



BillPlex™

Application Program Interface (API) Reference Manual

Release 2.2
June 1999

Billing And Customer Care Solutions For The 21st Century



ã Copyright Daleen Technologies, Inc. 1999 – The reproduction, transmission of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, created by patent, grant, or registration of a utility model or design are reserved.

BillPlex™ is a trademark of Daleen Technologies, Inc. MS Windows® is a registered trademark of Microsoft Corporation. UNIX® is a registered trademark of X/Open Company, Limited. Crystal Reports® is a registered trademark of Seagate Software Inc. Oracle™ is a registered trademark of Oracle Corporation. Solaris™ is a registered trademark of Sun Microsystems, Inc., COMMTAX™ is a registered trademark of Vertex, Inc. MetaFrame™ is a registered trademark of Citrix Systems Inc. Other brands and their products are trademarks of registered trademarks and should be noted as such.

Contents

Document Conventions	2
Return Codes.....	5
Property Access Methods.....	10
IDTA Agent.....	11
IDTAConnectionNumber.....	15
IDTAConnectionNumberType	17
IDTAConnectionType.....	19
IDTAFeature.....	21
IDTAFeatureType.....	26
IDTAInvoice.....	29
IDTAMarketType	31
IDTAParameter	33
IDTAParameterType	36
IDTARatePlan.....	38
IDTAService.....	41
IDTAServiceBundle	46
IDTAServiceConnection	48
IDTAServiceOrder.....	51
IDTAServiceType.....	55
IDTPSessionMgr	58
IDTASubscriber.....	60
IDTASubscriberComment.....	77
IDTATransaction.....	79
Appendix A. Return Codes and Error Messages.....	81
Appendix B. Footnotes	87
Appendix C. Limitations.....	89

About the Application Program Interface (API)

1

API Overview The BillPlex Core API allows third party applications to communicate with BillPlex freeing the application programmer from directly dealing with the intricacies of the database.

Purpose of the API The API allows external systems to retrieve and update information in the BillPlex database and to perform certain BillPlex functions. The goal of this API is to make BillPlex more interoperable with other systems in the customer's enterprise and to make the data and functions of BillPlex more accessible from other systems in the customer's enterprise.

Uses of the API You can use the API to:

- Support product integration with other systems in an enterprise environment
- Support third-party client user interface
- Support Internet access to BillPlex, including bill viewing, payment, and service subscription

Document Conventions

About this Document

The BillPlex API Reference Manual lists all the business data objects (BDO) defined in the BillPlex API. The business data objects are listed in alphabetical order.

Common definitions, such as return code values and their descriptions that are used by multiple API methods, are listed together in Appendix A.

Footnotes are denoted by asterisked numbers and appear in Appendix B.

Audience

System integrators, developers, and customers who use the BillPlex API to integrate BillPlex with other software systems can use this document as a reference.

Acronym Listing

Below is a table listing the terminology and acronyms used consistently throughout this document.

Acronym	Definition
API	Application Programming Interface
ATL	Active Template Library
BDR	BillPlex Data Repository
BDO	Business Data Object
BPO	Business Process Object
BRO	Business Rule Object
COM	Component Object Model
DCOM	Distributed Component Object Model
IDL	Interface Definition Language
MTS	Microsoft Transaction Server

Interface Definition Language

All API methods are defined in Interface Definition Language (IDL). For more information about IDLs, refer to:

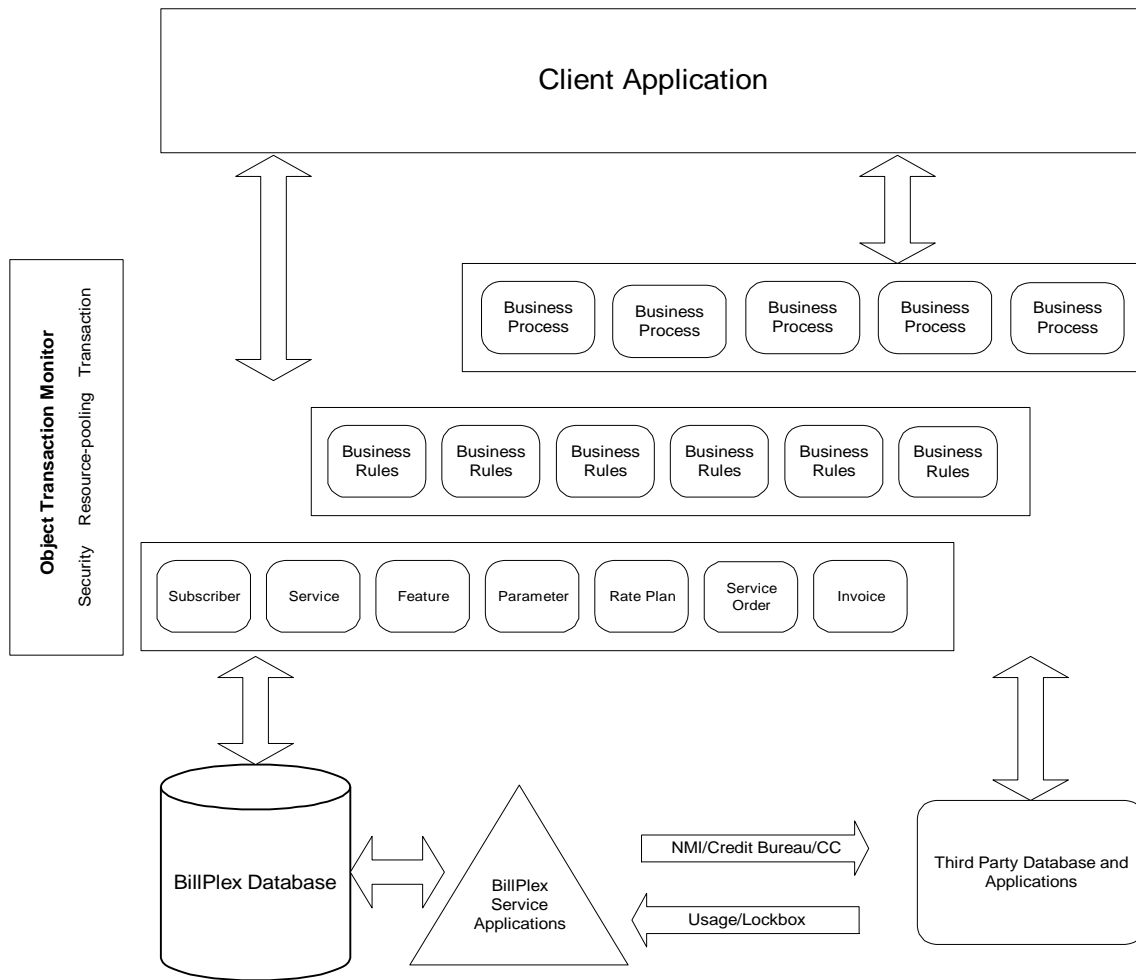
http://premium.microsoft.com/msdn/library/sdkdoc/com/custintf_1lr9.htm, and related pages on the Microsoft web site.

Business Data Objects (BDO)

2

Overview

BillPlex defines a set of business data objects to allow other applications to access and manipulate its data while encapsulating its database structure. These business data objects are only part of the overall BillPlex API. BillPlex business rule objects (BRO) and business process objects (BPO) complement the business data objects. See the diagram below.



**About
Business
Data Objects**

The business data objects include entities such as subscriber, service, feature, parameter, rate plan, service order, and invoice. These objects are mapped to BillPlex database records. Methods are defined for these business data objects allowing other applications to access and manipulate these data records.

With the objects and their defined methods, a third-party application can:

- Create a subscriber
 - Modify an existing subscriber
 - Provision services to a subscriber
 - Retrieve invoice information for a subscriber
 - Retrieve service order status for a subscriber
-

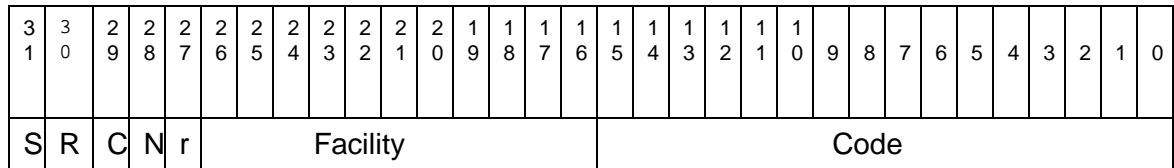
**The API
Environment**

BillPlex API is defined and implemented as a set of Microsoft COM (Component Object Model) interfaces. The API server runs on Microsoft Windows and Microsoft NT. You can invoke the API from any system where COM API can be invoked directly or through a gateway.

Return Codes

HRESULT Codes

BillPlex uses the HRESULT convention for COM components as defined by Microsoft. HRESULT codes are used to convey the result of an interface call in COM and OLE. They are 32 bit values with the following layout:



S	Severity – indicates success/failure 0 = Success 1 = Failure (COERROR)
R	Reserved portion of the facility code, corresponds to NT's second severity bit.
C	Reserved portion of the facility code, corresponds to NT's C field.
N	Reserved portion of the facility code. Used to indicate a mapped NT status value.
r	Reserved portion of the facility code. Reserved for internal use. Used to indicate HRESULT values that are not status values, but are instead message ids for display strings.
Facility	Represents the facility code.
Code	Represents the facility's status code.

Facility Code

The BillPlex Core API uses FACILITY_ITF, which is the predefined facility code for return codes specific to the interface that is returning the code. FACILITY_ITF should be used or all errors that occur in an interface, as opposed to some part of the Windows operating system. FACILITY_ITF has the numerical value of 4.

In addition to the errors occurring in and reported by the BillPlex API, other errors may occur. These errors include memory, network, and file system errors that all have their own error codes. The type of error may be determined both by the facility code and by the range in which the error code falls.

NOTE: the success codes do not have the Facility code set.

The following error number ranges are reserved:

0x#####0000-0x#####01FF	Windows
0x00000200-0x000002FF	BillPlex Core API success codes
0x#####0300-0x#####3FFF	BillPlex Core API failure codes
0x#####4000-0x#####40FF	Windows COM/Automation errors

See Appendix A for a complete listing of BillPlex result codes.

Evaluating HRESULT Codes

HRESULT codes convey errors and distinguish between several possible outcomes of a function, where different actions are taken although no actual error occurred. We are reserving 0x0200-0x02FF for BillPlex Core API success codes. For example:

```
#define S_NOCHANGE 0x0200 // Function succeeded. No data
has changed.
```

```
#define S_CHANGE 0x0201 // Function succeeded. Data has
changed.
```

Because of the way HRESULT codes are structured, it is important that you use the predefined macros (from Winerror.h) for evaluating HRESULTS; for example:

```
if (SUCCEEDED(hr)){...}
```

-or-

```
if (FAILED(hr)){...}
```

-or-

```
if (IS_ERROR(hr)){...}
```

If you use a language other than C/C++, you will have to use similar functions to parse the HRESULT. Below is an example of how to evaluate different successful outcomes:

```
if (SUCCEEDED(hr))
{
    switch (HRESULT_CODE(hr))
    {
        case S_NOCHANGE:
            // do something
            break;
        case S_CHANGE:
            // do something else
            break;
    }
}
```

**Standard
Macro
Definitions
from
Winerror.h**

Generic test for success on any status value (non-negative numbers indicate success):

```
#define SUCCEEDED(Status) ((HRESULT)(Status) >= 0)
```

- and the inverse:

```
#define FAILED(Status) ((HRESULT)(Status)<0)
```

Generic test for error on any status value.

```
#define IS_ERROR(Status) ((unsigned long)(Status) >> 31 == \
SEVERITY_ERROR)
```

The following macro returns the result/error code part of the HRESULT:

```
#define HRESULT_CODE(hr) ((hr) & 0xFFFF)
```

To extract the facility code:

```
#define HRESULT_FACILITY(hr) (((hr) >> 16) & 0x1fff)
```

ERRORInfo

In addition to the HRESULT code, the BillPlex Core API uses the standard mechanism in COM/DCOM for providing rich error information to a client: the ErrorInfo object. "Rich" in this context means that it not only provides an error number and a description of the error, but also the source of it. In later versions it will also provide a help file name and a help context number for additional information about how to deal with each error. Each thread has one and only one ErrorInfo object.

The error object (or component) implements the following two interfaces: "ICreateErrorInfo" to set values and "IErrorInfo" to retrieve them.

```
interface ICreateErrorInfo: IUnknown
{
    HRESULT SetGUID([in] REFGUID rguid);
    HRESULT SetSource([in] LPOLESTR szSource);
    HRESULT SetDescription([in] LPOLESTR szDescription);
    HRESULT SetHelpFile([in] LPOLESTR szHelpFile);
    HRESULT SetHelpContext([in] DWORD dwHelpContext);
}

interface IErrorInfo: IUnknown
{
    HRESULT GetGUID([out] GUID* pGUID);
    HRESULT GetSource([out] BSTR* pBstrSource);
    HRESULT GetDescription([out] BSTR* pBstrDescription);
    HRESULT GetHelpFile([out] BSTR* pBstrHelpFile);
    HRESULT GetHelpContext([out] DWORD* pdwHelpContext);
}
```

**Syntax for
Retrieving
Error
Information in
Visual Basic**

On the client side, processing `ErrorInfo` is quite straightforward. Since the object is already created, it's just a question of using the supplied methods to collect the error information. In Visual Basic (VB) it is even simpler because all of the information is automatically stuffed into the `Err` object. This mechanism is also supported in VBScript.

