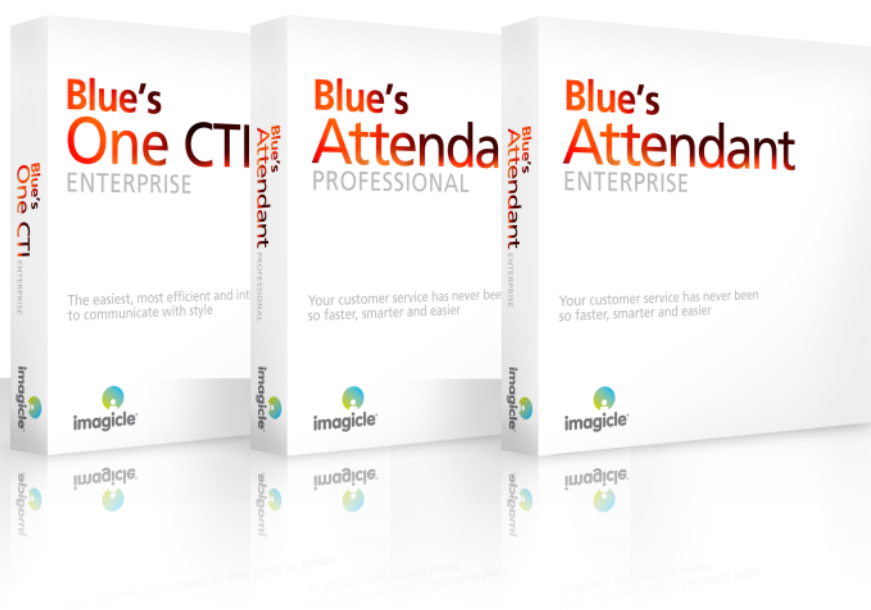




Blue's One CTI Enterprise Blue's Attendant Pro/Enterprise SDK - plugin creating tools

rel. 1.2 ITA 01-06-2012



Introduzione alla creazione del plug-in

Questo documento ha lo scopo di illustrare come creare un plug-in di ricerca da utilizzare con Blue's One CTI Enterprise, Blue's Attendant Professional ed Blue's Attendant Enterprise (da ora in poi Blue's One CTI - Attendant).

Il progetto che si va a realizzare dovrà essere composto da una o più dll a seconda dei casi. La dll che implementerà l'interfaccia ISearch di cui dopo daremo ampia spiegazione dovrà chiamarsi con un nome con la seguente forma:

Ricerca<nome plug-in>.dll

Assieme al presente documento verrà fornito un progetto generico sul quale sarà possibile avviare la creazione del proprio plug-in.

Per lo sviluppo del plug-in è richiesto come requisito l'utilizzo di Microsoft .NET Framework 2.0.

Le dll Plug-in vanno salvate nella cartella di installazione del programma sotto la cartella **SearchPlugin** (in genere {programmi}\Imagicle Blue's Attendant\SearchPlugin)

Contenuti

Le strutture	3
L'interfaccia del plug-in.....	5
Eventi di comunicazione del plug-in	8
La form delle opzioni	8
La nazionalizzazione delle risorse	9
Avvisi informativi per l'utente	9
La classe TelcenLog.....	9

Le strutture

L'unica struttura utilizzata che viene passata dal programma principale al plug-in e viceversa è la struttura `Contact` che altro non è che un contenitore di informazioni relative ad un contatto di una rubrica.

Esistono inoltre due enumeratori; `PhoneNumberType` per la definizione del tipo di telefono e `LanguageOption` per la lingua da utilizzare per la nazionalizzazione del plug-in.

La struttura `Contact` ha la seguente forma:

```
public struct Contact
{
    public bool bEmptyRecord;
    public Icon m_oIcon;

    public String sContactID;
    public String sIDRubrica;
    public String sContact;
    public String sPhoneNumber;
    public String sCompany;

    public String sMailAddress;
    public String sMailAddress1;
    public String sMailAddress2;

    public String sCustomType;
    public PhoneNumberType oNumType;

    public void InitializeContactStruct()
}
```

All'interno della struttura `Contact` è presente una funzione che ha lo scopo di inizializzare la struttura e che dovrà essere chiamata ogni qual volta viene creato un oggetto di tipo `Contact`.

