

# I moduli GLOBO - Guido allo sviluppo della modulistica Globo per STU 2 / STU 3

***“BP-OR-01- Manuale funzionale per STU  
 v2.0 - v3.0” del kit di riuso del progetto  
 PON-GOV “SPRINT”***

Per conto di:

Amin Khayam, Stefano Cortinovis <i>Redatto</i>	Marco Deligios <i>Verificato e autorizzato</i>	6/3/2020 <i>Data d'emissione</i>	2 <i>Revisione</i>
I_Ricerca_Sviluppo <i>Commessa</i>	Protocollo <i>Protocollo</i>	Pagina 1 di 36	



Le informazioni contenute in questo documento sono di proprietà intellettuale della GLOBO srl, sotto licenza [CC-BY-4.0](#). Prodotti o aziende indicate nel documento possono essere marchi o marchi registrati dei rispettivi proprietari.

La GLOBO srl governa i processi e i dati all'interno di un sistema integrato di gestione della qualità e della privacy conforme alla normativa UNI EN ISO 9001:2015, al [Decreto legislativo 30/06/2003, n. 196](#) "Codice in materia di protezione dei dati personali" e al [Regolamento \(UE\) 27/04/2016, n. 679](#), "General data protection regulation (GDPR)".

Accogliendo la raccomandazione dell'Unione Europea nell'ambito della certificazione ambientale, che suggerisce di *"migliorare l'efficienza ambientale di un'impresa consentendole di avere una conoscenza reale degli aspetti ambientali più rilevanti nella propria attività"*, questo documento è impaginato per essere stampato in modalità fronte e retro.



## SOMMARIO

0	STORIA DELLE REVISIONI	5
1	SCOPO E CAMPO D'APPLICAZIONE	5
1.1	Scopo	5
1.2	Campo d'applicazione	5
2	RIFERIMENTI NORMATIVI	5
3	TERMINI E DEFINIZIONI	5
4	CONTENUTI	6
4.1	Il modulo di Globo s.r.l.	6
4.1.1	Struttura generale del documento	6
4.1.2	Specifiche di integrazione per la struttura del documento	7
4.1.2.1	Specifiche funzionali sul naming dei campi	7
4.1.2.2	Specifiche di stile	8
4.2	Il meta tag "GLOBO_modulo"	9
4.3	Il meta tag "GLOBO_modulofiglio"	13
4.4	Il meta tag "GLOBO_allegato"	14
4.5	Il meta tag "GLOBO_mappa" e l'inserimento di una mappa nel modulo	16
4.6	Il meta tag "GLOBO_immagine"	19
4.7	Il meta tag "GLOBO_report"	21
4.8	Il meta-tag "GLOBO_function"	22
4.9	I campi di override	23
4.10	Javascript: funzioni particolari	24
4.11	Interagire con la libreria PD4ML	26
4.11.1	Il tag globo_salto_pagina	26
4.11.2	Il tag globo_fondo_pagina	26
4.11.3	Nascondere o mostrare elementi nel PDF	27
4.12	Integrazione con SOLO1	28
4.12.1	La tassonomia	28
4.12.1.1	Dichiarazione di radio button all'interno del modulo HTML:	28
4.12.1.2	Dichiarazione di checkbox all'interno del modulo HTML:	28
4.12.2	I parametri	28
4.12.2.1.1	Input	28
4.12.2.1.2	Select	29
4.12.2.1.3	Textarea	29
4.12.3	Esempi	29
5	APPENDICE B: I FIRMATARI DEGLI ALLEGATI	30
6	APPENDICE C: LO SCRIPT DI ATTIVAZIONE MAPPA	32
7	APPENDICE C: GESTIONE DELLE VERSIONI E LA POSIZIONE DI GLOBO	33
7.1	Gestione della versione da parte di Globo s.r.l.	33
8	APPENDICE D: LA SINTASSI DELLE SUBSTITUTION	34
8.1	Namespace "valore"	34
8.1.1	Esempi:	34
8.1.1.1	Dichiarazione di una Substitution	34

8.1.1.2	Richiamare una variabile dichiarata nella Substitution	35
8.2	Namespace "istanza"	35
8.3	Namespace "funzione"	36

## 0 STORIA DELLE REVISIONI

Rev.	Data	Redatto	Descrizione
0	29/09/2014	Andrea Cividini	Prima versione del documento
1	30/10/2019	Amin Khayam, Stefano Cortinovis	Aggiornamento e revisione del documento redatto da Andrea Cividini
2	6/3/2020	Amin Khayam, Stefano Cortinovis	Revisione del documento

## 1 SCOPO E CAMPO D'APPLICAZIONE

### 1.1 Scopo

Lo scopo di questo documento è di fornire una guida tecnica e funzionale riguardante la struttura dei tag meta della modulistica telematica redatta di Globo s.r.l.

### 1.2 Campo d'applicazione

L'intento è di costituire un riferimento completo per la redazione, modifica o anche solo analisi dei moduli HTML secondo lo standard di Globo s.r.l. e, più precisamente, del codice necessario per il funzionamento di questi con lo Sportello Telematico Unificato

## 2 RIFERIMENTI NORMATIVI

## 3 TERMINI E DEFINIZIONI

- STU: Sportello Telematico Unificato, sistema che permette la presentazione telematica di istanze agli enti.
- Name dei campi: valore dell'attributo HTML name dato al campo di input
- Istanza: insieme di compilazioni ed allegati da presentare all'ente nell'ambito di una singola comunicazione/richiesta
- Override: sovrascrittura dei comportamenti determinata o dal file di personalizzazione o dalla compilazione dei relativi campi nascosti

## 4 CONTENUTI

### 4.1 Il modulo di Globo s.r.l.

PREMESSA: generalmente il comportamento di uno modulo Globo è equivalente su STU2 e STU3 in fase di compilazione, in quanto il motore di Validator è de facto il medesimo.

