

## **SERVER**

## Introduzione

Il motore server di ViaDeiMercanti è un componente ActiveX che mette a disposizione del webmaster i seguenti oggetti:

- Generale
- Catalogo
- Carrello
- Ordine
- Ricerca
- Forum
- Contenuti
- Chat
- Banner
- Area riservata
- Aggiornamento

Ciascun oggetto espone metodi e proprietà che consentono di realizzare rapidamente e con poca esperienza tutte le funzionalità di un sito di commercio elettronico. Gli oggetti messi a disposizione infatti, gestiscono in maniera totalmente automatica le diverse problematiche: lingue, valute, archivio prodotti (taglie, colori, varianti, prodotti correlati), carrello virtuale, spese di spedizione (in funzione del peso della merce ordinata), esenzione IVA, dati cliente, interfacciamento al gateway per la validazione della carta di credito, invio notifiche via posta elettronica. Il motore server di fatto costituisce l'interfaccia tra il webmaster e la struttura dati sottostante che consente di realizzare le diverse funzionalità restituendo i dati in maniera semplice ed efficace. Questo consente allo sviluppatore di concentrarsi sulla realizzazione del sito appoggiandosi alle funzioni del motore server per l'implementazione delle funzionalità standard.

Ulteriore compito particolarmente importante è la gestione del dialogo con l'applicativo client di ViaDeiMercanti. L'oggetto *Aggiornamento* infatti espone i metodi necessari al client per sincronizzare il proprio archivio con quello presente sul server (download e upload dei dati). Quest' aspetto è particolarmente importante perché l'applicativo client ed il motore server insieme costituiscono uno strumento molto potente che risponde sia alle esigenze del webmaster che può sviluppare il sito in maniera rapida e veloce, sia alle esigenze dell'operatore che dovrà mantenere aggiornati i dati relativi al negozio on line.

Sebbene i modelli grafici di ViaDeiMercanti sono realizzati tutti seguendo il medesimo schema, in realtà il motore server consente una grande flessibilità permettendo di creare siti la cui struttura può essere completamente differente. Nel seguito di questo manuale, si farà riferimento allo schema dei modelli grafici preconfezionati per fornire esempi di utilizzo del motore server.

**Oggetti:**

- **Generale**
- **Catalogo**
- **Carrello**
- **Ordine**
- **AreaRiservata**
- **Dizionario**

## Proprietà oggetto Generale

### Dati aziendali esercente

- **RagioneSociale** (String)
- **Indirizzo** (String)
- **Cap** (String)
- **Localita** (String)
- **Provincia** (String)
- **Stato** (String)
- **TelefonoInfo** (String)
- **TelefonoSupporto** (String)
- **TelefonoOrdini** (String)
- **Fax** (String)
- **PartitaIVA** (String)
- **RegistroImprese** (String)
- **EmailInfo** (String) - a questo indirizzo vengono inviati i messaggi provenienti dalla pagina *Contattaci* dei modelli grafici di ViaDeiMercanti.
- **EmailSupporto** (String)
- **EmailOrdini** (String) - a questo indirizzo vengono inviate le notifiche di ordini effettuati sul sito.
- **PercorsoLogoAzienda** (String) - contiene il percorso completo del logo aziendale. Può essere utilizzato in un tag img. Questa proprietà tiene conto della lingua selezionata.

### Dati generali

- **AttivaCategorie** (Bool) - TRUE se in configurazione si è scelto di abilitare la gestione delle categorie merceologiche.
- **AttivaNovita** (Bool) - TRUE se in configurazione si è scelto di abilitare la sezione Novità.
- **AttivaOfferte** (Bool) - TRUE se in configurazione si è scelto di abilitare la sezione Offerte.
- **AttivaDisponibilita** (Bool) - TRUE se in configurazione si è scelto di abilitare la visualizzazione della disponibilità prodotto.
- **AttivaIndirizzoDestinatario** (Bool) - TRUE se in configurazione si è scelto di abilitare la possibilità di specificare un indirizzo del destinatario della merce diverso da quello del cliente che effettua l'ordine (gestione regali).
- **AttivaRicerca** (Bool) - TRUE se in configurazione si è scelto di abilitare la sezione Ricerca.
- **AttivaMailingList** (Bool) - TRUE se in configurazione si è scelto di abilitare la sezione MailingList.
- **AttivaAreaRiservata** (Bool) - TRUE se in configurazione si è scelto di abilitare la sezione AreaRiservata.
- **AttivaPrezziIvaCompresa** (Bool) - TRUE se in configurazione si è scelto di mostrare sul sito i prezzi IVA compresa.

## Modalità di pagamento

- **AttivaBonifico** (Bool) - TRUE se in configurazione si è scelto di abilitare il bonifico bancario come modalità di pagamento.
- **Banca** (String) - l'intestazione della banca sulla quale si desidera ricevere il bonifico.
- **ABI** (String) - ABI della banca sulla quale si desidera ricevere il bonifico.
- **CAB** (String) - CAB della banca sulla quale si desidera ricevere il bonifico.
- **CCBancario** (String) - conto corrente sul quale si desidera ricevere il bonifico (insieme alle tre proprietà precedenti contiene le coordinate bancarie alle quali effettuare il bonifico).
- **AttivaCCPostale** (Bool) - TRUE se in configurazione si è scelto di abilitare il versamento in C/C Postale come modalità di pagamento.
- **Intestatarario** (String) - l'intestatario del C/C postale sul quale si desidera ricevere il versamento.
- **CCPostale** (String) - il numero di C/C postale sul quale si desidera ricevere il versamento.
- **AttivaContrassegno** (Bool) - TRUE se in configurazione si è scelto di abilitare il contrassegno come modalità di pagamento.
- **TipoSpeseContrassegno** (Integer) - vale 0 se le spese di contrassegno sono rappresentate da un importo fisso, 1 se sono calcolate in percentuale rispetto al totale ordine.
- **SpeseContrassegno** (Double) - contiene l'importo relativo alle spese di contrassegno se TipoSpeseContrassegno = 0, contiene la percentuale da calcolare sul totale ordine se TipoSpeseContrassegno = 1. Nel primo caso l'importo è indicato nella valuta assegnata alla lingua selezionata dall'utente.
- **SpeseContrassegnoVL** (Double) - come sopra nella valuta di listino (nel caso in cui TipoSpeseContrassegno = 0).
- **MinimoContrassegno** (Double) - contiene l'importo minimo da applicare alle spese di contrassegno se TipoSpeseContrassegno = 1. L'importo è indicato nella valuta assegnata alla lingua selezionata dall'utente.
- **MinimoContrassegnoVL** (Double) - come sopra nella valuta di listino.
- **AttivaCartaCredito** (Bool) - TRUE se in configurazione si è scelto di abilitare la carta di credito come modalità di pagamento.

## Lingue

- **CodiceLingua** (String) - codice della lingua selezionata dall'utente. Questa proprietà viene utilizzata per comporre i percorsi degli elementi grafici presenti sul sito. Infatti tali elementi vengono memorizzati in sottocartelle della cartella *images* il cui nome è uguale al codice lingua cui si riferiscono.
- **CodiceLinguaPredefinita** (String) - codice della lingua indicata come predefinita in configurazione. Il sito viene mostrato in questa lingua quando l'utente non ha ancora effettuato una scelta differente.
- **DescrizioneLingua** (String)
- **SimboloValuta** (String) - contiene il simbolo della valuta specificata per la lingua correntemente selezionata dall'utente.

- **AttivaPrezziInEuro** (Bool) - TRUE se si è scelto di visualizzare i prezzi anche in Euro quando l'utente visita il sito nella lingua corrente.

## Metodi oggetto generale

### Inizializzazione

- Inizializza
- IdentificaUtente
- ImpostaLunghezzaPaginaProdotti
- ImpostaLunghezzaPaginaRicerca
- ImpostaLunghezzaListaOrdini
- ImpostaLunghezzaPaginaArticoli
- ImpostaLunghezzaPaginaDiscussioni
- ImpostaLunghezzaPaginaRisposte
- ImpostaLunghezzaPaginaMessaggi
- ImpostaDurataMessaggi
- ImpostaFrequenzaAggiornamentoChat
- ImpostaLunghezzaCodaChat
- ImpostaLunghezzaPaginaRiordino
- ImpostaLunghezzaElencoClientiAgente
- ImpostaURLPaginaSezioni
- ImpostaURLPaginaArgomentiForum
- ImpostaURLPaginaArgomentiChat

### Inizializza()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** inizializza la sessione utente leggendo da database tutte le informazioni di configurazione. Imposta opportunamente tutte le proprietà dell'oggetto Generale. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### IdentificaUtente()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** legge ed imposta i cookies per identificare l'utente che si collega al sito. Consente di visualizzare nel modulo d'ordine i dati utente memorizzati in un ordine precedente. Consente inoltre di ricavare dati statistici dall'analisi dei files di log del web server.

**Esempi:** vedi default.asp nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaPaginaProdotti(LPP)

**Input:** numero di prodotti desiderati per ogni pagina di indice.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire l'indice prodotti leggendo il numero di prodotti specificato. Tale valore viene in particolare utilizzato dal metodo ApriPaginaProdotti. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaPaginaRicerca(LPP)

**Input:** numero di risultati desiderati per ogni pagina di ricerca.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire la pagina ricerca leggendo il numero di risultati specificato. Tale valore viene in particolare utilizzato dal metodo ApriPaginaRicerca. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaListaOrdini(LLO)

**Input:** numero di ordini desiderati per ogni pagina di indice.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire la lista ordini prodotti leggendo il numero di ordini specificato. Tale valore viene in particolare utilizzato dal metodo ApriListaOrdini. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

---

### ImpostaLunghezzaPaginaArticoli(PS)

**Input:** numero di articoli desiderati per ogni pagina di articoli.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire la pagina articoli leggendo il numero di articoli specificato. Tale valore viene in particolare utilizzato dal metodo ApriPaginaArticoli dell'oggetto contenuti. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaPaginaDiscussioni(PS)

**Input:** numero di discussioni desiderate per ogni pagina di Discussioni.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire la pagina delle discussioni leggendo il numero di discussioni specificato. Tale valore viene in particolare utilizzato dal metodo ApriPaginaDiscussioni dell'oggetto Forum. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaPaginaRisposte(PS)

**Input:** numero di risultati desiderati per ogni pagina di Risposte a discussioni del Forum.

**Output:** nessuno.

**Commenti:** predispose il motore server a gestire la pagina delle risposte leggendo il numero di interventi specificato. Tale valore viene in particolare utilizzato dal metodo ApriPaginaRisposte dell'oggetto Forum. Normalmente questo metodo viene invocato nella funzione Session\_OnStart del file global.asa.

**Esempi:** vedi global.asa nei modelli grafici di ViaDeiMercanti.

### ImpostaLunghezzaPaginaMessaggi(PS)

**Input:** numero di risultati desiderati per ogni pagina di Messaggi.

**Output:** nessuno.

**Commenti:** predisporre il motore server a gestire la pagina dei messaggi leggendo il numero di interventi specificato. Tale valore viene in particolare utilizzato dal metodo `ApriPaginaInterventi` dell'oggetto `Chat`. Normalmente questo metodo viene invocato nella funzione `Session_OnStart` del file `global.asa`.

**Esempi:** vedi `global.asa` nei modelli grafici di `ViaDeiMercanti`.

### ImpostaDurataMessaggi (PS)

**Input:** Numero di Secondi per i quali un messaggio deve permanere nella chat

**Output:** Nessuno

**Commenti:** I messaggi vengono cancellati automaticamente quando sono trascorsi i secondi impostati con questo metodo.

### ImpostaFrequenzaAggiornamentoChat (PS)

**Input:** Numero di secondi, che trascorrono, tra un refresh e l'altro della pagina dei messaggi relativi alla chat. Più si diminuisce questo parametro, maggiore sarà l'impressione dell'utente, di trovarsi in una Chat aggiornata in tempo reale. Un valore troppo basso, tuttavia, aumenterebbe il numero dei refresh generando maggiore traffico di rete.

**Output:** Nessuno

**Commenti:** Il metodo imposta la proprietà `FrequenzaAggiornamento` disponibile nell'oggetto `Chat`.

### ImpostaLunghezzaPaginaRiordino (PS)

**Input:** Numero di prodotti contenuti nella pagina del riordino.

**Output:** Nessuno

**Commenti:** predisporre il motore server a gestire la pagina ricerca leggendo il numero di risultati specificato. Tale valore viene in particolare utilizzato dal metodo `ApriPaginaRicerca`. Normalmente questo metodo viene invocato nella funzione `Session_OnStart` del file `global.asa`.

### ImpostaLunghezzaElencoClientiAgente (PS)

**Input:** numero di clienti desiderati per ogni pagina di elenco clienti relativi ad un agente.

**Output:** nessuno.

**Commenti:** predisporre il motore server a gestire la pagina dei messaggi leggendo il numero di risultati specificato. Tale valore viene in particolare utilizzato dal metodo `ApriElencoClientiAgente` dell'oggetto `AreaRiservata`. Normalmente questo metodo viene invocato nella funzione `Session_OnStart` del file `global.asa`.

**Esempi:** vedi `global.asa` nei modelli grafici di ViaDeiMercanti.

### ImpostaURLPaginaSezioni (URL)

**Input:** URL = Indirizzo Url della pagina sezioni

**Output:** Nessuno

**Commenti:** l'Url impostato con questo metodo viene usato dalla proprietà **PercorsoSezioni** dell'oggetto contenuti al fine di restituire la barra di navigazione delle sezioni.

### ImpostaURLPaginaArgomentiForum (URL)

**Input:** URL = Indirizzo Url della pagina argomenti forum.

**Output:** Nessuno

**Commenti:** l'Url impostato impostato con questo metodo viene usato dalla proprietà **PercorsoArgomenti** dell'oggetto forum al fine di restituire la barra di navigazione degli argomenti del forum.

### ImpostaURLPaginaArgomentiChat (URL)

**Input:** URL = Indirizzo Url della pagina argomenti chat.

**Output:** Nessuno

**Commenti:** L'Url impostato impostato con questo metodo viene usato dalla proprietà **PercorsoArgomenti** dell'oggetto Chat al fine di restituire la barra di navigazione degli argomenti della Chat.

## Lingue

- **LeggiPrimaLingua**
- **LeggiLinguaSuccessiva**
- **ImpostaLingua**

### LeggiPrimaLingua()

**Input:** nessuno.

**Output:** FALSE se non ci sono lingue attive per il sito. TRUE altrimenti.

**Commenti:** legge la prima lingua attiva per il sito. Unitamente al metodo LeggiLinguaSuccessiva serve a scorrere tutte le lingue attive per creare la lista di selezione sulla home page.

**Esempi:** vedi default.asp nei modelli grafici di ViaDeiMercanti.

### LeggiLinguaSuccessiva()

**Input:** nessuno.

**Output:** FALSE se non ci sono ulteriori lingue attive per il sito (la lingua corrente è l'ultima). TRUE altrimenti.

**Commenti:** legge la successiva lingua attiva per il sito. Unitamente al metodo LeggiPrimaLingua serve a scorrere tutte le lingue attive per creare la lista di selezione sulla home page.