Le variabili contenute all'interno della struttura sono:

- `bool` `bEmptyRecord`:
Flag che indica se il record contiene dati validi o meno. Di default è uguale a false.
- `Icon` `m_oIcon`:
Icona associata al contatto che verrà mostrata nella lista di ricerca contatti. Le sue dimensioni devono essere 32x32 con 256 colori.
- `String` `sIDRubrica`:
ID univoco della rubrica creata. Una stringa contenente un valore di al massimo 6 caratteri e che verrà utilizzato per distinguere un plug-in dall'altro.
- `String` `sContactID`:
ID relativo ad uno specifico contatto. Questo valore verrà poi utilizzato per l'apertura della scheda
- `String` `sContact`:
Valore del contatto, in genere Nome e Cognome. Verrà utilizzato dal software formattato come viene passato dal plug-in. (di default vuoto)



- **String** sPhoneNumber:
Numero di telefono relativo al contatto (di default vuoto)
- **String** sCompany:
Nome dell'azienda relativa al contatto (di default vuota)
- **String** sMailAddress:
Indirizzo di mail primario relativo al contatto
- **String** sMailAddress1:
Indirizzo di mail secondario relativo al contatto
- **String** sMailAddress2:
Indirizzo di mail secondario relativo al contatto
- **String** sCustomType: Tipo di numero di telefono personalizzabile. Le voci standard sono Lavoro, Fax, Casa, Cellulare, Altro, Telefono. E' quindi possibile aggiungere un tipo custom in questa variabile.

La variabile **PhoneNumberType** della struttura deve essere impostata come PHN_TYPE_CUSTOM

- **PhoneNumberType** oNumType:
Tipo di numero di telefono. I tipi selezionabili sono definiti dall'enumeratore **PhoneNumberType** descritto di seguito.
- **public** InitializeContactStruct:
Funzione per l'inizializzazione della struttura

L'enumeratore PhoneNumberType:

Questo enumeratore definisce il tipo relativo ad un numero di telefono.

Può avere i seguenti valori:

- PHN_TYPE_HOME: Tipo di numero di telefono 'Casa'
- PHN_TYPE_BUSINESS: Tipo di numero di telefono 'Lavoro'
- PHN_TYPE_MOBILE: Tipo di numero di telefono 'Cellulare'
- PHN_TYPE_FAX: Tipo di numero di telefono 'Fax'
- PHN_TYPE_OTHER: Tipo di numero di telefono 'Altro'
- PHN_TYPE_GENERIC_PHON: Tipo di numero di telefono 'Telefono'
- PHN_TYPE_CUSTOM: Tipo di telefono custom. Se impostato il tipo di telefono diventa il valore contenuto nella variabile sCustomType
- PHN_TYPE_SDIALER: Tipo di numero di telefono "

L'enumeratore LanguageOption

Questo enumeratore serve ad indicare quale lingua utilizzare per la nazionalizzazione del plug-in.

Può avere i seguenti valori:

- IT: Lingua Italiana
- EN: Lingua Inglese
- ES: Lingua Spagnola
- FR: Lingua Francese
- DE: Lingua Tedesca

L'interfaccia del plug-in

Il plug-in deve implementare l'interfaccia **ISearch** le cui funzioni sono sotto elencate e che nel loro insieme svolgono il compito di ricerca di contatti nella rubrica, ricerca di un numero nella rubrica, apertura della scheda contatto.

