(0x000000DF)]
HRESULT _stdcall ReportColour([in] long Value);
[propget, id(0x000000E0)]
HRESULT _stdcall FontColour([out, retval] long* Value);
[propput, id(0x000000E0)]
HRESULT _stdcall FontColour([in] long Value);
[propget, id(0x000000E1)]
HRESULT _stdcall EmailCarrierId([out, retval] long* Value);
[propput, id(0x000000E1)]
HRESULT _stdcall EmailCarrierId([in] long Value);
[propget, id(0x000000E2)]
HRESULT _stdcall EmailTo([out, retval] IStrings** Value);
[propput, id(0x000000E2)]
HRESULT _stdcall EmailTo([in] IStrings* Value);
[propget, id(0x000000E3)]
HRESULT _stdcall IsAutoReport([out, retval] VARIANT_BOOL* Value);
[propget, id(0x000000E4)]
HRESULT _stdcall AutoReportType([out, retval] long* Value);
[propget, id(0x000000E5)]
HRESULT _stdcall AutoInterval([out, retval] long* Value);
[propget, id(0x000000E6)]
HRESULT _stdcall AutoStartTime([out, retval] DATE* Value);
[id(0x000000E7)]
HRESULT _stdcall SetAutoReport([in] VARIANT_BOOL autoEnabled, [in] long
  autoReportType, [in] long reportingInterval, [in] DATE autoStartDate);
[id(0x000000E8)]
HRESULT _stdcall SetReportDates([in] DATE dateFrom, [in] DATE dateTo);
[id(0x000000E9)]
HRESULT _stdcall SetReport([in] long reportId);
[id(0x000000EA)]
```

```
    HRESULT _stdcall Save(void);
    [propget, id(0x000000EB)]
    HRESULT _stdcall AllowEmpty([out, retval] VARIANT_BOOL* Value);
    [propput, id(0x000000EB)]
    HRESULT _stdcall AllowEmpty([in] VARIANT_BOOL Value);
    [propget, id(0x000000EC)]
    HRESULT _stdcall IncludeHeader([out, retval] VARIANT_BOOL* Value);
    [propput, id(0x000000EC)]
    HRESULT _stdcall IncludeHeader([in] VARIANT_BOOL Value);
  };
```

IMsgSettings

```
  [
    uuid(2B791897-E2B5-4795-8707-A36CF0582411),
    helpstring("Dispatch interface for MsgSettings Object"),
    dual,
    oleautomation
  ]
  interface IMsgSettings: IDispatch
  {
    [propget, id(0x000000C9)]
    HRESULT _stdcall MsgPriority([out, retval] long* Value);
    [propput, id(0x000000C9)]
    HRESULT _stdcall MsgPriority([in] long Value);
    [propget, id(0x000000CA)]
    HRESULT _stdcall MsgAge([out, retval] long* Value);
    [propput, id(0x000000CA)]
    HRESULT _stdcall MsgAge([in] long Value);
    [propget, id(0x000000CB)]
    HRESULT _stdcall TextAffixType([out, retval] long* Value);
    [propput, id(0x000000CB)]
    HRESULT _stdcall TextAffixType([in] long Value);
    [propget, id(0x000000CC)]
    HRESULT _stdcall DateAffixType([out, retval] long* Value);
    [propput, id(0x000000CC)]
    HRESULT _stdcall DateAffixType([in] long Value);
    [propget, id(0x000000CD)]
    HRESULT _stdcall MsgShowType([out, retval] long* Value);
    [propput, id(0x000000CD)]
    HRESULT _stdcall MsgShowType([in] long Value);
    [propget, id(0x000000CE)]
    HRESULT _stdcall ShowMsgNo([out, retval] long* Value);
    [propput, id(0x000000CE)]
    HRESULT _stdcall ShowMsgNo([in] long Value);
    [propget, id(0x000000CF)]
    HRESULT _stdcall ShowMsgsFrom([out, retval] DATE* Value);
    [propput, id(0x000000CF)]
    HRESULT _stdcall ShowMsgsFrom([in] DATE Value);
```

```
[propget, id(0x000000D0)]
HRESULT _stdcall ShowMsgsTo([out, retval] DATE* Value);
[propput, id(0x000000D0)]
HRESULT _stdcall ShowMsgsTo([in] DATE Value);
[propget, id(0x000000D1)]
HRESULT _stdcall EmailSubject([out, retval] BSTR* Value);
[propput, id(0x000000D1)]
HRESULT _stdcall EmailSubject([in] BSTR Value);
[propget, id(0x000000D2)]
HRESULT _stdcall TextAffix([out, retval] BSTR* Value);
[propput, id(0x000000D2)]
HRESULT _stdcall TextAffix([in] BSTR Value);
[propget, id(0x000000D3)]
HRESULT _stdcall DateAffix([out, retval] BSTR* Value);
[propput, id(0x000000D3)]
HRESULT _stdcall DateAffix([in] BSTR Value);
[propget, id(0x000000D4)]
HRESULT _stdcall ClearMsgOnSent([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D4)]
HRESULT _stdcall ClearMsgOnSent([in] VARIANT_BOOL Value);
[propget, id(0x000000D5)]
HRESULT _stdcall ClearContactsOnSent([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D5)]
HRESULT _stdcall ClearContactsOnSent([in] VARIANT_BOOL Value);
[propget, id(0x000000D6)]
HRESULT _stdcall AllowEmptyMsg([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D6)]
HRESULT _stdcall AllowEmptyMsg([in] VARIANT_BOOL Value);
[propget, id(0x000000D7)]
HRESULT _stdcall ShowSysContacts([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D7)]
HRESULT _stdcall ShowSysContacts([in] VARIANT_BOOL Value);
[propget, id(0x000000D8)]
HRESULT _stdcall ShowSysGroups([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D8)]
HRESULT _stdcall ShowSysGroups([in] VARIANT_BOOL Value);
[propget, id(0x000000D9)]
HRESULT _stdcall ShowSysTemplates([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000D9)]
HRESULT _stdcall ShowSysTemplates([in] VARIANT_BOOL Value);
[propget, id(0x000000DA)]
HRESULT _stdcall ShowSysSchedules([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000DA)]
HRESULT _stdcall ShowSysSchedules([in] VARIANT_BOOL Value);
[propget, id(0x000000DB)]
HRESULT _stdcall ShowSysReports([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000DB)]
HRESULT _stdcall ShowSysReports([in] VARIANT_BOOL Value);
[propget, id(0x000000DC)]
```

```
HRESULT _stdcall DisplayContactsFirst([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000DC)]
HRESULT _stdcall DisplayContactsFirst([in] VARIANT_BOOL Value);
[id(0x000000DD)]
HRESULT _stdcall SetMsgSettings([in] BSTR accountId);
[id(0x000000DE)]
HRESULT _stdcall Save(void);
[propget, id(0x000000DF)]
HRESULT _stdcall LastMsgId([out, retval] long* Value);
[propput, id(0x000000DF)]
HRESULT _stdcall LastMsgId([in] long Value);
[propget, id(0x000000E0)]
HRESULT _stdcall ContactDysplayType([out, retval] long* Value);
[propput, id(0x000000E0)]
HRESULT _stdcall ContactDysplayType([in] long Value);
[propget, id(0x000000E1)]
HRESULT _stdcall MsgDeleteType([out, retval] long* Value);
[propput, id(0x000000E1)]
HRESULT _stdcall MsgDeleteType([in] long Value);
[propget, id(0x000000E2)]
HRESULT _stdcall DeleteInterval([out, retval] long* Value);
[propput, id(0x000000E2)]
HRESULT _stdcall DeleteInterval([in] long Value);
[propget, id(0x000000E3)]
HRESULT _stdcall UseSameMsg([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000E3)]
HRESULT _stdcall UseSameMsg([in] VARIANT_BOOL Value);
[propget, id(0x000000E4)]
HRESULT _stdcall DuplicateContact([out, retval] VARIANT_BOOL* Value);
[propput, id(0x000000E4)]
HRESULT _stdcall DuplicateContact([in] VARIANT_BOOL Value);
[propget, id(0x000000E5)]
HRESULT _stdcall ContactSearchType([out, retval] long* Value);
[propput, id(0x000000E5)]
HRESULT _stdcall ContactSearchType([in] long Value);
[propget, id(0x000000E6)]
HRESULT _stdcall GroupSearchType([out, retval] long* Value);
[propput, id(0x000000E6)]
HRESULT _stdcall GroupSearchType([in] long Value);
[propget, id(0x000000E7)]
HRESULT _stdcall TemplateSearchType([out, retval] long* Value);
[propput, id(0x000000E7)]
HRESULT _stdcall TemplateSearchType([in] long Value);
[propget, id(0x000000E8)]
HRESULT _stdcall ContactSearchStr([out, retval] BSTR* Value);
[propput, id(0x000000E8)]
HRESULT _stdcall ContactSearchStr([in] BSTR Value);
[propget, id(0x000000E9)]
HRESULT _stdcall GroupSearchStr([out, retval] BSTR* Value);
```