Quello che cambia sostanzialmente è la gestione e funzionalità dei tag meta con i relativi attributi: molti comportamenti che in backend STU2 erano determinati dalla valorizzazione dei meta attributi in STU3 sono governati da logiche definite da altre configurazioni. Indicheremo attributo per attributo la pertinenza rispetto alle API STU2 e STU3.

#### 4.1.1 Struttura generale del documento

Un modulo telematico, redatto secondo lo standard Globo e funzionante con lo STU alla versione 2.0, è come la precedente versione una pagina di codice HTML; ne conserva ed estende anche tutte le potenzialità e funzionalità. La sua struttura generale è così composta:

- La dichiarazione del tipo di documento, uguale per tutti:  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
- L'apertura del tag html e del tag head; all'interno della *testa* del documento posso essere a loro volta presenti:
  - Il meta "GLOBO\_modulo", che definisce le proprietà generali del modulo e/o dell'istanza
  - Un numero variabile di meta "GLOBO\_modulofiglio" che definiscono eventuali dipendenze e relazioni tra i diversi moduli
  - Un numero variabile di meta "GLOBO\_allegato" che definiscono eventuali documenti allegabili alla compilazione del prospetto telematico
  - Un numero variabile di meta "GLOBO\_mappa" che indicano la presenza di mappe all'interno del modulo, su cui disegnare geometrie relative all'istanza in presentazione
  - Un numero variabile di meta "GLOBO\_report" che indicano la presenza di uno o N report riassuntivi da generare al termine al completamento dell'istanza (non obbligatori)
  - Un numero variabile di meta "Globo\_function" di supporto ai report, che definiscono le eventuali funzioni necessarie per i valori calcolati del report
  - Un'eventuale porzione di codice Javascript nella quale è possibile inserire i comportamenti specifici del modulo ed eventuali routine di controllo
  - Il tag title per intitolare correttamente il documento anche nel browser
  - Due tag script e un tag link **obbligatori** per l'inclusione della libreria jQuery ver. 1.8.1, dello script di utilità moduli-driver.js e dello stile testato compatibile con la stampa PDF, moduli.css; ecco il codice:  
<script src="/appoggio/js/jquery-1.8.1.min.js">

```
<script src="/appoggio/js/moduli-driver.js">
<link href="/appoggio/css/moduli.css" rel="stylesheet" type="text/css"/>
```

- La chiusura del tag head e l'apertura del tag body, contenente il corpo del modulo; la sua organizzazione generale è molto meno rigida, ma vi debbono essere determinate caratteristiche e alcune porzioni standard:

- Il paragrafo dei bottoni, contenente i pulsanti necessari all'invio della compilazione al server, così definito:

```
<p class="Modulo">
  <input id="SalvaModulo" name="SalvaModulo" style="height:30px;width:120px;"
  title="Verifica la corretta compilazione dei campi e salva il modulo nel faldone virtuale
  dell'istanza." type="submit" value="Salva modulo"/>
  <input id="SalvaCampi" name="SalvaCampi" onclick="salvaCampi();"
  style="height:30px;width:120px;" title="Salva il contenuto dei campi del modulo sul
  computer locale, per riutilizzarli successivamente." type="button" value="Salva campi"/>
  <input id="CaricaCampi" name="CaricaCampi" onclick="caricaCampi();"
  style="height:30px;width:120px;" title="Riempie i campi del modulo utilizzando il
  contenuto dei campi precedentemente salvati." type="button" value="Carica
  campi"/>
</p>
```

- Infine la chiusura del tag body e del tag html, definendo così la fine del documento

## 4.1.2 Specifiche di integrazione per la struttura del documento

### 4.1.2.1 Specifiche funzionali sul naming dei campi

Di seguito saranno presentati una serie di comportamenti ed automatismi utili durante la creazione della struttura del documento e della naming convention del campo, oltre che della logica degli script:

- Tutti i campi il cui nome contiene "CodiceISTAT" saranno automaticamente compilati con il codice Istat del comune destinatario dell'istanza
- Tutti i campi il cui name inizia per "Luogo" o combacia il *pattern* "Luogo[0-9]\*\_PDF" saranno automaticamente compilati con il nome esteso dell'ente destinatario
- I campi il cui nome rispecchia il pattern "Immobile[0-9]\*\_CatComune" saranno valorizzati con il codice catastale del comune destinatario
- I campi il cui nome rispecchia il pattern "Immobile[0-9]\*\_ Comune" saranno valorizzati con nome esteso del comune destinatario
- Se è presente un campo "Ente" esso sarà valorizzato come "Luogo"; se non presente verrà creato come campo nascosto e inserito automaticamente nel documento
- La data odierna è automaticamente inserita in tutti i campi i cui name rispecchiano i pattern "Data[0-9]" e "Data[0-9]\_PDF"
- Un campo 'GLOBO\_codice\_modulo' conterrà il codice alfanumerico del modulo HTML inserito nel suo meta principale
- Su tutti i campi il cui nome contiene "CodiceFiscale" verrà attivata la funzionalità di controllo e decodifica del codice fiscale