**Esempi:** vedi default.asp nei modelli grafici di ViaDeiMercanti.

### ImpostaLingua()

**Input:** legge il CodiceLingua da attivare direttamente tramite metodo POST o GET.

**Output:** nessuno.

**Commenti:** imposta come lingua di visualizzazione quella il cui codice è passato tramite metodo GET o POST. Normalmente la selezione avviene sulla home page tramite un URL tipo: <http://www.nomesito.com/default.asp?CodiceLingua=uk>. Il metodo imposta anche la corretta valuta.

**Esempi:** vedi default.asp nei modelli grafici di ViaDeiMercanti.

## Chi siamo

- **LeggiChiSiamo**
- **ControlloLogoAzienda**

### LeggiChiSiamo()

**Input:** nessuno.

**Output:** stringa vuota se il testo *Chi siamo* non è stato inserito in archivio. Il testo nella lingua corrente, se esiste, altrimenti nella lingua predefinita.

**Esempi:** vedi chisiamo.asp nei modelli grafici di ViaDeiMercanti.

### ControlloLogoAzienda()

**Input:** nessuno.

**Output:** TRUE se esiste il logo aziendale per la lingua selezionata o per la lingua predefinita. FALSE se non esiste per alcuna lingua.

**Commenti:** il valore restituito da questo metodo può essere utilizzato per decidere se mostrare il logo aziendale. Dopo aver invocato questo metodo, la proprietà `PercorsoLogoAzienda` contiene il percorso del logo aziendale completo di nome file. Questa proprietà potrà quindi essere utilizzata all'interno di un tag `img`.

**Esempi:** vedi chisiamo.asp nei modelli grafici di ViaDeiMercanti.

## Contattaci

- **InviaMessaggi**

### InviaMessaggi()

**Input:** tramite metodo POST questa funzione riceve i seguenti parametri: Mittente, Email, Messaggio (utilizzare i nomi esatti). E' inoltre possibile passare ulteriori campi il cui nome è a scelta dello sviluppatore (si veda paragrafo *Commenti*)

**Output:** nessuno.

**Commenti:** il metodo invia un messaggio di posta elettronica all'indirizzo specificato in archivio come Email info. Il corpo del messaggio è il seguente:

[Mittente] scrive:

Email: [Email]

...

[Messaggio]

Tra il rigo Email ed il messaggio digitato dall'utente vengono inseriti (uno per rigo) gli eventuali altri campi passati dallo sviluppatore secondo lo schema

NomeCampo: ValoreCampo

L'oggetto del messaggio viene letto dal Dizionario alla voce OGGETTOMAILINFO.

**Esempi:** vedi contattaci.asp nei modelli grafici di ViaDeiMercanti.

## Controllo sessioni

- SetOkSession
- TestOkSession

### SetOkSession()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** ViaDeiMercanti utilizza le Sessioni per memorizzare i dati relativi alla navigazione dell'utente sul sito. Le variabili di Sessione vengono automaticamente azzerate dal web server trascorso un determinato periodo di inattività da parte dell'utente (solitamente 20 minuti). E' necessario quindi controllare che la sessione iniziata dall'utente sia ancora attiva. Per inizializzare tale controllo deve essere invocato questo metodo. A questo scopo non è consigliabile utilizzare il file global.asa, in quanto allo scadere della sessione l'utente viene reindirizzato sulla home page (vedi metodo TestOkSession) che dovrebbe resettare il controllo chiamando appunto il metodo SetOkSession. Pertanto la home page risulta essere il punto migliore dove invocare questo metodo.

**Esempi:** vedi default.asp nei modelli grafici di ViaDeiMercanti.

### TestOkSession()

**Input:** nessuno.

**Output:** TRUE se la sessione corrente è ancora attiva, FALSE altrimenti.

**Commenti:** ogni pagina del sito dovrebbe invocare questo metodo per controllare se la sessione iniziata dall'utente è ancora attiva. In caso contrario dovrebbe reindirizzare l'utente alla home page. Per il corretto funzionamento del metodo è necessario invocare all'inizio della navigazione (nella home page) il metodo SetOkSession.

**Esempi:** vedi una qualsiasi pagina nei modelli grafici di ViaDeiMercanti.

## Proprietà oggetto Catalogo

Tutte le proprietà seguenti vengono gestite in base alla lingua selezionata. Qualora le informazioni relative alla lingua selezionata non fossero state inserite, il software fa riferimento alla lingua predefinita.

### Categorie

- **IDCategoria** (Long) - identificativo univoco della categoria.
- **NomeCategoria** (String)
- **DescrizioneCategoria** (String)
- **NumeroProdottiCategoria** (Long) - numero di prodotti appartenenti alla categoria corrente
- **FlagSottoCategorie** (Boolean) - indica se esistono sottocategorie per la categoria corrente.

### Prodotti

- **LunghezzaPaginaProdotti** (Integer) - numero di prodotti per pagina indice.
- **PaginaCorrenteProdotti** (Long) - numero della pagina indice corrente.
- **TotalePagineProdotti** (Long) - numero totale di pagine indice.
- **IDProdotto** (Long) - identificativo univoco del prodotto.
- **CodiceProdotto** (String)
- **NomeProdotto** (String) - in lingua.
- **DescrizioneFatturaProdotto** (String) - è la descrizione, in lingua, che compare al momento dell'ordine.
- **DescrizioneBreveProdotto** (String) - è una descrizione breve, in lingua, e può essere utilizzata per esempio in una pagina indice.
- **DescrizioneEstesaProdotto** (String) - è una descrizione estesa, in lingua, e può essere utilizzata per esempio in una scheda dettaglio del prodotto.
- **PrezzoInizialeVL** (Double)
- **PrezzoProdottoVL** (Double) - l'importo contenuto è ottenuto dall'applicazione di tutti i calcoli necessari (offerta, iva). Il prezzo è espresso nella valuta di listino.
- **PrezzoProdotto** (Double) - come precedente ma espresso nella valuta associata alla lingua scelta dall'utente.
- **PrezzoProdottoInEuro** (Double) - come precedente ma espresso in Euro.
- **PrezzoListino** (Double) - prezzo prodotto prima dell'applicazione di una eventuale offerta promozionale (viene espresso nella valuta selezionata dall'utente).
- **PrezzoListinoInEuro** (Double) - come precedente ma espresso in Euro.
- **SimboloValuta** (Long) - contiene il simbolo prescelto per la valuta nella quale è espresso il PrezzoProdotto.
- **UMProdotto** (String) - unità di misura.
- **ConfezioneProdotto** (Single) - Quantità di prodotti per ogni Confezione.
- **IVAProdotto** (Single) - aliquota IVA.

- **ImpostaProdottoVL** (Double) - importo dell'IVA sul prodotto. Questa proprietà contiene il valore corretto anche quando si sceglie di visualizzare i prezzi IVA inclusa. L'importo è espresso nella valuta di listino.
- **ImpostaProdotto** (Double) - come precedente ma espresso nella valuta associata alla lingua scelta dall'utente.
- **ImpostaProdottoInEuro** (Double) - come precedente ma espresso in Euro.
- **PesoProdotto** (Single) - peso prodotto in kg. Viene utilizzato per calcolare il peso totale del pacco e quindi le spese di spedizione.
- **DisponibilitaProdotto** (Long) - valore indicato in archivio prodotti.
- **Attivo** (Boolean) - indica se il prodotto deve essere pubblicato.
- **Novita** (Boolean) - indica se il prodotto è una novità.
- **Offerta** (Boolean) - indica se il prodotto è in offerta.
- **ValiditaScontoDal** (Date) - data di inizio dell'offerta.
- **ValiditaScontoAl** (Date) - data di fine dell'offerta.
- **ScontoProdotto** (Single) - percentuale di sconto applicata al prezzo di listino (gli importi calcolati nelle proprietà PrezzoProdottoVL, PrezzoProdotto e PrezzoProdottoInEuro sono già comprensivi di questo sconto).
- **NumeroTaglie** (Long) - numero di taglie disponibili per questo prodotto.
- **NumeroColori** (Long) - numero di colori disponibili per questo prodotto.
- **NumeroCampiCustom** (Long) - numero di campi custom per questo prodotto.
- **NomeCampoCustom** (String) - nome del campo custom letto dal metodo LeggiCampoCustom invocato passando come parametro l'indice del campo.
- **ValoreCampoCustom** (String) - valore del campo custom letto dal metodo LeggiCampoCustom.
- **FlagNovita** (Integer) - indica quali prodotti vengono mostrati dal Metodo ApriPaginaProdotti (0 = tutti i prodotti, -1 = solo prodotto non novità, 1 = solo prodotti novità).
- **FlagOfferte** (Integer) - indica quali prodotti vengono mostrati dal Metodo ApriPaginaProdotti (0 = tutti i prodotti, -1 = solo prodotto non in offerta, 1 = solo prodotti in offerta).

## Immagini

- **PercorsoFotoCategoria** (String) - dopo aver chiamato il metodo ControlloFotoCategorie questa proprietà contiene il percorso completo (incluso il nome file) dell'immagine relativa alla categoria corrente. E' pertanto possibile utilizzare tale proprietà all'interno di un tag img.
- **PercorsoFotoProdotto** (String) - dopo aver chiamato il metodo ControlloFotoProdotti questa proprietà contiene il percorso completo (incluso il nome file) della foto grande del prodotto. E' pertanto possibile utilizzare tale proprietà all'interno di un tag img.
- **PercorsoThumbnailProdotto** (String) - come PercorsoFotoProdotto per la foto piccola del prodotto.
- **PercorsoFotoColore** (String) - dopo aver chiamato il metodo ControlloFotoColore questa proprietà contiene il percorso completo (incluso il nome file) della foto grande del prodotto relativa al colore corrente impostato chiamando gli opportuni metodi (LeggiPrimoColore, LeggiColoreSuccessivo, LeggiColore). E' pertanto possibile utilizzare tale proprietà all'interno di un tag img.

- **PercorsoThumbnailColore** (String) - come PercorsoFotoColore per la foto piccola del colore prodotto.
- **PercorsoDettaglioColore** (String) - come PercorsoFotoProdotto per la foto del dettaglio colore del prodotto. E' tuttavia necessario chiamare il metodo ControlloDettaglioColore invece della funzione ControlloFotoColore.

## Files

- **PercorsoFileProdotto** (String) - dopo aver chiamato il metodo ControlloFileProdotti questa proprietà contiene il percorso completo (incluso il nome file) del file corrispondente al codice richiesto. E' pertanto possibile utilizzare tale proprietà per specificare il percorso di un download o all'interno di un opportuno tag.

## Colori

- **IDColoreProdotto** (Long) - identificativo univoco del colore prodotto.
- **CodiceColoreProdotto** (String)
- **DescrizioneColoreProdotto** (String)

## Sconti per quantità

- **FlagScontoQuantitaAttivo** (Boolean) - indica se sul prodotto corrente sono attivi sconti per quantità per l'utente attualmente collegato.
- **LimiteQuantita1** (Single) - valore limite per la prima fascia di sconto.
- **Sconto1** (String) - percentuale di sconto per la prima fascia.
- **PrezzoPromo1VL** (Double) - prezzo promozionale per la prima fascia espresso in valuta di listino.
- **PrezzoPromo1** (Double) - come precedente ma espresso nella valuta associata alla lingua scelta dall'utente.
- **LimiteQuantita2** (Single) - questa proprietà e le seguenti hanno lo stesso significato delle corrispondenti per la prima fascia.
- **Sconto2** (String)
- **PrezzoPromo2VL** (Double)
- **PrezzoPromo2** (Double)
- **LimiteQuantita3** (Single)
- **Sconto3** (String)
- **PrezzoPromo3VL** (Double)
- **PrezzoPromo3** (Double)
- **LimiteQuantita4** (Single)
- **Sconto4** (String)
- **PrezzoPromo4VL** (Double)
- **PrezzoPromo4** (Double)
- **LimiteQuantita5** (Single)

- **Sconto5** (String)
- **PrezzoPromo5VL** (Double)
- **PrezzoPromo5** (Double)
- **LimiteQuantita6** (Single)
- **Sconto6** (String)
- **PrezzoPromo6VL** (Double)
- **PrezzoPromo6** (Double)
- **LimiteQuantita7** (Single)
- **Sconto7** (String)
- **PrezzoPromo7VL** (Double)
- **PrezzoPromo7** (Double)
- **LimiteQuantita8** (Single)
- **Sconto8** (String)
- **PrezzoPromo8VL** (Double)
- **PrezzoPromo8** (Double)
- **LimiteQuantita9** (Single)
- **Sconto9** (String)
- **PrezzoPromo9VL** (Double)
- **PrezzoPromo9** (Double)

## Pagine

- **LunghezzaPaginaProdotti** (Integer) - numero di prodotti per pagina indice.
- **PaginaCorrenteProdotti** (Long) - numero della pagina indice corrente.
- **TotalePagineProdotti** (Long) - numero totale di pagine indice.

## Valori formattati

Per tutte le proprietà contenenti valori relativi ad importi, ne esiste una versione formattata con separatore delle migliaia. Ciascuna di queste proprietà ha il medesimo nome della corrispondente preceduto dalla lettera F. Per esempio del campo PrezzoProdotto=12345,67 esiste la versione FPrezzoProdotto=12.345,67.

## Metodi oggetto Catalogo

### Categorie

- **ApriPaginaCategorie**
- **LeggiCategoriaSuccessiva**
- **DisattivaCategorie**
- **ImpostaCalcoloNumeroProdottiCategoria**
- **ChiudiPaginaCategorie**

### ApriPaginaCategorie(IDCategoriaPadre)

**Input:** IDCategoriaPadre = identificativo della categoria di cui leggere le sottocategorie. Se tale parametro viene omesso, vengono lette le categorie di primo livello.

**Output:** FALSE se non ci sono categorie in archivio, TRUE altrimenti.

**Commenti:** inizializza la pagina delle categorie posizionandosi sulla prima in archivio.

**Esempi:** vedi catalogo.asp nei modelli grafici di ViaDeiMercanti.

### LeggiCategoriaSuccessiva()

**Input:** nessuno.

**Output:** FALSE se non ci sono categorie ulteriori categorie in archivio (la categoria corrente è l'ultima), TRUE altrimenti.

**Commenti:** legge la categoria successiva a quella corrente.

**Esempi:** vedi catalogo.asp nei modelli grafici di ViaDeiMercanti.

### DisattivaCategorie()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** se è stata attivata la gestione delle categorie merceologiche il metodo `ApriPaginaProdotti` inizializza una pagina indice prodotti per la categoria passata come parametro. Questo metodo disattiva temporaneamente la gestione delle categorie merceologiche consentendo di ottenere una pagina indice che riporti tutti i prodotti. E' utile, ad esempio, per ottenere una pagina di tutti i prodotti in novità a prescindere dalla categoria cui appartengono.

L'effetto del metodo è temporaneo in quanto influenza soltanto la prossima chiamata al metodo `ApriPaginaProdotti` e non le successive.