```
  [propput, id(0x000000E9)]
  HRESULT _stdcall GroupSearchStr([in] BSTR Value);
  [propget, id(0x000000EA)]
  HRESULT _stdcall TemplateSearchStr([out, retval] BSTR* Value);
  [propput, id(0x000000EA)]
  HRESULT _stdcall TemplateSearchStr([in] BSTR Value);
};
```

Events

```
[
  uuid(421EE7A2-7651-48C2-92BC-DE6774DACFF5),
  helpstring("Events interface for IPageServerCom Object")
]
dispinterface IIPageServerComEvents
{
  properties:
  methods:
  [id(0x000000C9)]
  HRESULT OnConnect([in] VARIANT_BOOL success, [in] long connectionResult,
    [in] BSTR description, [in] BSTR accountId);
  [id(0x000000CA)]
  HRESULT OnLogin([in] VARIANT_BOOL success, [in] long loginResult, [in]
    BSTR description, [in] BSTR accountId);
  [id(0x000000CB)]
  HRESULT OnIPMessage([in] long transactionId, [in] BSTR accountId, [in]
    BSTR msgAddress, [in] BSTR resultStr, [in] BSTR msg, [in] long
    sendResult, [in] DATE dateSent, [in] DATE dateReceived, [in] long
    addressType);
};
```

CoClasses

```
[
    uuid(9DF339C5-57F4-4F9D-97A4-7D15BF345221),
    helpstring("IPageServerCom Object")
  ]
  coclass IPageServerCom
  {
    [default] interface IIPageServerCom;
    [default, source] dispinterface IIPageServerComEvents;
  };

  [
    uuid(717DEBCE-7574-4AA3-A0A6-2735B6BCBF30),
    helpstring("Account Object")
  ]
  coclass Account
  {
    [default] interface IAccount;
  };

  [
    uuid(E3A8035D-7DF3-4DE8-901F-E62E55274474),
    helpstring("Carrier Object")
  ]
  coclass Carrier
  {
    [default] interface ICarrier;
  };

  [
    uuid(5409E1F6-8435-43A2-A9F7-643159E98CE7),
    helpstring("Contact Object")
  ]
  coclass Contact
  {
    [default] interface IContact;
  };

  [
    uuid(4C0A85F6-91ED-433A-B067-2E2984CA8296),
    helpstring("Group Object")
  ]
  coclass Group
  {
    [default] interface IGroup;
  };
```

```
    [
      uuid(7A82C804-6EEF-4CA0-8380-03EA1D095535),
      helpstring("Template Object")
    ]
    coclass Template
    {
      [default] interface ITemplate;
    };

    [
      uuid(6D01D1D4-1923-46F2-B1F0-F2F6C0D457D8),
      helpstring("Schedule Object")
    ]
    coclass Schedule
    {
      [default] interface ISchedule;
    };

    [
      uuid(8033451C-BF61-414D-AAAB-A1C2F6B3D6CF),
      helpstring("Report Object")
    ]
    coclass Report
    {
      [default] interface IReport;
    };

    [
      uuid(803CD4F6-8BAA-4A05-87A9-52CB57DC50B5),
      helpstring("MsgSettings Object")
    ]
    coclass MsgSettings
    {
      [default] interface IMsgSettings;
    };

  };
```

Pascal

unit IPageServer_TLB;

Library and Constants

```
LIBID: {9CA17DCD-E648-4009-941B-B9F88AB8F427}
HelpString: i-Page Server 4.0 Library
```

Dependency:

(1) v2.0 stdole, (C:\Windows\system32\stdole2.tlb)

(2) v4.0 StdVCL, (stdvcl40.dll)

Compiler:

```
{$TYPEDADDRESS OFF}//Unit must be compiled without type-checked
                    pointers.
{$WARN SYMBOL_PLATFORM OFF}//Compiled for Windows platform
{$WRITEABLECONST ON}
{$VARPROPSETTER ON}
{$ALIGN 4}
```

Uses units:

```
Windows, ActiveX, Classes, Graphics, OleServer, StdVCL, Variants;
```

GUIDS declared in the TypeLibrary use following prefixes:

```
Type Libraries:    LIBID_xxxx
CoClasses:         CLASS_xxxx
DISPInterfaces:    DIID_xxxx
Non-DISP interfaces:     IID_xxxx
```

```
const
      IPageServerMajorVersion = 4;//TypeLibrary major version
      IPageServerMinorVersion = 0;//TypeLibrary minor version

      LIBID_IPageServer: TGUID = '{9CA17DCD-E648-4009-941B-B9F88AB8F427}';
      IID_IIPageServerCom: TGUID = '{05FC6A4E-1D8E-440A-87DC-BB82B6B2F6D8}';
      DIID_IIPageServerComEvents: TGUID = '{421EE7A2-7651-48C2-92BC-
            DE6774DACFF5}';
      CLASS_IPageServerCom: TGUID = '{9DF339C5-57F4-4F9D-97A4-7D15BF345221}';
      IID_IAccount: TGUID = '{49DCE0D5-FDD6-4424-B77D-6AC68A3FD7D6}';
      CLASS_Account: TGUID = '{717DEBCE-7574-4AA3-A0A6-2735B6BCBF30}';
      IID_ICarrier: TGUID = '{25FC3D06-4448-4B22-9357-36450F6952EC}';
```

```
    CLASS_Carrier: TGUID = '{E3A8035D-7DF3-4DE8-901F-E62E55274474}';
    IID_IContact: TGUID = '{82C8C40F-AF9F-4FA4-8366-F1C31BCC287C}';
    CLASS_Contact: TGUID = '{5409E1F6-8435-43A2-A9F7-643159E98CE7}';
    IID_IGroup: TGUID = '{F8ADE1C5-B2BB-4255-94A2-73C2B762CC68}';
    CLASS_Group: TGUID = '{4C0A85F6-91ED-433A-B067-2E2984CA8296}';
    IID_ITemplate: TGUID = '{07BBC3EC-F61E-4A81-BC82-B90E6532A937}';
    CLASS_Template: TGUID = '{7A82C804-6EEF-4CA0-8380-03EA1D095535}';
    IID_ISchedule: TGUID = '{43AE1889-83E0-4418-B97F-FA021624F2CE}';
    CLASS_Schedule: TGUID = '{6D01D1D4-1923-46F2-B1F0-F2F6C0D457D8}';
    IID_IReport: TGUID = '{ADCB8D15-62B2-438E-BC26-D51B6D66AF7F}';
    CLASS_Report: TGUID = '{8033451C-BF61-414D-AAAB-A1C2F6B3D6CF}';
    IID_IMsgSettings: TGUID = '{2B791897-E2B5-4795-8707-A36CF0582411}';
    CLASS_MsgSettings: TGUID = '{803CD4F6-8BAA-4A05-87A9-52CB57DC50B5}';
    IID_IFolder: TGUID = '{2C53703A-31D4-4B18-9C83-3F5BBF405517}';
    CLASS_Folder: TGUID = '{831AEAA7-D342-4BAB-8646-C0E036E6BC43}';

type
      TSysObjectType = TOleEnum;
const  //Constants for enum TSysObjectType
    sotContact = $00000000;
    sotGroup = $00000001;
    sotTemplate = $00000002;
    sotSchedule = $00000003;
    sotReport = $00000004;
    sotFolder = $00000005;
```

Interfaces

```
  IIPageServerCom = interface;
  IIPageServerComDisp = dispinterface;
  IIPageServerComEvents = dispinterface;
  IAccount = interface;
  IAccountDisp = dispinterface;
  ICarrier = interface;
  ICarrierDisp = dispinterface;
  IContact = interface;
  IContactDisp = dispinterface;
  IGroup = interface;
  IGroupDisp = dispinterface;
  ITemplate = interface;
  ITemplateDisp = dispinterface;
  ISchedule = interface;
  IScheduleDisp = dispinterface;
  IReport = interface;
  IReportDisp = dispinterface;
  IMsgSettings = interface;
  IMsgSettingsDisp = dispinterface;
  IFolder = interface;
  IFolderDisp = dispinterface;
```

CoClasses

Each CoClass is matched to its default Interface