- In base ai parametri di atterraggio alla pagina possono avvenire i seguenti eventi:
  - Creazione di un campo 'GLOBO\_id\_padre' in cui verrà inserito l'id numerico della compilazione da cui dipende quella in svolgimento
  - Creazione di un campo 'GLOBO\_recupera\_dati' in cui verrà inserito l'id della compilazione attualmente in modifica
  - Il campo 'DichiarazioneCF' viene valorizzato con il codice fiscale del firmatario dell'istanza principale
  - Il campo 'Codice\_istanza' viene valorizzato con il codice istanza nelle compilazioni secondarie
  - L'eventuale tabella con id 'Tabella\_Procura' verrà mostrata solamente se le informazioni sul modulo indicano la possibilità di presentare l'istanza in procura

Inoltre, i seguenti valori sono riservati a particolari funzionalità di controllo e/o parametrizzazione:

- Name non concessi: SempreSI e tutti quelli combacianti con la naming convention di cui al punto 4.9
- Id non concessi: ATECO\_Validation, Parametri\_Codice\_ATECO\_attivita\_prevalen, Parametri\_Attivita\_1, Parametri\_Attivita\_1\_note, Parametri\_Attivita\_2, Parametri\_Attivita\_2\_note, Parametri\_Attivita\_3, Parametri\_Attivita\_3\_note, GLOBO\_controlla\_sempre
- Valori che verranno ignorati: "gg/mm/aaaa"

#### 4.1.2.2 Specifiche di stile

L'inclusione delle definizioni di stile presenti nel file moduli.css mette a disposizione una serie di template testati per la stampa PDF e la compatibilità cross-browser, nonché la maggiore corrispondenza possibile con i template cartacei. Qui presentiamo i più utilizzati:

- La presenza della classe "Modulo" su un elemento ne fissa la larghezza 19 cm; alle tabelle conferisce la colorazione grigia tipica dei layout di Globo e fissa le altezze delle righe.
- La classe "Labeltabella" applicata ai paragrafi li rende ideali per i titoli delle tabelle; la relativa riga dovrà avere la classe "Rigaintestazione"
- La classe "Labelcella" applicata ai paragrafi li rende ideali per i singoli titoli dei campi; la relativa riga dovrà avere la classe "Rigaintestazioneridotta"
- La classe 'cornice' applicata agli input o alle select permette di riquadrare di nero (contorno di 1px) l'area destinata al valore sia nella stampa PDF vuota, sia nel modulo compilato (destinata all'uso nei moduli non aderenti allo standard Globo)

All'interno di tale foglio di stile sono definite ulteriori regole specifiche per i modelli regionali come la SCIA, gli header e altre peculiarità.



## 4.2 Il meta tag "GLOBO\_modulo"

Viene convenzionalmente definito tag "GLOBO\_modulo" un meta-tag il cui name è valorizzato a "GLOBO\_modulo", posto nella *head* del documento.

Tale tag, e più precisamente la corretta creazione dei suoi attributi, permettono di determinare diversi meccanismi e parametrizzare alcune funzionalità del modulo o dell'istanza intera.

È **obbligatorio**, così come un determinato sottoinsieme dei suoi attributi, e la sua assenza determina il rifiuto del modulo al caricamento sul sistema di modulistica online.

Lo schema seguente espone il set completo di attributi, il loro dominio, la loro obbligatorietà e la *feature* a cui sono correlati:

| Attributo                    | Obbligatorio/Statico     | Formato/Dominio        | Descrizione  | Default |
|------------------------------|--------------------------|------------------------|--|---------|
| name                         | SI/SI                    | "GLOBO_modulo"         | Indica che il meta definisce i comportamenti del modulo e dell'istanza<br>API: STU2/STU3   | N.A     |
| campi_firma                  | SI (se firmare != NO)/NO | Stringa, nessun limite | Deve contenere un elenco di stringhe separate da virgola, senza spazi; queste stringhe debbono essere i <i>name</i> dei campi che conterranno, una volta compilato il modulo, i codici fiscali dei firmatari richiesti per quella compilazione<br>API: STU2/STU3 | ""      |
| campi_firma_label            | NO/NO                    | Stringa, nessun limite | Indicare le etichette descrittive dei vari ruoli dei firmatari indicati con il precedente campo, separandole con una virgola<br>API: STU2/STU3   | N.A.    |
| classi_modulo                | NO/NO                    | Stringa, nessun limite | (Deprecato) Elenco di stringhe separate da virgola indicanti gli ambiti di competenza dell'attributo   | ""      |
| codice_modulo                | SI/SI                    | Stringa, 100           | Codice che identifica univocamente il modulo all'interno del portale<br>API: STU2/STU3   | N.A     |
| descrizione                  | SI/SI                    | Stringa, nessun limite | Descrizione testuale del modulo<br>API: STU2/STU3  | N.A     |
| destinatario_amministrazione | NO/NO                    | Stringa, 50            | Codice identificativo dell'Ente destinatario<br>API: STU2  | ""      |
| destinatario_ufficio         | SI/NO                    | Stringa, 50            | Codice identificativo dell'UOO destinataria<br>API: STU2   | ""      |
| dinamico                     | NO/SI                    | SI<br>NO               | SI: indica che il comportamento del modulo è modificabile tramite le tecniche di override durante la compilazione<br>NO: indica che il modulo non cambia dinamicamente i suoi attributi<br>API: STU2   | NO      |
| email_a                      | NO/NO                    | Stringa, nessun limite | Deve contenere un eventuale destinatario aggiuntivo per l'istanza quando l'invio avviene via mail<br>API: STU2/STU3  | ""      |