Le funzioni dell'interfaccia sono:

- **bool** IsLoading;
- **bool** InitializeComponent(**String** sFileIni, **String** sLogFolder, **bool** bEnableLog, **LanguageOption** oLanguage);
- **void** EnableLog(**bool** bEnable);
- **void** SetPlugInLanguage(**LanguageOption** oLanguage);
- **bool** CloseComponent();
- **void** SetMaxResultCount(**int** iMaxResult);
- **bool** ReloadContactData();
- **String** GetSearchDescription();
- **Icon** GetPhoneBookIcon();
- **Icon** GetPhoneBookIconForOption();
- **Control** GetOptionControlForm(**DelegateSettingsChanged** oDelegate, **LanguageOption** oLanguage);
- **String** GetPhoneBookID();
- **String** GetOutgoingPrefix();
- **bool** SaveOptionSettings();
- **Contact** FindContact(**String** sPhoneNumero);
- **List<Contact>** FindContact(**String** sContactName, **DelegateProcessContact** oHwnWnd);
- **bool** OpenContactCard(**DelegateContactAdd** oDelegateAddContact, **Contact** oContact);
- **bool** IsEnabled(**ref int** iNumeroRubricheAttive);

Di seguito vedremo in dettaglio l'utilizzo delle varie funzioni.

bool IsLoading

- **Descrizione:** Proprietà che indica se il plug-in è in fase di inizializzazione o meno.
- **Restituisce:** **bool** che indica lo stato del plug-in
- **Parametri:** Nessuno

bool InitializeComponent(**String** sFileIni, **String** sLogFolder, **bool** bEnableLog, **LanguageOption** oLanguage)

- **Descrizione:** Funzione di inizializzazione del plug-in. Viene chiamata dal programma alla partenza.
- **Restituisce:** Un **bool** che indica l'esito della funzione
- **Parametri:**
 - **String** sFileIni: File ini completo di path dove dovranno essere lette e salvate le impostazioni relative
 - **String** sLogFolder: Cartella dove verranno salvati eventuali log relativi al plug-in
 - **bool** bEnableLog: Flag che indica se abilitare o meno la scrittura del file di log
 - **LanguageOption** oLanguage: Parametro che indica la lingua da utilizzare all'interno del plug-in

void EnableLog(**bool** bEnable)

Descrizione: Funzione che abilita o meno, in base al flag passato, la scrittura del file di log

Restituisce: Nulla

Parametri: **bool** bEnable: Flag che imposta come abilitato o meno la scrittura del file di Log

void SetPlugInLanguage(**LanguageOption** oLanguage)

Descrizione: Funzione che imposta la lingua da utilizzare all'interno del Plug-in

Restituisce: Nulla

Parametri: **LanguageOption** oLanguage: Tipo di lingua da utilizzare

bool CloseComponent()

Descrizione: Funzione chiamata quando viene chiuso Blue's One CTI - Attendant. Serve al plug-in nell'eventualità per chiudere eventuali thread aperti oppure deallocare oggetti allocati durante il normale funzionamento.

Restituisce: Un **bool** che indica l'esito della funzione

Parametri: Nessuno

void SetMaxResultCount(int iMaxResult)

Descrizione: Funzione che imposta il limite massimo di record che il plug-in può restituire ad ogni ricerca. Se non impostata verrà utilizzata una costante interna al plugin come limite massimo

Restituisce: Un **bool** che indica l'esito della funzione

Parametri: **int** iMaxResult: Il numero massimo di record che durante una ricerca di un contatto il plug-in può restituire.

bool ReloadContactData()

Descrizione: Funzione che all'occorrenza serve a far sì che il plug-in ricarichi o aggiorni eventuali strutture contenenti all'interno la lista dei contatti. Ritorna **true** subito se non viene utilizzata.

Restituisce: Un **bool** che indica l'esito della funzione

Parametri: Nessuno

String GetSearchDescription()

Descrizione: Funzione che restituisce il nome della rubrica a cui il Plug-in si connette. Tale nome verrà visualizzato all'interno di un'etichetta di un Tab all'interno della sezione Rubriche del menù Opzioni di Blue's One CTI - Attendant.