```
IPageServerCom = IIPageServerCom;
Account = IAccount;
Carrier = ICarrier;
Contact = IContact;
Group = IGroup;
Template = ITemplate;
Schedule = ISchedule;
Report = IReport;
MsgSettings = IMsgSettings;
Folder = IFolder;
```

Interfaces

IIPageServerCom

```
IIPageServerCom = interface(IDispatch)
['{05FC6A4E-1D8E-440A-87DC-BB82B6B2F6D8}']
  procedure ConnectToServer(const accountId: WideString; const ipAddress:
    WideString); safecall;
  function Start: WordBool; safecall;
  function Stop: WordBool; safecall;
  procedure Terminate; safecall;
  procedure Login(const userName: WideString; const password: WideString;
                  const ipAddress: WideString; ecoding: Integer); safecall;
  procedure Logout(const accountId: WideString; const ipAddress:
    WideString); safecall;
  procedure ClientReady(const accountId: WideString; const accountIP:
    WideString;isReady: WordBool); safecall;
  procedure SendMsg(const accountId: WideString; const accountIP:
    WideString;const msgData: IStrings); safecall;
  function RunReport(reportId: Integer; out resultMsg: WideString):
    WordBool; safecall;
  function NewMsgObject(objType: TSysObjectType; const accountId:
    WideString): IDispatch; safecall;
  function FindMsgObjects(objType: TSysObjectType; const accountId:
    WideString;searchType: Integer; const searchStr: WideString): IStrings;
    safecall;
  procedure AssignMsgObjects(objType: TSysObjectType; const accountId:
    WideString;const objIds: IStrings); safecall;
  function DeleteMsgObject(objId: Integer; objType: TSysObjectType; const
    accountId: WideString): WordBool; safecall;
  function Get_MsgObjectTypeName(objType: TSysObjectType): WideString;
    safecall;
  function Get_IsActive: WordBool; safecall;
  function Get_Account(const accountId: WideString): IAccount; safecall;
  function Get_Carriers: IStrings; safecall;
  function Get_Carrier(carrierId: Integer): ICarrier; safecall;
  function Get_Contact(contactId: Integer): IContact; safecall;
  function Get_Group(groupId: Integer): IGroup; safecall;
  function Get_Template(templateId: Integer): ITemplate; safecall;
  function Get_Schedule(scheduleId: Integer): ISchedule; safecall;
  function Get_Report(reportId: Integer): IReport; safecall;
  function LoginWeb(const userName: WideString; const password: WideString;
                    const ipAddress: WideString; const browser: WideString;
                    out errorMsg: WideString): WordBool; safecall;
  procedure LogoutWeb(const account: WideString; const ipAddress:
    WideString); safecall;
  function Get_WebTimeout: Integer; safecall;
```

```
    function ReportOnMsgs(const accountId: WideString; msgShowType: Integer;
      noOfMsgs: Integer;fromDate: TDateTime; toDate: TDateTime): IStrings;
      safecall;
    function ReturnMsgResultWeb(const accountId: WideString; const ipAddress:
      WideString): IStrings; safecall;
    function Get_Folder(folderId: Integer): IFolder; safecall;
    property MsgObjectTypeName[objType: TSysObjectType]: WideString read
      Get_MsgObjectTypeName;
    property IsActive: WordBool read Get_IsActive;
    property Account[const accountId: WideString]: IAccount read Get_Account;
    property Carriers: IStrings read Get_Carriers;
    property Carrier[carrierId: Integer]: ICarrier read Get_Carrier;
    property Contact[contactId: Integer]: IContact read Get_Contact;
    property Group[groupId: Integer]: IGroup read Get_Group;
    property Template[templateId: Integer]: ITemplate read Get_Template;
    property Schedule[scheduleId: Integer]: ISchedule read Get_Schedule;
    property Report[reportId: Integer]: IReport read Get_Report;
    property WebTimeout: Integer read Get_WebTimeout;
    property Folder[folderId: Integer]: IFolder read Get_Folder;
  end;
```

IIPageServerComDisp

```
  IIPageServerComDisp = dispinterface
  ['{05FC6A4E-1D8E-440A-87DC-BB82B6B2F6D8}']
    procedure ConnectToServer(const accountId: WideString; const ipAddress:
      WideString); dispid     201;
    function Start: WordBool; dispid 202;
    function Stop: WordBool; dispid 203;
    procedure Terminate; dispid 204;
    procedure Login(const userName: WideString; const password: WideString;
      const ipAddress: WideString; ecoding: Integer); dispid 205;
    procedure Logout(const accountId: WideString; const ipAddress:
      WideString); dispid 206;
    procedure ClientReady(const accountId: WideString; const accountIP:
      WideString;isReady: WordBool); dispid 207;
    procedure SendMsg(const accountId: WideString; const accountIP:
      WideString;const msgData: IStrings); dispid 208;
    function RunReport(reportId: Integer; out resultMsg: WideString):
      WordBool; dispid 209;
    function NewMsgObject(objType: TSysObjectType; const accountId:
      WideString): IDispatch; dispid 210;
    function FindMsgObjects(objType: TSysObjectType; const accountId:
      WideString;searchType: Integer; const searchStr: WideString): IStrings;
      dispid 211;
    procedure AssignMsgObjects(objType: TSysObjectType; const accountId:
      WideString;const objIds: IStrings); dispid 212;
    function DeleteMsgObject(objId: Integer; objType: TSysObjectType; const
      accountId: WideString): WordBool; dispid 213;
    property MsgObjectTypeName[objType: TSysObjectType]: WideString readonly
      dispid 214;
    property IsActive: WordBool readonly dispid 215;
    property Account[const accountId: WideString]: IAccount readonly
      dispid 216;
    property Carriers: IStrings readonly dispid 217;
    property Carrier[carrierId: Integer]: ICarrier readonly dispid 218;
    property Contact[contactId: Integer]: IContact readonly dispid 219;
    property Group[groupId: Integer]: IGroup readonly dispid 220;
    property Template[templateId: Integer]: ITemplate readonly dispid 221;
    property Schedule[scheduleId: Integer]: ISchedule readonly dispid 222;
    property Report[reportId: Integer]: IReport readonly dispid 223;
    function LoginWeb(const userName: WideString; const password: WideString;
                      const ipAddress: WideString; const browser: WideString;
                      out errorMsg: WideString): WordBool; dispid 224;
    procedure LogoutWeb(const account: WideString; const ipAddress:
      WideString); dispid 225;
    property WebTimeout: Integer readonly dispid 226;
    function ReportOnMsgs(const accountId: WideString; msgShowType: Integer;
      noOfMsgs: Integer;fromDate: TDateTime; toDate: TDateTime): IStrings;
      dispid 227;
    function ReturnMsgResultWeb(const accountId: WideString; const ipAddress:
      WideString): IStrings; dispid 228;
    property Folder[folderId: Integer]: IFolder readonly dispid 229;
  end;
```

IAccount

```
IAccount = interface(IDispatch)
['{49DCE0D5-FDD6-4424-B77D-6AC68A3FD7D6}']
  function Get_AccountId: WideString; safecall;
  function Get_AccountRights: Integer; safecall;
  function Get_PagerId: WideString; safecall;
  procedure Set_PagerId(const Value: WideString); safecall;
  function Get_MobileNo: WideString; safecall;
  procedure Set_MobileNo(const Value: WideString); safecall;
  function Get_Email: WideString; safecall;
  procedure Set_Email(const Value: WideString); safecall;
  function Get_DisplayName: WideString; safecall;
  procedure Set_DisplayName(const Value: WideString); safecall;
  procedure SetAccount(const accountId: WideString); safecall;
  procedure Save; safecall;
  function ChangePassword(const oldPassword: WideString; const newPassword:
    WideString;const confirmPassword: WideString; encoding: Integer;out
    errorMsg: WideString): WordBool; safecall;
  procedure AssignSystemObject(objId: Integer; objType: TSysObjectType);
    safecall;
  function RemoveSystemObject(objId: Integer; objType: TSysObjectType; out
    errorMsg: WideString): WordBool; safecall;
  function Get_MsgObjects(objType: TSysObjectType): IStrings; safecall;
  procedure DeleteAllMessages; safecall;
  procedure DeleteOldMessages(olderThanDays: Integer); safecall;
  procedure DeleteSelectedMessage(msgId: Integer); safecall;
  function Get_MsgSettings: IMsgSettings; safecall;
  function GetMsgObjects(objType: TSysObjectType; searchType: Integer; const
    searchStr: WideString): IStrings; safecall;
  property AccountId: WideString read Get_AccountId;
  property AccountRights: Integer read Get_AccountRights;
  property PagerId: WideString read Get_PagerId write Set_PagerId;
  property MobileNo: WideString read Get_MobileNo write Set_MobileNo;
  property Email: WideString read Get_Email write Set_Email;
  property DisplayName: WideString read Get_DisplayName write
    Set_DisplayName;
  property MsgObjects[objType: TSysObjectType]: IStrings read
    Get_MsgObjects;
  property MsgSettings: IMsgSettings read Get_MsgSettings;
end;
```

IAccountDisp