The following example shows how to retrieve error information in Visual Basic:

```
On Error GoTo ErrorHandler

[... code ...]

ErrorHandler:

Dim ErrDesc

ErrDesc = Err.Source + " " + Hex$(Err.Number) + ": " +
Err.Description

MsgBox ErrDesc, vbOKOnly, Err.Source

Err.Clear                ' Clear Err object fields

End Sub
```

Property Access Methods

Overview

Each data object has one or more properties. These properties hold information that is available through the interface. They are accessible through two methods, `put` and `get`. The actual invocation syntax depends on the programming language you use.

Populate the property `SessionID` before executing any methods in any data object. An application creates a session by executing `beginSession` method of the `SessionMgr`, which returns a session id. Use this id to populate the `SessionID` property of each object invoked. The application should end with closing the session; for example, executing the `endSession` method of the `SessionMgr`.

C++ Syntax

When calling these methods using C++, the calling application must invoke the following methods to retrieve and set the value for a property `xxxx` respectively:

```
pInterface->get_xxxx(&variable);  
pInterface->put_xxxx(variable);
```

where `pInterface` is the interface pointer to a business object.

Visual Basic Syntax

Note that BillPlex API supports the dual interface as defined by COM. As such, it supports a variety of programming languages: Visual Basic, Java, VBScript, etc.

When calling the property access methods using Visual Basic, the calling application must invoke the following methods to retrieve and set the value for a property `xxxx` respectively:

```
variable = obj.xxxx  
obj.xxxx = variable
```

where `obj` is declared as a business data object.

IDTA Agent

Description In BillPlex, agents are used to capture an entity that has special access to the system or a special relationship with other subscribers. For example, an agent can be a sales agent who brings in new subscribers or sells new services to existing subscribers.

Properties The following table contains the properties for the IDTA Agent.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Address1	BSTR	First line of agent's address.		60	Y	Y
Address2	BSTR	Second line of agent's address.		60	Y	Y
AgentCompany	BSTR	Company for which agent works.	<i>Examples:</i> Radio Shack, Sears	30	Y	Y
AgentID	Long	Unique identifier for agent. Generated by BillPlex when agent is created.	1,2,3		Y	Y
AgentReference	BSTR	Reference for agent (password)		30	N	Y
AgentReferenceID	BSTR	Agent password		20	N	Y
AgentType	BSTR	Type of agent	<i>Examples:</i> DLR = Dealer, INS = Installer, RSL = Reseller, OUL = Outlet.	3	Y	Y
Channel	BSTR	Distribution channel for agent.		30	Y	Y
City	BSTR	City part of Agent's address.	<i>Examples:</i> Boca Raton Palm Beach *1	50	Y	Y
Company	BSTR	Company through which agent collects payment from subscriber.				
Country	BSTR	Name of country where agent does business or resides.	<i>Examples:</i> U.S.A. France *2	50	Y	Y
Email	BSTR	E-mail address of agent's business.	*3	40	Y	Y
Emp	BSTR	Employee, true or false: T=employee, F=non- employee.	T, F *4	1	Y	Y

*See table footnotes in Appendix B.

IDTAAgent – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
EndDate	Date	Date from which agent stopped its activities.	mm-dd-yyyy *5		Y	Y
FaxAreaCode	BSTR	Area code of agent's business fax.	xxxxxx	6	Y	Y
FaxCountryCode	BSTR	Country code of agent's business fax.	xxxxxx *3	6		
FaxNumber	BSTR	Fax number of agent's business.	xxxxxxxxxx *3	10	Y	Y
FirstName	BSTR	First name of agent.		25	Y	Y
LastName	BSTR	Last name of agent.		35	Y	Y
MiddleInitial	BSTR	Middle initial of agent.		1	Y	Y
PhoneAreaCode	BSTR	Area code of agent's business.	*3	6	Y	Y
PhoneCountryCode	BSTR	Country code of agent's business.	*3	6	Y	Y
PhoneExtension	BSTR	Extension of agent's business	*3	5	Y	Y
PhoneNumber	BSTR	Telephone number of agent's business	*3	10	Y	Y
PostalCode	BSTR	Postal code of agent's business or residence (5 or 9 digit Zip Code or International formatted Postal Code).	*3	15	Y	Y
SessionID	Long	Unique identifier of database connection, created by the SessionMgr.	*8		Y	Y
StartDate	Date	Date from which the agent started its activities	mm-dd-yyyy *5		Y	Y
State	BSTR	State in which the agent's business is located.	<i>Examples:</i> Florida, Colorado *6	50	Y	Y
Title	BSTR	Title of the agent.	Mr. Mrs. Dr. *4	30	Y	Y

* See table footnotes in Appendix B.

Methods

The IDTAAgent methods are:

- authenticate
- changePassword
- getAll
- getDetails

The following tables provide detailed information about each of the IDTAAgent methods.

IDTAAgent – Methods (Cont.)

authenticate	
Synopsis	HRESULT authenticate ([in] BSTR password, [out] long * result)
Description	Validate if the caller has the correct password of the agent.
Prerequisite	Set AgentID of the agent.
Parameters	password The agent's password.
	result The output status. 1 – Wrong password 0 – Correct password
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

changePassword	
Synopsis	HRESULT changePassword ([in] BSTR oldPassword, [in] BSTR newPassword, [out] long* result)
Description	Changes the agent's password if the agent is authorized with the old password.
Prerequisite	Set AgentID of the agent.
Parameters	oldPassword The agent's old password.
	newPassword The agent's new password.
	result The output status. 1 – Wrong password 0 – Correct password
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAAgent – Methods (Cont.)

getAll	
Synopsis	HRESULT getAll([out] BSTR* pAgntTyp)
Description	Retrieves the list of existing agents in the BillPlex database.
Prerequisite	None.
Parameters	pAgntTyp Agent types in comma delimited format. Example: "1 = 001, 2 = 002"
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_All([out, retval] VARIANT* pVal)
See Also	None.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves values of all the Agent attributes from the database.
Prerequisite	Set AgentID of the agent.
Parameters	None
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAConnectionNumber

Description The IDTAConnection Number represents the number of possible connection numbers. In traditional wireline telephony, the telephone number is the identifier to correlate usage to a subscriber. BillPlex takes a more generic approach. BillPlex has a concept of connection number which correlates the usage of a service to a subscriber.

Properties The following table contains the properties for IDTAConnectionNumber.

Property	Data Type	Description	Allowable Values	Max	Get	Put
ConnectionNumber	BSTR	Number that represents a link to a subscriber's actual service. For example, the connection number for a telephone would be the telephone number.	Account N.23400, 456789, 00100000001, 2001110002 *1	30	Y	Y
ConnectionNumberStatus	BSTR	Status of ConnectionNumber, Stored as a single character code that can be displayed as such. For example, "A" means "Assigned".	Assigned, Inactive, Permanently Retired, Retired, Unassigned *4	1	Y	Y
ConnectionNumberType	Long	Unique identification of ConnectionNumberType to which this ConnectionNumber is associated.	1, 2, 3, 21		Y	Y
RetiredDateTime	BSTR	Date and time from which the number started the aging process	Mm-dd-yyyy hh:mm:ss *5		Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y

* See table footnotes in Appendix B.

Method

The method for IDTA ConnectionNumber is:

- `getDetails`

The following table provides detailed information about the IDTAConnectionNumber method.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves values of all ConnectionNumber attributes from database.
Prerequisite	Set ConnectionNumber attribute of the Connection Number.
Parameters	None
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAConnectionNumberType

Description The IDTAConnectionNumberType is used to specify what syntax a connection number should have. This is similar to data types in programming languages.

Properties The following table contains the properties for IDTAConnectionNumberType.

Property	Data Type	Description	Allowable Values	Max	Get	Put
ConnectionNumberName	BSTR	Name of connection number type	POTS, ESN, MIN Daleen PBX. *1	50	Y	Y
ConnectionNumberType	Long	Unique system generated id of ConnectionNumberType.	Integer *7		Y	Y
RetiredDuration	Long	Default amount of time that a connection number of this type will be retired before returning to the status of "unassigned" making it available for use.	Integer		Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y

**See table footnotes in Appendix B.*

Methods The IDTAConnectionNumberType Methods are:

- getConnectionNumberList
- getDetails
- change (not yet implemented)
- create (not yet implemented)

The following tables provide detailed information about IDTAConnectionNumberType methods.

IDTAConnectionNumberType – Methods (Cont.)

getConnectionNumberList	
Synopsis	HRESULT getConnectionNumberList ([out] SAFEARRAY(IDispatch*) *ConnectionNumberList)
Description	Retrieves the list of ConnectionNumbers associated with the ConnectionNumberType.
Prerequisite	Set ConnectionNumberType attribute of the Connection NumberType.
Parameters	ConnectionNumberList List of ConnectionNumbers as object.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript. HRESULT getVar_ConnectionNumberList ([out, retval] VARIANT* pVal)
See Also	None.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of a ConnectionNumberType from the database.
Prerequisite	Set ConnectionNumber attribute of the Connection NumberType.
Parameters	None
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAConnectionType

Description IDTAConnectionType represents a type of connection. A given service may have multiple connection numbers. To distinguish the purpose of each connection number, BillPlex uses connection type. Each connection number has a connection type.

Properties The following table contains the properties for IDTAConnectionType.

Property	Data Type	Description	Allowable Values	Max	Get	Put
ConnectionName	BSTR	The name of the ConnectionType.	ESN, LIN, HOST_ID *1	50	Y	Y
ConnectionType	Long	Unique system generated identifier for the connection type.	Integer *7		Y	Y
SessionID	Long	Unique identifier for the database connection, created by the SessionMgr.	*8		Y	Y

*See table footnotes in Appendix B.

Methods The IDTAConnectionType methods are:

- change
- create
- getConnNumTypList
- getDetails

The following tables provide detailed information about IDTAConnectionType methods.

change	
Synopsis	HRESULT change ()
Description	Physically changes in BillPlex data repository any changes on the connection type properties. Execute the getDetails method prior to the “change” method execution. If you disregard this procedure, the BDR will only be updated with the attributes populated at the time of execution and uninitialized values will delete corresponding database values.
Prerequisite	Set ConnectionType attribute of the ConnectionType.
Parameters	None
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAConnectionType – Methods (Cont.)

create	
Synopsis	HRESULT create (Idispatch* serviceID)
Description	Physically creates a feature in the BDR for the specified service. Upon successful completion of this method, the ConnectionType will be populated.
Prerequisite	None.
Parameters	serviceID The ID of the service for which the feature will be created.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	Service, ServiceType, FeatureType

getConnNumTypList	
Synopsis	HRESULT getConnNumTypList([out] SAFEARRAY(IDispatch*) *ConnNumTypList)
Description	Retrieves the list of ConnectionNumberTypes associated with the ConnectionType.
Prerequisite	Set ConnectionType attribute of the ConnectionType.
Parameters	AConnNumTypList List of ConnectionNumberTypes.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ConnNumTypList([out, retval] VARIANT* pVal)
See Also	None.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of a ConnectionType from the database.
Prerequisite	Set ConnectionType attribute of the ConnectionType.
Parameters	None
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAFeature

Description The IDTAFeature represents a feature of a service belonging to a subscriber; for example, a Wake Up Call feature of a Pager service.

Properties The following table contains the properties for IDTAFeature.

Property	Data Type	Description	Allowable Values	Max	Get	Put
So_ReqEffectiveDttm	BSTR	Date feature is effective.	mm-dd-yyyy hh:mm:ss *5			
EndDateTime	BSTR	Date from which feature stopped being active.	mm-dd-yyyy hh:mm:ss *5		Y	Y
FeatureID	Long	Unique system generated feature id.	Integer *7		Y	Y
FeatureSoFlag	Long				Y	Y
FeatureType	Long	See FeatureType component for more details.	Integer		Y	Y
LastInvoiceDateTime	BSTR	Date and time of last Invoice.	mm-dd-yyyy hh:mm:ss *5		Y	Y
OrderDateTime	BSTR	Date from which feature was ordered.	mm-dd-yyyy hh:mm:ss *5		Y	Y
SessionID	Long	ID for the database connection, created by the SessionMgr.	*8		Y	Y
So_Provisioning Dttm	BSTR	Date and time of provisioning.			Y	Y
So_SoActionClass	BSTR	Service order action class.			Y	Y
So_Status	BSTR	Service order status.			Y	Y
So_SoID	Long	ID of service order generated to activate feature.			Y	Y
StartDateTime	BSTR	Date from which feature becomes active.	Mm-dd-yyyy hh:mm:ss *5		Y	Y
Status	BSTR	Feature status, stored as a single character code. For example, A means Active.	Active, Deactivate *4	1	Y	Y
SusEndTime	BSTR	End date and time of suspension.			Y	Y
SusStartTime	BSTR	Start date and time of suspension.			Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAFeature methods are:

- createParameter
- createRatePlan
- change
- create
- getDetails
- getParameterList
- getRatePlanList
- AddFeatureToSo
- AddParmInfoToSo
- getServiceOrderDetails

The following tables provide detailed information about of the IDTAFeature methods.

createParameter			
Synopsis	<code>HRESULT createParameter (Idispatch** pParameter)</code>		
Description	Adds a parameter to the SAFEARRAY that contains all other parameters of the feature. All the attributes of the added parameter component are initialized to zeros and NULLs.		
Prerequisite	Instantiation of the feature component.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;"><code>PParameter</code></td> <td>The reference to the parameter component that is added to the SAFEARRAY.</td> </tr> </table>	<code>PParameter</code>	The reference to the parameter component that is added to the SAFEARRAY.
<code>PParameter</code>	The reference to the parameter component that is added to the SAFEARRAY.		
Return Value	See common return codes in Appendix A.		
Notes	<p>This component also has the following method to retrieve the result as Variant to support VBScript:</p> <pre>HRESULT createVar_Parameter ([out, retval] VARIANT* pParameter)</pre>		
See Also	None.		