Restituisce: Una **String** contenente il nome della rubrica che verrà poi mostrata come etichetta di un Tab all'interno delle Opzioni di Blue's One CTI - Attendant nella sezione di configurazione delle Rubriche

Parametri: Nessuno

Icon GetPhoneBookIcon()

Descrizione: Funzione che restituisce l'icona che verrà associata a tutti i contatti restituiti dal plug-in e poi mostrata all'interno delle varie liste di ricerca

Restituisce: Una **Icon** di dimensioni 32x32 e 256 colori

Parametri: Nessuno

Icon GetPhoneBookIconForOption()

Descrizione: Funzione che restituisce l'icona che verrà aggiunta come nuova sezione all'interno della sezione Opzioni di Blue's One CTI - Attendant.

Restituisce: Una **Icon** di dimensioni 48x48, **null** se si vuole far sì che il plug-in venga visualizzato all'interno della sezione Rubriche nel menù Opzioni.

Parametri: Nessuno

Control GetOptionControlForm(DelegateSettingsChanged oDelegate,LanguageOption oLanguage)

Descrizione: Funzione che restituisce il form control che dovrà contenere gli eventuali settaggi relativi al plug-in creato. Il settaggio minimo che deve essere creato nella form è una Checkbox che ha lo scopo di abilitare o disabilitare la ricerca tramite il plug-in. La dimensione del controllo non dovrà superare la dimensione di 350x300. I parametri della configurazione dovranno essere salvati in una Section relativa al plug-in e quindi non ancora presente nel file ini passato al plug-in al momento dell'inizializzazione

Restituisce: Un **Control** che al suo interno contiene tutte le impostazioni ed i settaggi necessari al funzionamento del Plug-in. La dimensione del **Control** deve essere al massimo 350x300.

Parametri:

- **DelegateSettingsChanged** oDelegate: Evento utilizzato per indicare all'applicazione principale che all'interno della form di gestione delle opzioni è stata fatta una modifica da parte dell'utente.
- **LanguageOption** oLanguage: Lingua da utilizzare per la nazionalizzazione della form di setup del plugin.

String GetPhoneBookID()

Descrizione: Funzione che restituisce l'ID univoco della rubrica. Questo ID serve al software principale per distinguere un plug-in dall'altro.

Restituisce: Una [String](#) che indica un codice univoco che rappresenta la rubrica. La stringa sarà al massimo di 5 caratteri

Parametri: Nessuno

String GetOutgoingPrefix()

Descrizione: Questa funzione restituisce il codice di connessione da anteporre al numero da chiamare. E' vuoto questo codice se si vuole che venga anteposto il codice di connessione di default. Il valore -1 indica che non devono essere aggiunti codici di connessione al numero di telefono.

Restituisce: Una [String](#) che indica il codice di uscita da anteporre al numero telefonico da chiamare.

Parametri: Nessuno

bool SaveOptionSettings()

Descrizione: Questa funzione viene chiamata quando vengono salvate le impostazioni delle Rubriche modificate dall'utente dal menù Opzioni. Il salvataggio deve essere fatto nell'apposita Section del file ini passato all'inizio al plug-in tramite la funzione Inizializza.

Restituisce: Un bool che indica l'esito della procedura di salvataggio delle impostazioni.

Parametri: Nessuno

Contact FindContact(String sPhoneNumero)

Descrizione: Questa funzione viene chiamata nel momento che il software ha necessità di associare una chiamata entrante oppure uscente ad un contatto. NOTA: la variabile sPhoneNumero passata dalla funzione rappresenta un numero telefonico che può essere anche nella forma "+390584123456" oppure "00390584123456". E' quindi necessaria la gestione del codice nazione durante la ricerca del numero all'interno delle rubriche.

Restituisce: Una struttura [Contact](#) che contiene i vari parametri relativi al contatto trovato all'interno della rubrica. Se non è stato trovato alcun contatto nella rubrica il flag bEmptyRecord della struttura [Contact](#) è uguale a false.