### ImpostaCalcoloNumeroProdottiCategoria(F)

**Input:**

- F = 1: viene attivato il calcolo del numero prodotti per categoria.
- F = 0: il calcolo viene disattivato.

**Output:** nessuno.

**Commenti:** poichè il calcolo dei prodotti per categoria può richiedere molte risorse, può essere utile disattivarlo nei casi in cui questo non sia necessario o gradito. L'impostazione selezionata invocando questo metodo rimane fissata fino ad un eventuale nuova impostazione. L'impostazione predefinita è F = 1 e pertanto a meno di reimpostazioni il calcolo viene effettuato.

### ChiudiPaginaCategorie()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude la pagina delle categorie.

**Esempi:** vedi `catalogo.asp` nei modelli grafici di ViaDeiMercanti.

## Prodotti

- **ImpostaPaginaNovita**
- **ImpostaPaginaOfferte**
- **ImpostaSelezioneCustom**
- **ApriPaginaProdotti**
- **LeggiPrimoProdotto**
- **LeggiProdottoSuccessivo**
- **LeggiProdotto**
- **LeggiCampoCustom**
- **ControlloPrimaPaginaProdotti**
- **ControlloUltimaPaginaProdotti**
- **ChiudiPaginaProdotti**
- **ApriListaProdottiCasuali**
- **LeggiProdottoCasualeSuccessivo**
- **ChiudiListaProdottiCasuali**

## ImpostaPaginaNovita(N)

### Input:

- $N = -1$ : saranno letti solo i prodotti non novità.
- $N = 1$ : saranno letti solo i prodotti novità.
- $N = 0$  saranno letti tutti i prodotti.

**Output:** nessuno.

**Commenti:** predisporre la lettura dell'archivio prodotti in base alla proprietà Novità.

**Esempi:** vedi catalogo.asp oppure prodotti.asp nei modelli grafici di ViaDeiMercanti.

## ImpostaPaginaOfferte(O)

### Input:

- $O = -1$  saranno letti solo i prodotti non in offerta.
- $O = 1$  saranno letti solo i prodotti in offerta
- $O = 0$  saranno letti tutti i prodotti.

**Output:** nessuno.

**Commenti:** predisporre la lettura dell'archivio prodotti in base alla proprietà Offerta. Per valutare i prodotti in offerta il motore server tiene conto della data di validità (data di inizio offerta precedente ad oggi e data di fine offerta seguente ad oggi).

**Esempi:** vedi catalogo.asp oppure prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ImpostaSelezioneCustom(SC)

**Input:**

- SC = testo di selezione. Vedi commenti.

**Output:** nessuno.

**Commenti:** questo metodo funziona solo se si è scelto un ordinamento per l'indice prodotti diverso da quello custom ossia per codice, nome, descrizione fattura, breve o estesa. In questo caso è possibile selezionare (attraverso i normali metodi ApriPaginaProdotti, LeggiProdottoSuccessivo) i soli prodotti il cui campo impostato per l'ordinamento inizia con la stringa passata come parametro a questo metodo.

Supponiamo quindi che per l'ordinamento si sia scelto il campo *NomeProdotto*. Invocando il metodo ImpostaSelezioneCustom passando la stringa "AM" vengono restituiti i soli prodotti il cui nome inizia per "AM".

### ApriPaginaProdotti()

**Input:** i parametri input vengono letti direttamente tramite metodo GET o POST (utilizzare il nome esatto sotto riportato).

- IDCategoria: identificativo univoco della categoria relativa ai prodotti da visualizzare. Se la gestione delle categorie merceologiche non è abilitata non è necessario passare questo parametro.
- Action: vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva.

**Output:** TRUE se esistono prodotti per la pagina, FALSE altrimenti.

**Commenti:** inizializza la pagina prodotti in base alle impostazioni definite dai metodi ImpostaPaginaNovita e ImpostaPaginaOfferte posizionandosi sul primo prodotto.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### LeggiPrimoProdotto()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** legge il primo prodotto della pagina corrente e si posiziona su di esso. Non è necessario chiamare tale metodo dopo ApriPaginaProdotti.

### LeggiProdottoSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono prodotti successivi (il prodotto corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge il prodotto successivo al prodotto corrente e si posiziona su di esso.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### LeggiProdotto(IDProdotto)

**Input:** IDProdotto = identificativo del prodotto da leggere. Se tale parametro viene omesso, il metodo leggerà il valore IDProdotto (utilizzare il nome esatto) tramite il metodo POST o GET.

**Output:** nessuno.

**Commenti:** legge il prodotto identificato dal parametro passato.

**Esempi:** vedi schedaprodotti.asp nei modelli grafici di ViaDeiMercanti.

### LeggiCampoCustom(W)

**Input:** il parametro W può essere sia il nome del campo custom da leggere sia il suo indice. In quest'ultimo caso il valore dell'indice deve essere compreso tra 1 ed il valore della proprietà NumeroCampiCustom.

**Output:** il valore del campo custom richiesto.

**Commenti:** il metodo restituisce sempre il valore del campo custom richiesto. Tale valore viene restituito anche nella proprietà ValoreCampoCustom. Se il metodo viene invocato passando l'indice del campo, viene impostata anche la proprietà NomeCampoCustom per la

visualizzazione. E' quindi possibile utilizzare il metodo senza conoscere a priori i campi impostati sul prodotto.

### ControlloPrimaPaginaProdotti()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è la prima. FALSE altrimenti.

**Commenti:** utile nella pagina di indice prodotti per abilitare il collegamento alla pagina precedente.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ControlloUltimaPaginaProdotti()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è l'ultima. FALSE altrimenti.

**Commenti:** utile nella pagina di indice prodotti per abilitare il collegamento alla pagina successiva.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ChiudiPaginaProdotti()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude la pagina prodotti.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ApriListaProdottiCasuali (IDCategoria)

**Input:** IdCategoria = Identificativo della categoria prodotti per la quale si intende aprire la lista casuale. Se viene omesso, vengono letti i prodotti di tutte le categorie.

**Output:** .TRUE se la lista risultante non è vuota, FALSE altrimenti.

**Commenti:** . Inizializza la pagina prodotti in base alle impostazioni definite dai metodi ImpostaPaginaNovita e ImpostaPaginaOfferte posizionandosi sul primo prodotto.

### LeggiProdottoCasualeSuccessivo()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** Legge il prodotto successivo al prodotto corrente e si posiziona su di esso. Se il prodotto corrente è l'ultimo della lista legge il primo della lista.

### ChiudiListaProdottiCasuali ()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude la lista prodotti casuali.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

## Immagini

- **ControlloFotoCategorie**
- **ControlloFotoProdotti**
- **ControlloFotoColore**
- **ControlloDettaglioColore**

### ControlloFotoCategorie()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per la categoria corrente. FALSE altrimenti.

**Commenti:** il metodo è utile per abilitare la visualizzazione dell'immagine categoria. Il metodo imposta anche le proprietà PercorsoFotoCategoria che conterrà il percorso completo (incluso il nome file) dell'immagine della categoria corrente. Questa proprietà può essere utilizzata all'interno del tag img.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ControlloFotoProdotti()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per il prodotto corrente. FALSE altrimenti.

**Commenti:** il metodo è utile per abilitare la visualizzazione dell'immagine prodotto. Il metodo imposta anche le proprietà PercorsoFotoProdotto e PercorsoThumbnailProdotto che conterranno il percorso completo (incluso il nome file) rispettivamente della foto grande e della foto piccola del prodotto corrente. Queste proprietà possono essere utilizzate all'interno del tag img.

**Esempi:** vedi prodotti.asp nei modelli grafici di ViaDeiMercanti.

### ControlloFotoColore()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per il colore corrente del prodotto corrente. FALSE altrimenti.

**Commenti:** il metodo è utile per abilitare la visualizzazione dell'immagine del colore corrente del prodotto corrente. Il metodo imposta anche le proprietà PercorsoFotoColore e PercorsoThumbnailColore che conterranno il percorso completo (incluso il nome file) rispettivamente della foto grande e della foto piccola del colore corrente del prodotto corrente. Queste proprietà possono essere utilizzate all'interno del tag img.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

### ControlloDettaglioColore()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per il dettaglio del colore corrente del prodotto corrente. FALSE altrimenti.

**Commenti:** il metodo è utile per abilitare la visualizzazione dell'immagine del dettaglio del colore corrente del prodotto corrente. Il metodo imposta anche le proprietà PercorsoDettaglioColore che conterrà il percorso completo (incluso il nome file) dell'immagine del dettaglio del colore corrente del prodotto corrente. Questa proprietà può essere utilizzata all'interno del tag img.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

## Files

- **ControlloFileProdotti**

### ControlloFileProdotti(Codice)

**Input:** Codice = identificativo attribuito dell'utente.

**Output:** TRUE se esiste un file per il prodotto corrente con il codice richiesto. FALSE altrimenti.

**Commenti:** il metodo consente di verificare l'esistenza di un determinato file associato al prodotto corrente e corrispondente al codice richiesto. Il metodo imposta la proprietà PercorsoFileProdotto che conterrà il percorso completo (incluso il nome file). Questa proprietà può essere utilizzata all'interno di un tag a per predisporre un download.

## Colori

- **LeggiPrimoColore**
- **LeggiColoreSuccessivo**
- **LeggiColore**

### LeggiPrimoColore()

**Input:** nessuno.

**Output:** FALSE se non esistono colori per il prodotto corrente. TRUE altrimenti.

**Commenti:** legge le informazioni relative al primo colore del prodotto corrente e le memorizza nelle proprietà IDColoreProdotto, CodiceColoreProdotto e DescrizioneColoreProdotto. Questo colore diventa il colore corrente.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

### leggiColoreSuccessivo ()

**Input:** nessuno.

**Output:** FALSE se non esistono colori successivi per il prodotto corrente (il colore corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge le informazioni relative al colore successivo del prodotto corrente e le memorizza nelle proprietà IDColoreProdotto, CodiceColoreProdotto e DescrizioneColoreProdotto. Questo colore diventa il colore corrente.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

### LeggiPrimoColore()

**Input:** Nessuno

**Output:** FALSE se non esiste alcun colore per il prodotto corrente. TRUE altrimenti.

**Commenti:** legge le informazioni relative al primo colore relativo al prodotto corrente e le memorizza nelle proprietà IDColoreProdotto, CodiceColoreProdotto e

DescrizioneColoreProdotto. Questo colore **non** diventa il colore corrente e pertanto non è possibile chiamare la funzione LeggiColoreSuccessivo().

## Taglie

- LeggiTaglia

### LeggiTaglia(i)

**Input:** indice della taglia da leggere. Se la proprietà NumeroTaglie è maggiore di zero per il prodotto corrente, tale valore può variare da 1 a NumeroTaglie.

**Output:** una stringa contenente la i-esima taglia.

**Commenti:** l'output di questo metodo può essere utilizzato per popolare un tag <select>.

**Esempi:** vedi schedaprodotti.asp nei modelli grafici di ViaDeiMercanti.

## Varianti

- **LeggiPrimaVariante**
- **LeggiVarianteSuccessiva**

### LeggiPrimaVariante(ID)

**Input:** ID (identificativo univoco) del prodotto di cui leggere la prima variante. Se tale parametro viene omesso il motore server leggerà la prima variante del prodotto corrente.

**Output:** FALSE se non esistono varianti per il prodotto corrente. TRUE altrimenti.

**Commenti:** legge la prima variante (diventa la variante corrente) del prodotto specificato. Dopo aver chiamato questo metodo le proprietà relative ai prodotti contengono i valori corretti per la variante selezionata.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

### LeggiVarianteSuccessiva()

**Input:** nessuno.

**Output:** FALSE se non esistono varianti successive per il prodotto corrente (la variante corrente è l'ultima). TRUE altrimenti.

**Commenti:** legge la variante successiva (diventa la variante corrente) del prodotto specificato. Dopo aver chiamato questo metodo le proprietà relative ai prodotti contengono i valori corretti per la variante selezionata.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

## Prodotti correlati

- **LeggiPrimoProdottoCorrelato**
- **LeggiProdottoCorrelatoSuccessivo**

### LeggiPrimoProdottoCorrelato(ID)

**Input:** ID (identificativo univoco) del prodotto di cui leggere il primo prodotto correlato. Se tale parametro viene omesso il motore server leggerà il primo prodotto correlato del prodotto corrente.

**Output:** FALSE se non esistono prodotti correlati per il prodotto corrente. TRUE altrimenti.

**Commenti:** legge il primo prodotto correlato (diventa il prodotto correlato corrente) del prodotto specificato. Dopo aver chiamato questo metodo le proprietà relative ai prodotti contengono i valori corretti per il prodotto correlato selezionato.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

### LeggiProdottoCorrelatoSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non esistono prodotti correlati successivi per il prodotto corrente (il prodotto correlato corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge il prodotto correlato successivo (diventa il prodotto correlato corrente) del prodotto specificato. Dopo aver chiamato questo metodo le proprietà relative ai prodotti contengono i valori corretti per il prodotto correlato selezionato.

**Esempi:** vedi schedaprodotto.asp nei modelli grafici di ViaDeiMercanti.

## Proprietà oggetto Carrello

Tutte le proprietà seguenti vengono gestite in base alla lingua selezionata. Qualora le informazioni relative alla lingua selezionata non fossero state inserite, il software fa riferimento alla lingua predefinita.

### Generali

- **NumeroOggettiNelCarrello** (Integer)
- **IDProdotto** (Long) - ID (identificativo univoco) del prodotto corrente.
- **Quantita** (Single) - quantità del prodotto corrente contenuta nel carrello.
- **IDColore** (Long) - ID (identificativo univoco) del colore del prodotto corrente.
- **Taglia** (String) - taglia del prodotto corrente.
- **NumeroCampiUtente** (Integer) - il carrello virtuale può contenere, per ogni prodotto, ulteriori informazioni diverse da quelle standard. Questa proprietà indica il numero di campi utente associati al prodotto corrente. Si utilizzi il metodo `LeggiCampoUtente` per leggerne i valori.
- **EsenzioneIVA** (Bool) - TRUE se l'utente ha selezionato una destinazione esente IVA.
- **SimboloValuta** (String)

### Prodotti

Relativamente ai prodotti, l'oggetto carrello espone le medesime proprietà dell'oggetto catalogo. In questo modo, iterando su tutti gli oggetti contenuti nel carrello, è possibile disporre delle medesime informazioni. Di seguito sono riportati i nomi delle proprietà. Per la descrizione di queste proprietà si faccia riferimento alla corrispondente proprietà dell'oggetto catalogo.

`CodiceProdotto`, `NomeProdotto`, `DescrizioneFatturaProdotto`, `DescrizioneBreveProdotto`, `DescrizioneEstesaProdotto`, `PrezzoInizialeVL`, `PrezzoProdottoVL`, `PrezzoProdotto`, `UMProdotto`, `ConfezioneProdotto`, `IVAProdotto`, `ImpostaProdottoVL`, `ImpostaProdotto`, `PesoProdotto`, `DisponibilitaProdotto`, `Attivo`, `Novita`, `Offerta`, `ValiditaScontoDal`, `ValiditaScontoAl`, `ScontoProdotto`, `IDColoreProdotto`, `CodiceColoreProdotto`, `DescrizioneColoreProdotto`, `PercorsoFotoProdotto`, `PercorsoThumbnailProdotto`, `PercorsoFotoColore`, `PercorsoThumbnailColore`, `PercorsoDettaglioColore`.