IDTAFeature – Methods (Cont.)

createRatePlan			
Synopsis	HRESULT createRatePlan (Idispatch** pRatePlan)		
Description	Adds a RatePlan to the SAFEARRAY that contains all other RatePlans of the Feature. One Feature has only one RatePlan at any given time. All of the attributes of the added rateplan component are initialized to zeros and NULLs.		
Prerequisite	Instantiation of the feature component.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">PratePlan</td> <td>The reference to the RatePlan that is added to the SAFEARRAY.</td> </tr> </table>	PratePlan	The reference to the RatePlan that is added to the SAFEARRAY.
PratePlan	The reference to the RatePlan that is added to the SAFEARRAY.		
Return Value	See common return codes in Appendix A.		
Notes	<p>This component also has the following method to retrieve the result as Variant to support VBScript:</p> <p>HRESULT createVar_RatePlan ([out, retval] VARIANT* pRatePlan)</p>		
See Also	Feature Type, ParameterType, RatePlan		

change	
Synopsis	HRESULT change ()
Description	Physically changes BillPlex data repository any changes on the Feature properties, including lists. Execute the getDetails method prior to the "change" method execution. If you disregard this procedure, the BDR will only be updated with the attributes populated at the time of execution and uninitialized values will delete corresponding database values.
Prerequisite	Set FeatureID of the Feature.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

create			
Synopsis	HRESULT create (long serviceID)		
Description	Physically creates a feature in the BDR for the specified service. Upon successful completion of this method, the FeatureID is populated.		
Prerequisite	Set FeatureID of the Feature.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">ServiceID</td> <td>The ID of the service for which the feature will be created.</td> </tr> </table>	ServiceID	The ID of the service for which the feature will be created.
ServiceID	The ID of the service for which the feature will be created.		
Return Value	See common return codes in Appendix A.		
Notes	None.		
See Also	Service, ServiceType, FeatureType		

IDTAFeature – Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the feature from the database.
Prerequisite	Set FeatureID of the Feature.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getParameterList	
Synopsis	HRESULT getParameterList([out] SAFEARRAY(IDispatch*) *ParameterList)
Description	Retrieves the list of Parameters associated with the Feature.
Prerequisite	Set FeatureID of the Feature.
Parameters	ParameterList List of Parameters.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ParameterList([out, retval] VARIANT* pVal)
See Also	None.

getRatePlanList	
Synopsis	HRESULT getRatePlanList([out] SAFEARRAY(IDispatch*) *RatePlanList)
Description	Retrieves the historical view of RatePlans associated with the Feature. One Feature has only one RatePlan at any given time.
Prerequisite	Set FeatureID of the Feature.
Parameters	RatePlanList List of RatePlans.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_RatePlanList([out, retval] VARIANT* pVal)
See Also	None.

IDTAFeature – Methods (Cont.)

AddFeatureToSo	
Synopsis	HRESULT AddFeatureToSo(long serviceID)
Description	Physically creates a new service order of this new or changed Feature in the BillPlex data repository.
Prerequisite	Set FeatureID of the Feature.
Parameters	serviceID The service ID to which this Feature belongs.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

AddParmInfoToSo	
Synopsis	HRESULT AddParmInfoToSo(long serviceID)
Description	Physically creates a new service-feature-parameter information in the BillPlex data repository.
Prerequisite	Set serviceID to which this Feature belongs.
Parameters	serviceID The service ID to which this Feature belongs.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

getServiceOrderDetails	
Synopsis	HRESULT getServiceDetails([in]long svcId)
Description	Retrieves the attribute values of the Service Order of this feature object from the database.
Prerequisite	Set FeatureID and ServiceOrderID of the feature object.
Parameters	long svcId The service ID of the service associated with this feature.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAFeatureType

Description The IDTAFeatureType represents a characteristic provided by the operator that can be rated or provisioned.

Properties The following table contains the properties for the IDTAFeatureType component interface.

Property	Data Type	Description	Allowable Values	Max	Get	Put
AbstractFunctionID	Long	The id of the function that serves as the mapping between BillPlex service/feature and VNED functions.	*9		Y	Y
BillStartDateOvrd	BSTR	Bill Start Date Override.			Y	Y
BillStartInterval	Long	Bill Start interval.			Y	Y
EndDateTime	BSTR	Date from which feature type can no longer be used in the system.	mm-dd-yyyy hh:mm:ss *5		Y	Y
FeatureName	BSTR	Name of feature type	<i>Examples:</i> Call forwarding, Call waiting, Caller Identification	50	Y	Y
FeatureType	Long	Feature type key.	Integer		Y	Y
FeatureTypeDescription	BSTR	Brief description of the feature type.		255	Y	Y
RatePlanRequired	BSTR		T,F *9	1	Y	Y
Required	BSTR	Indicates whether or not the feature type is required to be assigned to a specific service type.	T,F *9	1	Y	Y
ServiceTypeChangedBY	BSTR	Application that modified the ServiceType to which the FeatureType is associated.		30	Y	Y
ServiceTypeChangedDate	Date	Date and time the ServiceType to which the FeatureType is associated with is modified.	mm-dd-yyyy *5		Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y
StartDateTime	BSTR	Date from which the feature type can be used by the system.	mm-dd-yyyy hh:mm:ss *5		Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAFeatureType methods include:

- getDetails
- getParameterTypeList
- getRatePlanTypeList
- getServiceFeatureTypeJoinDetails
- create (not yet implemented)
- change (not yet implemented)

The following tables provide detailed information about each of the IDTAFeatureType methods.

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the feature type from the database.
Prerequisite	Set FeatureType attribute of the Feature Type.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getParameterTypeList	
Synopsis	HRESULT getParameterTypeList([out] SAFEARRAY(IDispatch*) *ParameterTypeList)
Description	Retrieves the list of ParameterTypes associated with the Feature Type.
Prerequisite	Set FeatureType attribute of the Feature Type.
Parameters	ParameterTypeList List of ParameterTypes.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ParameterTypeList([out, retval] VARIANT* pVal)
See Also	None.

IDTAFeatureType – Methods (Cont.)

getRatePlanTypeList	
Synopsis	HRESULT getRatePlanTypeList([out, retval] BSTR *pVal)
Description	Retrieves the list of RatePlanTypes associated with the Feature Type.
Prerequisite	Set FeatureType attribute of the Feature Type.
Parameters	Pval List of RatePlanTypes as a string.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_RatePlanTypeList([out, retval] VARIANT* pVal)
See Also	None.

getServiceFeatureTypeJoinDetails	
Synopsis	HRESULT getServiceFeatureTypeJoinDetails(long aServiceType)
Description	Retrieves the information of a FeatureType associated with a service type, which is passed in. This function helps the service type component get a list of associated Feature Types.
Prerequisite	Need a service type passed
Parameters	aServiceType The value of a service type.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

IDTAINVOICE

Description The IDTAINVOICE represents subscriber's bill.

Properties The following table contains the properties for IDTAINVOICE.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Adjustment	Long	Monetary amount of adjustments since last invoice billing run.		14,2	Y	Y
DueDate	BSTR	Date the invoice is due.	mm-dd-yyyy *5		Y	Y
EndDate	BSTR	Ending date of transactions included in the invoice.	mm-dd-yyyy *5		Y	Y
ExchangeRateUnitDollar	Long	Exchange rate in unit dollars.		14,7	Y	Y
InvoiceAmount	Long	Current invoice amount.		14,2	Y	Y
InvoiceDateTime	BSTR	Date and time of invoice generation.	mm-dd-yyyy hh:mm:ss *5		Y	Y
InvoiceID	Long	The system generated unique ID of the invoice.		10	Y	Y
InvoiceLocation	BSTR	The location of invoice.			Y	Y
InvoiceStatus	BSTR	Indicates the completion status of an Invoice, stored as a single character code that can be displayed as such. For example, "A" means "Assigned".	Ready, Pending, Canceled, Invoiced	1	Y	Y
PaymentReceived	Long	Last payment received.		14,2	Y	Y
PreviousBalance	Long	The subscriber's previous balance prior to the date that the invoice was generated.		14,2	Y	Y
ReferenceNumber	BSTR	A system assigned number that references an invoice.		30		
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y
StartDate	BSTR	The starting date of transactions included in the invoice.	mm-dd-yyyy *5		Y	Y
SubscriberSequence	Long					
Terms	BSTR	Indicates payment grace period.		25	Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAInvoice methods are:

- getDetails
- create (not yet implemented)
- change (not yet implemented)
- getInvLoc (not yet implemented)

The following table contains detailed information about the IDTAInvoice method.

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the invoice from the database.
Prerequisite	Set InvoiceID of the Invoice.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAMarketType

Description The IDTAMarketType represents a group of subscribers that have common term of taxes, treatments, and services needs.

Properties The following table contains the properties for IDTAMarketType.

Property	Data Type	Description	Allowable Values	Max	Get	Put
MarketName	BSTR	The name of market type	Business, Residence, Government *1	50	Y	Y
MarketType	Long	Market type key.	Integer *7		Y	Y
SessionID	Long	Unique identifier for the database connection, created by the SessionMgr.	*8		Y	Y

*See table footnotes in Appendix B.

Methods The IDTAMarketType methods are:

- getAll
- getDetails
- getServiceTypeList

The following tables provide detailed information about each of the IDTAMarketType methods.

getAll	
Synopsis	HRESULT getAll ([out] BSTR* pMarketTyp)
Description	Retrieves the existing MarketType objects in the BillPlex database.
Prerequisite	None.
Parameters	PMarketTyp List of market types as string in comma delimited format.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_All([out, retval] VARIANT* pVal)
See Also	None.

IDTAMarketType – Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the market type from the database.
Prerequisite	Set the MarketType attribute of the MarketType.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getServiceTypeList	
Synopsis	HRESULT getServiceTypeList ([out] SAFEARRAY (IDispatch*) *ServiceTypeList)
Description	Retrieves the list of Service Type objects that are linked to it.
Prerequisite	Set the MarketType attribute of the MarketType.
Parameters	ServiceTypeList List of Service Types
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceTypeList([out, retval] VARIANT* pVal)
See Also	None.

IDTAParameter

Description An IDTAParameter is a modifier on a feature that appears on the Service Tree; for example, a Pager service with a Wake Up Call feature may include a parameter setting that indicates the Time of Wake Up Call.

Properties The following table contains the properties for IDTAParameter.

Property	Data Type	Description	Allowable Values	Max	Get	Put
So_ReqEffectiveDttm	BSTR		mm-dd-yyyy hh:mm:ss *5		Y	Y
EndDateTime	BSTR	Date from which parameter starts being inactive.	mm-dd-yyyy hh:mm:ss *5		Y	Y
ParameterID	BSTR	Unique identifier of a parameter component.	Integer *7	10	Y	Y
ParameterSoFlag	Long				Y	Y
ParameterType	Long	Name of parameter type.			Y	Y
ParameterValue	BSTR	Value of the parameter.	800000001, 800000002	30	Y	Y
SessionID	Long	Unique identifier of database connection, created by the SessionMgr.	*8		Y	Y
So_SoID	Long	Unique identifier of the service order generated to activate the feature.			Y	Y
So_NewValue	BSTR	New value.			Y	Y
So_OldValue	BSTR	Old value.			Y	Y
So_ProvisioningDttm	BSTR	Date and time of provisioning.			Y	Y
So_SoActionClass	BSTR	Service order action class.			Y	Y
So_Stauts	BSTR	Service order status.			Y	Y
StartDateTime	BSTR	The date from which the parameter start being active.	mm-dd-yyyy hh:mm:ss *5		Y	Y
Status	BSTR	The status of the parameter, stored as a single character code that can be displayed as such. For example, "P" means "Pending".	Active, Inactive, Suspended, Pending *4	1	Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAPParameter methods are:

- change
- create
- getDetails
- AddParameterToSo
- GetServiceOrderDetails

The following tables provide detailed information about each of the IDTAPParameter methods.

change	
Synopsis	HRESULT change ()
Description	Physically changes the BillPlex data repository and any changes in the parameter properties. Execute the <code>getDetails</code> method prior to executing the <code>change</code> method. If you disregard this procedure, the BDR will only be updated with the attributes populated at the time of execution and uninitialized values will delete corresponding database values.
Prerequisite	Set the ParameterID of the Parameter.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	Subscriber, Service, ServiceType, ParameterType.

create	
Synopsis	HRESULT create (long FeatureID)
Description	Physically creates Parameter in the BDR. Upon successful completion of this method, the ParameterID is populated.
Prerequisite	None.
Parameters	FeatureID The Feature for which the Parameter is being created.
Return Value	See common return codes in Appendix A.
Notes	If defined, the end date must be later than the start date. This is an internal method.
See Also	None.

IDTAParameter - Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the parameter from the database.
Prerequisite	Set the ParameterID of the Parameter.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

AddParameterToSo	
Synopsis	HRESULT AddParameterToSo(long grandParentID, long parentID)
Description	Physically creates a new service order of this new or changed parameter in the BillPlex data repository.
Prerequisite	Pass ServiceID and FeatureID to which this parameter belongs.
Parameters	GrandParentID The ServiceID to which this parameter belongs.
	parentID The FeatureID to which this parameter belongs.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

getServiceOrderDetails	
Synopsis	HRESULT getServiceDetails([in]long srvclId, [in]long featId)
Description	Retrieves the attribute values of the Service Order of this parameter object from the database.
Prerequisite	Set the ParameterID and Service OrderID of the feature object.
Parameters	Long srvclId The ServiceID of the service associated with this parameter.
	Long featId The FeatureID of the feature associated with this parameter.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAPParameterType

Description The IDTAPParameterType represents a type of parameter.