Parametri: [String](#) sPhoneNumero: Numero di telefono da cercare

List<Contact> FindContact(String sContactName, DelegateProcessContact oHwnWnd) :

Descrizione: Questa funzione ha lo scopo di cercare tutti i contatti all'interno della rubrica che contengono all'interno dei campi (ove presenti) nome, numero di telefono, azienda il valore passatomi sContactName. Ogni qual volta che si andrà a popolare la lista di Contact da restituire è necessario chiamare l'evento [DelegateProcessContact](#) illustrato nella sezione 'Eventi del plug-in'. Questo evento che prende come parametro un Contact ha lo scopo di verificare se dall'interfaccia principale l'utente ha interrotto la ricerca. Per questo se l'esito della chiamata dell'evento [DelegateProcessContact](#) è uguale a [true](#) vuol dire che la ricerca è stata interrotta e la funzione restituirà la lista parziale con i contatti fino a li trovati.

Restituisce: Una [List<Contact>](#). Una lista di strutture [Contact](#) che contengono tutti i contatti che corrispondono nel campo nome, numero telefono, azienda al valore passatomi sContactName.

Parametri:

- [String](#) sContactName: Nome del contatto da ricercare nella rubrica
- [DelegateProcessContact](#) oHwnWnd: Evento passato dall'applicazione e chiamato ogni qual volta dal plug-in viene trovato un elemento corrispondente. Se quando chiamato ritorna [true](#) significa che la ricerca è stata interrotta e quindi non devo cercare oltre o restituire altri record anche se questi sono già stati trovati.

bool OpenContactCard(DelegateContactAdd oDelegateAddContact, Contact oContact)

Descrizione: Funzione che permette l'apertura della scheda relativa al contatto passato. All'interno della struttura [Contact](#) deve essere presente l'ID che permetterà di aprire la scheda relativa al contatto passato.

Restituisce: Un bool che indica l'esito dell'apertura della scheda del contatto.

Parametri:

- DelegateContactAdd oDelegateAddContact : Evento chiamato dopo l'inserimento di un nuovo contatto (NON utilizzato)

- Contact oContact: La struttura Contact che contiene i vari parametri relativi al contatto di cui dobbiamo aprire la scheda

bool IsEnabled(ref int iNumeroRubricheAttive)

Descrizione: Questa funzione serve per il controllo sull'abilitazione del plug-in di ricerca. Come ulteriore informazione restituisce il numero di rubriche attive all'interno del plug-in. Questo perché un plug-in potrebbe contenere l'integrazione con più di una rubrica.

Restituisce: L'esito della funzione che indica se la rubrica è attiva oppure no. Per riferimento restituisce il numero di rubriche attive; questo perché in un plug-in ci possono essere ricerche in più rubriche.

Parametri: ref int iNumeroRubricheAttive: Numero di rubriche attive restituite dalla funzione.

Eventi di comunicazione del plug-in

All'interno del plug-in sono presenti tre delegate per la gestione di eventi da comunicare all'applicazione principale. Solo due di questi eventi vengono gestiti.

Gli eventi presenti sono:

- public delegate bool DelegateProcessContact(Contact oContactFound)
Viene chiamato ogni qual volta il plug-in trova un contatto da restituire durante la procedura di ricerca di un contatto. Questo serve per poter interrompere dal software principale l'operazione di ricerca. Ritorna true se è stata richiesta l'interruzione della procedura di ricerca del contatto.
- public delegate void DelegateSettingsChanged()
Viene chiamato quando dall'interfaccia delle impostazioni all'interno del plug-in l'utente modifica dei valori. L'evento ha lo scopo di comunicare avvenuti cambiamenti all'applicazione principale.
- public delegate void DelegateContactAdd(Contact oNewContactAdd);
Non è attualmente disponibile per l'integrazione di nuovi plug-in.