Relativamente ai prodotti, l'oggetto carrello espone anche le seguenti proprietà:

- **IDCategoria** (Long) - identificativo univoco della categoria di appartenenza del prodotto.
- **PrezzoProdottoInEuro** (Double)
- **ImpostaProdottoInEuro** (double)
- **ImportoProdotto** (Double) - quantità per prezzo. Il valore contenuto in questa proprietà è espresso nella valuta associata alla lingua scelta dall'utente.
- **ImportoProdottoVL** (Double) - come sopra in valuta di listino.
- **ImportoProdottoInEuro** (Double) - come sopra in Euro.

## Spese

- **ImportoSpeseSpedizione** (Double) - costo della spedizione per la destinazione selezionata. Il valore contenuto in questa proprietà è espresso nella valuta associata alla lingua scelta dall'utente.
- **ImportoSpeseSpedizioneVL** (Double) - come sopra in valuta di listino.
- **ImportoSpeseSpedizioneInEuro** (Double) - come sopra in Euro.
- **SpeseContrassegno** (Double) - costo della spedizione in contrassegno. Il valore contenuto in questa proprietà è espresso nella valuta associata alla lingua scelta dall'utente.
- **SpeseContrassegnoVL** (Double) - come sopra in valuta di listino.
- **SpeseContrassegnoInEuro** (Double) - come sopra in Euro.
- **AttivaSpeseSpedizionePersonalizzate** (Boolean) - indica se sono attive le spese di spedizione personalizzate (non lette dall'archivio spesa ma impostate con il metodo `ImpostaSpeseSpedizione`).
- **ImportoSpeseSpedizionePersonalizzate** (Double) - importo impostato con il metodo `ImpostaSpeseSpedizione`.

## Totali

- **TotalePesoProdotti** (Single) - peso totale dei prodotti calcolato come somma dei singoli pesi moltiplicati per le rispettive quantità acquistate. Questo valore viene utilizzato per determinare la fascia di prezzo corrispondente per le spese di spedizione.
- **TotaleProdotti** (Double) - totale prezzo prodotti escluso IVA.
- **TotaleProdottiVL** (Double) - come sopra in valuta di listino.
- **TotaleProdottiInEuro** (Double) - come sopra in Euro.
- **TotaleIVA** (Double) - oltre all'IVA sui prodotti include anche l'IVA sulle spese di spedizione. In caso di esenzione IVA non viene sommata al totale ordine.
- **TotaleIVAVL** (Double) - come sopra in valuta di listino.
- **TotaleIVAINEuro** (Double) - come sopra in Euro.
- **TotaleOrdine** (Double) - totale a pagare. Da tale importo mancano soltanto le spese di contrassegno eventualmente sommate in fase di conferma ordine quando l'utente selezionerà la modalità di pagamento.
- **TotaleOrdineVL** (Double) - come sopra in valuta di listino.
- **TotaleOrdineInEuro** (Double) - come sopra in Euro.

## Campi utente

- **NomeCampoUtente** (String) - il nome del campo utente letto dal metodo `LeggiCampoUtente`.

## Valori formattati

Per tutte le proprietà contenenti valori relativi ad importi, ne esiste una versione formattata con separatore delle migliaia. Ciascuna di queste proprietà ha il medesimo nome della corrispondente preceduto dalla lettera F. Per esempio del campo ImportoProdotto=12345,67 esiste la versione

## Metodi oggetto Carrello

### Principale

- **EseguiOperazione**

**EseguiOperazione(Operazione, IndiceOggetto, IDProdotto, Quantita, IDColore, Taglia, Prezzo, IDSpese)**

**Input:** i parametri possono essere passati direttamente al metodo. In caso contrario essi vengono letti tramite metodo POST o GET. I parametri utilizzati sono i seguenti:

- Operazione da effettuare: utilizzando il metodo GET il nome del parametro è Operazione e può valere: aggiungi, modifica, elimina, svuota, spese. Se passato come parametro al metodo, il valore da specificare è il medesimo. Utilizzando il metodo POST, le diverse operazioni corrisponderanno a differenti tag <input type="submit"...> il cui attributo name potrà valere: aggiungi, modifica, elimina, svuota, spese. Il significato delle operazioni è il seguente:
  - aggiungi: inserisce un nuovo prodotto nel carrello. E' necessario passare anche i parametri IDProdotto, Quantita ed opzionalmente IDColore e Taglia.
  - modifica: modifica la quantità relativa ad un determinato prodotto già presente nel carrello. E' necessario passare i parametri IndiceOggetto e Quantita.
  - elimina: elimina un determinato prodotto già presente nel carrello. E' necessario passare il parametro IndiceOggetto.
  - svuota: elimina l'intero contenuto del carrello. Non sono necessari ulteriori parametri.
  - spese: imposta la destinazione di spedizione. E' necessario passare il parametro IDSpese
- IDProdotto: ID (identificativo univoco) del prodotto da aggiungere al carrello.
- Quantita: quantità del prodotto aggiunto o modificato.
- IDColore: ID (identificativo univoco) del colore del prodotto da aggiungere al carrello.
- Taglia: taglia del prodotto da aggiungere al carrello.
- Prezzo: normalmente il prezzo di un oggetto inserito nel carrello è quello definito dal listino e dagli eventuali ulteriori calcoli necessari. E' tuttavia possibile passare al carrello un prezzo calcolato diversamente.
- IndiceOggetto: indice del prodotto da modificare o eliminare.
- IDSpese: ID (identificativo univoco) della destinazione specificata dall'utente.

E' possibile inoltre passare ulteriori parametri definiti *Campi utente* per memorizzare nel carrello ulteriori proprietà non standard. Tali campi saranno coppie (Nome, Valore). Per leggere il valore di questo tipo di campi si utilizzi il metodo *LeggiCampoUtente*.

**Output:** nessuno.

**Commenti:** questo metodo costituisce il cuore della pagina carrello che viene richiamata passando gli opportuni parametri tramite metodo POST o GET. La pagina carrello dei modelli grafici di ViaDeiMercanti è implementata in modo da visualizzare il contenuto del carrello quando questo metodo viene richiamato senza richiedere alcuna operazione. Al contrario quando alla pagina viene richiesta una qualsiasi operazione, essa viene eseguita prima della visualizzazione in modo tale che la pagina stessa mostri il contenuto del carrello già modificato.

**Esempi:** vedi *carrello.asp* nei modelli grafici di ViaDeiMercanti.

## Oggetti

- **LeggiOggetto**
- **LeggiCampoUtente**

### LeggiOggetto(i)

**Input:** indice dell'oggetto di cui leggere i dati. Se la proprietà NumeroOggettiNelCarrello è maggiore di zero, il parametro può valere da 1 a NumeroOggettiNelCarrello.

**Output:** nessuno.

**Commenti:** dopo aver invocato questo metodo le proprietà relative ai prodotti contengono le medesime informazioni contenute nelle corrispondenti proprietà dell'oggetto catalogo. Iterando tramite ciclo for su tutti gli oggetti del carrello è possibile visualizzarne l'intero contenuto.

**Esempi:** vedi carrello.asp nei modelli grafici di ViaDeiMercanti.

### LeggiCampoUtente(W)

**Input:** a questo metodo è possibile passare come parametro sia l'indice del campo utente sia il nome con cui questo campo è stato identificato nella coppia (Nome, Valore). Se il parametro passato è di tipo numerico, il metodo assume che si tratti dell'indice del campo, in caso contrario assume si tratti del nome. Se la proprietà NumeroCampiUtente (impostata dal metodo LeggiOggetto) è maggiore di zero, l'indice può valere da 1 a NumeroCampiUtente.

**Output:** il valore del campo utente specificato.

**Commenti:** questo metodo deve essere invocato dopo aver chiamato il metodo LeggiOggetto poichè fa riferimento all'oggetto corrente (l'ultimo letto).

## Immagini

- **ControlloFotoProdotti**
- **ControlloFotoColore**
- **ControlloDettaglioColore**

### ControlloFotoProdotti()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per il prodotto corrente. FALSE altrimenti.

**Commenti:** questo metodo ha le stesse funzioni del corrispondente metodo dell'oggetto catalogo.

**Esempi:** vedi carrello.asp nei modelli grafici di ViaDeiMercanti.

### ControlloFotoColore()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per il colore corrente del prodotto corrente. FALSE altrimenti.

**Commenti:** questo metodo ha le stesse funzioni del corrispondente metodo dell'oggetto catalogo.

**Esempi:** vedi carrello.asp nei modelli grafici di ViaDeiMercanti.

### ControlloDettaglioColore()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per dettaglio del colore corrente del prodotto corrente. FALSE altrimenti.

**Commenti:** questo metodo ha le stesse funzioni del corrispondente metodo dell'oggetto catalogo.

**Esempi:** vedi carrello.asp nei modelli grafici di ViaDeiMercanti.

## Spese spedizione

- **OpzioniSpeseSpedizione**
- **SpeseSpedizioneOk**
- **ImpostaSpeseSpedizione**

### OpzioniSpeseSpedizione(P)

**Input:** il primo valore da inserire nella lista delle destinazioni. Tale valore corrisponde ad una frase del tipo "Selezionare la destinazione". Per adeguare la frase alla lingua corrente è necessario utilizzare il dizionario.

**Output:** frammento di HTML contenente i tag necessari alla costruzione di un menu a tendina (<select><option>...</option></select>) per la selezione delle destinazioni. L'attributo value di ciascun tag <option> viene impostato all'ID (identificativo univoco) della destinazione corrispondente.

**Commenti:** questo metodo costruisce autonomamente tutto l'HTML necessario a visualizzare la lista delle destinazioni in accordo con il tipo di parametri da passare al metodo EseguiOperazione per la relativa funzione di impostazione.

**Esempi:** vedi carrello.asp nei modelli grafici di ViaDeiMercanti.

### SpeseSpedizioneOk()

**Input:** nessuno.

**Output:** TRUE se l'utente ha già selezionato la destinazione della merce. FALSE altrimenti.

**Commenti:** questo metodo è utile per effettuare un controllo sui dati necessari ad evadere l'ordine.

**Esempi:** vedi condizionivendita.asp nei modelli grafici di ViaDeiMercanti.

### ImpostaSpeseSpedizione(importo)

**Input:** importo delle spese di spedizione da impostare. Se il parametro viene omesso vengono impostate le spese di spedizione come definite in archivio.

**Output:** nessuno.

**Commenti:** questo metodo è utile per impostare spese di spedizione diverse da quelle specificate in archivio. Si consideri ad esempio il caso in cui si vogliono gestire spese di spedizione gratuite per ordini superiori ad un determinato importo.

## Proprietà oggetto Ordine

### Cliente

- **Ragione sociale** (String)
- **Cognome** (String)
- **Nome** (String)
- **Indirizzo** (String)
- **Cap** (String)
- **Localita** (String)
- **Provincia** (String)
- **Stato** (String)
- **CFPI** (String) - codice fiscale o partita IVA
- **Telefono** (String)
- **Email** (String) - a questo indirizzo di posta elettronica, inserito dall'utente durante la compilazione del modulo d'ordine, viene inviata notifica della transazione effettuata.

### Destinatario

- **D\_RagioneSociale** (String)
- **D\_Cognome** (String)
- **D\_Nome** (String)
- **D\_Indirizzo** (String)
- **D\_Cap** (String)
- **D\_Localita** (String)
- **D\_Provincia** (String)
- **D\_Stato** (String)

### Ordine

- **IDOrdine** (Long) - identificativo univoco dell'ordine.
- **IDClienteOrdine** (Long) - identificativo univoco del cliente.
- **NumeroOrdine** (String)
- **DataOrdine** (Date)
- **OraOrdine** (Date)
- **CodiceLinguaOrdine** (String) - codice della lingua selezionata dall'utente durante la visita in cui ha effettuato l'ordine.
- **FCValutaOrdine** (Single) - fattore di conversione della valuta ordine al momento della conferma dell'ordine stesso. Tale valore potrebbe infatti cambiare nel tempo.
- **NumeroFatturaOrdine** (Long)
- **DataFatturaOrdine** (Date)
- **StatoPagamento** (Boolean) - indica se l'ordine è stato pagato.

- **AutorizzazioneCartaCredito** (String) - contiene il codice di autorizzazione assegnato alla transazione dal circuito bancario in caso di pagamento con carta di credito.
- **EsenzioneIVA** (Boolean) - indica se l'ordine è esente IVA.
- **SimboloValutaOrdine** (String) - contiene il simbolo della valuta utilizzata per effettuare l'ordine.

### Dettaglio ordine

- **IDProdottoDettaglioOrdine** (Long) - identificativo univoco del prodotto relativo alla corrente riga d'ordine.
- **IDColoreProdottoDettaglioOrdine** (Long) - identificativo univoco del colore del prodotto relativo alla corrente riga d'ordine.
- **CodiceProdottoDettaglioOrdine** (String) - codice del prodotto. Questo dato è memorizzato nel dettaglio ordine piuttosto che fare riferimento all'archivio prodotti per motivi di coerenza dell'ordine. Infatti l'archivio prodotti potrebbe essere stato modificato dopo l'ordine. La stessa tecnica viene adottata per le informazioni riportate di seguito.
- **NomeProdottoDettaglioOrdine** (String)
- **DescrizioneProdottoDettaglioOrdine** (String)
- **UMProdottoDettaglioOrdine** (String)
- **IVAProdottoDettaglioOrdine** (Single)
- **PesoProdottoDettaglioOrdine** (Single)
- **QuantitaDettaglioOrdine** (Single) - quantità ordinata.
- **QuantitaEvasaDettaglioOrdine** (Single) - quantità evasa.
- **TagliaProdottoDettaglioOrdine** (String)
- **PrezzoProdottoDettaglioOrdine** (Single) - prezzo unitario
- **ImportoProdottoDettaglioOrdine** (Single) - importo prodotto
- **TotaleImponibileOrdine** (Single)
- **TotaleImpostaOrdine** (Single)
- **SpeseSpedizioneOrdine** (Single)
- **SpeseContrassegnoOrdine** (Single)
- **TotaleOrdineOrdine** (Single)

### Stato ordine

- **DataStatoOrdine** (Date) - dopo aver chiamato il metodo LeggiStatoOrdine contiene la data dello stato dell'ordine.
- **OraStatoOrdine** (Date) - come DataStatoOrdine per l'ora.
- **DescrizioneStatoOrdine** (String) - come DataStatoOrdine per la descrizione.

### Pagine

- **NumeroOrdini** (Long) - numero totale di ordini per il cliente corrente.