```
  IAccountDisp = dispinterface
  ['{49DCE0D5-FDD6-4424-B77D-6AC68A3FD7D6}']
    property AccountId: WideString readonly dispid 201;
    property AccountRights: Integer readonly dispid 202;
    property PagerId: WideString dispid 203;
    property MobileNo: WideString dispid 204;
    property Email: WideString dispid 205;
    property DisplayName: WideString dispid 206;
    procedure SetAccount(const accountId: WideString); dispid 212;
    procedure Save; dispid 213;
    function ChangePassword(const oldPassword: WideString; const newPassword:
      WideString;const confirmPassword: WideString; encoding: Integer;out
      errorMsg: WideString): WordBool; dispid 214;
    procedure AssignSystemObject(objId: Integer; objType: TSysObjectType);
      dispid 215;
    function RemoveSystemObject(objId: Integer; objType: TSysObjectType; out
      errorMsg: WideString): WordBool; dispid 216;
    property MsgObjects[objType: TSysObjectType]: IStrings readonly
      dispid 217;
    procedure DeleteAllMessages; dispid 207;
    procedure DeleteOldMessages(olderThanDays: Integer); dispid 208;
    procedure DeleteSelectedMessage(msgId: Integer); dispid 209;
    property MsgSettings: IMsgSettings readonly dispid 210;
    function GetMsgObjects(objType: TSysObjectType; searchType: Integer; const
      searchStr: WideString): IStrings; dispid 211;
  end;
```

ICarrier

```
ICarrier = interface(IDispatch)
['{25FC3D06-4448-4B22-9357-36450F6952EC}']
  function Get_CarrierId: Integer; safecall;
  function Get_CarrierName: WideString; safecall;
  function Get_CarrierType: Integer; safecall;
  function Get_CarrierTypeStr: WideString; safecall;
  function Get_AddressType: Integer; safecall;
  function Get_AddressTypeStr: WideString; safecall;
  function Get_MaxMsgLength: Integer; safecall;
  function Get_CarrierInfo: IStrings; safecall;
  procedure SetCarrier(carrierId: Integer); safecall;
  property CarrierId: Integer read Get_CarrierId;
  property CarrierName: WideString read Get_CarrierName;
  property CarrierType: Integer read Get_CarrierType;
  property CarrierTypeStr: WideString read Get_CarrierTypeStr;
  property AddressType: Integer read Get_AddressType;
  property AddressTypeStr: WideString read Get_AddressTypeStr;
  property MaxMsgLength: Integer read Get_MaxMsgLength;
  property CarrierInfo: IStrings read Get_CarrierInfo;
end;
```

ICarrierDisp

```
ICarrierDisp = dispinterface
['{25FC3D06-4448-4B22-9357-36450F6952EC}']
  property CarrierId: Integer readonly dispid 201;
  property CarrierName: WideString readonly dispid 202;
  property CarrierType: Integer readonly dispid 203;
  property CarrierTypeStr: WideString readonly dispid 204;
  property AddressType: Integer readonly dispid 205;
  property AddressTypeStr: WideString readonly dispid 206;
  property MaxMsgLength: Integer readonly dispid 207;
  property CarrierInfo: IStrings readonly dispid 208;
  procedure SetCarrier(carrierId: Integer); dispid 209;
end;
```

IContact

```
IContact = interface(IDispatch)
['{82C8C40F-AF9F-4FA4-8366-F1C31BCC287C}']
  function Get_ContactId: Integer; safecall;
  function Get_ContactName: WideString; safecall;
  procedure Set_ContactName(const Value: WideString); safecall;
  function Get_AccountOwner: WideString; safecall;
  procedure Set_AccountOwner(const Value: WideString); safecall;
  function Get_IsSystemObject: WordBool; safecall;
  function Get_Notes: WideString; safecall;
```

```
    procedure Set_Notes(const Value: WideString); safecall;
    function Get_ContactAddress: WideString; safecall;
    procedure Set_ContactAddress(const Value: WideString); safecall;
    function Get_CarrierId: Integer; safecall;
    procedure Set_CarrierId(Value: Integer); safecall;
    function Get_CarrierName: WideString; safecall;
    function Get_CarrierType: WideString; safecall;
    function Get_AddressType: Integer; safecall;
    function Get_AddressTypeStr: WideString; safecall;
    procedure SetContact(contactId: Integer); safecall;
    procedure Save; safecall;
    function Get_EncryptKey: WideString; safecall;
    procedure Set_EncryptKey(const Value: WideString); safecall;
    function Get_EncryptLanguage: Integer; safecall;
    procedure Set_EncryptLanguage(Value: Integer); safecall;
    property ContactId: Integer read Get_ContactId;
    property ContactName: WideString read Get_ContactName write
      Set_ContactName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property ContactAddress: WideString read Get_ContactAddress write
      Set_ContactAddress;
    property CarrierId: Integer read Get_CarrierId write Set_CarrierId;
    property CarrierName: WideString read Get_CarrierName;
    property CarrierType: WideString read Get_CarrierType;
    property AddressType: Integer read Get_AddressType;
    property AddressTypeStr: WideString read Get_AddressTypeStr;
    property EncryptKey: WideString read Get_EncryptKey write Set_EncryptKey;
    property EncryptLanguage: Integer read Get_EncryptLanguage write
      Set_EncryptLanguage;
  end;
```

IContactDisp

```
  IContactDisp = dispinterface
  ['{82C8C40F-AF9F-4FA4-8366-F1C31BCC287C}']
    property ContactId: Integer readonly dispid 201;
    property ContactName: WideString dispid 202;
    property AccountOwner: WideString dispid 203;
    property IsSystemObject: WordBool readonly dispid 204;
    property Notes: WideString dispid 205;
    property ContactAddress: WideString dispid 206;
    property CarrierId: Integer dispid 207;
    property CarrierName: WideString readonly dispid 208;
    property CarrierType: WideString readonly dispid 209;
    property AddressType: Integer readonly dispid 210;
    property AddressTypeStr: WideString readonly dispid 211;
    procedure SetContact(contactId: Integer); dispid 212;
    procedure Save; dispid 213;
    property EncryptKey: WideString dispid 214;
    property EncryptLanguage: Integer dispid 215;
  end;
```

IGroup

```
  IGroup = interface(IDispatch)
  ['{F8ADE1C5-B2BB-4255-94A2-73C2B762CC68}']
    function Get_GroupId: Integer; safecall;
    function Get_GroupName: WideString; safecall;
    procedure Set_GroupName(const Value: WideString); safecall;
    function Get_AccountOwner: WideString; safecall;
    procedure Set_AccountOwner(const Value: WideString); safecall;
    function Get_IsSystemObject: WordBool; safecall;
    function Get_Notes: WideString; safecall;
    procedure Set_Notes(const Value: WideString); safecall;
    function Get_GroupContacts: IStrings; safecall;
    function Get_GroupContact(contactId: Integer): IContact; safecall;
    function Get_GroupContactByName(const contactName: WideString): IContact;
      safecall;
    procedure SetGroupContacts(const contactIds: IStrings); safecall;
    procedure SetGroup(groupId: Integer); safecall;
    procedure Save; safecall;
    property GroupId: Integer read Get_GroupId;
    property GroupName: WideString read Get_GroupName write Set_GroupName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property GroupContacts: IStrings read Get_GroupContacts;
    property GroupContact[contactId: Integer]: IContact read Get_GroupContact;
    property GroupContactByName[const contactName: WideString]: IContact read
      Get_GroupContactByName;
  end;
```

IGroupDisp