Properties The following table contains the properties for IDTAPParameterType.

Property	Data Type	Description	Allowable Values	Max	Get	Put
DefaultValue	BSTR	Default parameter value.	*10	50	Y	Y
EndDateTime	BSTR	Date from which parameter type cannot be assigned to any feature being created in the system.	mm-dd-yyyy hh:mm:ss *5		Y	Y
FeatureType ChangedBy	BSTR	Application that modified feature type to which the parameter type belongs.		30	Y	Y
FeatureType ChangedDateTime	BSTR	Date and time the feature type to which the parameter type belongs is modified.	mm-dd-yyyy hh:mm:ss *5		Y	Y
ParameterName	BSTR	Parameter type name.	Incoming Call Barring control, Length of Time for Installation. *1	50	Y	Y
ParameterType	Long	Parameter type key.	*7		Y	Y
PromptUser	BSTR	Whether or not the client should be prompted while provisioning the service for the value of the parameter.	T, F *10	1	Y	Y
RegularExpression	BSTR	Whether or not parameter type is a value or is an expression.	T, F *10	1	Y	Y
Required	BSTR	Whether or not a value needs to be specified to fully characterize the parameter.	T, F *10	1	Y	Y
SessionID	Long	Unique identifier for database connection, created by the SessionMgr.	*8		Y	Y
StartDateTime	BSTR	Date from which parameter type can be assigned to a new feature in the system.	mm-dd-yyyy hh:mm:ss *5		Y	Y
TypeTag	BSTR	Parameter provisioning classification. Based on this tag the system performs specific automatic operations. Stored as a single character and can be displayed as such. For example, P means provisioning.	Provisioning, Billing, Both(Billing and Provisioning) *4	1	Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAPParameterType methods are:

- getDetails
- getFeatureParameterTypeJoinDetails
- create (not yet implemented)
- change (not yet implemented)

The following tables provide detailed information about each of the IDTAPParameterType methods.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the ParameterType from the database.
Prerequisite	Set the ParameterType attribute of the ParameterType.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getFeatureParameterTypeJoinDetails	
Synopsis	HRESULT getFeatureParameterTypeJoinDetails(long aFeatureType)
Description	Retrieves the information of the Parameter Type associated with a Feature Type, which is passed on. This function helps the Feature Type component to get a list of associated Parameter Types.
Prerequisite	Pass a Feature Type.
Parameters	aFeatureType The value of a Feature Type.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

IDTARatePlan

Description IDTARatePlan represents a plan to bill a subscriber.

Properties The following table contains the properties for IDTARatePlan.

Property	Data Type	Description	Allowable Values	MAX	Get	Put
EndDateTime	BSTR	The date from which the rate plan can no longer be used to rate any subscriber activities.	mm-dd-yyyy hh:mm:ss *5		Y	Y
FeatureRatePlan EndDateTime	BSTR		Mm-dd-yyyy hh:mm:ss *5		Y	Y
FeatureRatePlan StartDateTime	BSTR		Mm-dd-yyyy hh:mm:ss *5		Y	Y
RatePlan Description	BSTR	The description associated to the rate plan.		50	Y	Y
RatePlan FeatureID	Long	The identification of the rate plan which is already assigned to certain feature. Then this id is inserted into v_feat_rate_pln when createFeatureRatePlan is executed. If rate plan id is not NULL, it indicates this rate plan exists in v_feat_rate_pln table.			Y	Y
RatePlanID	Long	System generated unique identification of the rate plan component.	Integer *7		Y	Y
RatePlanName	BSTR	Rate plan name.	Basic Service Plan, Basic Service Indemnified Plan, Premium Service Plan *1	50	Y	Y
RatePlanType	Long	Rate plan type.				
RatePlanTypeNa me	BSTR	Rate plan type name.	Activation Billing, Period Pending, AirTime/Event Rating, Late Payment Rating *1	50		
SessionID	Long	Unique database connection identifier, created by the SessionMgr.	*8		Y	Y
StartDateTime	BSTR	The date from which the rate plan can be used to rate any subscriber activities to which this rate plan was assigned.	Mm-dd-yyyy hh:mm:ss *5		Y	Y

*See table footnotes in Appendix B.

Methods

The IDTARatePlan methods are:

- changeFeatureRatePlan
- createFeatureRatePlan
- getAll
- getDetails
- getDetailsFeatureRatePlan
- create (not yet implemented)
- change (not yet implemented)
- getRatingControlGroupList

The following tables provide detailed information about each of the IDTARatePlan methods.

changeFeatureRatePlan	
Synopsis	HRESULT changeFeatureRatePlan ()
Description	Updates the values in v_feat_rate_pln.
Prerequisite	Set the RatePlanID of the RatePlan.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

createFeatureRatePlan	
Synopsis	HRESULT createFeatureRatePlan (long parentID)
Description	This function adds the record into table v_feat_rate_pln.
Prerequisite	Set the RatePlanID of the RatePlan.
Parameters	ParentID The ID of the feature for which the rate plan is created.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

IDTARatePlan – Methods (Cont.)

getAll	
Synopsis	HRESULT getAll([out] BSTR *pVal)
Description	Retrieves the existing RatePlans in the BillPlex database associated with a specific RatePlanType.
Prerequisite	Set the RatePlanType of the RatePlan.
Parameters	Pval List of RatePlan as string in comma delimited format. Example: "Emergency Rating = 11, Periodic Rating = 29"
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_All([out, retval] VARIANT* pVal)
See Also	None.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the RatePlan from the database.
Prerequisite	Set the RatePlanID attribute of the RatePlan.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getDetailsFeatureRatePlan	
Synopsis	HRESULT getDetailsFeatureRatePlan ()
Description	Selects the startDateTime and endDateTiime from v_feat_rate_pln table.
Prerequisite	None
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAService

Description IDTAService represents a service of a subscriber.

Properties The following table contains the properties for IDTAService.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Comment	BSTR	Comment that may be entered during the provisioning of the service.		255	Y	Y
So-ReqEffectiveDttm	BSTR		mm-dd-yyyy hh:mm:ss *5		Y	Y
EndDateTime	BSTR	Date from which service stopped being active.	mm-dd-yyyy hh:mm:ss *5		Y	Y
OrderDateTime	BSTR	Date and time of order.	mm-dd-yyyy hh:mm:ss *5		Y	Y
ServiceID	Long	System generated unique identification of the service. This attribute should be set before any methods in the service component are executed.			Y	Y
ServiceSoFlag	Long				Y	Y
ServiceType	Long	Service type. Should be consistent with market type of subscriber. Failure to do this will result in BDR inconsistencies.			Y	Y
SessionID	Long	Unique identifier for database connection, created by the SessionMgr.	*8		Y	Y
So_DueDateTime	BSTR	Track the expected due date for an order to complete.			Y	Y
So_ProvisioningDttm	BSTR	Date and time of provisioning.			Y	Y
So_Status	BSTR	Service order status.			Y	Y
So_SoActionClass	BSTR	Service order action class.			Y	Y
So_SoID	BSTR	System generated unique id of the service order.			Y	Y

IDTAService – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
So_VnedID	Long	System-generated VNED ID.			Y	Y
StartDateTime	BSTR	The date from which the service can be fully used, i.e., when all features that composed the service were successfully provisioned.	mm-dd-yyyy hh:mm:ss *5		Y	Y
Status	BSTR	The status of the service, stored as a single character code that can be displayed as such. For example, "P" means "Pending".	Active, Inactive, Suspended, Pending *4	1	Y	Y
SusStartDateTime	BSTR	The start date and time of suspension.			Y	Y
SusEndDateTime	BSTR	The end date and time of suspension.			Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAService methods are:

- createFeature
- createServiceConnection
- change
- create
- getDetails
- getFeatureList
- getServiceConnectionList
- AddServiceToSo
- GetServiceOrderDetails

The following tables provide detailed information about each of the IDTAService methods.

getFeature			
Synopsis	HRESULT createFeature(IDispatch** pFeature)		
Description	Adds a feature to the SAFEARRAY that contains all the features of the service. All the attributes of the added feature component are initialized to zeros and NULLs.		
Prerequisite	Instantiation of the service component.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Pfeature</td> <td>The reference to the feature component that is added to the SAFEARRAY.</td> </tr> </table>	Pfeature	The reference to the feature component that is added to the SAFEARRAY.
Pfeature	The reference to the feature component that is added to the SAFEARRAY.		
Return Value	See common return codes in Appendix A.		
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT createVar_Feature([out, retval] VARIANT* pFeature)		
See Also	None.		

IDTAService – Methods (Cont.)

createServiceConnection	
Synopsis	HRESULT createServiceConnection (IDispatch**pServiceConnection)
Description	Appends a service connection component to the existing service connection list (SAFEARRAY) of the service. All the attributes of the added serviceConnection component are initialized to zeros and NULLs.
Prerequisite	Instantiate the service component.
Parameters	PserviceConnection The reference to the ServiceConnection component that is added to the SAFEARRAY.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT createVar_ServiceConnection([out, retval] VARIANT* pServiceConnection)
See Also	None.

change	
Synopsis	HRESULT change()
Description	Physically propagates the changes in attribute values to the BDR. Execute the <code>getDetails</code> method before executing the <code>change</code> method. If you disregard this, the BDR will only be updated with the attributes populated at the time of execution and uninitialized values will delete corresponding database values.
Prerequisite	Set the ServiceID of the Service.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

create	
Synopsis	HRESULT create(long SubscriberID)
Description	Physically creates a service in the BillPlex data repository. Upon successful completion of this method, the ServiceID is updated.
Prerequisite	Set the ServiceType of the Service.
Parameters	SubscriberID The subscriber for which the service is being created.
Return Value	See common return codes in Appendix A.
Notes	If defined, make the end date later than the start date. This is an internal method.
See Also	None.

IDTAService – Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the Service from the database.
Prerequisite	Set the ServiceID of the Service.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getFeatureList	
Synopsis	HRESULT getFeatureList([out] SAFEARRAY(IDispatch*) *FeatureList)
Description	Retrieves the list of Features that are linked to the service.
Prerequisite	Set the ServiceID of the Service.
Parameters	FeatureList The list of Features associated to the service.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_FeatureList([out, retval] VARIANT* pVal)
See Also	None.

getServiceConnectionList	
Synopsis	HRESULT getServiceConnectionList([out] SAFEARRAY(IDispatch*) *ServiceConnectionList)
Description	Retrieves the list of ServiceConnection components that are linked to the service.
Prerequisite	Set the ServiceID of the Service.
Parameters	ServiceConnectionList The ServiceConnectionList associated with this service.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceConnectionList([out, retval] VARIANT* pVal)
See Also	None.

IDTAService – Methods (Cont.)

AddServiceToSo	
Synopsis	HRESULT AddServiceToSo()
Description	Physically creates a new service order in the Billplex data repository. Upon this method's successful completion, the ServiceOrderID is populated.
Prerequisite	None.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

getServiceOrderDetails	
Synopsis	HRESULT getServiceDetails()
Description	Retrieves the attribute values of the Service Order of this service object from the database.
Prerequisite	Set the ServiceOrderID of the Service.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAServiceBundle

Description The IDTAServiceBundle represents a set of services assembled together under the same tax plan.

Properties The following table contains the properties for IDTAServiceBundle.

Property	Data Type	Description	Allowable Values	Max	Get	Put
ServiceBundleName	BSTR	Service bundle name	Basic Service Plan, Premium Svcs Plan, Basic Indemnified Svcs Plan. *1	32	Y	Y
ServiceBundleType	Long	Service bundle type key.	Integer *7		Y	Y
SessionID	Long	Unique identifier for the database connection, created by the SessionMgr.	*8		Y	Y

**See table footnotes in Appendix B.*

Methods The methods for IDTAServiceBundle are:

- getAll
- getDetails
- getServiceTypeList
- create (not yet implemented)
- change (not yet implemented)

The following tables provide detailed information about each of the IDTAServiceBundle methods.

IDTAServiceBundle – Methods (Cont.)

getAll	
Synopsis	HRESULT getAll([out] BSTR *pVal)
Description	Retrieves the existing ServiceBundles from the database.
Prerequisite	None.
Parameters	PVal The list of ServiceBundles as a string in comma delimited format. Example: "Basic Service Plan = 1, Premium Svcs Plan = 2, Basic Indemnified Svcs Plan = 3."
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_All ([out, retval] VARIANT* pVal)
See Also	None.

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the ServiceBundle from the database.
Prerequisite	Set the ServiceBundleName of the ServiceBundle.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getServiceTypeList	
Synopsis	getServiceTypeList([out] SAFEARRAY(Idispatch*) *ServiceTypeList)
Description	Retrieves the list of ServiceTypes associated with the ServiceBundle.
Prerequisite	Set the ServiceBundleName of the ServiceBundle.
Parameters	ServiceTypeList List of Service Types
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceTypeList ([out, retval] VARIANT* pVal)
See Also	None.

IDTAServiceConnection

Description IDTAServiceConnection represents a physical connection number.