- **LunghezzaListaOrdini** (Long) - numero di ordini mostrato in una singola pagina di lista ordini.
- **PaginaCorrenteOrdini** (Long) - numero della pagina di lista ordini corrente per il cliente corrente.
- **TotalePagineOrdini** (Long) - numero totale di pagine di lista ordini per il cliente corrente.
- **NumeroProdottiRiordino** (Long) – Numero di prodotti presenti nell'elenco dei prodotti già ordinati dall'utente loggato. Tale elenco viene creato per facilitare un eventuale riordino di prodotti ordinati di frequente.
- **LunghezzaPaginaRiordino** (Long) – Lunghezza pagina riordino, proprietà impostata tramite il metodo `ImpostaLunghezzaPaginaRiordino` dell'oggetto Generale.
- **PaginaCorrenteRiordino** (Long) – Numero della pagina corrente dei prodotti riordinabili
- **TotalePagineRiordino** (Long) – Numero di pagine di prodotti riordinabili

### Varie

- **Note** (String) - il testo digitato dall'utente nel campo *Note* del modulo d'ordine.
- **TipoPagamento** (String) - la modalità di pagamento scelta dall'utente.
- **Errore** (String) - la descrizione di un eventuale errore commesso dall'utente durante la compilazione del modulo d'ordine. I messaggi di errore sono memorizzati nel dizionario e sono pertanto modificabili. Per l'elenco dei codici di dizionario si veda il metodo `ControlloDatiPersonali`

### Valori formattati

Per tutte le proprietà contenenti valori relativi ad importi, ne esiste una versione formattata con separatore delle migliaia. Ciascuna di queste proprietà ha il medesimo nome della corrispondente preceduto dalla lettera F. Per esempio del campo `PrezzoProdottoDettaglioOrdine=12345,67` esiste la versione `FPrezzoProdottoDettaglioOrdine=12.345,67`.

## Metodi oggetto Ordine

### Condizioni di vendita

- LeggiCondizioniVendita

### LeggiCondizioniVendita()

**Input:** nessuno.

**Output:** stringa vuota se il testo *Condizioni di Vendita* non è stato inserito in archivio. Il testo nella lingua corrente altrimenti.

**Commenti:** il testo restituito da questo metodo viene solitamente presentato all'utente quando decide di confermare l'ordine. E' naturalmente possibile collegare anche altrove la relativa pagina per consentire la presa visione delle condizioni di vendita anche prima di effettuare un ordine.

**Esempi:** vedi condizionidivendita.asp nei modelli grafici di ViaDeiMercanti.

## Modulo d'ordine

- **AzzeraDatiPersonali**
- **ControlloDatiPersonali**
- **MemorizzaCookieCliente**

### AzzeraDatiPersonali()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** l'oggetto ordine memorizza temporaneamente i dati personali dell'utente durante la compilazione del modulo d'ordine. Questo evita all'utente che ha commesso un errore di dover ridigitare nuovamente tutte le informazioni. Il nome del metodo è improprio in quanto in realtà inizializza il contenuto di tutte le proprietà predisponendo il motore server ad un nuovo ordine. L'inizializzazione può essere un semplice azzeramento (comportamento iniziale del metodo da cui il nome) oppure una lettura dei dati cliente in base ad un cookie o all'ID dell'utente validato nell'area riservata.

**Esempi:** vedi orderform.asp nei modelli grafici di ViaDeiMercanti.

### ControlloDatiPersonali()

**Input:** i seguenti parametri (utilizzare i nomi esatti) vengono letti direttamente tramite metodo GET o POST: RagioneSociale, Cognome, Nome, Indirizzo, Cap, Localita, Provincia, Stato, CFPI, Telefono, Email, D\_RagioneSociale, D\_Nome, D\_Cognome, D\_Indirizzo, D\_Cap, D\_Localita, D\_Provincia, D\_Stato, Note, TipoPagamento. Il significato dei parametri è il medesimo delle corrispondenti proprietà. Viene inoltre letto il parametro Submit (è necessario impostare correttamente il tag che provoca l'invio del modulo d'ordine: <input type="submit" name="Submit" value="...">) che può valere *Annulla* per indicare che l'utente ha scelto di annullare l'ordine.

**Output:** 0 = dati ok, 1 = l'utente ha scelto di annullare l'ordine, 2 = Errore.

**Commenti:** il metodo controlla che l'utente abbia inserito tutti i dati necessari. Le informazioni relative al destinatario non sono obbligatorie (è possibile controllare la visualizzazione del relativo form tramite la proprietà AttivaIndirizzoDestinatario dell'oggetto Generale). Tuttavia se l'utente compila uno solo dei campi relativi al destinatario, allora diventano tutti obbligatori. In caso di errore la proprietà Errore conterrà la descrizione del problema selezionando dal dizionario l'opportuna frase in accordo alla lingua corrente. I codici utilizzati sono i seguenti:

- ERRNOME: non è stato digitato il nome del cliente.
- ERRCOGNOME: non è stato digitato il cognome del cliente.
- ERRINDIRIZZO: non è stato digitato l'indirizzo del cliente.
- ERRCAP: non è stato digitato il C.A.P. del cliente.
- ERRLOCALITA: non è stato digitato la località del cliente.
- ERRPROVINCIA: non è stato digitato la provincia del cliente.
- ERRSTATO: non è stato digitato lo stato del cliente.
- ERRCFPI: non è stato digitato il codice fiscale o la partita IVA del cliente.
- ERREMAIL: non è stato digitato l'indirizzo email del cliente.
- ERRTIPOPAGAMENTO: non è stato selezionato il tipo di pagamento.
- ERRDCOGNOME: non è stato digitato il cognome del destinatario.
- ERRDINDIRIZZO: non è stato digitato l'indirizzo del destinatario.
- ERRDCAP: non è stato digitato il C.A.P. del destinatario.
- ERRDLOCALITA: non è stato digitato la località del destinatario.
- ERRDPROVINCIA: non è stato digitato la provincia del destinatario.
- ERRDSTATO: non è stato digitato lo stato del destinatario.

**Esempi:** vedi controllodati.asp nei modelli grafici di ViaDeiMercanti.

### MemorizzaCookieCliente()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** invia un cookie all'utente al fine di riconoscerlo ad un prossimo collegamento ed evitare il reinserimento dei dati personali.

**Esempi:** vedi controllodati.asp nei modelli grafici di ViaDeiMercanti.

## Carte di credito

- URLValidazioneCartaCredito
- URLValidazioneCartaCredito1
- ControlloEsitoValidazione
- ControlloEsitoValidazione1

### URLValidazioneCartaCredito()

**Input:** nessuno.

**Output:** URL completo di parametri da passare tramite metodo GET a Payment Gateway, la prima versione del gateway Banca Sella per la gestione dei pagamenti on line.

**Commenti:** il metodo prepara i parametri da passare al gateway Banca Sella (codice esercente, importo codificato (moltiplicato per il TID), OTP e numero d'ordine) e genera il corrispondente URL cui redirigere l'utente. Provvede inoltre alla corretta gestione dell'archivio locale delle OTP.

Per interfacciarsi a GestPay, il nuovo gateway Banca Sella, si utilizzi il metodo URLValidazioneCartaCredito1

**Esempi:** vedi controllodati.asp nei modelli grafici di ViaDeiMercanti.

### URLValidazioneCartaCredito1()

**Input:** nessuno.

**Output:** URL completo di parametri da passare tramite metodo GET a GestPay, il nuovo gateway Banca Sella per la gestione dei pagamenti on line.

**Commenti:** il metodo ha la stessa funzione del metodo URLValidazioneCartaCredito ma lavora con GestPay, il nuovo gateway di Banca Sella.

**Esempi:** vedi controllodati.asp nei modelli grafici di ViaDeiMercanti.

### ControlloEsitoValidazione()

**Input:** parametri passati dal gateway Banca Sella se la transazione prevedeva il pagamento tramite carta di credito.

**Output:** TRUE se pagamento OK. FALSE altrimenti.

**Commenti:** il metodo controlla la validità della transazione avvenuta con Banca Sella verificando la OTP restituita dal gateway. Provvede alla gestione delle OTP locali e memorizza in archivio il codice transazione da visualizzare nell'apposito modulo di gestione ordini.

Per interfacciarsi a GestPay, il nuovo gateway Banca Sella, si utilizzi il metodo ControlloEsitoValidazione1

**Esempi:** vedi esitoordine.asp nei modelli grafici di ViaDeiMercanti.

### ControlloEsitoValidazione1()

**Input:** se la transazione prevedeva il pagamento tramite carta di credito, il metodo riceve in input i parametri passati da GestPay, il nuovo gateway Banca Sella.

**Output:** TRUE se pagamento OK. FALSE altrimenti.

**Commenti:** come per il metodo ControlloEsitoValidazione, controlla la validità della transazione avvenuta con Banca Sella utilizzando però il nuovo gateway GestPay. Memorizza in archivio il codice transazione da visualizzare nell'apposito modulo di gestione ordini.

**Esempi:** vedi esitoordine.asp nei modelli grafici di ViaDeiMercanti.

## Notifiche

- **NotificaEsercente**
- **NotificaCliente**

### NotificaEsercente(N, FUserID)

**Input:** entrambi i parametri sono opzionali.

- N = note aggiuntive da includere nel testo della mail di notifica.
- FUserID = True se si desidera generare automaticamente e comunicare all'utente il proprio Nome Utente e Password nella mail di notifica.

**Output:** nessuno.

**Commenti:** il metodo memorizza in archivio i dati relativi all'ordine (dati cliente, eventuale destinatario, dati generali ordine, dettaglio prodotti). Provvede inoltre ad inviare messaggio di notifica tramite posta elettronica al gestore (l'indirizzo utilizzato è quello indicato come email ordini in configurazione). Il corpo del messaggio è così strutturato:

- Dati cliente
- Eventuali dati destinatario
- Dati ordine (lista prodotti ordinati)
- Riepilogo ordine (totali)
- Eventuali dati non standard passati come parametro a questo metodo
- Note (digitate dall'utente nel modulo d'ordine)

Il corpo del messaggio così descritto sarà riportato, unitamente ad altre informazioni per le quali si rimanda al metodo NotificaCliente, all'interno del corpo messaggio inviato al cliente. Per questo motivo frasi e descrizioni dipendono dalla lingua selezionata e pertanto vengono utilizzati i seguenti codici del dizionario (con conseguente possibilità di modifica):

- DATICLIENTE: intestazione dati cliente.
  - CODICEFISCALE: intestazione codice fiscale o partita iva.
  - DATIDEST: intestazione dati destinatario.
  - DATIORDINE: intestazione dati ordine.
  - CODICEPRODOTTO, CODICECOLORE, TAGLIA, PREZZO, QUANTITA, IMPORTO: intestazioni dati prodotti.
  - RIEPILOGOORDINE: intestazione riepilogo ordine.
  - TOTALEPRODOTTI, TOTALEIVA, TOTALEORDINE: intestazioni totali nel riepilogo ordine.
  - SPESESPEDIZIONE, SPESECONTRASSEGNO: intestazioni spese.
  - TIPOPAGAMENTO: intestazione modalità di pagamento.
-

- MPCARTACREDITO, MPCCPOSTALE, MPBONIFICO, MPCONTRASSEGNO: descrizione tipo di pagamento.
- NOTIFICAORDINE: oggetto messaggio di notifica.

Se il parametro FUserID è True, la mail di notifica inviata all'utente contiene il suo Nome utente e la sua Password. Se si tratta di un nuovo utente che sta effettuando il suo primo ordine, viene generata automaticamente una coppia di valori (il nome utente viene ricavato dal nome e dal cognome, la password viene generata a caso).

**Esempi:** vedi controllodati.asp nei modelli grafici di ViaDeiMercanti.

### NotificaCliente()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** il metodo invia all'utente tramite posta elettronica un messaggio di notifica della transazione effettuata. Il messaggio viene inviato all'indirizzo specificato nel modulo d'ordine. L'indirizzo del mittente è quello specificato in configurazione come Email ordini. Il corpo del messaggio è così strutturato:

- Dati esercente
- Corpo del messaggio indirizzato all'esercente (si faccia riferimento al metodo NotificaEsercente)
- Testo di notifica ordine (in lingua come digitato in archivio)
- Firma

Fraasi e descrizioni dipendono dalla lingua selezionata e pertanto vengono utilizzati i seguenti codici del dizionario (con conseguente possibilità di modifica):

- HAIACQUISTATODA: intestazione dati esercente.
- TELEFONO, FAX, PARTITAIVA: intestazioni campi esercente.
- ESTREMIORDINE: intestazione estremi ordine.
- FIRMANOTORDINE: firma.
- CONFERMARICORDINE: oggetto messaggio posta elettronica.

**Esempi:** vedi esitoordine.asp nei modelli grafici di ViaDeiMercanti.

## Tracking ordine

- **ApriListaOrdini**
- **LeggiOrdineSuccessivo**
- **ChiudiListaOrdini**
- **ControlloPrimaPaginaOrdini**
- **ControlloUltimaPaginaOrdini**
- **LeggiOrdine**
- **LeggiPrimoRigoDettaglio**
- **LeggiRigoDettaglioSuccessivo**
- **LeggiStatoOrdine**

### ApriListaOrdini()

**Input:** i parametri input vengono letti direttamente tramite metodo GET o POST (utilizzare il nome esatto sotto riportato).

- **Action:** vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva.

**Output:** TRUE se esistono ordini per il cliente corrente, FALSE altrimenti.

**Commenti:** inizializza la lista ordini del cliente corrente (quello che ha effettuato l'accesso all'area riservata). Il metodo si posiziona sul primo ordine. La lista viene generata in ordine decrescente di data dell'ordine (dal più recente al più vecchio).

**Esempi:** vedi listaordini.asp nei modelli grafici di ViaDeiMercanti.

### LeggiOrdineSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono ordini successivi (l'ordine corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge l'ordine successivo all'ordine corrente e si posiziona su di esso.

**Esempi:** vedi listaordini.asp nei modelli grafici di ViaDeiMercanti.

### ChiudiListaOrdini()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude la lista ordini.

**Esempi:** vedi listaordini.asp nei modelli grafici di ViaDeiMercanti.

### ControlloPrimaPaginaOrdini()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è la prima della lista ordini. FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista ordini per abilitare il collegamento alla pagina precedente.

**Esempi:** vedi listaordini.asp nei modelli grafici di ViaDeiMercanti.

### ControlloUltimaPaginaOrdini()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è l'ultima della lista ordini. FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista ordini per abilitare il collegamento alla pagina successiva.

**Esempi:** vedi listaordini.asp nei modelli grafici di ViaDeiMercanti.

### LeggiOrdine(IDOrdine)

**Input:** IDOrdine = identificativo dell'ordine da leggere. Se tale parametro viene omissso, il metodo leggerà il valore IDOrdine (utilizzare il nome esatto) tramite il metodo POST o GET.

**Output:** nessuno.

**Commenti:** legge l'ordine identificato dal parametro passato.

**Esempi:** vedi schedaordine.asp nei modelli grafici di ViaDeiMercanti.

### LeggiPrimoRigoDettaglio()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** legge il primo rigo di dettaglio (prodotti ordinati) dell'ordine corrente.

**Esempi:** vedi schedaordine.asp nei modelli grafici di ViaDeiMercanti.

### LeggiRigoDettaglioSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non esistono ulteriori righe di dettaglio per l'ordine corrente (la riga corrente è l'ultima). TRUE altrimenti.