La form delle opzioni

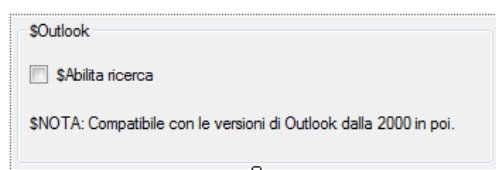
Il plug-in di ricerca dovrà contenere al suo interno anche una form che visualizzerà le impostazioni necessarie al corretto funzionamento e che l'utente avrà la possibilità di modificare dal menù Opzioni nella sezione Rubriche.

La form implementata dovrà essere di tipo [UserControl](#) con dimensioni non superiori a 350x300 . La nazionalizzazione della form è a carico del plug-in e dovrà utilizzare la lingua passata al momento della chiamata della funzione `GetOptionControlForm` da Blue's One CTI - Attendant.

Ogni modifica alle impostazioni che l'utente eseguirà all'interno della form, dovrà essere seguito dalla chiamata al messaggio [DelegateSettingsChanged](#), delegate passato al plug-in dalla chiamata `GetOptionControlForm`.

La form delle opzioni del plug-in dovrà contenere come minimo una Checkbox che dovrà permettere all'utente di abilitare o meno la ricerca attraverso il plug-in.

La forma grafica dovrà essere sullo stile dell'immagine riportata di seguito:



La nazionalizzazione delle risorse

La nazionalizzazione delle risorse è totalmente a carico del plug-in; sia per quanto riguarda la form delle opzioni sia per eventuali messaggi di avviso per l'utente. Le risorse dovranno essere contenute all'interno del file EmbeddedResource.resx all'interno del plug-in e già presente all'interno del progetto struttura allegato alla documentazione. Per la nazionalizzazione si consiglia di utilizzare degli ID di risorsa seguendo il seguente schema:

```
ID_LBL_RISORSA_IT  
ID_LBL_RISORSA_EN  
ID_LBL_RISORSA_FR  
ID_LBL_RISORSA_DE  
ID_LBL_RISORSA_ES
```

Utilizzando poi l'informazione sulla lingua passata da Blue's One CTI - Attendant durante l'inizializzazione del Plug-in sarà possibile selezionare la corretta risorsa nazionalizzata in base alla lingua.

Avvisi informativi per l'utente

All'interno del plug-in possono essere presenti avvisi informativi visualizzabili per l'utente ma devono essere assolutamente non bloccanti in nessun modo. Non è quindi permesso utilizzare MessageBox per avvisi di nessun tipo.

Per avvisare l'utente di eventuali malfunzionamenti del plug-in o di qualsiasi altra comunicazione che si ritiene necessaria è permesso il solo utilizzo di Baloon ToolTip. La comunicazione verrà visualizzata tramite questo strumento.

La classe TelcenLog

La struttura di ogni singolo evento salvato all'interno del file di log dovrà essere del seguente tipo:

```
|<DATA> <ORA> | <TIPO LOG> | <CLASSE> | <FUNZIONE> | <MESSAGGIO> | <CODICE ERRORE>|
```

Dove TIPO LOG può avere i seguenti valori: INFO, DEBUG, WARNING, ERROR, FATAL, GENERIC.

Il nome del file di log è scelto dal creatore del plug-in ma deve avere una estensione .log

La cartella di salvataggio del file di log e l'abilitazione alla scrittura o meno di tale file sono dati che verranno forniti nella fase di inizializzazione del plug-in tramite la funzione InitializeComponent.

All'interno del progetto allegato al documento è presente una classe che gestisce la scrittura del log con il metodo sopra specificato.

L' utilizzo tipico della classe è il seguente:

```
//Inizializzo l'oggetto  
TelcenLog m_oLog = new TelcenLog();  
m_oLog.EnableLog(true);  
m_oLog.SetClassName("Class Name");  
m_oLog.SetLogFile(@"c:\FileLog.log");  
m_oLog.SetLogMode(LogType.DEBUG_LOG);  
  
//Scrivo un messaggio di log  
m_oLog.Log(LogType.DEBUG_LOG, "Function Name", "Message to log");  
  
//Scrivo il log di una eccezione  
m_oLog.LogException(ex);
```