|                         |       |  |  |       |
|-------------------------|-------|--|--|-------|
| ente_abilitato          | NO/NO | TUTTI<br>Stringa, nessun limite                      | TUTTI: indica che l'istanza in considerazione è compilabile per qualsiasi ente del portale<br>o<br>Elenco di stringhe, separate da virgola, indicanti i codici amministrativi degli Enti abilitati alla ricezione dell'istanza<br>API: STU2  | TUTTI |
| filigrana               | NO/NO | SI<br>NO   | Determina la possibilità di presentare il modulo in cartaceo.<br>La parola chiave "SI" determina la creazione di una filigrana sullo sfondo riportante la scritta "FAC-SIMILE"<br>API: STU2/STU3   | SI    |
| firmare                 | NO/NO | NO<br>P7M_ELETTRONICA<br>P7M_DIGITALE<br>AUTENTICATO | NO: non è necessario che il modulo sia firmato per essere presentabile; non è necessario indicare campi firma.<br>P7M_ELETTRONICA: per la presentabilità del modulo è necessario che tutti i firmatari abbiano apposto una firma elettronica o "debole" sul pdf<br>P7M_DIGITALE: per la presentabilità del modulo è necessario che tutti i firmatari abbiano apposto una firma digitale o "forte" sul pdf<br>(SOLO STU3) AUTENTICATO: richiede la firma elettronica solamente se l'intestatario del documento è diverso dall'utente connesso<br>API: STU2/STU3 | NO    |
| istruzioni_compilazione | NO/NO | Stringa, nessun limite                               | Testo che verrà comunicato tramite una finestra <i>pop-up</i> all'utente al termine delle routine di preparazione alla compilazione del modulo<br>API: STU2  | ""    |
| istruzioni_invio        | NO/NO | Stringa, nessun limite                               | Chiave indicante il testo mail che deve essere compilato e usato come ricevuta per l'invio dell'istanza tramite PEC. Nell'installazione basa sono disponibili due testi: mail_pec (default) e mail_suap (come vecchia versione).<br>API: STU2/STU3   | "1 "  |
| metodo_invio            | NO/NO | WS<br>MAIL<br>MAIL_ZIP<br>MINIMALE<br>SERVER         | WS: la presentazione dell'istanza passerà per ComUnica StarWeb<br>MAIL: viene inviata una mail con tutti i file, compilazioni e allegati, allegati al messaggio<br>MAIL_ZIP: l'unico allegato alla posta elettronica è un zip contenente tutti i file necessari<br>MINIMALE: viene allegato alla mail solamente il modulo principale firmato, i restanti file vengono referenziati con un link<br>SERVER: tutti i file, compresa la compilazione principale, sono  | MAIL  |

|                      |       |   |   |             |
|----------------------|-------|---|---|-------------|
|                      |       |   | referenziati con un link e la mail risulta priva di allegati<br>API: STU2/STU3  |             |
| procura              | NO/NO | Stringa, come da descrizione                  | Indica la possibilità di presentare in procura il modulo. Vi sono due configurazioni possibili:<br>"NO,,": indica che non è ammessa procura<br>"SI,<cod_modulo>,<campo_firma>": ammette procura e indica dove trovare il codice fiscale del firmatario riportando il codice del modulo (<cod_modulo>) di dichiarazione e il campo, all'interno di tale HTML, che conterrà il codice fiscale del soggetto abilitato (<campo_firma>).<br>API: STU2/STU3 | "NO,,"      |
| tipo_modulo          | SI/SI | PRINCIPALE<br>SECONDARIO                      | Indica qualora il modulo può essere usato per aprire un'istanza (PRINCIPALE) o se è solamente compilabile come modulo accessorio ad un altro (SECONDARIO)<br>API: STU2/STU3   | N.A         |
| urlhelp              | NO/NO | Stringa, 255                                  | Url relativo alla pagina informativa sul modulo o sulla scheda procedimento<br>API: STU2  | ""          |
| verifica_pagamento   | NO/NO | SI<br>NO                                      | Se attivata la funzionalità di "Reversale web" blocca la presentazione dell'istanza fino a quando non sarà compilata una reversale associata o dichiarata esente da pagamenti<br>API: STU2  | NO          |
| versione             | NO/SI | Stringa, 100<br>Formato:<br><numero>   <data> | Indica la versione di rilascio del modulo e la relativa data di modifica, utile per la gestione di sistemi qualità che richiedono le revisioni;<br>API: STU2/STU3   | N.A.        |
| contesto_procedurale | NO/SI | Stringa, 50                                   | La stringa da mostrare nell'eventuale porzione di contesto procedurale della segnatrice<br>API: STU2  | NULL        |
| segnatura            | NO/NO | Stringa, 50                                   | Prefisso dei campi, appartenenti ad una porzione anagrafica, che verranno usati per compilare la segnatrice<br>API: STU2/STU3   | "Titolar e" |
| in_bollo             | NO/NO | SI<br>NO                                      | Indica qualora la presentazione del modulo debba essere accompagnata dall'annullamento di una marca da bollo o meno. Attributo solamente semantico, non funzionale.<br>API: STU2  | NO          |
| forzabile            | NO/NO | SI<br>NO                                      | Indica la possibilità, da parte di utenti privilegiati, di modificare il metodo di invio di un'istanza in modo forzoso.<br>API: STU2/STU3   | NO          |
| generabile_da        | NO/NO | Stringa, 255                                  | Elenco, separato da virgola, delle istanze che possono generare questo  | NULL        |

|              |       |              |   |      |
|--------------|-------|--------------|---|------|
|              |       |              | modulo con la tecnica del 'Pratica da Pratica'<br>API: STU2/STU3  |      |
| urn          | SI/SI | Stringa, 255 | Nuovo identificativo univoco parlante introdotto in STU3 che collega il modulo al relativo Document<br>API: STU3  | N.A. |
| sub_urn      | NO/NO | Stringa, 255 | Sottoclassificazione del modulo secondo la nuova gestione di codifica STU3<br>API: STU3   | ""   |
| sub_urn_list | NO/SI | Stringa, 255 | Contiene l'elenco, separato da virgola senza spazi, delle possibili sotto classificazioni del modulo<br>API: STU3   | ""   |
| uo_globo     | NO/SI | Stringa, 255 | L'ufficio – secondo la classificazione Globo – che di default riceverà la modulistica se nessuna esplicita configurazione è stata precedentemente creata<br>API: STU3 | ""   |