```
  IGroupDisp = dispinterface
  ['{F8ADE1C5-B2BB-4255-94A2-73C2B762CC68}']
    property GroupId: Integer readonly dispid 201;
    property GroupName: WideString dispid 202;
    property AccountOwner: WideString dispid 203;
    property IsSystemObject: WordBool readonly dispid 204;
    property Notes: WideString dispid 205;
    property GroupContacts: IStrings readonly dispid 206;
    property GroupContact[contactId: Integer]: IContact readonly dispid 207;
    property GroupContactByName[const contactName: WideString]: IContact
      readonly dispid 208;
    procedure SetGroupContacts(const contactIds: IStrings); dispid 209;
    procedure SetGroup(groupId: Integer); dispid 210;
    procedure Save; dispid 211;
  end;
```

ITemplate

```
  ITemplate = interface(IDispatch)
  ['{07BBC3EC-F61E-4A81-BC82-B90E6532A937}']
    function Get_TemplateId: Integer; safecall;
    function Get_TemplateName: WideString; safecall;
    procedure Set_TemplateName(const Value: WideString); safecall;
    function Get_AccountOwner: WideString; safecall;
    procedure Set_AccountOwner(const Value: WideString); safecall;
    function Get_IsSystemObject: WordBool; safecall;
    function Get_Notes: WideString; safecall;
    procedure Set_Notes(const Value: WideString); safecall;
    function Get_TemplateText: WideString; safecall;
    procedure Set_TemplateText(const Value: WideString); safecall;
    procedure SetTemplate(templateId: Integer); safecall;
    procedure Save; safecall;
    property TemplateId: Integer read Get_TemplateId;
    property TemplateName: WideString read Get_TemplateName write
      Set_TemplateName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property TemplateText: WideString read Get_TemplateText write
      Set_TemplateText;
  end;
```

ITemplateDisp

```
  ITemplateDisp = dispinterface
  ['{07BBC3EC-F61E-4A81-BC82-B90E6532A937}']
    property TemplateId: Integer readonly dispid 201;
    property TemplateName: WideString dispid 202;
    property AccountOwner: WideString dispid 203;
    property IsSystemObject: WordBool readonly dispid 204;
    property Notes: WideString dispid 205;
    property TemplateText: WideString dispid 206;
    procedure SetTemplate(templateId: Integer); dispid 207;
    procedure Save; dispid 208;
  end;
```

ISchedule

```
ISchedule = interface(IDispatch)
['{43AE1889-83E0-4418-B97F-FA021624F2CE}']
  function Get_ScheduleId: Integer; safecall;
  function Get_ScheduleName: WideString; safecall;
  procedure Set_ScheduleName(const Value: WideString); safecall;
  function Get_AccountOwner: WideString; safecall;
  procedure Set_AccountOwner(const Value: WideString); safecall;
  function Get_IsSystemObject: WordBool; safecall;
  function Get_Notes: WideString; safecall;
  procedure Set_Notes(const Value: WideString); safecall;
  function Get_ScheduleEnabled: WordBool; safecall;
  procedure Set_ScheduleEnabled(Value: WordBool); safecall;
  function Get_ScheduleType: Integer; safecall;
  procedure Set_ScheduleType(Value: Integer); safecall;
  function Get_ScheduleInterval: Integer; safecall;
  procedure Set_ScheduleInterval(Value: Integer); safecall;
  function Get_ScheduleEndType: Integer; safecall;
  procedure Set_ScheduleEndType(Value: Integer); safecall;
  function Get_NoOfMessages: Integer; safecall;
  procedure Set_NoOfMessages(Value: Integer); safecall;
  function Get_EndDate: TDateTime; safecall;
  procedure Set_EndDate(Value: TDateTime); safecall;
  function Get_StartDate: TDateTime; safecall;
  procedure Set_StartDate(Value: TDateTime); safecall;
  function Get_SentMessagesNo: Integer; safecall;
  function Get_LastMessageSent: TDateTime; safecall;
  function Get_MessagesPaused: Integer; safecall;
  procedure Set_MessagesPaused(Value: Integer); safecall;
  function Get_MessagePriority: Integer; safecall;
  procedure Set_MessagePriority(Value: Integer); safecall;
  function Get_MessageAgeLimit: Integer; safecall;
  procedure Set_MessageAgeLimit(Value: Integer); safecall;
  function Get_EmailSubject: WideString; safecall;
  procedure Set_EmailSubject(const Value: WideString); safecall;
  function Get_MessageText: WideString; safecall;
  procedure Set_MessageText(const Value: WideString); safecall;
  function Get_ScheduleContacts: IStrings; safecall;
  function Get_ScheduleContactsStr: WideString; safecall;
  function Get_ScheduleContact(contactId: Integer): IContact; safecall;
  procedure SetScheduleContacts(const contactIds: IStrings); safecall;
  procedure SetScheduleContactsStr(const contactIds: WideString); safecall;
  procedure SetSchedule(scheduleId: Integer); safecall;
  procedure Save; safecall;
  function Get_AllowEmptyMsg: WordBool; safecall;
  procedure Set_AllowEmptyMsg(Value: WordBool); safecall;
  function HasCorrectSettings(out errorMsg: WideString): WordBool; safecall;
  function Test(out errorMsg: WideString): IStrings; safecall;
```

```
    function Get_Criteria: WideString; safecall;
    property ScheduleId: Integer read Get_ScheduleId;
    property ScheduleName: WideString read Get_ScheduleName write
      Set_ScheduleName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property ScheduleEnabled: WordBool read Get_ScheduleEnabled write
      Set_ScheduleEnabled;
    property ScheduleType: Integer read Get_ScheduleType write
      Set_ScheduleType;
    property ScheduleInterval: Integer read Get_ScheduleInterval write
      Set_ScheduleInterval;
    property ScheduleEndType: Integer read Get_ScheduleEndType write
      Set_ScheduleEndType;
    property NoOfMessages: Integer read Get_NoOfMessages write
      Set_NoOfMessages;
    property EndDate: TDateTime read Get_EndDate write Set_EndDate;
    property StartDate: TDateTime read Get_StartDate write Set_StartDate;
    property SentMessagesNo: Integer read Get_SentMessagesNo;
    property LastMessageSent: TDateTime read Get_LastMessageSent;
    property MessagesPaused: Integer read Get_MessagesPaused write
      Set_MessagesPaused;
    property MessagePriority: Integer read Get_MessagePriority write
      Set_MessagePriority;
    property MessageAgeLimit: Integer read Get_MessageAgeLimit write
      Set_MessageAgeLimit;
    property EmailSubject: WideString read Get_EmailSubject write
      Set_EmailSubject;
    property MessageText: WideString read Get_MessageText write
      Set_MessageText;
    property ScheduleContacts: IStrings read Get_ScheduleContacts;
    property ScheduleContactsStr: WideString read Get_ScheduleContactsStr;
    property ScheduleContact[contactId: Integer]: IContact read
      Get_ScheduleContact;
    property AllowEmptyMsg: WordBool read Get_AllowEmptyMsg write
      Set_AllowEmptyMsg;
    property Criteria: WideString read Get_Criteria;
  end;
```

IScheduleDisp

```
  IScheduleDisp = dispinterface
    ['{43AE1889-83E0-4418-B97F-FA021624F2CE}']
    property ScheduleId: Integer readonly dispid 201;
    property ScheduleName: WideString dispid 202;
    property AccountOwner: WideString dispid 203;
    property IsSystemObject: WordBool readonly dispid 204;
    property Notes: WideString dispid 205;
    property ScheduleEnabled: WordBool dispid 206;
    property ScheduleType: Integer dispid 207;
    property ScheduleInterval: Integer dispid 208;
    property ScheduleEndType: Integer dispid 209;
    property NoOfMessages: Integer dispid 210;
    property EndDate: TDateTime dispid 211;
    property StartDate: TDateTime dispid 212;
    property SentMessagesNo: Integer readonly dispid 213;
    property LastMessageSent: TDateTime readonly dispid 214;
    property MessagesPaused: Integer dispid 215;
    property MessagePriority: Integer dispid 216;
    property MessageAgeLimit: Integer dispid 217;
    property EmailSubject: WideString dispid 218;
    property MessageText: WideString dispid 219;
    property ScheduleContacts: IStrings readonly dispid 220;
    property ScheduleContactsStr: WideString readonly dispid 221;
    property ScheduleContact[contactId: Integer]: IContact readonly dispid
      222;
    procedure SetScheduleContacts(const contactIds: IStrings); dispid 223;
    procedure SetScheduleContactsStr(const contactIds: WideString); dispid
      224;
    procedure SetSchedule(scheduleId: Integer); dispid 225;
    procedure Save; dispid 226;
    property AllowEmptyMsg: WordBool dispid 227;
    function HasCorrectSettings(out errorMsg: WideString): WordBool; dispid
      228;
    function Test(out errorMsg: WideString): IStrings; dispid 229;
    property Criteria: WideString readonly dispid 230;
  end;
```

IReport