Properties The following table contains the properties for IDTAServiceConnection.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Connection Description	BSTR	The description of the service connection		255	Y	Y
ConnectionNumber	BSTR	The connection number associated to the service. The connection number needs to be a valid connection number for the connection number type.		30	Y	Y
Connection NumberID	Long	The unique identification of the ConnectionNumber. One service may have during the history of its activity several several connections.			Y	Y
ConnectionNumber Type	BSTR	The connection number type of the connection assigned to the service.	POTS, Daleen PBX, ESN Number, LIN Number. *1	30	Y	Y
ConnectionType	BSTR	The type of the connection associated to the service. The connection type needs to be compatible with the service type of the service. In order to know the connection types associated with a service type, please see getConnectionTypeList of the ServiceType Interface.	Wline, Customer LIN, Host LIN *1	30	Y	Y
So_ReqEffective Dttm	BSTR		mm-dd-yyyy hh:mm:ss *5		Y	Y
EndDateTime	BSTR	Date from which the connection stops being active.	Mm-dd-yyyy hh:mm:ss *5		Y	Y
ServiceConnection ID	Long	The system generated unique identification of the service connection. This attribute should be set before executing any method in this component.			Y	Y

IDTAServiceConnection – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
ServiceConnectionSoFlag	Long				Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y
So_ProvisioningDttm	BSTR	Date and time of provisioning.			Y	Y
So_SoActionClass	BSTR	Service order action class.			Y	Y
So_Status	BSTR	Service order status.			Y	Y
So_SoID	Long				Y	Y
StartDateTime	BSTR	Date from which the service connection starts to be active.	mm-dd-yyyy hh:mm:ss *5		Y	Y
Status	BSTR	Status of service connection, stored as a single character code that can be displayed as such. For example, A means assigned. (It represents v_ct_ct_num.stat)	Assigned, Inactive, Permanently Retired, Retired, Unassigned. *4	1	Y	Y

*See table footnotes in Appendix B.

Methods

The IDTAServiceConnection methods are:

- change
- getDetails
- AddServiceConnectionToSo
- GetServiceOrderDetails

The following tables provide detailed information about the IDTAServiceConnection methods.

change	
Synopsis	HRESULT change
Description	Physically propagates the changes in attribute values to the BDR. Execute the <code>getDetails</code> method before executing the <code>change</code> method. If this procedure is disregarded, the BillPlex data repository is only updated with the attributes populated at the time of execution and uninitialized values delete corresponding database values.
Prerequisite	Set the <code>ServiceConnectionID</code> of the <code>ServiceConnection</code> .
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAServiceConnection – Methods (Cont.)

create	
Synopsis	HRESULT create(long ServiceID)
Description	Physically creates a ServiceConnection in the BillPlex data repository. Upon successful completion of this method, the ServiceConnectionID is updated.
Prerequisite	Set the ServiceConnectionID of the ServiceConnection.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

AddServiceConnectionToSo	
Synopsis	HRESULT AddServiceConnectionToSo(long ParentID)
Description	Calls a stored procedure to add service connection records into service order provisioning.
Prerequisite	Set the ServiceID attribute of the Service.
Parameters	ParentID Service ID of the associated service.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

getServiceOrderDetails	
Synopsis	HRESULT getServiceDetails([in] long srvcID)
Description	Retrieves the attribute values of the Service Order of this ServiceConnection object from the database.
Prerequisite	Set the FeatureID and ServiceOrderID of the ServiceConnection object.
Parameters	Long srvcID Service ID of the associated service associated with this ServiceConnection.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTAServiceOrder

Description IDTAServiceOrder represents an activation of a subscriber.

Properties The following table contains the properties for IDTAServiceOrder.

Property	Data Type	Description	Allowable Values	Max	Get	Put
InitialSolID	Long	Initial service order ID.			Y	Y
ProvisioningDateTime	BSTR	Date and time of provisioning.			Y	Y
ServiceOrderID	Long	The system generated unique identification of the service order component. This attribute should be set before executing any method in the service order.	Integer		Y	Y
ServiceOrderStatus	BSTR	The order status of the service order generated to activate the service, stored as a single character code that can be displayed as such. For example, P means pending.	Pending, Complete, Error *4	1	Y	Y
SubscriberID	Long	Subscriber ID of this service order.			Y	Y
SessionID	Long	Unique identifier of the database connection, created by the SessionMgr.	*8		Y	Y

*See table footnotes in Appendix B.

Methods The methods for IDTAServiceOrder are:

- getDetails
- getServiceList
- getServiceSolInfo
- getFeatureSolInfo (not yet implemented)
- getParameterSolInfo
- getServiceConnectionSolInfo
- create (not yet implemented)
- change (not yet implemented)
- getDepositList (not yet implemented)
- getRequestData (not yet implemented)

The following tables provide detailed information about each of the IDTAServiceOrder methods.

IDTAServiceOrder - Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the Service Order from the database.
Prerequisite	Set the ServiceOrderID of the ServiceOrder.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getServiceList	
Synopsis	HRESULT getServiceList([out] SAFEARRAY (IDispatch** ServiceList))
Description	Retrieves the list of services associated to the ServiceOrder.
Prerequisite	Set the ServiceOrderID of the ServiceOrder.
Parameters	ServiceList List of Services.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript.
See Also	None.

getServiceSoInfo	
Synopsis	HRESULT getServiceSoInfo([out, retval] VARIANT* pVal)
Description	Retrieves the attribute values of the Service Order of this ServiceConnection object from the database.
Prerequisite	Set the FeatureID and ServiceOrderID of the ServiceConnection object.
Parameters	ServiceList List of Services.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceList ([out, retval] VARIANT* pVal) This method has not been implemented.
See Also	None.

IDTAServiceOrder - Methods (Cont.)

getServiceSoInfo			
Synopsis	HRESULT getServiceSoInfo ([out, retval] VARIANT* pService)		
Description	Retrieves the VARIANT type wrapping the serviceOrder information.		
Prerequisite	Set the ServiceOrderID of the ServiceOrder.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">pService</td> <td>Pointer to the VARIANT type wrapping the ServiceOrder information.</td> </tr> </table>	pService	Pointer to the VARIANT type wrapping the ServiceOrder information.
pService	Pointer to the VARIANT type wrapping the ServiceOrder information.		
Return Value	See common return codes in Appendix A.		
Notes	None.		
See Also	None.		

getFeatureSoInfo					
Synopsis	HRESULT getFeatureSoInfo ([in]long serviceId, [out]SAFEARRAY (IDispatch*)*pFeatureSoList)				
Description	Retrieves the list of features associated to the service with the serviceID as serviceID.				
Prerequisite	Set the ServiceOrderID of the ServiceOrder.				
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">serviceID</td> <td>The serviceID of the service associated with these features.</td> </tr> <tr> <td>pFeatureSoList</td> <td>List of Features with service order information.</td> </tr> </table>	serviceID	The serviceID of the service associated with these features.	pFeatureSoList	List of Features with service order information.
serviceID	The serviceID of the service associated with these features.				
pFeatureSoList	List of Features with service order information.				
Return Value	See common return codes in Appendix A.				
Notes	<p>This component also has the following method to retrieve the result as Variant to support VBScript:</p> <p>HRESULT getVar_FeatureSoInfo([in]long serviceId, [out, retval]VARIANT* pFeatureSoList)</p>				
See Also	None.				

IDTAServiceOrder - Methods (Cont.)

getParameterSoInfo	
Synopsis	<code>HRESULT getParameterSoInfo ([in]long serviceId, [out]SAFEARRAY (IDispatch*)*pParmSoList)</code>
Description	Retrieves the list of parameters associated to the service with the serviceID as serviceID; and the feature with the featureID.
Prerequisite	Set the ServiceOrderID of the ServiceOrder.
Parameters	serviceID The serviceID of the service associated with these features.
	FeatID The featureID of the feature associated with these parameters.
	pParmSoList List of Parameters with service order information.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: <code>HRESULT getVar_ParameterSoInfo ([in]long serviceId, [in]long featId, [out, retval]VARIANT* pParmSoList)</code>
See Also	None.

getServiceConnectionSoInfo	
Synopsis	<code>HRESULT getServiceConnectionSoInfo ([in]long serviceId, [out]SAFEARRAY (IDispatch*)*pSrcvConnSoList)</code>
Description	Retrieves the list of service connections associated to the service with the serviceID as serviceID.
Prerequisite	Set the ServiceOrderID of the ServiceOrder.
Parameters	serviceID The serviceID of the service associated with these service connections.
	pSrcvConnSoList List of Service Connections with service order information.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: <code>HRESULT getVar_ServiceConnectionSoInfo([in]long serviceId, [out, retval]VARIANT* pFeatureSoList)</code>
See Also	None.

IDTAServiceType

Description IDTAServiceType represents a type of service offered by the operator.

Properties The following table contains the properties for IDTAServiceType.

Property	Data Type	Description	Allowable Values	Max	Get	Put
EndDateTime	BSTR	Date from which service type can no longer be used as a template to provision new services.	Mm-dd-yyyy hh:mm:ss *5		Y	Y
EventClass	Long	This field should not be surfaced.			Y	Y
EventType	Long	This field should not be surfaced.			Y	Y
ExpectedDaysTo Complete	Long	Number of days service type can still be used.	Integer		Y	Y
MaxCnctNumReq	long	Maximum number of connection required.			Y	Y
MinCnctNumReq	long	Minimum number of connection required.			Y	Y
ServiceName	BSTR	Name of service type.	AutoLink Services, Equipment Services, CCG Services.	50	Y	Y
ServiceType	Long	Service type key.	Integer *7		Y	Y
ServiceTypeDescription	BSTR	Brief description of the service type.		255	Y	Y
SessionID	Long	Unique identifier of the database connection, created by the SessionMgr.	*8		Y	Y
StartDateTime	BSTR	Date from which the service type can be used as a template to provision new services.	mm-dd-yyyy hh:mm:ss *5		Y	Y

*See table footnotes in Appendix B.

Methods

The methods for the IDTAServiceType component interface are:

- getAll
- getConnectionTypeList
- getDetails
- getFeatureTypeList
- getFeatureTypeListMarketType
- create (not yet implemented)
- change (not yet implemented)

The following tables provide detailed information about each of the IDTAServiceType methods.

getAll	
Synopsis	HRESULT getAll([out] BSTR *pVal)
Description	Retrieves the existing ServiceTypes in the BillPlex database..
Prerequisite	None.
Parameters	pVal ServiceTypes as a string in a comma delimited format. Example: "Can Do Road Side Assistance Tier 1 = 1, Can Do Emergency Tier 1 = 2, Autolink OEM Business = 22, Autolink OEM Residence = 23, Autolink Employee = 24"
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_All([out, retval] VARIANT* pVal)
See Also	None.

getConnectionTypeList	
Synopsis	HRESULT getConnectionTypeList([out] SAFEARRAY(Idispatch*) *ConnectionTypeList)
Description	The list of ConnectionTypes associated to the ServiceType.
Prerequisite	Set the ServiceType of the Service.
Parameters	ConnectionTypeList List of ConnectionTypes.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ConnectionTypeList([out, retval] VARIANT* pVal)
See Also	None.

IDTAServiceType - Methods (Cont.)

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the ServiceType from the database.
Prerequisite	Set the ServiceType of the ServiceType.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getFeatureTypeList	
Synopsis	HRESULT getFeatureTypeList([out] SAFEARRAY(Idispatch*) *FeatureTypeList)
Description	Retrieves the list of FeatureTypes associated to the ServiceType.
Prerequisite	Set the ServiceType attribute of the ServiceType.
Parameters	FeatureTypeList List of FeatureTypes.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_FeatureTypeList([out, retval] VARIANT* pVal)
See Also	None.

getFeatureTypeListMarketType	
Synopsis	HRESULT getFeatureTypeListMarketType([out] SAFEARRAY(Idispatch*) *FeatureTypeList, BSTR marketType)
Description	Returns the list of FeatureTypes associated to the ServiceType, and valid for a specific MarketType.
Prerequisite	Set the ServiceType attribute of the ServiceType.
Parameters	FeatureTypeList List of FeatureTypes. MarketType The name of the MarketType.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_FeatureTypeListMarketType([in] BSTR marketType,[out, retval] VARIANT* pVal)
See Also	None.

IDTPSessionMgr

Description The SessionMgr optimizes the database connection reuse and ensures that the connection is reused with integrity.

Purpose A user may access BillPlex database multiple times during a single session. For example, the user may inquire current invoice information and then decide to change the service subscription. To minimize the overhead of establishing database connections, it is desirable to reuse the same database connection. Since the database connection also implies who created the database connection in the first place, the database may use the information to decide what data the user is allowed to access through this connection. Therefore, the connections can only be reused among users who have identical privileges.

Methods The methods for the IDTPSessionMgr Component Interface are:

- beginSession
- endSession

The following tables provide detailed information about each of the IDTPSessionMgr methods.

beginSession	
Synopsis	HRESULT beginSession ([out, retval] long *pVal)
Description	Creates a database connection for an application. When an application creates a session, the application passes a user-id and a password. This user-id and password are used to login to the BillPlex database and to create a database connection. A session id is created, associated with the database connection, and returned to the application on the successful completion of this method.
Prerequisite	None.
Parameters	pVal The SessionID for the session
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTPSessionMgr – Methods (Cont.)

endSession	
Synopsis	HRESULT endSession ([in] long newVal)
Description	Ends the session.
Prerequisite	None.
Parameters	newVal The Session ID for the session
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTASubscriber

Description A subscriber can be any individual, department, or company that BillPlex maintains in the subscriber database.