### 4.3 Il meta tag "GLOBO\_modulofiglio"

Contrariamente al tag descritto nel precedente capitolo, quello indicato come tag "GLOBO\_modulofiglio" non è obbligatorio e può comparire diverse volte all'interno dell'intestazione del modulo.

Ogni occorrenza di questo meta crea una dipendenza tra il modulo in cui è contenuto e quello indicato nel tag stesso con la corretta configurazione dei suoi attributi.

Non è necessario che tutti i moduli secondari siano dipendenti da uno principale (potrebbero a loro volta dipendere da un secondario creando un terzo livello di compilazioni), ma non permesso indicare un modulo principale come accessorio di una qualsiasi altra compilazione.

Gli attributi disponibili sono i seguenti:

| Attributo            | Obbligatorio/Statico | Formato/Dominio  | Descrizione  | Default      |
|----------------------|----------------------|--|--|--------------|
| name                 | SI/SI                | "GLOBO_modulofiglio"                                     | Indica che il meta determina una dipendenza tra moduli e ne definisce le peculiarità<br>API: STU2/STU3   | N.A.         |
| codice_modulo_figlio | SI/SI                | Stringa, un codice modulo valido                         | Indica il codice di un modulo HTML dello standard Globo che è possibile compilare contestualmente alla presentazione del modulo attuale.<br>API: STU2/STU3   | N.A.         |
| numero obbligatorio  | SI/NO<br>SI/NO       | Intero positivo<br>OBBLIGATORIO<br>OPZIONALE<br>DINAMICO | Le diverse combinazioni di questi parametri possono creare le seguenti condizioni, bloccanti o meno per la presentazione dell'istanza:<br>DINAMICO, qualsiasi numero: non mostra il blocco relativo al modulo nell'interfaccia semaforica e non richiede compilazioni.<br>OPZIONALE, 0: è possibile allegare liberamente un numero di copie imprecisato da 0 a infinite ipotetiche<br>OPZIONALE, 1+: è possibile allegare fino a n copie, dove n è il valore del parametro numero<br>OBBLIGATORIO, 0: indica la necessità di allegare obbligatoriamente almeno una copia dell'allegato, non limitando tuttavia il numero massimo<br>OBBLIGATORIO, 1+: è richiesto esattamente il numero di copie indicato dal parametro numero<br>API: STU2/STU3 | N.A.<br>N.A. |
| urn_modulo_figlio    | SI/SI                | Stringa, 255   | Indica l'urn di un modulo HTML dello standard Globo che è possibile compilare contestualmente alla presentazione del modulo attuale.<br>API: STU3  |              |

#### 4.4 Il meta tag "GLOBO\_allegato"

L'ultimo meta appartenente al sistema Globo è il tag "GLOBO\_allegato" (questo valore deve diventare il name del meta per scatenare le funzionalità descritte).

Ogni occorrenza di questo elemento definisce una tipologia di documento che è possibile allegare ad ogni compilazione del modulo, rimanendo tuttavia opzionale se non è necessaria o prevista la possibilità di compiere tale azione.

Ecco il set completo di attributi definibili:

| Attributo          | Obbligatorio/Statico | Formato/Dominio                       | Descrizione   | Default |
|--------------------|----------------------|---------------------------------------|---|---------|
| name               | SI/SI                | "GLOBO_allegato"                      | Indica che il meta definisce una nuova categoria di documento allegabile alla compilazione<br>API: STU2/STU3  | N.A     |
| codice_allegato    | SI/SI                | Stringa, 100                          | Codice che distingue univocamente il tipo di allegato dagli altri disponibili nel modulo<br>API: STU2/STU3  | N.A     |
| descrizione_estesa | SI/SI                | Stringa, 255                          | Breve descrizione testuale dei contenuti previsti dall'allegato, per aiutare l'utente<br>API: STU2/STU3   | N.A.    |
| filename           | SI/SI                | Stringa, 255                          | Nome con cui verrà automaticamente rinominato il file una volta accettato nel sistema; se firmato, questo parametro non deve riportare alcuna estensione p7m<br>API: STU2/STU3  | N.A     |
| firmato            | NO/NO                | SI<br>NO                              | SI: indica che il file caricato dovrà essere firmato nelle stesse modalità con cui è richiesta la firma per la compilazione da cui dipende<br>NO: l'allegato può essere caricato sprovvisto di firma<br>API: STU2/STU3  | SI      |
| istruzioni         | SI/NO                | Stringa, nessun limite                | Url relativo alla pagina informativa sull'allegato<br>API: STU2   | N.A     |
| numero             | SI/NO                | Intero positivo                       | Il comportamento derivante dalla combinazione dei due attributi è del tutto simile a quello descritto per il meta "GLOBO_modulofoglio", con la peculiarità che al posto delle compilazioni si rende obbligatorio o opzionale il caricamento di un determinato numero di copie per l'allegato.<br>API: STU2/STU3 | N.A     |
| obbligatorio       | SI/NO                | OBBLIGATORIO<br>OPZIONALE<br>DINAMICO |   | N.A     |
| tipo_mime          | SI/NO                | Stringa, 50                           | Indica quale MIME TYPE dovrà avere il file per essere accettato nel sistema come allegato.<br>API: STU2/STU3  | N.A     |
| firmatari          | NO/NO                | Stringa, 255                          | Contiene le indicazioni sugli eventuali firmatari particolari dell'allegato. La sintassi specifica è contenuta nell'appendice A.  | N.A.    |