**Commenti:** legge il rigo di dettaglio ordine successivo (diventa il rigo corrente).

**Esempi:** vedi schedaordine.asp nei modelli grafici di ViaDeiMercanti.

### LeggiStatoOrdine(Idx)

**Input:** Idx = indice della riga di stato da leggere.

**Output:** TRUE se esiste una riga di stato dell'ordine con l'indice richiesto. FALSE altrimenti.

**Commenti:** legge la riga di stato dell'ordine specificata dall'indice passato come parametro. Per leggere tutte le righe si passi al metodo un valore crescente, partendo da 1, fino a ricevere FALSE come output.

**Esempi:** vedi schedaordine.asp nei modelli grafici di ViaDeiMercanti.

## Riordino

- **ApriPaginaRiordino**
- **LeggiProdottoRiordinoSuccessivo**
- **ChiudiPaginaRiordino**
- **ControlloPrimaPaginaRiordino**
- **ControlloUltimaPaginaRiordino**

### ApriPaginaRiordino()

**Input:** Nessuno

**Output:** TRUE se esistono ordini per il cliente corrente, FALSE altrimenti.

**Commenti:** Apre la pagina contenente i prodotti già ordinati dall'utente. Tali prodotti vengono presentati a partire da quelli ordinati più di recente. Per ogni prodotto viene proposta la quantità ordinata nell'ultimo ordine, mentre il prezzo presentato, è quello di listino.

### LeggiProdottoRiordinoSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non esistono altri prodotti già ordinati dall'utente (il prodotto corrente è l'ultimo della lista). TRUE altrimenti.

**Commenti:** legge il rigo di dettaglio ordine successivo (diventa il rigo corrente).

### ChiudiPaginaRiordino()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude la lista dei prodotti riordinabili.

### ControlloPrimaPaginaRiordino()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è la prima della lista dei riordinabili. FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista del riordino per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPaginaRiordino()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è l'ultima della lista ordini. FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista del riordino per abilitare il collegamento alla pagina successiva.

## Proprietà oggetto AreaRiservata

### Utenti

- **IDUtente** (Long) - ID (identificativo univoco) dell'utente connesso.
- **IDGruppo** (Long) - ID (identificativo univoco) del gruppo cui appartiene l'utente connesso.
- **Gruppo** (String) - nome del gruppo cui appartiene l'utente connesso.
- **RagioneSociale** (String) - questa proprietà ed le seguenti si riferiscono all'utente connesso.
- **Cognome** (String)
- **Nome** (String)
- **Indirizzo** (String)
- **Cap** (String)
- **Localita** (String)
- **Provincia** (String)
- **Stato** (String)
- **Telefono** (String)
- **Fax** (String)
- **Cellulare** (String)
- **Email** (String)
- **Web** (String) - URL sito utente.
- **CFPI** (String) - codice fiscale o partita Iva
- **UserID** (String)
- **IDListino** (Long) - identificativo listino assegnato all'utente o al gruppo.
- **ScontoPredefinito** (String) - è la percentuale di sconto predefinita.
- **ValiditaScontoDal** (Date) - inizio periodo di validità sconto periodo.
- **ValiditaScontoAl** (Date) - fine periodo di validità sconto periodo.
- **ScontoPeriodo** (String) - è la percentuale di sconto valida nel periodo specificato.
- **TipoUtente** – (Byte) Identifica il tipo di utente, in particolare 0 è B2B, 1 è B2C, 2 è Mailing, 3 è Agente e 4 è Destinatario.

### Elenco Clienti per Agente

- **LunghezzaElencoClienti** (Integer) – Numero di clienti per ogni pagina dell'elenco.
- **PaginaCorrente** (Long)
- **TotalePagine** (Long)
- **NumeroClienti** (Long)
- **FiltroRagioneSociale** (String) – Filtro, applicato all'elenco clienti per agente, sul campo RagioneSociale. Se si applica un filtro, vengono restituiti soltanto quei records i cui rispettivi campi iniziano con i valori impostati nelle proprietà *Filtro<Nome Campo>*
- **FiltroCap** (String) – Filtro, applicato all'elenco clienti per agente, sul campo Cap.
- **FiltroLocalita** (String) – Filtro, applicato all'elenco clienti per agente, sul campo Località.
- **FiltroProvincia** (String) – Filtro, applicato all'elenco clienti per agente, sul campo Provincia.
- **FiltroTelefono** (String) – Filtro, applicato all'elenco clienti per agente, sul campo Telefono.

### Interessi

- **IDInteresse** (Long) - Identificativo univoco dell'interesse.
- **NomeInteresse** (String)
- **DescrizioneInteresse** (String)

### Varie

- **Errore** (String) - messaggio di errore generato in fase di registrazione.

## Metodi oggetto areariservata

### Validazione

- **ValidazioneUtente**
- **ControlloUtenteCollegato**

### ValidazioneUtente()

**Input:** i parametri input vengono letti direttamente tramite metodo GET o POST.

- **UserID** : Account dell'Utente per il quale si intende effettuare la validazione
- **Password** : Password dell'Utente per il quale si intende effettuare la validazione
- **IDUtente**: Nel caso ci sia già un utente loggiato di tipo agente, sarà sufficiente passare IDUtente per ottenere la validazione. In questo caso, il metodo verifica che IIDUtente passato, identifichi effettivamente un utente legato all'agente attualmente loggato.

**Output:** TRUE se la validazione è andata a buon fine, FALSE altrimenti.

**Commenti:** questo metodo controlla la validità dei parametri inseriti (Nome Utente e Password) e predispone il motore server ad operare con i parametri di sconto impostati per l'utente collegato.

**Esempi:** vedi intranet.asp nei modelli grafici di ViaDeiMercanti.

### ControlloUtenteCollegato()

**Input:** nessuno.

**Output:** TRUE se c'è un utente collegato, FALSE altrimenti.

## Registrazione

- **AzzeraDatiUtente**
- **RegistraUtente**

### AzzeraDatiUtente()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** l'oggetto areariservata memorizza temporaneamente i dati personali dell'utente da registrare durante la compilazione del modulo. Questo evita all'utente che ha commesso un errore di dover ridigitare nuovamente tutte le informazioni. Questo metodo azzera il contenuto di tutte le proprietà predisponendo il motore server ad un nuova registrazione.

**Esempi:** vedi registrazione.asp nei modelli grafici di ViaDeiMercanti.

### RegistraUtenti(N, FUserID)

**Input:** entrambi i parametri sono opzionali.

- N = note aggiuntive da includere nel testo della mail di notifica.
- FUserID = True se si desidera generare automaticamente e comunicare all'utente il proprio Nome Utente e Password nella mail di notifica.

I seguenti parametri (utilizzare i nomi esatti) vengono letti direttamente tramite metodo GET o POST:

- IDUtente
- RagioneSociale
- Cognome
- Nome
- Indirizzo
- Cap
- Localita
- Provincia
- Stato
- Telefono
- Fax

- Cellulare
- Email
- Web
- CFPI
- Note

Il significato dei parametri è il medesimo delle corrispondenti proprietà. Viene inoltre letto il parametro Submit (è necessario impostare correttamente il tag che provoca l'invio del modulo d'ordine: `<input type="submit" name="Submit" value="...">`) che può valere *Annulla* per indicare che l'utente ha scelto di annullare la registrazione.

Tramite metodo GET o POST vengono inoltre letti i seguenti parametri:

- IDUtente: se specificato il metodo modifica i dati dell'utente individuato da tale ID. In caso contrario il metodo provvede all'inserimento di un nuovo utente.
- IDGruppo: se specificato l'utente viene inserito nel gruppo indicato.
- TipoUtente: se specificato all'utente viene assegnato tale tipo (0=B2B, 1=B2C, 2=Mailing).
- ImpostaInteressi: se diverso da stringa nulla il metodo provvede a leggere ed ad impostare gli interessi dell'utente
- "INT"<numinteresse>: Variabile composta dalla stringa "INT" e dal numero interesse. Es: INT1, INT2, ecc.

Tramite metodo GET o POST vengono infine letti eventuali parametri relativi agli interessi. Il nome dei parametri deve essere del tipo INTxx dove xx è l'ID dell'interesse. Se tale parametro è diverso da stringa nulla l'interesse corrispondente viene selezionato per l'utente.

**Output:** 0 = dati ok, 1 = l'utente ha scelto di annullare la registrazione, 2 = Errore.

**Commenti:** il metodo controlla che l'utente abbia inserito tutti i dati necessari. In caso di errore la proprietà Errore conterrà la descrizione del problema selezionando dal dizionario l'opportuna frase in accordo alla lingua corrente. I codici utilizzati sono i seguenti:

- ERRRAGIONESOCIALE
- ERRNOME
- ERRCOGNOME
- ERRINDIRIZZO
- ERRCAP
- ERRLOCALITA
- ERRPROVINCIA
- ERRSTATO
- ERRCFPI
- ERRTELEFONO
- ERRFAX
- ERRCELLULARE
- ERREMAIL
- ERRWEB

**Esempi:** vedi registrazione.asp nei modelli grafici di ViaDeiMercanti.

## Interessi

- **ApriElencoInteressi**
- **LeggiInteresseSuccessivo**
- **ChiudiElencoInteressi**
- **ControlloCheckModuloInteressi**

### ApriElencoInteressi()

**Input:** nessuno

**Output:** TRUE se esistono interessi, FALSE altrimenti.

**Commenti:** inizializza la lettura dell'elenco interessi. Il metodo si posiziona sul primo interesse.

**Esempi:** vedi mailing.asp nei modelli grafici di ViaDeiMercanti.

### LeggiInteresseSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono interessi successivi (l'interesse corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge l'interesse successivo all'interesse corrente e si posiziona su di esso.

**Esempi:** vedi mailing.asp nei modelli grafici di ViaDeiMercanti.

### ChiudiElencoInteressi()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude l'elenco interessi.

**Esempi:** vedi mailing.asp nei modelli grafici di ViaDeiMercanti.

---

### ControlloCheckModuloInteressi()

**Input:** nessuno.

**Output:** TRUE se l'interesse corrente è selezionato dall'utente collegato. FALSE altrimenti.

**Commenti:** il metodo è utile per mostrare il modulo di selezione degli interessi già compilato con i dati presenti nel database.

**Esempi:** vedi mailing.asp nei modelli grafici di ViaDeiMercanti.

## Elenco Clienti per Agente

- **ApriElencoClientiAgente**
- **LeggiClienteAgenteSuccessivo**
- **ChiudiElencoClientiAgente**
- **ControlloPrimaPaginaElencoClienti**
- **ControlloUltimaPaginaElencoClienti**

### ApriElencoClientiAgente ()

**Input:** i seguenti parametri vengono letti direttamente tramite i metodi GET e POST:

- FiltroRagioneSociale
- FiltroCap
- FiltroLocalita
- FiltroProvincia
- FiltroTelefono

Questi paramerti consentono di effettuare un filtro, nei campi evidenziati nei rispettivi nomi. E' possibile immettere stringhe parziali.

- **Action:** vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva.

**Output:** TRUE se esistono clienti per l'agente corrente, FALSE altrimenti.

**Commenti:** inizializza la lettura dell'elenco clienti per l'agente corrente. Il metodo si posiziona sul primo cliente.

### LeggiClientiAgenteSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono clienti successivi (il cliente corrente è l'ultimo) per la selezione corrente. TRUE altrimenti.

**Commenti:** legge il cliente successivo al cliente corrente e si posiziona su di esso.

### ChiudiElencoClientiAgente ()

**Input:** nessuno.

**Output:** nessuno.

**Commenti:** chiude l'elenco clienti l'agente corrente.

### ControlloPrimaPaginaElencoClienti ()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è la prima. FALSE altrimenti.

**Commenti:** utile nella pagina di elenco clienti per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPaginaElencoClienti ()

**Input:** nessuno.

**Output:** TRUE se la pagina corrente è l'ultima. FALSE altrimenti.

**Commenti:** utile nella pagina di elenco clienti per abilitare il collegamento alla pagina successiva.

## Oggetto dizionario

L'oggetto dizionario espone soltanto il metodo **Traduci** (non espone proprietà). Questo metodo riceve in Input il codice di una frase contenuta nel dizionario e restituisce tale frase nella lingua selezionata dall'utente se esiste, nella lingua predefinita altrimenti.

In realtà essendo il metodo **Traduci** quello predefinito per l'oggetto è possibile utilizzare la seguente sintassi con ovvi vantaggi di leggibilità:

```
<%Response.Write Dizionario("CODICEDATRADURRE")%>
```

## Proprietà Oggetto Banner

### Banner

- **Link** (String) - Link URL associato al banner
- **Percorso** (String) - Questa proprietà contiene il percorso completo (incluso il nome file) dell'immagine o file relativo al banner. E' pertanto possibile utilizzare tale proprietà all'interno dell'appropriato tag html.

## Metodi Oggetto Banner

### Banner

- **Seleziona**

### Seleziona(Tipo)

**Input:** Tipo = Codice del tipo Banner

**Output:** TRUE se il Banner è stato selezionato , FALSE altrimenti.

**Commenti:** Questo metodo seleziona il prossimo banner, del tipo richiesto, da pubblicare. La selezione viene effettuata a rotazione tra quelli disponibili, tenendo conto dei parametri impostati: periodo di pubblicazione e numero di comparse. Questo metodo imposta le proprietà **Link** e **Percorso** dell'oggetto banner. Essendo noto il tipo di banner selezionato, sarà possibile utilizzare il valore della proprietà percorso all'interno dell'opportuno tag HTML (ad esempio *img* per gif animati, *object* per banner in tecnologia flash ecc)

## Proprietà Oggetto Chat

### Argomenti

- **IDArgomento** (Long) – Identificativo unico dell'argomento
- **FlagSottoArgomenti** (Boolean) – Indica se esistono sottoargomenti per l'argomento corrente
- **NomeArgomento** (String)
- **DescrizioneArgomento** (String)
- **PercorsoArgomenti** (String) – Questa proprietà restituisce la Barra di navigazione dell'argomento corrente. Tale barra comprende una lista di Link, che rappresenta tutti gli argomenti padre precedenti quello corrente.
- **NumeroUtentiArgomento** (Long) – Numero di utenti impegnato in chat per l'argomento corrente.

### Interventi

- **IDUtente** (Long) – Identificativo dell'utente autore dell'intevento.
- **Nickname** (String)
- **DataIntervento** (Date)
- **OralIntervento** (Date)
- **TestoIntervento** (String)
- **IndirizzoIP** (String) – Indirizzo IP di provenienza dell'utente che ha inviato l'intervento.

### Varie

- **Errore** (String) – Stringa di errore
- **LunghezzaPaginaMessaggi** (Long) – Numero di messaggi da mantenere attivi a paritire dall'ultimo. I precedenti vengono persi.
- **FrequenzaAggiornamento** (Long) – Numero di secondi, che trascorrono, tra un refresh e l'altro della pagina dei messaggi relativi alla chat. Vedi metodo `ImpostaFrequenzaAggiornamentoChat` dell'oggetto generale.