Properties The following table contains the properties for IDTASubscriber.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Address1	BSTR	First line of subscriber's address.		60	Y	Y
Address2	BSTR	Second line of subscriber's address.		60	Y	Y
AddressGeoCode	BSTR	Address geography code.			Y	Y
AddressStatus	Long	Integer value of address status (for multi-language functionality).			Y	Y
AuthorizationIndicator	BSTR	Authorization Indicator.			Y	Y
BalanceChangedBy	BSTR	Application that modified the balance.		30	Y	N
BalanceChangedDate	Date	Date balance modified.	mm-dd-yyyy *5		Y	N
Billable	BSTR	Whether subscriber is responsible for the bill. Top level subscribers must be billable. If the subscriber is billable, then a Term Type, Period Type, and Currency Type will be required.	T, F	1	Y	Y
BillableParentSubscriberID	Long	Unique identifier of the parent subscriber responsible for the bill. A "Billable Subscriber" includes customers at any hierarcial level that are responsible for paying their own bill. If a Billable Subscriber has billing and oversight responsibility for one or more Non-Billable Subscribers, they become "Billable Parent Subscribers". An example includes departments within a company.	Integer		Y	Y
BillingAddress1	BSTR	First line of the street address for use in a paper invoicing method.		60	Y	Y
BillingAddress2	BSTR	Second line of the street address for use in a paper invoicing method.		60	Y	Y
BillingChangedBy	BSTR	Application that modified the Billing.		30	Y	N

IDTASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
BillingChangedDate	Date	Date billing is modified.	mm-dd-yyyy *5		Y	N
BillingCity	BSTR	Name of the for use in paper invoicing method.		50	Y	Y
BillingCompany Name	BSTR	Name of company for use in paper invoicing method.		50	Y	Y
BillingCountry	BSTR	Name of a country to show on the paper invoice.	*2	50	Y	Y
BillingJobTitle	BSTR	Job title to show on paper invoice.		50	Y	Y
BillingName	BSTR	Name of the person who receives the paper invoice		60	Y	Y
BillingPostalCode	BSTR	Postal code to show on the paper invoice – 5 or 9 digit ZIP Code or International formatted Postal Code.	Xxxxxxxxx *3	15	Y	Y
BillingState	BSTR	Name of state to show on letter invoice.		50	Y	Y
BillingSuffix	BSTR	Suffix to show on paper invoice.	Jr., Sr.	15	Y	Y
BillingTitleType	BSTR	Title to show on letter invoice. This property is editable when the user sets the SubscriberType to Individual.	Mr., Mrs., Dr. *2	15	Y	Y
BillOrStatement	BSTR	If T, the invoice has to be a bill. If F, the invoice can either be a bill or a statement.	T, F	1	Y	Y
BirthDateTime	BSTR	Date and time subscriber birth.			Y	Y
CardHolder	BSTR	Name of the credit card holder as it appears on credit card.		65	Y	Y
City	BSTR	City of subscriber's residence or business.	<i>Examples:</i> Boca Raton, Palm Beach	50	Y	Y
CompanyName	BSTR	Subscriber's company name.	<i>Examples:</i> Shell, IBM *1	50	Y	Y
Country	BSTR	Country of subscriber's residence or business.	<i>Examples:</i> U.S.A., France, Germany *6	50	Y	Y
County	BSTR	County of subscriber's residence or business..	<i>Examples:</i> Broward, Palm Beach		Y	Y
CreditCardChangedBy	BSTR	Application that modified the credit card information.		30	Y	N
CreditCardChangedDate	Date	Date and time credit card information of the subscriber is modified.	mm-dd-yyyy *5		Y	N
CreditCard Company	BSTR	Name of subscriber's credit card company.	<i>Examples:</i> VISA, MASTER CARD	30	Y	Y
CreditCardNumber	BSTR	Subscriber's credit card number.		20	Y	Y

IDTASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
CreditCardEndDate	Date	Date from which credit card stops being active.	mm-dd-yyyy *5		Y	Y
CreditCardStart Date	Date	Date from which credit card starts to be active.	mm-dd-yyyy *5		Y	Y
CreditClass	BSTR	Subscriber's credit risk.	High, Medium, Low *2	10	Y	Y
CreditLimit	Long	Subscriber's credit limit.	9999999999.99, 100000.00	12,2	Y	Y
Currency	BSTR	Currency to be used to bill this subscriber.	US Dollars, Francs *1	50	Y	Y
CurrencyFormat	Long	Format of currency	Number	10	Y	Y
CurrencySymbol	BSTR	Currency symbol.		20	Y	Y
DaysInPeriod	Long	Number of days in period.			Y	Y
DecimalSeparator	BSTR	Number of decimal places.			Y	Y
DefaultCurrency Code	BSTR	Default currency code.			Y	Y
DriverLicense Number	BSTR	Driver license number or other identification of the subscriber.	Xxxxxxxxxx *3	15	Y	Y
EDIChangedBy	BSTR	Application that modified the EDI information.		30	Y	N
EDIChangedDate	Date	Date the EDI information of the subscriber is modified.	mm-dd-yyyy *5		Y	N
EDIDescription	BSTR	A brief description of the EDI.		32	Y	Y
EDIType	Long	Type of electronic data interchange. This field is generated internally by the system when the subscriber component is created.			Y	Y
EmailChangedBy	BSTR	Application that modified the subscriber e-mail.		30	Y	N
EmailChangedDate	Date	Date the subscriber e-mail information is modified.	mm-dd-yyyy *5		Y	N
EndDateTime	BSTR	Date and time from which subscriber's account becomes Inactive, Suspended, or Written Off to profit and loss.	mm-dd-yyyy hh:mm:ss *5		Y	Y
ExpirationDate	Date	Credit card expiration date.	mm-dd-yyyy *5		Y	Y
FaxChangedBy	BSTR	Application that modified the subscriber's fax information.		30	Y	N

IDTASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
FaxChangedDate	Date	Date subscriber's fax information is modified	mm-dd-yyyy *5		Y	N
FederalIDNum	BSTR	Federal ID number of the subscriber.			Y	Y
FirstName	BSTR	First name of the subscriber		25	Y	Y
InTreatment	BSTR	Whether or not subscriber is under any treatment action.	T, F	1	Y	Y
InvoiceBalance	Long	Total amount of all the open invoices.	99999999.99, 10000.00, 20.50		Y	Y
InvoiceDay	Long	Day of the billing cycle on which subscriber is billed.	The days include 1-2-3-4-5-6-7.	25	Y	Y
InvoiceDeliverType	BSTR	Delivery type of Invoice.	Bulk Mailing, E-Mail Delivery, Fax Delivery. *2, *4	50	Y	Y
InvDirFmtGrpCls	BSTR	Invoice format group class.			Y	Y
InvDirHasToBeBill	BSTR	Indicates billable subscriber.			Y	Y
InvFmtFmtGrpCls	BSTR	Invoice format group class.			Y	Y
InvoiceFormatClass	BSTR	The invoice format class.			Y	Y
InvoiceFormatType	BSTR	Defines name and class of an invoice format. An Invoice Class specifies whether an invoice is bill or statement.	Monthly Cellular Bill, Monthly Pager Bill *2	50	Y	Y
InvMethHasToBe Bill	BSTR	Indicates that subscriber must be billable.			Y	Y
InvoiceMethodType	BSTR	Defines method that invoice is generated.	Credit Card, E-Mail, Fax File, Paper Mail *2, *4	50	Y	Y
JobTitle	BSTR	Subscriber's job title.	<i>Examples:</i> Manager, President *2	50	Y	Y
LastInvoiceDate Time	BSTR	Date and time of last invoice.	mm-dd-yyyy hh:mm:ss *5		Y	Y
LastName	BSTR	Last name of subscriber. Should only be editable when user sets SubscriberType to Individual. <i>This is client responsibility. API does not enforce this rule.</i>	<i>Examples:</i> Smith, Gates	35	Y	Y

IDTASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
LastUpdated	Date		mm-dd-yyyy *5		Y	N
MarketType	BSTR	Defines a particular type of customer.	<i>Examples:</i> Business, Residence, Government *2	50	Y	Y
MiddleInitial	BSTR	Middle initial of subscriber's name. Can be edited when user sets SubscriberType to Individual.	X	1	Y	Y
NegativeCurrency Format	Long	The negative currency format.			Y	Y
NumberOfDecimal Place	Long	Number of decimal places.			Y	Y
OuterMost SubscriberID	Long	Identification of subscriber who has highest hierarchically responsible position in tree, on which subscriber is a leaf.	Integer		Y	Y
PaprDataAddGeo Code	BSTR	Address geography code.			Y	Y
PaprDataAddStatus	Long	Stores an integer value of address status (for multi-language functionality.)	Un-Validated, Known But Unvalidated, Changed requiring Geocode, Validated.		Y	Y
ParentSubscriberID	Long	Identification of subscriber who has immediate hierarchically responsible position for the subscriber.	Integer		Y	Y
PaymentBalance	Long	Amount in subscriber's account. Can be accumulated in two ways. By overpayment, i.e., the balance of payment after closing all the open invoices. If the sum of current payment and existing account balance is not enough to close oldest open invoice, it is added to subscriber's account.	9999.99, 200.00	14,2	Y	Y
Period	BSTR	Billing cycle as a period of time.	<i>Examples:</i> Monthly, Quarterly, By-Monthly *1	50	Y	Y
Phone1AreaCode	BSTR	Area code of subscriber's primary phone.	Xxxx *3	6	Y	Y
Phone1Country Code	BSTR	Country code of subscriber's primary phone.	Xxxx *3	4	Y	Y
Phone1Extension	BSTR	Extension of subscriber's primary phone.	Xxxx *3	5	Y	Y
Phone1Number	BSTR	Telephone number of subscriber's primary phone.	Xxxxxxxxxx *3	10	Y	Y

ITDASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
Phone2AreaCode	BSTR	Area code of subscriber's secondary phone.	Xxxx *3	6	Y	Y
Phone2Country Code	BSTR	Country code of subscriber's secondary phone.	Xxxx *3	4	Y	Y
Phone2Extension	BSTR	Extension of subscriber's secondary phone.	Xxxx *3	5	Y	Y
Phone2Number	BSTR	Telephone number of subscriber's secondary phone.	Xxxxxxxxx *3	6	Y	Y
PostalCode	BSTR	Postal code of subscriber's business or residence - 5 or 9 digit ZIP Code or International formatted Postal Code.	Xxxxxxxx *3	15	Y	Y
PrimaryEmailAddress	BSTR	E-mail address of the subscriber.	*3	30	Y	Y
PrimaryFaxNumber	BSTR	Fax number of the subscriber.	*3	15	Y	Y
SessionID	Long	Unique identifier for the database connection, created by the SessionMgr.	*8		Y	Y
SocialSecurityNumber	BSTR	The social security number of the subscriber.	xxxxxxxxxx *3	15	Y	Y
StartDateTime	BSTR	The date and time a subscriber first achieved active status in BillPlex. The user cannot change information in the Start Date field.	mm-dd-yyyy hh:mm:ss *5		Y	Y
State	BSTR	The name of the state where the subscriber does business or resides.	<i>Examples:</i> Florida, Colorado, California *6	50	Y	Y
Status	BSTR	The status of the subscriber, stored as a single character code that can be displayed as such. For example, "A" means "Active". Active - an active account in good standing. Inactive - the subscriber either voluntarily or for some other reason, is no longer using the services. Suspended - a subscriber was involuntarily suspended due to a Customer Service problem, or an overdue bill in the Treatment and Collection process. Written Off - reserved for subscribers that were written off to profit and loss by Collections department.	Active, Inactive, Suspended, Written Off. *4	1	Y	Y
SubscriberID	Long	The system generated unique identification of the subscriber. This attribute should be set before any methods are executed in the subscriber component.	Integer		Y	Y

ITDASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
SubscriberName	BSTR	Name of subscriber. This property is editable when the user sets a Company, Department or Location as the "SubscriberType".		255	Y	Y
SubscriberTaxPlanTypeID	Long	System generated tax plan id, identifying the combination of the SubscriberID, TaxPlanType, and StartDate.			Y	Y
SubscriberType	BSTR	Type of subscriber.	Department, Individual, Location *1	50	Y	Y
Suffix	BSTR	Ending suffix in a subscriber's name. This property is editable when the user sets the SubscriberType to "Individual".	Jr., Sr. *1	15	Y	Y
SusStartDateTime	BSTR	Start date and time of suspension.			Y	Y
SusEndDateTime	BSTR	End date and time of suspension.			Y	Y
TaxExemptionChangedBy	BSTR	Application that modified the TaxExemption.		30	Y	N
TaxExemptionChangedDate	Date	Date the TaxExemption is modified.	mm-dd-yyyy *5		Y	N
TaxExemptionEndDate	Date	Date from which the tax plan exemption stops being active.	mm-dd-yyyy *5		Y	Y
TaxExemptionStartDate	Date	Date from which the tax exemption stars being active.	mm-dd-yyyy *5		Y	Y
TaxPlanChangedBy	BSTR	Application that modified the TaxPlan.		30	Y	N
TaxPlanChangedDate	Date	Date the TaxPlan is modified.	mm-dd-yyyy *5		Y	N
TaxPlanEndDate	Date	Date from which the tax plan stops being active.	mm-dd-yyyy *5		Y	Y
TaxPlanStartDate	Date	Date from which the tax plan starts to be active.	mm-dd-yyyy *5		Y	Y
TaxPlanType	Long	Tax plan type.			Y	Y
TaxPlanValidateCode	Long	Tax plan validation code.			Y	Y
Terms	BSTR	Subscriber's payment grace period.	25 days, 15 days, 0 days	50	Y	Y
ThousandSeparator	BSTR	Thousandths separator for multi-language functionality.			Y	Y
TitleType	BSTR	The title of an individual subscriber. This property is editable when the user sets the SubscriberType to "Individual".	Mr., Mrs., Dr. *2	15	Y	Y
Track1	BSTR	First strip from the back of the credit card magnetic strip.		80	Y	Y
Track2	BSTR	Second strip from the back of the credit card magnetic strip.		42	Y	Y
Track3	BSTR	Third strip from the back of the credit card magnetic strip.		108	Y	Y

IDTASubscriber – Properties (Cont.)