|                        |       |                |  |                           |
|------------------------|-------|----------------|--|---------------------------|
|                        |       |                | API: STU2/STU3   |                           |
| firmatari_label        | NO/NO | Stringa, 255+  | Indicare le etichette descrittive dei ruoli dei firmatari indicati con il precedente campo, separandole con una virgola<br>API: STU2/STU3        | N.A.                      |
| dimensione_m<br>assima | NO/SI | Reale positivo | Definisce la dimensione massima, in MB, di ogni singola copia dell'allegato, sovrascrivendo l'impostazione globale del portale<br>API: STU2/STU3 | Da configurazione portale |
| urn_allegato           | SI/SI | Stringa, 255   | Urn che distingue univocamente il tipo di allegato dagli altri disponibili nel modulo<br>API: STU3   |                           |

#### 4.5 Il meta tag "GLOBO\_mappa" e l'inserimento di una mappa nel modulo

NB. Questa funzionalità esiste solo per STU2.

Dalla versione 2.3 del prodotto Sportello Telematico è possibile inserire delle mappe, una o più a seconda delle necessità e soggette o meno a validazione, all'interno del modulo HTML in modo che l'utente possa:

- scorrere la visualizzazione e cambiare lo zoom della porzione di mappa tematica visualizzata
- specificare una delle seguenti geometrie:
  - Punto
  - Linea spezzata
  - Poligono chiuso
- cancellare, in caso di errori, la figura disegnata e specificarne una nuova
- 'zoomare' in automatico su un indirizzo o un immobile specificati nel modulo

Gli attributi specificabili per questo meta sono divisi in due tipologie: quelli specifici del modulo e quelli di override del portale. I primi, riportati nella tabella sottostante, sono specifici e immutabili per il modulo:

| Attributo | Obbligatorio/Statico | Formato/Dominio | Descrizione  | Default |
|-----------|----------------------|-----------------|--|---------|
| name      | SI/SI                | "GLOBO_mappa"   | Indica che il meta definisce una nuova porzione di modulo adibita alla visualizzazione cartografica  | N.A     |
| prefisso  | SI/SI                | Stringa, 255    | Specifica il prefisso che i campi di appoggio per tale mappa avranno, in modo da poter essere individuati univocamente ed utilizzati dagli script Esri   | N.A     |
| geometrie | SI/SI                | Stringa, 255    | Nel campo vengono indicate quali geometrie possono essere disegnate sulla mappa. Per abilitare la geometria è sufficiente scriverne il relativo placeholder; in caso di più tipologie di dato tracciabile separare i diversi valori con una virgola. <ul style="list-style-type: none"> <li>• point: si può specificare un punto</li> <li>• line: si può specificare una linea spezzata aperta</li> <li>• polygon: si può specificare un poligono</li> </ul> | N.A.    |

gli altri invece sono da riportare, con la notazione e modalità di override statico, nell'xml specifico del portale:

| Attributo        | Obbligatorio/Statico | Formato/Dominio | Descrizione   | Default |
|------------------|----------------------|-----------------|---|---------|
| url              | SI/SI                | url http        | Indica il MapServer che fornirà il servizio di cartografia da mostrare all'interno del campo mappa          | N.A     |
| layer_particelle | NO/SI                | Stringa, 255    | Il layer utilizzato dal CatViewWS per la validazione di particelle e UIU, necessario in caso siano presenti | N.A     |



|              |       |              |   |      |
|--------------|-------|--------------|---|------|
|              |       |              | pulsanti di auto-focalizzazione. Deve utilizzare le colonne COMUNE, SEZIONE, FOGLIO, NUMERO, SUBALTERNO   |      |
| layer_civici | NO/SI | Stringa, 255 | Il layer utilizzato dal GeoStreetsWS per la validazione di civi e indirizzi, necessario in caso siano presenti pulsanti di auto-focalizzazione. Deve utilizzare le colonne COMUNE, TOPONIMO, CIVICO | N.A. |

**n.b.:** i servizi pubblicati per le diverse funzionalità del widget debbono rispettare due vincoli:

- la tipologia del servizio deve essere *'tiled'*; per capire se la condizione risulta rispettata è sufficiente accedere tramite browser all'url specificato e, nell'elenco esposto, cercare le voci "Tile Info > Levels of Detail": se presenti il sistema scelto è conforme
- il sistema di coordinate deve essere lo stesso per l'url del MapServer e quello di identificazione features; per ottenere questa meta-informazione è sufficiente accedere attraverso il proprio browser agli url specificati e controllare la corrispondenza dei valori di *spatialReference*

Tuttavia la sola definizione del meta non è sufficiente ad attivare la funzionalità. È necessario infatti anche inserire, nella head del documento, la seguente dichiarazione di importazione script e stili di utilità:

```
<link rel="stylesheet" href="http://js.arcgis.com/3.7/js/dojo/dijit/themes/nihilo/nihilo.css">
<link rel="stylesheet" href="http://js.arcgis.com/3.7/js/esri/css/esri.css">
<script src="http://js.arcgis.com/3.7/"></script>
<script src="./validator/js/map_utils.js"></script>
```

seguita poi dallo script *inline* riportato nell'appendice C: lo script di attivazione mappa.