## Metodi Oggetto Chat

### Argomenti

- **ImpostaCalcoloUtentiArgomento**
- **LeggiArgomento**
- **ApriPaginaArgomenti**
- **LeggiArgomentoSuccessivo**
- **ChiudiPaginaArgomenti**

### ImpostaCalcoloUtentiArgomento (F)

**Input:** TRUE se si desidera che venga calcolato il numero di utenti connessi per l'argomento corrente, FALSE per disabilitare questo calcolo. Il valore di Default è TRUE

**Output:** Nessuno

**Commenti:** Se impostato a TRUE viene valorizzata la proprietà **NumeroUtentiArgomento**. Poichè il questo calcolo può richiedere molte risorse, può essere utile disattivarlo nei casi in cui questo non sia necessario o gradito. L'impostazione selezionata invocando questo metodo rimane fissata fino ad un eventuale nuova impostazione. L'impostazione predefinita è F = 1 e pertanto a meno di reimpostazioni il calcolo viene effettuato.

### LeggiArgomento (IDArg)

**Input:** IDArg = identificativo dell'argomento da leggere.

**Output:** nessuno.

**Commenti:** legge l'argomento identificato dal parametro passato.

### ApriPaginaArgomenti(IDArg)

**Input:** IDArg = identificativo dell'argomento padre da cui leggere i sottoargomenti. Se tale parametro viene omissso, vengono letti tutti gli argomenti di primo livello.

**Output:** FALSE se non ci sono argomenti in archivio, per l'argomento padre richiesto. TRUE altrimenti.

**Commenti:** inizializza la pagina degli argomenti posizionandosi sul primo di essi.

### LeggiArgomentoSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono argomenti successivi (l'argomento corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge l'argomento successivo a quello corrente e si posiziona su di esso.

### ChiudiPaginaArgomenti()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Chiude la pagina degli argomenti

## Utenti

- **IngressoUtente**
- **AggiornaMessaggi**

### IngressoUtente()

**Input:** I seguenti parametri vengono passati direttamente tramite i metodi GET e POST.

- **Nickname:** Nickname dell'utente che vuole entrare in una stanza (argomento) di discussione.
- **IDArgomento:** Identificativo dell' argomento relativo alla stanza di discussione, dove il nuovo utente intende entrare.

**Output:** TRUE se l'utente *<NickName>* è entrato con successo, FALSE in caso contrario, in questo caso viene valorizzata la proprietà Errore.

**Commenti:** L'errore viene restituito se il parametro NickName non viene passato oppure se esiste già un utente con soprannome *<Nickname>* nella stanza identificata da IDArgomento.

### AggiornaMessaggi()

**Input:** I seguenti parametri vengo passati direttamente tramite i metodi GET e POST

- **Testo:** Testo del messaggio.
- **IDUtente:** Identificativo dell'utente autore del messaggio.

**Output:** Nessuno

**Commenti:** Viene inserito il messaggio contenuto nel parametro testo relativamente all'argomento corrente.

## Interventi

- **ApriPaginaInterventi**
- **LeggiInterventoSuccessivo**
- **ChiudiPaginaInterventi**

### ApriPaginaInterventi()

**Input:** Il seguente parametro viene letto direttamente tramite i metodi GET e POST

- IDUtente = Identificativo dell'utente per il quale si desidera aprire la pagina degli interventi.

**Output:** TRUE se ci sono interventi in archivio, legati all'utente chiesto. FALSE altrimenti

**Commenti:** Apre la pagina degli interventi e si posiziona sul primo di essi.

### LeggiInterventoSuccessivo()

**Input:** nessuno.

**Output:** FALSE se non ci sono interventi successivi (l'intervento corrente è l'ultimo). TRUE altrimenti.

**Commenti:** legge l'intervento successivo all'intervento corrente e si posiziona su di esso.

### ChiudiPaginaInterventi()

**Input:** Nessuno

**Output:** Nessuno

**Commenti:** Chiude la pagina degli Interventi

## Proprietà Oggetto Contenuti

### Sezioni

- **IDSezione** (Long): Identificativo unico della sezione.
- **FlagSottoSezioni** (Boolean): Indica se sono presenti sottosezioni per la sezione corrente.
- **CodiceSezione** (String)
- **NomeSezione** (String)
- **DescrizioneSezione** (String)
- **PercorsoFotoSezione** (String) - Questa proprietà contiene il percorso completo (incluso il nome file) dell'immagine relativa alla categoria corrente. E' pertanto possibile utilizzare tale proprietà all'interno di un tag img.
- **NumeroArticoliSezione** (Long) - Numero di articoli presenti nella sezione corrente.
- **PercorsoSezioni** (String) - Questa proprietà restituisce la Barra di navigazione della sezione corrente. Tale barra comprende una lista di Link, che rappresenta tutte le sezioni padre precedenti quella corrente.
- **FlagDataPubblicazioneArticolo** (Boolean) – Indica se gli articoli della sezione corrente devono essere pubblicati in ordine decrescente di data pubblicazione.

### Articoli

- **IDArticolo** (Long) - Identificativo unico dell'articolo.
- **CodiceArticolo** (String)
- **TitoloArticolo** (String)
- **SottotitoloArticolo** (String)
- **TestoArticolo** (String)
- **EstrattoArticolo** (String) – Primi n caratteri (esclusi i tag HTML) dell'articolo. Il valore di n può essere impostato con il metodo **ImpostaLunghezzaEstrattoArticolo**. Questa proprietà può essere utile per pubblicare un indice di una sezione di articoli.
- **AutoreArticolo** (String)
- **Attivo** (Boolean) - Indica se l'articolo è attivo.
- **Novità** (Boolean) - Indica se l'articolo è una novità.
- **Mailing** (Boolean) - scopi futuri
- **DataPubblicazioneArticolo** (String) – Data di pubblicazione dell'articolo.
- **PubblicazioneDallaDataArticolo** (String) – Data inizio periodo di validità per la pubblicazione dell'articolo.
- **PubblicazioneAllaDataArticolo** (String) – Data fine periodo di validità per la pubblicazione dell'articolo.
- **FlagStampaArticolo** (Boolean) – Indica se sulla pagina web contenente l'articolo, deve essere presente il comando per eseguire la stampa dell'articolo stesso.
- **FlagSegnalazioneArticolo** (Boolean) – Indica se sulla pagina web contenente l'articolo, deve essere presente la possibilità di inviare un' e-mail, ad un amico, contenente il Link all'articolo corrente.

- **FlagVotoArticolo** (Boolean) – Indica se sulla pagina web contenente l'articolo, deve essere presente la possibilità esprimere un parere, riguardo l'articolo corrente.
- **NumeroVotantiArticolo** (Long) – Numero di utenti che hanno votato l'articolo.
- **VotiArticolo** (Long) – Somma dei voti ricevuti dall'articolo.
- **VotoMedioArticolo** (Single) Calcolato come **VotiArticolo** diviso **NumeroVotantiArticoli**.
- **SegnalazioniArticolo** (Long) – Scopi futuri.
- **VisiteArticolo** (Long) - Scopi futuri
- **DataUltimaVisitaArticolo** (String) - Scopi futuri
- **LunghezzaPaginaArticoli** (Integer) - Numero di articoli per pagina
- **PaginaCorrenteArticoli** (Long)
- **TotalePagineArticoli** (Long)
- **FlagNovita** (Integer) – assume i seguenti valori: -1 se la selezione corrente, effettuata tramite il metodo **ImpostaPaginaNovita**, si riferisce soltanto ad articoli non novità, 0 a tutti, 1 soltanto agli articoli novità.

### Link Articoli

- **IDLinkArticolo** (Long) - Identificativo unico del Link articolo.
- **TitoloLinkArticolo** (String)
- **DescrizioneLinkArticolo** (String)
- **URLLinkArticolo** (String) – Url a cui il Link si riferisce.
- **FlagEsternoLinkArticolo** (Boolean) – Può essere utilizzato per aprire un'altra finestra del Browser per Link esterni al sito.
- **VisiteLinkArticolo** (Long) - Scopi futuri
- **DataUltimaVisitaLinkArticolo** (String) - Scopi futuri

### File Articoli

- **IDFileArticolo** (Long) – Identificativo del file articolo.
- **CodiceFileArticolo** (String)
- **TitoloFileArticolo** (String)
- **DescrizioneFileArticolo** (String)
- **PercorsoFileArticolo** – Percorso completo del file. In base al tipo di file (noto in virtù del codice), sarà utilizzato un opportuno tag HTML.
- **DimensioneFileArticolo** (Single) – Dimensione del file.
- **UMDimensioneFileArticolo** (String) – Unità di misura usata per la dimensione, i valori saranno bytes, Kbytes, o Mbytes.

### Segnalazioni

- **MittenteSegnalazione** (String) – Nome del mittente inserito nel modulo di segnalazione.
- **EmailMittenteSegnalazione** (String) – Indirizzo E-Mail del mittente della segnalazione

- **EmailDestinatarioSegnalazione** (String) -
- **NoteSegnalazione** (String) – Testo inserito dal mittente nel modulo di segnalazione

**Varie**

- **Errore** (String)

## Metodi Oggetto Contenuti

### Sezioni

- **LeggiSezione**
- **ApriPaginaSezioni**
- **LeggiSezioneSuccessiva**
- **ControlloSezionePadre**
- **ChiudiPaginaSezioni**
- **ControlloFotoSezione**

### LeggiSezione (IDSez)

**Input:** IDSez = identificativo della sezione da leggere. Se tale parametro non è numerico rappresenta il codice della sezione, se invece è di tipo numerico allora è l'identificativo della sezione.

**Output:** nessuno.

**Commenti:** legge la sezione identificata dal parametro passato.

### ApriPaginaSezioni(IDSez)

**Input:** IDSez = identificativo della sezione padre da leggere. Se tale parametro non è numerico rappresenta il codice della sezione, se invece è di tipo numerico allora è l'identificativo della sezione.

**Output:** FALSE se non ci sono sezioni in archivio per la sezione padre richiesta, TRUE altrimenti.

**Commenti:** inizializza la pagina delle sezioni posizionandosi sulla prima in archivio.

### LeggiSezioneSuccessiva()

**Input:** nessuno.

**Output:** FALSE se non ci sono sezioni successive (la sezione corrente è l'ultima). TRUE altrimenti.

**Commenti:** legge la sezione successiva a quella corrente e si posiziona su di essa.

### ControlloSezionePadre(IDTestSez, IDCurrSez)

**Input:**

- IDTestSez: Identificativo sezione padre
- IDCurrSez: Identificativo sezione corrente

Entrambi i parametri, se non sono numerici rappresentano il codice della sezione, se invece sono di tipo numerico allora rappresentano l'identificativo della sezione.

**Output:** TRUE se la sezione indicata da IDCurrSez è discendente della sezione IDTestSez.

**Commenti:** Per discendenti si intendono tutti i figli diretti, i figli dei propri figli, i figli dei figli dei propri figli e così via.

### ChiudiPaginaSezioni()

**Input:** Nessuno

**Output:** Nessuno

**Commenti:** Chiude la pagina delle sezioni

### ControlloFotoSezione()

**Input:** nessuno.

**Output:** TRUE se esiste l'immagine per la sezione corrente. FALSE altrimenti.

**Commenti:** il metodo è utile per abilitare la visualizzazione dell'immagine della sezione. Il metodo imposta anche la proprietà PercorsoFotoSezione che conterrà il percorso completo (incluso il nome file) dell'immagine della sezione corrente. Questa proprietà può essere utilizzata all'interno del tag img.

## Articoli

- ImpostaPaginaNovita
- ImpostaCalcoloNumeroArticoliSezione
- ImpostaLunghezzaEstrattoArticolo
- ImpostaOrdinamentoArticoliPerDataPubblicazione
- ImpostaOrdinamentoArticoliCustom
- ApriPaginaArticoli
- LeggiArticoloSuccessivo
- LeggiPrimoArticolo
- ChiudiPaginaArticoli
- LeggiArticolo
- VotaArticolo
- ControlloPrimaPaginaArticoli
- ControlloUltimaPaginaArticoli

## ImpostaPaginaNovita (N)

### Input:

- N = -1: saranno letti solo articoli non novità.
- N = 1: saranno letti solo articoli novità.
- N = 0 saranno letti tutti gli articoli.

**Output:** nessuno.

**Commenti:** predispone la lettura dell'archivio articoli in base alla proprietà Novità.

## ImpostaCalcoloNumeroArticoliSezione (F)

**Input:** TRUE se si desidera che venga calcolato il numero di articoli presenti in questa sezione, FALSE per disabilitare questo calcolo. Il valore di Default è TRUE

**Output:** Nessuno

**Commenti:** poichè il calcolo degli articoli per sezione può richiedere molte risorse, può essere utile disattivarlo nei casi in cui questo non sia necessario o gradito. L'impostazione selezionata invocando questo metodo rimane fissata fino ad un eventuale nuova impostazione.

L'impostazione predefinita è  $F = 1$  e pertanto a meno di reimpostazioni il calcolo viene effettuato. Se impostato a TRUE viene valorizzata la proprietà **NumeroArticoliSezione**

### ImpostaLunghezzaEstrattoArticolo (L)

**Input:**

- L: numero di caratteri desiderati per la lunghezza dell'estratto per l'articolo corrente.

**Output:** Nessuno

**Commenti:** Il valore impostato con questo metodo viene utilizzato dalla proprietà **EstrattoArticolo** per prelevare i primi L caratteri dal testo dell'Articolo. Tale estratto non conterrà codice HTML eventualmente presente nel testo costituente l'articolo. E' possibile, infine, che la proprietà restituisca più di L caratteri se risulta necessario completare l'ultima parola dell'articolo.

### ImpostaOrdinamentoArticoliPerDataPubblicazione()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Predisporre la lettura degli articoli in ordine decrescente di data di pubblicazione

### ImpostaOrdinamentoArticoliCustom()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Scopi futuri

### ApriPaginaArticoli(Act, IDSezPar)

**Input:**

---

- Act = vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva.
- IDSezPar = Se non è numerico rappresenta il codice della sezione, altrimenti, se numerico, l'identificativo della sezione.

Per entrambi i parametri c'è la possibilità di passarli al metodo, direttamente tramite i metodi GET e POST usando i nomi Action e IDSezione.

**Output:** TRUE se esistono articoli per la sezione richiesta, FALSE altrimenti.

**Commenti:** Apre la pagina degli articoli e si posiziona sul primo di essi.

### LeggiArticoloSuccessivo()

**Input:** nessuno

**Output:** FALSE se non ci sono articoli successivi (l'articolo corrente è l'ultimo). TRUE altrimenti

**Commenti:** legge l'articolo successivo all'articolo corrente e si posiziona su di esso.

### LeggiPrimoArticolo()

**Input:** nessuno

**Output:** nessuno

**Commenti:** legge il primo articolo dell'ordine corrente

**Esempi:**

### ChiudiPaginaArticoli()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Chiude la pagina articoli

### LeggiArticolo(IDArticolo)

**Input:**

- IDArticolo = se non numerico rappresenta il codice articolo, altrimenti rappresenta l'identificativo

E' possibile passare il parametro anche tramite i metodi GET e POST usando il nome esatto.