Property	Data Type	Description	Allowable Values	Max	Get	Put
TransactionBalance	Long	Invoice Balance + the sum of each transaction amount that occurred after the last invoice.	9999999999.99 150.00	14,2	Y	Y
Treatment	BSTR	The name of the treatment plan of the subscriber	VIP, Normal *2	10	Y	Y
TreatmentChanged By	BSTR	The application that modified the Treatment.		30	Y	N
TreatmentChanged Date	Date	The date the Treatment is modified.	mm-dd-yyyy *5		Y	N
TreatmentCode Comment	BSTR	A comment to explain why the subscriber has an exceptional treatment code.		255	Y	Y

*See table footnotes in Appendix B.

Methods

The methods for the IDTASubscriber component interface are:

- createService
- authenticate
- authenticateByConn
- changeSubscriber
- createSubscriber
- getCharacteristic
- getCharacteristicList
- getInvoiceList
- getInvoiceMsgList (not yet implemented)
- getServiceList
- getServiceOrderList
- getSubscriberCommentList
- getSubscriberDetails
- getTaxExemptList (not yet implemented)
- getTaxPlanList (not yet implemented)
- getTransactionList
- searchSubscriber
- setCharacteristic
- setCharacteristicList
- createServiceOrder

The following tables provide detailed information about each of the IDTASubscriber methods.

IDTASubscriber – Methods (Cont.)

createService			
Synopsis	HRESULT createService (IDispatch** pService)		
Description	Adds a service component to the SAFEARRAY of contains all the service of the subscriber. All the attributes of the added service component are initialized to zeros and NULLs.		
Prerequisite	Instantiation of the subscriber component.		
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">pService</td> <td>The reference to the service component that is added to the SAFEARRAY of services.</td> </tr> </table>	pService	The reference to the service component that is added to the SAFEARRAY of services.
pService	The reference to the service component that is added to the SAFEARRAY of services.		
Return Value	See common return codes in Appendix A.		
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: STDMETHODCALLTYPE(createVar_Service) (/*[out, retval]*/ VARIANT * pServiceObj)		
See Also	None.		

authenticate							
Synopsis	HRESULT authenticate([in] long subscriberID, [in] BSTR password, [out] long* result)						
Description	Authenticates the subscriber based on the SubscriberID.						
Prerequisite	None.						
Parameters	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">subscriberID</td> <td>The ID of the subscriber to be authenticated.</td> </tr> <tr> <td>password</td> <td>The subscriber's password.</td> </tr> <tr> <td>result</td> <td>The output status: 1 – Wrong password 0 – Correct password</td> </tr> </table>	subscriberID	The ID of the subscriber to be authenticated.	password	The subscriber's password.	result	The output status: 1 – Wrong password 0 – Correct password
subscriberID	The ID of the subscriber to be authenticated.						
password	The subscriber's password.						
result	The output status: 1 – Wrong password 0 – Correct password						
Return Value	See common return codes in Appendix A.						
Notes	None.						
See Also	None.						

IDTASubscriber – Methods (Cont.)

authenticateByConn		
Synopsis	<code>HRESULT authenticateByConn([in] BSTR connectionNumber, [in] BSTR connectionType, [in] BSTR password, [out] long* result)</code>	
Description	Authenticates the subscriber based on the ConnectionNumber.	
Prerequisite	None.	
Parameters	connectionNumber	The connection number of the subscriber to be authenticated.
	connectionType	The ConnectionType of the ConnectionNumber.
	password	The subscriber's password.
	result	The output status: 1 – Wrong password 0 – Correct password
Return Value	See common return codes in Appendix A.	
Notes	None.	
See Also	None.	

changeSubscriber	
Synopsis	<code>HRESULT changeSubscriber()</code>
Description	Physically changes the BillPlex data repository for any changes on the subscriber properties, including lists. When changing subscriber properties, a check is made to determine if the corresponding objects inside the respective lists are changed. If they are changed, the <code>change</code> method on the respective object is called. Execute the <code>getDetails</code> method before executing the <code>change</code> method. If you disregard this procedure, the BillPlex data repository is only updated with the attributes populated at the time of execution and uninitialized values will delete corresponding database values.
Prerequisite	Set the Subscriber ID of the Subscriber.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTASubscriber – Methods (Cont.)

createSubscriber	
Synopsis	HRESULT createSubscriber()
Description	Physically creates a subscriber in the BillPlex data repository. Upon successful completion of this method, the SubscriberID is populated.
Prerequisite	In order to successfully create a subscriber, initialize the following attributes before you call this method: Billable, Country, CreditClass, Currency, InTreatment, LastName, MarketType, Period, State, Status, SubscriberType, Terms, Title, Treatment.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getCharacteristic		
Synopsis	HRESULT getCharacteristic([in] BSTR group, [in] BSTR name, [out] BSTR* pvalue)	
Description	Returns the value of a subscriber characteristic belonging to a specific characteristic group.	
Prerequisite	Set the SubscriberID of the Subscriber.	
Parameters	group	The group in which the subscriber characteristic belongs.
	name	The subscriber characteristic's name.
	pvalue	The subscriber characteristic's value.
Return Value	See common return codes in Appendix A.	
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_Characteristic([in] BSTR group, [in] BSTR name, [out, retval] VARIANT* pVal)	
See Also	None.	

IDTASubscriber – Methods (Cont.)

getCharacteristicList	
Synopsis	HRESULT getCharacteristicList ([in] BSTR group, [out] BSTR* pvalueList)
Description	Returns a comma delimited BSTR of characteristic name/value pairs of a subscriber characteristic group. The characteristic name/value pair are separated by the “=” character.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	group The group name of the subscriber characteristic from which the list should be retrieved.
	pvalueList The list of characteristic name/value pairs.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_CharacteristicList([in] BSTR group, [out, retval] VARIANT* pVal)
See Also	None.

getInvoiceList	
Synopsis	HRESULT getInvoiceList([out] SAFEARRAY (Idispatch** InvoiceList)
Description	Returns the list of invoices of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	InvoiceList The list of invoices.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_InvoiceList([out, retval] VARIANT* pVal)
See Also	GetServiceList

IDTASubscriber – Methods (Cont.)

getInvoiceMsgList	
Synopsis	HRESULT getInvoiceMsgList([out] SAFEARRAY (Idispatch** InvoiceList)
Description	Returns the list of invoice messages of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	InvoiceMsgList The list of invoice messages.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_InvoiceMsgList([out, retval] VARIANT* pVal) This method has not been implemented yet.
See Also	None.

getServiceList	
Synopsis	HRESULT getServiceList([out] SAFEARRAY (Idispatch** InvoiceList)
Description	Returns the list of servicess of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	ServiceList The list of subscriber services.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceList([out, retval] VARIANT* pVal)
See Also	None.

getServiceOrderList	
Synopsis	HRESULT getServiceOrderList([out] SAFEARRAY (Idispatch** InvoiceList)
Description	Returns the list of service orders of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	ServiceOrderList The list of service orders.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_ServiceOrderList([out, retval] VARIANT* pVal)
See Also	ServiceOrder component.

IDTASubscriber – Methods (Cont.)

getSubscriberCommentList	
Synopsis	HRESULT getSubscriberCommentList([out] SAFEARRAY (Idispatch*) *CommentList)
Description	Returns the list of subscriber comments of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	CommentList The list of comments.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_SubscriberCommentList([out, retval] VARIANT* pVal)
See Also	None.

getSubscriberDetails	
Synopsis	HRESULT getSubscriberDetails()
Description	Returns the attribute values of the subscriber from the database.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getTaxExemptList	
Synopsis	HRESULT getTaxExemptList([out] SAFEARRAY (Idispatch*) *TaxExemptList)
Description	Returns the list of TaxExemptions of the subscriber.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	TaxExemptList The list of TaxExemptions.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_TaxExemptList([out, retval] VARIANT* pVal) This method has not been implemented yet.
See Also	None.

IDTASubscriber – Methods (Cont.)

getTaxPlanList	
Synopsis	HRESULT getTaxPlanList([out] SAFEARRAY (Idispatch*) *TaxPlanList)
Description	Returns the list of TaxPlans of the subscriber. Each subscriber has only one tax plan at any given time. Use this method as a historical perspective of the subscriber's tax plan.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	TaxPlanList The list of TaxPlan.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_TaxPlanList([out, retval] VARIANT* pVal) This method has not been implemented yet.
See Also	None.

getTransactionList	
Synopsis	HRESULT getTransactionList([in] Long InvoiceID, [out] SAFEARRAY (BSTR*) *TransactionList)
Description	Retrieves the list of Transactions of the subscriber on a specific invoice.
Prerequisite	Set the SubscriberID of the Subscriber.
Parameters	InvoiceID The ID of the invoice on which the transactions are listed. TransactionList The list of transactions.
Return Value	See common return codes in Appendix A.
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_TransactionList([in] long InvoiceID, [out, retval] VARIANT* pVal)
See Also	None.

IDTASubscriber – Methods (Cont.)

searchSubscriber		
Synopsis	HRESULT searchSubscriber([out] long found, [out] SAFEARRAY (IDispatch) *result, [in] long howMany)	
Description	Returns a list of subscribers stored in the BillPlex data repository that meet specific criteria. The criteria is defined by setting the following attributes for the current system release: Phone1CountryCode, Phone1AreaCode, Phone1Number	
Prerequisite	Set the attributes defined in the search criteria.	
Parameters	found	The Number of subscribers matching the search criteria.
	result	The list of subscriber components found that match the search criteria.
	howMany	The number of subscriber components that should be listed in the result form the entire list that match the search criteria.
Return Value	See common return codes in Appendix A.	
Notes	This component also has the following method to retrieve the result as Variant to support VBScript: HRESULT getVar_searchSubscriber([in] long howMany, [out] VARIANT* found, [out, retval] VARIANT* pVal)	
See Also	None.	

setCharacteristic		
Synopsis	HRESULT setCharacteristics([in] BSTR Group, [in] BSTR name, [in] BSTR value)	
Description	Physically sets the value of the subscriber characteristic in the BillPlex data repository.	
Prerequisite	Set the SubscriberID of the Subscriber.	
Parameters	group	The characteristic group in which the characteristic whose value should be set belongs.
	name	The name of the characteristic whose value should be set.
	value	The value of the characteristic.
Return Value	See common return codes in Appendix A.	
Notes	Every market type has a list of characteristics groups associated. Characteristics are divided into groups. The BillPlex administrator defines characteristics for every group.	
See Also	None.	

IDTASubscriber - Methods

getCharacteristicList		
Synopsis	HRESULT getCharacteristicList([in] BSTR group, [in] BSTR valueList)	
Description	Physically sets the values of a list of subscriber characteristics that belong to a specific characteristic group in the BillPlex data repository.	
Prerequisite	Set the SubscriberID of the Subscriber.	
Parameters	group	The characteristic group in which the subscriber characteristics belong.
	valueList	The comma delimited string of characteristic name/value pairs. Example: valueList = ""Automobile Insurance Provider"" ""AutoLink"" ""Security Company Telephone Number"" ""autolink test""
Return Value	See common return codes in Appendix A.	
Notes	None.	
See Also	None.	

createServiceOrder	
Synopsis	HRESULT createServiceOrder()
Description	Checks whether a new service order needs adding. If yes, it physically creates a new service order for a service by calling service component's AddServiceToSo method.
Prerequisite	None.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	This is an internal method.
See Also	None.

IDTASubscriberComment

Description IDTASubscriberComment represents comments entered for a subscriber's account.

Properties The following table contains properties for the IDTASubscriberComment component interface.

Property	Data Type	Description	Allowable Values	Max	Get	Put
Comment	BSTR	The comment.		255	Y	Y
CommentClass	BSTR	A Comment Class refers to an area of system information where a comment applies.	Subscriber Phone Status. *1	50	Y	Y
CommentDateTime	BSTR	The date the comment was introduced. This date may be different from the creation date and refers to the date when the comment was thought to be appropriate.	mm-dd-yyyy hh:mm:ss *5		Y	Y
CommentType	BSTR	A Comment Type represents an action subset of the parent Comment Class.	Defective, Removed, Sold to Agent. *1	30	Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y
SubscriberCommentID	Long	The system generated unique identification of the SubscriberComment. This attribute should be set before executing any method in this component.	Integer *7		Y	Y

*See table footnotes in Appendix B.

Methods The methods for the IDTASubscriberComment component interface are:

- create
- getDetails
- change (not yet implemented)

The following tables provide detailed information about each of the IDTASubscriberComment methods.

IDTASubscriberComment – Methods (Cont.)

create	
Synopsis	HRESULT create (long SubscriberID)
Description	Creates a comment for the subscriber in the BillPlex data repository. Upon successful completion of this method, the SubscriberCommentID is populated.
Prerequisite	None.
Parameters	SubscriberID The ID of the Subscriber component for which the comment is being created.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

getDetails	
Synopsis	HRESULT getDetails ()
Description	Retrieves the attribute values of the SubscriberComment from the database.
Prerequisite	Set the SubscriberCommentID of the Subscriber.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

IDTATransaction

Description IDTATransaction represents a monetary action that BillPlex processes for a subscriber.

Properties The following table contains the properties for the IDTATransaction component interface.