```
  IReport = interface(IDispatch)
  ['{ADCB8D15-62B2-438E-BC26-D51B6D66AF7F}']
    function Get_ReportId: Integer; safecall;
    function Get_ReportName: WideString; safecall;
    procedure Set_ReportName(const Value: WideString); safecall;
    function Get_AccountOwner: WideString; safecall;
    procedure Set_AccountOwner(const Value: WideString); safecall;
    function Get_IsSystemObject: WordBool; safecall;
    function Get_Notes: WideString; safecall;
    procedure Set_Notes(const Value: WideString); safecall;
    function Get_DateRangeType: Integer; safecall;
    procedure Set_DateRangeType(Value: Integer); safecall;
    function Get_DateFrom: TDateTime; safecall;
    function Get_DateTo: TDateTime; safecall;
    function Get_IsSorted: WordBool; safecall;
    procedure Set_IsSorted(Value: WordBool); safecall;
```

```
function Get_SortAscending: WordBool; safecall;
procedure Set_SortAscending(Value: WordBool); safecall;
function Get_SortOnField: Integer; safecall;
procedure Set_SortOnField(Value: Integer); safecall;
function Get_MsgStatuses: Integer; safecall;
procedure Set_MsgStatuses(Value: Integer); safecall;
function Get_Priorities: Integer; safecall;
procedure Set_Priorities(Value: Integer); safecall;
function Get_CarrierTypes: Integer; safecall;
procedure Set_CarrierTypes(Value: Integer); safecall;
function Get_AddressTypes: Integer; safecall;
procedure Set_AddressTypes(Value: Integer); safecall;
function Get_MsgCarriers: WideString; safecall;
procedure Set_MsgCarriers(const Value: WideString); safecall;
function Get_MsgAddresses: WideString; safecall;
procedure Set_MsgAddresses(const Value: WideString); safecall;
function Get_MsgAccounts: WideString; safecall;
procedure Set_MsgAccounts(const Value: WideString); safecall;
function Get_TextSearchType: Integer; safecall;
procedure Set_TextSearchType(Value: Integer); safecall;
function Get_SearchText: WideString; safecall;
procedure Set_SearchText(const Value: WideString); safecall;
function Get_CaseSensitive: WordBool; safecall;
procedure Set_CaseSensitive(Value: WordBool); safecall;
function Get_PartialMatch: WordBool; safecall;
procedure Set_PartialMatch(Value: WordBool); safecall;
function Get_ReportColour: Integer; safecall;
procedure Set_ReportColour(Value: Integer); safecall;
function Get_FontColour: Integer; safecall;
procedure Set_FontColour(Value: Integer); safecall;
function Get_EmailCarrierId: Integer; safecall;
procedure Set_EmailCarrierId(Value: Integer); safecall;
function Get_EmailTo: IStrings; safecall;
procedure Set_EmailTo(const Value: IStrings); safecall;
function Get_IsAutoReport: WordBool; safecall;
function Get_AutoReportType: Integer; safecall;
function Get_AutoInterval: Integer; safecall;
function Get_AutoStartTime: TDateTime; safecall;
procedure SetAutoReport(autoEnabled: WordBool; autoReportType:
  Integer;reportingInterval: Integer; autoStartDate: TDateTime); safecall;
procedure SetReportDates(dateFrom: TDateTime; dateTo: TDateTime);
  safecall;
procedure SetReport(reportId: Integer); safecall;
procedure Save; safecall;
function Get_AllowEmpty: WordBool; safecall;
procedure Set_AllowEmpty(Value: WordBool); safecall;
function Get_IncludeHeader: WordBool; safecall;
procedure Set_IncludeHeader(Value: WordBool); safecall;
property ReportId: Integer read Get_ReportId;
```

```
    property ReportName: WideString read Get_ReportName write Set_ReportName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property DateRangeType: Integer read Get_DateRangeType write
      Set_DateRangeType;
    property DateFrom: TDateTime read Get_DateFrom;
    property DateTo: TDateTime read Get_DateTo;
    property IsSorted: WordBool read Get_IsSorted write Set_IsSorted;
    property SortAscending: WordBool read Get_SortAscending write
      Set_SortAscending;
    property SortOnField: Integer read Get_SortOnField write Set_SortOnField;
    property MsgStatuses: Integer read Get_MsgStatuses write Set_MsgStatuses;
    property Priorities: Integer read Get_Priorities write Set_Priorities;
    property CarrierTypes: Integer read Get_CarrierTypes write
      Set_CarrierTypes;
    property AddressTypes: Integer read Get_AddressTypes write
      Set_AddressTypes;
    property MsgCarriers: WideString read Get_MsgCarriers write
      Set_MsgCarriers;
    property MsgAddresses: WideString read Get_MsgAddresses write
      Set_MsgAddresses;
    property MsgAccounts: WideString read Get_MsgAccounts write
      Set_MsgAccounts;
    property TextSearchType: Integer read Get_TextSearchType write
      Set_TextSearchType;
    property SearchText: WideString read Get_SearchText write Set_SearchText;
    property CaseSensitive: WordBool read Get_CaseSensitive write
      Set_CaseSensitive;
    property PartialMatch: WordBool read Get_PartialMatch write
      Set_PartialMatch;
    property ReportColour: Integer read Get_ReportColour write
      Set_ReportColour;
    property FontColour: Integer read Get_FontColour write Set_FontColour;
    property EmailCarrierId: Integer read Get_EmailCarrierId write
      Set_EmailCarrierId;
    property EmailTo: IStrings read Get_EmailTo write Set_EmailTo;
    property IsAutoReport: WordBool read Get_IsAutoReport;
    property AutoReportType: Integer read Get_AutoReportType;
    property AutoInterval: Integer read Get_AutoInterval;
    property AutoStartTime: TDateTime read Get_AutoStartTime;
    property AllowEmpty: WordBool read Get_AllowEmpty write Set_AllowEmpty;
    property IncludeHeader: WordBool read Get_IncludeHeader write
      Set_IncludeHeader;
  end;
```

IReportDisp

```
IReportDisp = dispinterface
['{ADCB8D15-62B2-438E-BC26-D51B6D66AF7F}']
  property ReportId: Integer readonly dispid 201;
  property ReportName: WideString dispid 202;
  property AccountOwner: WideString dispid 203;
  property IsSystemObject: WordBool readonly dispid 204;
  property Notes: WideString dispid 205;
  property DateRangeType: Integer dispid 206;
  property DateFrom: TDateTime readonly dispid 207;
  property DateTo: TDateTime readonly dispid 208;
  property IsSorted: WordBool dispid 209;
  property SortAscending: WordBool dispid 210;
  property SortOnField: Integer dispid 211;
  property MsgStatuses: Integer dispid 212;
  property Priorities: Integer dispid 213;
  property CarrierTypes: Integer dispid 214;
  property AddressTypes: Integer dispid 215;
  property MsgCarriers: WideString dispid 216;
  property MsgAddresses: WideString dispid 217;
  property MsgAccounts: WideString dispid 218;
  property TextSearchType: Integer dispid 219;
  property SearchText: WideString dispid 220;
  property CaseSensitive: WordBool dispid 221;
  property PartialMatch: WordBool dispid 222;
  property ReportColour: Integer dispid 223;
  property FontColour: Integer dispid 224;
  property EmailCarrierId: Integer dispid 225;
  property EmailTo: IStrings dispid 226;
  property IsAutoReport: WordBool readonly dispid 227;
  property AutoReportType: Integer readonly dispid 228;
  property AutoInterval: Integer readonly dispid 229;
  property AutoStartTime: TDateTime readonly dispid 230;
  procedure SetAutoReport(autoEnabled: WordBool; autoReportType:
    Integer;reportingInterval: Integer; autoStartDate: TDateTime); dispid
    231;
  procedure SetReportDates(dateFrom: TDateTime; dateTo: TDateTime); dispid
    232;
  procedure SetReport(reportId: Integer); dispid 233;
  procedure Save; dispid 234;
  property AllowEmpty: WordBool dispid 235;
  property IncludeHeader: WordBool dispid 236;
end;
```

IMsgSettings