Per posizionare tale widget all'interno del modulo è sufficiente invece riportare questa porzione di codice HTML nel punto esatto dove si desidera che compaia il componente cartografico (sostituendo opportunamente il *placeholder* <prefisso> a quello specificato nel meta-tag che lo configura):

```
<div style='width:19cm'>
  <input style="pd4ml-display:none" id="<prefisso>_drawPoint" name="<prefisso>_drawPoint" type="button"
  value='Inserisci Punto' />
  <input style="pd4ml-display:none" id="<prefisso>_drawPolyline" name="<prefisso>_drawPolyline"
  type="button" value='Inserisci Linea' />
  <input style="pd4ml-display:none" id="<prefisso>_drawPolygon" name="<prefisso>_drawPolygon"
  type="button" value='Inserisci Poligono' />
  <input style="pd4ml-display:none" id="<prefisso>_cancelGeometry" name="<prefisso>_cancelGeometry"
  type="button" onclick="cancellaGeometry()" value='Cancella'>
  <input type="hidden" id="<prefisso>_serializedGeometry" name="<prefisso>_serializedGeometry" />
  <input type="hidden" id="<prefisso>_serializedExtent" name="<prefisso>_serializedExtent" />
</div>
<div id="<prefisso>_mapDiv" name="<prefisso>_mapDiv" style='width:19cm'></div>
```

Per invece disporre delle funzionalità di zoom a immobile, sia PT che UIU, o a indirizzo con o senza numero civico è sufficiente predisporre il seguente bottone (tra <> sono indicati i valori variabili della porzione di codice, analizzati sotto):

```
<input type='button' style="display:none; float: right" id="mappetta_doZoom<n>"  
name="mappetta_doZoom<n>" onClick='zoomToFeature("<prefisso>", "<tipologia> | <sezione>")'  
value="<Testo bottone>"/>
```

dove:

- n: numero intero positivo progressivo che identifica univocamente il numero di bottone per lo zoom automatico (utile in caso siano presenti diversi bottoni di auto-focus)
- prefisso: prefisso identificativo del widget mappa a cui applicare la visualizzazione
- tipologia: può valere "Catasto" o "Toponomastica", indica alla logica Javascript che tipologia di oggetto vuole focalizzare l'utente sulla cartografia *embedded*
- sezione: prefisso dei campi dai quali leggere i dati per la focalizzazione; abbinato a <tipologia> permette di specificare esattamente quali campi verranno utilizzati per l'interrogazione ai layer remoti

Per coinvolgere un campo mappa all'interno del sistema di validazione dati è sufficiente specificare le condizioni di obbligatorietà, la label identificativa e quanto altro necessario sul campo <prefisso>\_serializedGeometry.

## 4.6 Il meta tag "GLOBO\_immagine"

NB. Questa funzionalità esiste solo per STU2.

A partire dalla versione 2.3 del prodotto si dispongono delle funzionalità collegate al meta "GLOBO\_immagine": ovvero la possibilità di definire un sezione del modulo dedicata all'upload ed inserimento di un'immagine che verrà poi stampata nella versione PDF dello stesso; è possibile definire inoltre alcune validazioni circa la dimensione e il formato dell'immagine caricata.

Ogni meta-tag referencia un singolo widget di gestione delle figure e necessita dei seguenti attributi

Attributo	Obbligatorio/Statico	Formato/Dominio	Descrizione	Default
name	SI/SI	"GLOBO_immagine"	Indica che il meta definisce una nuova porzione di modulo dedicata al widget di upload e visualizzazione delle immagini	N.A
resize	SI/SI	SI NO	Specifica se il server deve ridimensionare un'immagine di dimensioni eccessive (valore 'SI') o rifiutarla indicando all'utente il limite (valore 'NO')	N.A
max_size	SI/SI	Stringa, 'WIDTHxHEIGHT'	Stringa contenente le dimensioni massime di altezza e larghezza dell'immagine da caricare	N.A
formati	SI/SI	Stringa, elenco separato da virgola	Elenco di <i>mime type</i> accettati; separare i valori con una virgola. Attualmente sono supportati: <ul style="list-style-type: none"> <li>image/jpeg</li> <li>image/gif</li> <li>image/png</li> <li>image/bmp, image/x-windows-bmp</li> </ul>	N.A
prefisso	SI/SI	Stringa, 255	Prefisso dei nomi/id dei campi di appoggio al funzionamento del widget	N.A