Property	Data Type	Description	Allowable Values	Max	Get	Put
AlternateTaxPlanType	Long				Y	Y
Amount	Long	The current monetary amount of the transaction.			Y	Y
BypassBalanceUpdate	BSTR			1	Y	Y
ConnectionNumberID	Long	Unique number identifying the connection to connection number record.			Y	Y
CurrencyCode	BSTR	Indicates the type of currency that a subscriber was billed.		10	Y	Y
FeatureID	Long	System generated unique id of the Feature.			Y	Y
FeatureType	Long	Type of Feature.	Basic, Call Forwarding.		Y	Y
InvoiceID	Long	System generated, unique number used to identify an invoice.			Y	Y
ServiceID	Long	System generated unique ID of the Service.			Y	Y
ServiceType	Long	Type of service	POTS, Internet, Pager.		Y	Y
SessionID	Long	The ID for the database connection, created by the SessionMgr.	*8		Y	Y
SURID	Long	Unique number identifying the Service Usage Record(SUR).			Y	Y
TableID	Long	The id of the table that contains all the basic usage fields including connection number and usage date/time. It is intended to be used as an abstract guideline, not a real table.			Y	Y
TaxType	Long	Indicates an individual tax type. An example includes Federal Tax.			Y	Y
TransactionClass	BSTR	Defines whether a selected transaction is an: <ul style="list-style-type: none"> • Adjustment • Credit • Payment • Charge • Refund 		1	Y	Y

Property	Data Type	Description	Allowable Values	Max	Get	Put
TransactionDate Time	Date	The date and time that a transaction was generated.	mm-dd-yyyy hh:mm:ss *5		Y	Y
TransactionDescription	BSTR	Description of transaction that can be system-generated or entered by operator.		255	Y	Y
TransactionID	Long	A system generated number that identifies a transaction.			Y	Y
TransactionStatus	BSTR	The completion state of a transaction , stored as a single character code that can be displayed as such. For example, "P" means "Pending".	Ready, Pending, Canceled, Invoiced.	1	Y	Y
TransactionType	BSTR	A Transaction Type indicates a type of subscriber transaction.	Peak usage, service charge.		Y	Y
TransactionTypeName	BSTR	Name of the transaction type.		50	Y	Y

*See table footnotes in Appendix B.

Methods

The IDTATransaction methods are:

- getDetails
- create (not yet implemented)
- change (not yet implemented)
- getAllowedTransactionStatus (not yet implemented)

The following table provides detailed information about the IDTATransaction method.

getDetails	
Synopsis	HRESULT getDetails()
Description	Retrieves the attribute values of the transaction from the database.
Prerequisite	Set the TransactionID of the Transaction.
Parameters	None.
Return Value	See common return codes in Appendix A.
Notes	None.
See Also	None.

Return Codes and Error Messages



Return Codes and Error Messages

Overview

Common return codes and corresponding error messages used throughout multiple BillPlex API methods are listed here. BillPlex uses the HRESULT convention as defined by Microsoft. You can use the HRESULT to convey an error status and to distinguish between several different outcomes that may be equally successful in that they do not signify an error, but may require different processing on return from the function. See Chapter 2 for a more extensive description of result codes and error information.

Return Codes and Error Message Table

The following table lists and describes the return codes and lists the corresponding error message for each return code.

Return Code	Description	Corresponding Error Message
0x00000000	S_OK:Successful.	None
0x00000000	NOERROR: same as S_OK.	None
0x00000001	S_FALSE:Successful.	None
0x00000200	S_NOCHANGE: Function succeeded. No data has changed.	None
0x00000201	S_CHANGE: Function succeeded. Data has changed.	None
0x8___0300	BDR error.	BillPlex Data Repository read failed
0x8___0301	BDR error.	BillPlex Data Repository write failed
0x8___0302	BDR error.	Call to BillPlex Data Repository was not successful

Return Codes and Error Message Table (continued)

Return Code	Description	Corresponding Error Message
0x8__0303	BDR error.	BillPlex Data Repository transaction rollback failure
0x8__0304	BDR error.	BillPlex Data Repository commit failed.
0x8__0305	BDR error.	BDR statement max length exceeded.
0x8__0306	BDR error.	Error starting BillPlex Data Repository transaction.
0x8__0307	BDR error.	No data returned by BDR selection.
0x8__0308	BDR error.	Error executing BPDR statement.
0x8__0309	BDR error.	BillPlex Data Repository returned more than one record.
0x8__030a	BDR error.	The BDR returned excessive information.
0x8__030b	BDR error.	BillPlex Data Repository update failed.
0x8__030c	BDR error.	BDR: cannot connect to Database
0x8__030d	BDR error.	BDR endSession: Could not free connection.
0x8__030e	BDR error.	BillPlex Data Repository: RemoveKey failed.
0x8__030f	BDR error.	BDR: not possible to parse the incoming data.
0x8__0310	BDR error.	BDR: stored procedure error
0x8__0340	General Core API error.	Create: ParentID is not defined!
0x8__0341	General Core API error.	Could not create Session Manager.
0x8__0342	General Core API error.	Service Type can not be null!
0x8__0343	General Core API error.	Did not find any Feature types.
0x8__0344	General Core API error.	The Service Type must be set before calling this function.
0x8__0345	General Core API error.	Service Type and ServiceName not specified.

Return Codes and Error Message Table (continued)

Return Code	Description	Corresponding Error Message
0x8__0346	General Core API error.	Parameter Type should be set before calling this function.
0x8__0347	General Core API error.	getDetails: ConnectionNumberType and ConnectionNumberName not specified.
0x8__0348	General Core API error.	Unable to find a connection number.
0x8__0349	General Core API error.	getDetails: ConnectionType and ConnectionName not specified.
0x8__034a	General Core API error.	Connection Type should be set before calling this function.
0x8__034b	General Core API error.	Market Name should be set before calling this function.
0x8__034c	General Core API error.	Change failed on Market Type.
0x8__034d	General Core API error.	Create: CommentType or CommentClass can not be null.
0x8__034e	General Core API error.	Session Dictionary can only hold 200 records.
0x8__034f	General Core API error.	Session ID is not in the list.
0x8__0350	General Core API error.	More than one value returned.
0x8__0351	General Core API error.	The input list is empty.
0x8__0352	General Core API error.	Last call setCharacteristic failed.
0x8__0353	General Core API error.	Unable to authenticate subscriber.
0x8__0354	General Core API error.	Unable to modify unique property.
0x8__0355	General Core API error.	The Subscriber ID has not been set.
0x8__0356	General Core API error.	SAFEARRAY lock error
0x8__0357	General Core API error.	SAFEARRAY redimensioning error
0x8__0358	General Core API error.	Required fields are not satisfied.
0x8__0359	General Core API error.	The Service ID must be set before calling this function.
0x8__035a	General Core API error.	Could not create the Service interface.

Return Codes and Error Message Table (continued)

Return Code	Description	Corresponding Error Message
0x8__035b	General Core API error.	Failed to create the ConnectionMgr.
0x8__035c	General Core API error.	Unable to get the length of the list.
0x8__035d	General Core API error.	Could not create the ServiceOrder interface.
0x8__035e	General Core API error.	Could not get the list.
0x8__035f	General Core API error.	Could not create the Subscriber Comment interface.
0x8__0360	General Core API error.	Could not create the Invoice interface.
0x8__0361	General Core API error.	RatePlanName and RatePlanID not specified.
0x8__0362	General Core API error.	getDetails: FeatureType and FeatureName not specified.
0x8__0363	General Core API error.	Cannot insert a null attribute OrderDateTime.
0x8__0364	General Core API error.	FeatureID should be set before calling this function.
0x8__0365	General Core API error.	Could not create the parameter interface.
0x8__0366	General Core API error.	This method has not been implemented.
0x8__0367	General Core API error.	ConnectionNumberType must be set before calling this function.
0x8__0368	General Core API error.	FeatureType must be set before calling this function.
0x8__0369	General Core API error.	No Service Type in list.
0x8__036A	General Core API error.	Not a billable subscriber.
0x8__036B	General Core API error.	OuterMostSubscriberID does not patch its parentSub's OuterMostSubscriberID.
0x8__036C	General Core API error.	BillableParenSubscriberId set wrong: it must be billable.
0x8__0370	Missing Mandatory Field.	Mandatory field missing: State.

Return Codes and Error Message Table (continued)

Return Code	Description	Corresponding Error Message
0x8__0371	Missing Mandatory Field.	Mandatory field missing: Country
0x8__0372	Missing Mandatory Field.	Mandatory field missing: TitleType.
0x8__0373	Missing Mandatory Field.	Mandatory field missing: Period.
0x8__0374	Missing Mandatory Field.	Mandatory field missing: Currency.
0x8__0375	Missing Mandatory Field.	Mandatory field missing: Status.
0x8__0376	Missing Mandatory Field.	Mandatory field missing: SubscriberType.
0x8__0377	Missing Mandatory Field.	Mandatory field missing: MarketType.
0x8__0378	Missing Mandatory Field.	Mandatory field missing: Treatment.
0x8__0379	Missing Mandatory Field.	Mandatory field missing: Terms.
0x8__037a	Missing Mandatory Field.	Mandatory field missing: CreditClass.
0x8__037b	Missing Mandatory Field.	Mandatory field missing: LastUpdated.
0x8__037c	Missing Mandatory Field.	Mandatory field missing: InvoiceMethodType.
0x8__037d	Missing Mandatory Field.	Mandatory field missing: InvoiceDeliverType.
0x8__037e	Missing Mandatory Field.	Mandatory field missing: InvoiceFormatType.
0x8__037f	Missing Mandatory Field.	Mandatory field missing: BillOrStatement.
0x8__0380	Missing Mandatory Field.	Mandatory field missing: TreatmentCodeLog.
0x8__0381	Missing Mandatory Field.	Mandatory field missing: BillingName
0x8__0382	Missing Mandatory Field.	Mandatory field missing: BillingState
0x8__0383	Missing Mandatory Field.	Mandatory field missing: BillingCountry.
0x8__0384	Missing Mandatory Field.	Mandatory field missing: BillingTitleType.
0x8__0385	Missing Mandatory Field.	Mandatory field missing: CreditCardNumber.
0x8__0386	Missing Mandatory Field.	Mandatory field missing: CreditCardCompany.

Return Codes and Error Message Table (continued)

Return Code	Description	Corresponding Error Message
0x8__0387	Missing Mandatory Field.	Mandatory field missing: ExpirationDate.
0x8__0388	Missing Mandatory Field.	Mandatory field missing: PrimaryEmailAddress.
0x8__0389	Missing Mandatory Field.	Mandatory field missing: PrimaryFaxNumber.
0x8__038A	Missing Mandatory Field.	Mandatory field missing: ConnectionType.
0x8__038B	Missing Mandatory Field.	Mandatory field missing: ConnectionNumberType.
0x8__038C	Missing Mandatory Field.	Mandatory field missing: EffectiveDateTime.
0x8__038D	Missing Mandatory Field.	Mandatory field missing: LastName
0x8__038E	Missing Mandatory Field.	Mandatory field missing: SubscriberName
0x8__038F	Missing Mandatory Field.	Mandatory field missing: TransactionBalance.
0x8__0390	Missing Mandatory Field.	Mandatory field missing: InvoiceBalance.
0x8__0391	Missing Mandatory Field.	Mandatory field missing: PaymentBalance.
0x8__0392	Missing Mandatory Field.	Mandatory field missing: ParentSubscriberID.
0x8__0393	Missing Mandatory Field.	Mandatory field missing: OuterMostSubscriberID.
0x8__0394	Missing Mandatory Field.	Mandatory field missing: BillableParentSubscriberID.

Footnotes

B

Footnotes

The following table contains footnotes from the Properties tables that appear throughout this manual.

Number	FootNote
*1	The above values are just examples and may vary from implementation to implementation.
*2	The API does not perform any check on the validity of this attribute. However, once the <code>create</code> method is executed, the system will perform a check. The above values are just examples. The values may vary from implementation to implementation and the BillPlex administrator defines the values during the product installation.
*3	No particular edits are performed on this attribute. It is the client's responsibility to make sure that a valid attribute value was entered.
*4	The above values are the only allowed values by the current system release.
*5	The only date mask available by the API.
*6	The API does not perform any check on the validity of this attribute. However, once the <code>create</code> method is executed, the system will perform a check to determine if the attribute is defined in the BillPlex Data Repository and if the state belongs to the defined country. The above values are just examples. The values may vary from implementation to implementation and the BillPlex administrator defines the values during the product installation.
*7	When the <code>create</code> method is executed, the system automatically creates the ID by incrementing the identification of the last created. This attribute should be set before any methods are executed in this component.
*8	See <code>SessionMgr</code> for details.
*9	This attribute is populated only after the method <code>getFeatureTypeList</code> of the service type interface is executed.
*10	This attribute is populated only after the method <code>getParameterTypeList</code> (if defined) is executed or during the assignment of a parameter type to a feature type.

Limitations



Limitations

The following table documents the different lists that can be retrieved and the maximum number of members that each list can hold.

Component	List	Max
DTAConnectionNumberType	ConnectionNumberList	50
DTAConnectionType	ConnNumTypList	50
DTAFeature	ParameterList	50
	RatePlanList	50
DTAFeatureType	ParameterTypeList	50
	RatePlanTypeList	50
DTAMarketType	ServiceTypeList	50
DTAService	FeatureList	50
	ServiceConnectionList	50
DTAServiceBundle	ServiceTypeList	50
DTAServiceType	FeatureTypeList	50
	ConnectionTypeList	50
DTASubscriber	ServiceList	50
	SubscriberCommentList	50
	TransactionList	150
	ServiceOrderList	50
	TaxPlanList	50
	CharacteristicList	50
	InvoiceList	50