```
IMsgSettings = interface(IDispatch)
['{2B791897-E2B5-4795-8707-A36CF0582411}']
```

```
function Get_MsgPriority: Integer; safecall;
procedure Set_MsgPriority(Value: Integer); safecall;
function Get_MsgAge: Integer; safecall;
procedure Set_MsgAge(Value: Integer); safecall;
function Get_TextAffixType: Integer; safecall;
procedure Set_TextAffixType(Value: Integer); safecall;
function Get_DateAffixType: Integer; safecall;
procedure Set_DateAffixType(Value: Integer); safecall;
function Get_MsgShowType: Integer; safecall;
procedure Set_MsgShowType(Value: Integer); safecall;
function Get_ShowMsgNo: Integer; safecall;
procedure Set_ShowMsgNo(Value: Integer); safecall;
function Get_ShowMsgsFrom: TDateTime; safecall;
procedure Set_ShowMsgsFrom(Value: TDateTime); safecall;
function Get_ShowMsgsTo: TDateTime; safecall;
procedure Set_ShowMsgsTo(Value: TDateTime); safecall;
function Get_EmailSubject: WideString; safecall;
procedure Set_EmailSubject(const Value: WideString); safecall;
function Get_TextAffix: WideString; safecall;
procedure Set_TextAffix(const Value: WideString); safecall;
function Get_DateAffix: WideString; safecall;
procedure Set_DateAffix(const Value: WideString); safecall;
function Get_ClearMsgOnSent: WordBool; safecall;
procedure Set_ClearMsgOnSent(Value: WordBool); safecall;
function Get_ClearContactsOnSent: WordBool; safecall;
procedure Set_ClearContactsOnSent(Value: WordBool); safecall;
function Get_AllowEmptyMsg: WordBool; safecall;
procedure Set_AllowEmptyMsg(Value: WordBool); safecall;
function Get_ShowSysContacts: WordBool; safecall;
procedure Set_ShowSysContacts(Value: WordBool); safecall;
function Get_ShowSysGroups: WordBool; safecall;
procedure Set_ShowSysGroups(Value: WordBool); safecall;
function Get_ShowSysTemplates: WordBool; safecall;
procedure Set_ShowSysTemplates(Value: WordBool); safecall;
function Get_ShowSysSchedules: WordBool; safecall;
procedure Set_ShowSysSchedules(Value: WordBool); safecall;
function Get_ShowSysReports: WordBool; safecall;
procedure Set_ShowSysReports(Value: WordBool); safecall;
function Get_DisplayContactsFirst: WordBool; safecall;
procedure Set_DisplayContactsFirst(Value: WordBool); safecall;
procedure SetMsgSettings(const accountId: WideString); safecall;
procedure Save; safecall;
function Get_LastMsgId: Integer; safecall;
procedure Set_LastMsgId(Value: Integer); safecall;
function Get_ContactDysplayType: Integer; safecall;
procedure Set_ContactDysplayType(Value: Integer); safecall;
function Get_MsgDeleteType: Integer; safecall;
procedure Set_MsgDeleteType(Value: Integer); safecall;
function Get_DeleteInterval: Integer; safecall;
```

```
procedure Set_DeleteInterval(Value: Integer); safecall;
function Get_UseSameMsg: WordBool; safecall;
procedure Set_UseSameMsg(Value: WordBool); safecall;
function Get_DuplicateContact: WordBool; safecall;
procedure Set_DuplicateContact(Value: WordBool); safecall;
function Get_ContactSearchType: Integer; safecall;
procedure Set_ContactSearchType(Value: Integer); safecall;
function Get_GroupSearchType: Integer; safecall;
procedure Set_GroupSearchType(Value: Integer); safecall;
function Get_TemplateSearchType: Integer; safecall;
procedure Set_TemplateSearchType(Value: Integer); safecall;
function Get_ContactSearchStr: WideString; safecall;
procedure Set_ContactSearchStr(const Value: WideString); safecall;
function Get_GroupSearchStr: WideString; safecall;
procedure Set_GroupSearchStr(const Value: WideString); safecall;
function Get_TemplateSearchStr: WideString; safecall;
procedure Set_TemplateSearchStr(const Value: WideString); safecall;
function Get_ShowSysFolders: WordBool; safecall;
procedure Set_ShowSysFolders(Value: WordBool); safecall;
function Get_AddObjectNamesToMsg: WordBool; safecall;
procedure Set_AddObjectNamesToMsg(Value: WordBool); safecall;
function Get_ObjectNamesDelimit: WideString; safecall;
procedure Set_ObjectNamesDelimit(const Value: WideString); safecall;
function Get_TvDisplayType: Integer; safecall;
procedure Set_TvDisplayType(Value: Integer); safecall;
function Get_FolderSearchType: Integer; safecall;
procedure Set_FolderSearchType(Value: Integer); safecall;
function Get_FolderSearchStr: WideString; safecall;
procedure Set_FolderSearchStr(const Value: WideString); safecall;
property MsgPriority: Integer read Get_MsgPriority write Set_MsgPriority;
property MsgAge: Integer read Get_MsgAge write Set_MsgAge;
property TextAffixType: Integer read Get_TextAffixType write
  Set_TextAffixType;
property DateAffixType: Integer read Get_DateAffixType write
  Set_DateAffixType;
property MsgShowType: Integer read Get_MsgShowType write Set_MsgShowType;
property ShowMsgNo: Integer read Get_ShowMsgNo write Set_ShowMsgNo;
property ShowMsgsFrom: TDateTime read Get_ShowMsgsFrom write
  Set_ShowMsgsFrom;
property ShowMsgsTo: TDateTime read Get_ShowMsgsTo write Set_ShowMsgsTo;
property EmailSubject: WideString read Get_EmailSubject write
  Set_EmailSubject;
property TextAffix: WideString read Get_TextAffix write Set_TextAffix;
property DateAffix: WideString read Get_DateAffix write Set_DateAffix;
property ClearMsgOnSent: WordBool read Get_ClearMsgOnSent write
  Set_ClearMsgOnSent;
property ClearContactsOnSent: WordBool read Get_ClearContactsOnSent write
  Set_ClearContactsOnSent;
```

```
    property AllowEmptyMsg: WordBool read Get_AllowEmptyMsg write
      Set_AllowEmptyMsg;
    property ShowSysContacts: WordBool read Get_ShowSysContacts write
      Set_ShowSysContacts;
    property ShowSysGroups: WordBool read Get_ShowSysGroups write
      Set_ShowSysGroups;
    property ShowSysTemplates: WordBool read Get_ShowSysTemplates write
      Set_ShowSysTemplates;
    property ShowSysSchedules: WordBool read Get_ShowSysSchedules write
      Set_ShowSysSchedules;
    property ShowSysReports: WordBool read Get_ShowSysReports write
      Set_ShowSysReports;
    property DisplayContactsFirst: WordBool read Get_DisplayContactsFirst
      write Set_DisplayContactsFirst;
    property LastMsgId: Integer read Get_LastMsgId write Set_LastMsgId;
    property ContactDysplayType: Integer read Get_ContactDysplayType write
      Set_ContactDysplayType;
    property MsgDeleteType: Integer read Get_MsgDeleteType write
      Set_MsgDeleteType;
    property DeleteInterval: Integer read Get_DeleteInterval write
      Set_DeleteInterval;
    property UseSameMsg: WordBool read Get_UseSameMsg write Set_UseSameMsg;
    property DuplicateContact: WordBool read Get_DuplicateContact write
      Set_DuplicateContact;
    property ContactSearchType: Integer read Get_ContactSearchType write
      Set_ContactSearchType;
    property GroupSearchType: Integer read Get_GroupSearchType write
      Set_GroupSearchType;
    property TemplateSearchType: Integer read Get_TemplateSearchType write
      Set_TemplateSearchType;
    property ContactSearchStr: WideString read Get_ContactSearchStr write
      Set_ContactSearchStr;
    property GroupSearchStr: WideString read Get_GroupSearchStr write
      Set_GroupSearchStr;
    property TemplateSearchStr: WideString read Get_TemplateSearchStr write
      Set_TemplateSearchStr;
    property ShowSysFolders: WordBool read Get_ShowSysFolders write
      Set_ShowSysFolders;
    property AddObjectNamesToMsg: WordBool read Get_AddObjectNamesToMsg write
      Set_AddObjectNamesToMsg;
    property ObjectNamesDelimit: WideString read Get_ObjectNamesDelimit write
      Set_ObjectNamesDelimit;
    property TvDisplayType: Integer read Get_TvDisplayType write
      Set_TvDisplayType;
    property FolderSearchType: Integer read Get_FolderSearchType write
      Set_FolderSearchType;
    property FolderSearchStr: WideString read Get_FolderSearchStr write
      Set_FolderSearchStr;
  end;
```

IMsgSettingsDisp