**Output:** TRUE se la lettura dell'articolo è avvenuta correttamente, FALSE altrimenti.

**Commenti:** Legge l'articolo individuato dal parametro passato.

### VotaArticolo(ID, Voto)

**Input:**

- ID = Identificativo articolo
- Voto = Voto da assegnare all'articolo

**Output:** nessuno

**Commenti:** Questo metodo, incrementa di una unità la proprietà **NumeroVotantiArticolo**, somma il valore *Voto* ai voti ricevuti dall'articolo specificato da ID. Il range dei voti assegnabili all'articolo quindi non è prestabilito, ma può essere stabilito a piacere chiamando il metodo con il parametro voto desiderato.

### ControlloPrimaPaginaArticoli()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è la prima. FALSE altrimenti.

**Commenti:** utile nella pagina di elenco Articoli per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPaginaArticoli()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è l'ultima. FALSE altrimenti.

**Commenti:** utile nella pagina di elenco clienti per abilitare il collegamento alla pagina successiva.

## Link Articoli

- **LeggiPrimoLinkArticolo**
- **LeggiLinkArticoloSuccessivo**

### LeggiPrimoLinkArticolo()

**Input:** nessuno

**Output:** TRUE se esiste almeno un Link per l'articolo corrente, FALSE altrimenti

**Commenti:** Legge il primo Link dell'articolo corrente

### LeggiLinkArticoloSuccessivo()

**Input:** nessuno

**Output:** TRUE se esiste un Link successivo al link corrente, FALSE altrimenti.

**Commenti:** Legge il Link successivo al Link corrente

## File Articoli

- **LeggiFileArticolo**
- **LeggiPrimoFileArticolo**
- **LeggiFileArticoloSuccessivo**

### LeggiFileArticolo (ID)

**Input:** ID = se non è numerico rappresenta il codice del file, altrimenti rappresenta l'identificativo.

**Output:** TRUE se il file esiste, FALSE altrimenti.

### LeggiPrimoFileArticolo()

**Input:** nessuno

**Output:** TRUE se esiste almeno un file per l'articolo corrente, FALSE altrimenti.

### LeggiFileArticoloSuccessivo()

**Input:** nessuno

**Output:** TRUE se esiste un file successivo al file corrente, FALSE se il file corrente è l'ultimo file presente in archivio.

## Segnalazione

- **SegnalazioneArticolo**

### SegnalazioneArticolo()

**Input:** I seguenti campi vengono letti direttamente tramite i metodi GET e POST

- Mittente = Mittente del messaggio.
- EmailMittente = E-Mail del mittente.
- EmailDestinatario = E-Mail del destinatario.
- Note = Note del messaggio.
- UlterioriNote = Ulteriori Note del messaggio. Utile per inserire note aggiuntive, non indicate dall'utente ma stabilite dal web-master

**Output:** TRUE se l'operazione va a buon fine, FALSE altrimenti, in oltre, in questo caso viene valorizzata la variabile Errore.

**Commenti:** Invia, tramite e-mail, all'indirizzo specificato nel parametro EmailDestinatario, il Link all'articolo corrente. Viene valorizzata la proprietà **errore** nei seguenti casi:

- manca l'E-Mail del mittente, il codice errore sarà NOMAILMITTSEGN.
- manca l'E-Mail del destinatario, il codice errore sarà NOMAILDESTSEGN.

## Proprietà Oggetto Forum

### Argomenti

- **IDArgomento** (Long) – Identificativo unico dell'argomento
- **FlagSottoArgomenti** (Boolean) – Indica se esistono sottoargomenti per l'argomento corrente.
- **NomeArgomento** (String)
- **DescrizioneArgomento** (String)
- **NumeroDiscussioniArgomento** (Long) – Numero di discussioni presenti per l'argomento corrente.
- **NumeroInterventiArgomento** (Long) – Numero di interventi presenti per l'argomento corrente.
- **PercorsoArgomenti** (String) Questa proprietà restituisce la Barra di navigazione dell'argomento corrente. Tale barra comprende una lista di Link, che rappresenta tutti gli argomenti padre precedenti quello corrente.

### Discussioni

- **LunghezzaPaginaDiscussioni** (Integer) – Numero di discussioni contenute nella pagina di discussioni.
- **PaginaCorrenteDiscussioni** (Integer) – Numero della pagina corrente di discussioni.
- **TotalePagineDiscussioni** (Integer) – Numero totale di pagina di discussioni per l'argomento corrente.

### Risposte

- **LunghezzaPaginaRisposte** (Integer) – Numero di interventi, inseriti come risposte ad un intervento che inizia una discussione, presenti nella pagina delle risposte.
- **PaginaCorrenteRisposte** (integer) – Numero della pagina corrente.
- **TotalePagineRisposte** (integer) – Numero totale di pagina di risposte per la discussione corrente.

### Interventi

- **IDIntervento** (Long) – Identificativo unico dell'intervento.
  - **IDInterventoPadre** (Long) - Identificativo unico dell'intervento padre per l'intervento corrente.
  - **MittenteIntervento** (String) – Mittente dell'intervento.
  - **EmailMittenteIntervento** (String) – EMail del mittente dell'intervento.
  - **OggettoIntervento** (String) – Oggetto dell'intervento.
-

- **TestoIntervento** (String)
- **NoteModeratoreIntervento** (String) – Note del moderatore per l'intervento corrente.
- **Attivo** (Boolean) – Indica se l'intervento è attivo
- **DataIntervento** (Date)
- **OralIntervento** (Date)
- **EstrattoIntervento** (String) – Testo iniziale dell'intervento, la lunghezza di questo testo viene impostata con il metodo **ImpostaLunghezzaEstrattoIntervento**.

#### **Varie**

- **Errore** (String) – Contiene la descrizione dell'errore.

## Metodi Oggetto Forum

### Argomenti

- **ImpostaCalcoloStatisticheArgomento**
- **LeggiArgomento**
- **ApriPaginaArgomenti**
- **LeggiArgomentoSuccessivo**
- **ChiudiPaginaArgomenti**

### ImpostaCalcoloStatisticheArgomento (F)

**Input:** TRUE se si desidera che vengano calcolate le statistiche per gli argomenti del forum, FALSE per disabilitare questo calcolo. Il valore di Default è TRUE

**Output:** Nessuno

**Commenti:** poichè il calcolo delle statistiche per argomento può richiedere molte risorse, può essere utile disattivarlo nei casi in cui questo non sia necessario o gradito. L'impostazione selezionata invocando questo metodo rimane fissata fino ad un eventuale nuova impostazione. L'impostazione predefinita è F = 1 e pertanto a meno di reimpostazioni il calcolo viene effettuato. Le proprietà valorizzate da questo calcolo sono **NumeroDiscussioniArgomento** e **NumeroInterventiArgomento**.

### LeggiArgomento (IDArg)

**Input:** IDArg = Se non è numerico rappresenta il codice dell'argomento, altrimenti l'identificativo.

**Output:** nessuno.

### ApriPaginaArgomenti(IDArg)

**Input:** IDArg = Se non è numerico rappresenta il codice dell'argomento, altrimenti l'identificativo dell'argomento padre

**Output:** TRUE se esistono argomenti per il parametro richiesto, FALSE altrimenti.

---

**Commenti:** Apre la pagina degli argomenti e si posiziona sul primo di essi.

### LeggiArgomentoSuccessivo()

**Input:** nessuno

**Output:** TRUE se esiste un argomento successivo, FALSE altrimenti.

**Commenti:** legge l'argomento successivo all'argomento corrente e si posiziona su di esso.

### ChiudiPaginaArgomenti()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Chiude la pagina Argomenti.

## Interventi

- **ImpostaLunghezzaEstrattoIntervento**
- **LeggiIntervento**
- **InvioIntervento**

### ImpostaLunghezzaEstrattoIntervento (L)

#### Input:

- L: numero di caratteri desiderati per la lunghezza dell'estratto intervento.

**Output:** Nessuno

**Commenti:** Il valore impostato con questo metodo viene utilizzato dalla proprietà **EstrattoIntervento** per prelevare i primi L caratteri dal testo dell'intervento. Tale estratto non conterrà codice HTML eventualmente presente nel testo costituente l'intervento. E' possibile, infine, che la proprietà restituisca più di L caratteri se risulta necessario completare l'ultima parola dell'intervento.

### LeggiIntervento (ID)

**Input:** ID = Identificativo dell'intervento, può essere passato anche con i metodi GET e POST con nome IDIntervento.

**Output:** TRUE se l'intervento richiesto esiste, FALSE altrimenti.

**Commenti:** Viene letto l'intervento specificato dal parametro ID.

### InvioIntervento()

**Input:** I seguenti parametri vengono passati tramite i metodi GET e POST:

- Mittente = Mittente del messaggio
- EmailMittente = E-Mail Mittente
- Oggetto = Oggetto del messaggio
- Testo = Testo dell'intervento inviato

**Output:** TRUE se sono presenti i parametri necessari all'invio, FALSE altrimenti, in questo caso viene anche valorizzata la variabile Errore.

**Commenti:** il metodo controlla che l'utente abbia inserito tutti i dati necessari. In caso di errore la proprietà Errore conterrà la descrizione del problema selezionando dal dizionario l'opportuna frase in accordo alla lingua corrente. I codici utilizzati sono i seguenti:

- NOMITTINT = Non è stato valorizzato il parametro Mittente.
- NOGGINT = Non è stato valorizzato il parametro Oggetto.
- NOTESTOINT = Non è stato valorizzato il parametro Testo.

## Discussioni

- **ApriPaginaDiscussioni**
- **LeggiDiscussioneSuccessiva**
- **ChiudiPaginaDiscussioni**
- **ControlloPrimaPaginaDiscussioni**
- **ControlloUltimaPaginaDiscussioni**

### ApriPaginaDiscussioni(Act, IDArgPar)

#### Input:

- Act: vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva. Se non viene passato assume valore di default "I".
- IDArgPar : Se non è numerico rappresenta il codice dell'argomento, altrimenti l'identificativo.

Entrambi i parametri possono essere passati direttamente con i metodi GET e POST con i nomi Action e IDArgomento

**Output:** TRUE se esistono interventi in archivio per i parametri richiesti, FALSE altrimenti.

**Commenti:** Le discussioni sono create da interventi che ne costituiscono l'inizio, e comprendono interventi che rispondono a questi primi. Cioè, gli interventi possono essere inseriti come risposte a qualche altro intervento, quindi entrando in una discussione esistente, oppure come intervento ex-novo, creando una nuova discussione. I metodi relativi alla categoria discussioni, leggono soltanto i primi interventi, cioè quelli che identificano la discussione, mentre nella categoria risposte, ci sono i metodi che leggono il secondo tipo di intervento.

### LeggiDiscussioneSuccessiva()

**Input:** nessuno

**Output:** FALSE se non ci sono interventi successive a quello corrente. TRUE altrimenti.

**Commenti:** legge l' intervento successivo a quello corrente. Unitamente al metodo ApriPaginaDiscussioni serve a creare la lista di interventi per l'argomento selezionato.

### ChiudiPaginaDiscussioni()

**Input:** nessuno

**Output:** nessuno

**Commenti:** Chiude la pagina delle discussioni.

### ControlloPrimaPaginaDiscussioni()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è la prima della lista corrente, FALSE altrimenti.

**Commenti:** utile per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPaginaDiscussioni()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è l'ultima della lista corrente, FALSE altrimenti.

**Commenti:** utile per abilitare il collegamento alla pagina successiva.

## Risoste

- **ApriPaginaRisposte**
- **LeggiRispostaSuccessiva**
- **ChiudiPaginaRisposta**
- **ControlloPrimaPaginaRisposte**
- **ControlloUltimaPaginaRisposte**

### ApriPaginaRisposte(Act, IDIntPar)

#### Input:

- Act: vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva. Se non viene passato assume valore di default "I"
- IDIntPar : Identificativo dell'intervento da leggere.

Entrambi i parametri possono essere passati direttamente con i metodi GET e POST con i nomi Action e IDIntervento.

**Output:** TRUE se esistono risposte in archivio per l'intervento richiesto in input, FALSE altrimenti.

**Commenti:** Apre la pagina, richiesta dal parametro Act, con le risposte relative all'intervento specificato da Idintervento.

### LeggiRispostaSuccessiva()

**Input:** nessuno

**Output:** FALSE se non ci sono risposte successive a quella corrente. TRUE altrimenti.

**Commenti:** legge la risposta successiva a quella corrente. Unitamente al metodo ApriPaginaRisposte serve a creare la lista di risposte per l'intervento selezionato.

### ChiudiPaginaRisposta()

**Input:** nessuno

---

**Output:** nessuno

**Commenti:** Chiude la pagina delle risposte.

### ControlloPrimaPaginaRisposte()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è la prima della lista corrente, FALSE altrimenti.

**Commenti:** utile per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPaginaRisposte()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è l'ultima della lista corrente, FALSE altrimenti.

**Commenti:** utile per abilitare il collegamento alla pagina precedente.

## Proprietà Oggetto Ricerca

### Record

- **TipoRecord** (Byte) – Vale 1 se la proprietà IDRecord si riferisce all'identificativo di un prodotto, 3 se si riferisce all'identificativo di un articolo.
- **IDRecord** (Long) – Identificativo unico del record. Vedi TipoRecord.

### Pagine

- **LunghezzaPagina** (Integer) – Numero di elementi contenuti nella pagina risultante dalla ricerca.
- **PaginaCorrente** (Long) – Numero della pagina corrente.
- **TotalePagine** (Long) – Numero di pagine risultanti dalla ricerca.
- **NumeroRisultati** (Long) – Numero di elementi risultanti dalla ricerca.

## Metodi Oggetto Ricerca

### Risultati

- **ApriPaginaRisultati**
- **LeggiRisultatoSuccessivo**
- **ChiudiPaginaRisultati**

### ApriPaginaRisultati

**Input:** I seguenti parametri vengono passati tramite i metodi GET e POST.

- Action = vale I per inizializzare la pagina, P per passare alla pagina precedente, S per passare alla pagina successiva.
- ParoleChiave = elenco di parole chiave da ricercare, separate da spazio.

**Output:** TRUE se ci sono risultati per la ricerca effettuata, FALSE altrimenti

**Commenti:** Apre la pagina richiesta e si posiziona sul primo elemento di essa.

### LeggiRisultatoSuccessivo

**Input:** nessuno

**Output:** TRUE se il risultato corrente è il primo della selezione corrente, FALSE altrimenti.

**Commenti:**

### ChiudiPaginaRisultati

**Input:** nessuno

**Output:** nessuno

**Commenti:** Chiude la pagina dei risultati.

## Pagine

- **ControlloPrimaPagina**
- **ControlloUltimaPagina**

### ControlloPrimaPagina()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è la prima, FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista dei risultati della ricerca, per abilitare il collegamento alla pagina precedente.

### ControlloUltimaPagina()

**Input:** nessuno

**Output:** TRUE se la pagina corrente è l'ultima, FALSE altrimenti.

**Commenti:** utile nella pagina che mostra la lista dei risultati della ricerca, per abilitare il collegamento alla pagina successiva.