```
IMsgSettingsDisp = dispinterface
['{2B791897-E2B5-4795-8707-A36CF0582411}']
  property MsgPriority: Integer dispid 201;
  property MsgAge: Integer dispid 202;
  property TextAffixType: Integer dispid 203;
  property DateAffixType: Integer dispid 204;
  property MsgShowType: Integer dispid 205;
  property ShowMsgNo: Integer dispid 206;
  property ShowMsgsFrom: TDateTime dispid 207;
  property ShowMsgsTo: TDateTime dispid 208;
  property EmailSubject: WideString dispid 209;
  property TextAffix: WideString dispid 210;
  property DateAffix: WideString dispid 211;
  property ClearMsgOnSent: WordBool dispid 212;
  property ClearContactsOnSent: WordBool dispid 213;
  property AllowEmptyMsg: WordBool dispid 214;
  property ShowSysContacts: WordBool dispid 215;
  property ShowSysGroups: WordBool dispid 216;
  property ShowSysTemplates: WordBool dispid 217;
  property ShowSysSchedules: WordBool dispid 218;
  property ShowSysReports: WordBool dispid 219;
  property DisplayContactsFirst: WordBool dispid 220;
  procedure SetMsgSettings(const accountId: WideString); dispid 221;
  procedure Save; dispid 222;
  property LastMsgId: Integer dispid 223;
  property ContactDysplayType: Integer dispid 224;
  property MsgDeleteType: Integer dispid 225;
  property DeleteInterval: Integer dispid 226;
  property UseSameMsg: WordBool dispid 227;
  property DuplicateContact: WordBool dispid 228;
  property ContactSearchType: Integer dispid 229;
  property GroupSearchType: Integer dispid 230;
  property TemplateSearchType: Integer dispid 231;
  property ContactSearchStr: WideString dispid 232;
  property GroupSearchStr: WideString dispid 233;
  property TemplateSearchStr: WideString dispid 234;
  property ShowSysFolders: WordBool dispid 235;
  property AddObjectNamesToMsg: WordBool dispid 236;
  property ObjectNamesDelimit: WideString dispid 237;
  property TvDisplayType: Integer dispid 238;
  property FolderSearchType: Integer dispid 239;
  property FolderSearchStr: WideString dispid 240;
end;
```

IFolder

```
IFolder = interface(IDispatch)
```

```
    ['{2C53703A-31D4-4B18-9C83-3F5BBF405517}']
    function Get_FolderId: Integer; safecall;
    function Get_FolderName: WideString; safecall;
    procedure Set_FolderName(const Value: WideString); safecall;
    function Get_AccountOwner: WideString; safecall;
    procedure Set_AccountOwner(const Value: WideString); safecall;
    function Get_IsSystemObject: WordBool; safecall;
    function Get_Notes: WideString; safecall;
    procedure Set_Notes(const Value: WideString); safecall;
    function Get_FolderGroups: IStrings; safecall;
    function Get_FolderGroup(groupId: Integer): IGroup; safecall;
    function Get_FolderGroupByName(const groupName: WideString): IGroup;
      safecall;
    function Get_FolderContacts: IStrings; safecall;
    function Get_FolderContact(contactId: Integer): IContact; safecall;
    function Get_FolderContactByName(const contactName: WideString): IContact;
      safecall;
    procedure SetFolderGroups(const groupIds: IStrings); safecall;
    procedure SetFolderContacts(const contactIds: IStrings); safecall;
    procedure SetFolder(folderId: Integer); safecall;
    procedure Save; safecall;
    property FolderId: Integer read Get_FolderId;
    property FolderName: WideString read Get_FolderName write Set_FolderName;
    property AccountOwner: WideString read Get_AccountOwner write
      Set_AccountOwner;
    property IsSystemObject: WordBool read Get_IsSystemObject;
    property Notes: WideString read Get_Notes write Set_Notes;
    property FolderGroups: IStrings read Get_FolderGroups;
    property FolderGroup[groupId: Integer]: IGroup read Get_FolderGroup;
    property FolderGroupByName[const groupName: WideString]: IGroup read
      Get_FolderGroupByName;
    property FolderContacts: IStrings read Get_FolderContacts;
    property FolderContact[contactId: Integer]: IContact read
      Get_FolderContact;
    property FolderContactByName[const contactName: WideString]: IContact read
      Get_FolderContactByName;
  end;
```

IFolderDisp

```
  IFolderDisp = dispinterface
    ['{2C53703A-31D4-4B18-9C83-3F5BBF405517}']
    property FolderId: Integer readonly dispid 201;
    property FolderName: WideString dispid 202;
    property AccountOwner: WideString dispid 203;
    property IsSystemObject: WordBool readonly dispid 204;
    property Notes: WideString dispid 205;
    property FolderGroups: IStrings readonly dispid 206;
    property FolderGroup[groupId: Integer]: IGroup readonly dispid 207;
    property FolderGroupByName[const groupName: WideString]: IGroup readonly
      dispid 208;
    property FolderContacts: IStrings readonly dispid 209;
    property FolderContact[contactId: Integer]: IContact readonly dispid 210;
    property FolderContactByName[const contactName: WideString]: IContact
      readonly dispid 211;
    procedure SetFolderGroups(const groupIds: IStrings); dispid 212;
    procedure SetFolderContacts(const contactIds: IStrings); dispid 213;
    procedure SetFolder(folderId: Integer); dispid 214;
    procedure Save; dispid 215;
  end;
```

Events

```
IIPageServerComEvents = dispinterface
['{421EE7A2-7651-48C2-92BC-DE6774DACFF5}']
  procedure OnConnect(success: WordBool; connectionResult: Integer;const
    description: WideString; const accountId: WideString); dispid 201;
  procedure OnLogin(success: WordBool; loginResult: Integer; const
    description: WideString;const accountId: WideString); dispid 202;
  procedure OnIPMessage(transactionId: Integer; const accountId:
    WideString;const msgAddress: WideString; const resultStr:
    WideString;const msg: WideString; const contactName:
    WideString;sendResult: Integer; dateSent: TDateTime; dateReceived:
    TDateTime;addressType: Integer); dispid 203;
end;
```

CoClasses

Every CoClass provides a Create and CreateRemote method to create instances of the default interface exposed by the CoClass. The functions are intended to be used by clients wishing to automate the CoClass objects exposed by the server of this type library.

```
CoIPageServerCom = class
  class function Create: IIPageServerCom;
  class function CreateRemote(const MachineName: string): IIPageServerCom;
end;

CoAccount = class
  class function Create: IAccount;
  class function CreateRemote(const MachineName: string): IAccount;
end;

CoCarrier = class
```

```
  class function Create: ICarrier;
  class function CreateRemote(const MachineName: string): ICarrier;
end;

CoContact = class
  class function Create: IContact;
  class function CreateRemote(const MachineName: string): IContact;
end;

CoGroup = class
  class function Create: IGroup;
  class function CreateRemote(const MachineName: string): IGroup;
end;

CoTemplate = class
  class function Create: ITemplate;
  class function CreateRemote(const MachineName: string): ITemplate;
end;

CoSchedule = class
  class function Create: ISchedule;
  class function CreateRemote(const MachineName: string): ISchedule;
end;

CoReport = class
  class function Create: IReport;
  class function CreateRemote(const MachineName: string): IReport;
end;

CoMsgSettings = class
  class function Create: IMsgSettings;
  class function CreateRemote(const MachineName: string): IMsgSettings;
end;

CoFolder = class
  class function Create: IFolder;
  class function CreateRemote(const MachineName: string): IFolder;
end;
```