La porzione di codice necessaria a creare il widget di caricamento immagini è la seguente (tra <> le porzioni variabili successivamente analizzate):

```
<table class="Modulo">
  <tr class="Rigaintestazione">
    <td>
      <p class='Labeltabella'><fotografia_titolo>
        <input type="button" style="float: right; margin: 1px;" name="<prefisso>_cancella"
          id="<prefisso>_cancella" value="Cancella" onclick='cancellaUpload("<prefisso>")' />
        <input type="submit" style="float: right; margin: 1px;" name="<prefisso>_submit"
          id="<prefisso>_submit" value="Upload" onclick='startUpload("<prefisso>")' />
        <input name="<prefisso>_file" id="<prefisso>_file" type="file" style="float: right;"/>
      </p>
    </td>
  </tr>
  <tr style='height: 360px'>
    <td style='text-align: center; align: center;'>
      <p style='pd4ml-display: none' id='<prefisso>_formato'><fotografia_formati></p>
      <p style='pd4ml-display: none' id='<prefisso>_dimensione'><fotografia_dimensioni></p>
      <p id="<prefisso>_upload_process" name="<prefisso>_upload_process">
```

```

class='upload_process'>Loading...<br/></p>
<p id="<prefisso>_result" name="<prefisso>_result" style='display: none'></p>
<input type='hidden' id="<prefisso>_valueField" name="<prefisso>_valueField" />
<iframe id="immagine1_upload_target" name="immagine1_upload_target" src="about:blank"
style="width:0;height:0;border:0px solid #fff;"></iframe>
<img id="<prefisso>_destImmagine" name="<prefisso>_destImmagine" style="height: 100%" src="#"/>
</td>
</tr>
</table>

```

dove:

- **<prefisso>**: prefisso indicato nell'omonimo attributo del meta-tag che parametrizza il funzionamento del widget
- **<fotografia\_titolo>**: breve descrizione del contenuto previsto per l'immagine che verrà caricata, funziona esattamente come l'etichetta di un campo
- **<fotografia\_formati>**: breve elencazione delle estensioni accettate dal sistema per quel widget immagine
- **<fotografia\_dimensioni>**: indicazioni testuali sulle dimensioni massime dell'immagine ed eventualmente della possibilità o meno di un resize dinamico

Per coinvolgere un campo immagine all'interno del sistema di validazione dati è sufficiente specificare le condizioni di obbligatorietà, la label identificativa e quanto altro necessario sul campo **<prefisso>\_valueField**.

## 4.7 Il meta tag "GLOBO\_report"

Ogni meta così nominato scatena l'automatismo di creazione di un report PDF – composto dai dati della compilazione del modulo che lo referencia – che verrà allegato e incluso nell'istanza. Questa funzionalità è disponibile dalla versione 2.5 del prodotto.

La sua presenza, oltre che nei moduli Globo, può essere riscontrata anche negli effettivi file che implementano il report stesso, ai quali sarà dedicato uno specifico paragrafo; utilizzato contestualmente ad un modulo principale o secondario indica le eventuali personalizzazioni che prevede tale modulo rispetto alle implementazioni standard fornite dal report di eventuali specifici attributi o meta attributi. Quindi a parte il codice identificativo ed il name nessun altro attributo è obbligatorio e costituisce esso stesso già un override

Attributo	Obbligatorio	Formato/Dominio	Descrizione	Default
name	SI	"GLOBO_report"	Indica che il meta determina la correlazione tra un modulo e la possibilità di generare report associati alle compilazione del modulo stesso API: STU2/STU3	N.A.
codice_report	SI	Stringa, 100 caratteri	Indica il codice identificativo univoco del report a livello di installazione API: STU2/STU3	N.A.
descrizione	NO	Stringa, 255 caratteri	Il valore di 'descrizione' verrà utilizzato come etichetta parlante per referenziare il file all'utente – nell'interfaccia grafica API: STU2/STU3	N.A.
substitutions	NO	Area di testo, illimitato	Le regole di calcolo e sostituzione dei segnaposto all'interno del report; la specifica sintassi è trattata nel paragrafo dedicato API: STU2/STU3	N.A.
firmare	NO	Stringa, 'SI' o 'NO'	Indica la necessità o meno che l'utente fornisca una versione firmata del PDF del report API: STU2/STU3	N.A.
orientation	NO	Stringa, 'portrait' o 'landscape'	Specifica se il PDF generato sarà un documento verticale (portrait) o orizzontale (landscape) API: STU2/STU3	N.A.
format	NO	Stringa, <px> <template>	Stabilisce il rapporto di scalatura tra le dimensioni in px del documento renderizzato dal browser e le dimensioni di stampa; consulta <a href="#">questa pagina</a> cercando informazioni sui parametri htmlWidth pageFormatName API: STU2/STU3	N.A.
Insets	NO	Stringa, "top,left,bottom,right, mm   pt"	Indica i margini di rientro da applicare nella stampa PDF. 0,0,0,0,mm genera un documento a dimensione pagina. API: STU2/STU3	N.A.
urn_report	SI/SI	Stringa, 255	Indica l'urn identificativo univoco del report a livello di installazione API: STU3	

## 4.8 Il meta-tag "GLOBO\_function"

Di appoggio al meta relativo ai report troviamo invece il meta di dichiarazione delle funzioni di calcolo dei valori complessi, identificato come nel titolo.

Oltre al *name* identificativo, tale tag richiede la presenza solamente di 2 attributi:

- *function*: qui dichiariamo il nome funzione, quello che sarà indicato quando il valore di un placeholder dovrà essere calcolato tramite una logica complessa
- *code*: in quest'altro attributo scriviamo il codice PHP che implementa la logica di calcolo del valore; il valore che si vuole esporre come risultato deve essere oggetto di un'operazione *return* (no *print*, no *echo*)

Esattamente come la dichiarazione del report, è obbligatoria all'interno dei file di report stessi ma facoltativa all'interno dei moduli compilabili, se l'implementazione di default è sufficiente

## 4.9 I campi di override

Per redattore, al fine di modificare delle meta informazioni circa il modulo, i rapporti con i moduli secondari e gli allegati, è stata creata una naming convention sui campi di input in modo che, se aderente a questa convenzione, il suo valore sia interpretato in modo da scatenare l'override delle funzionalità.

È permessa la sovrascrittura dinamica dei soli attributi che, nelle tabelle ai paragrafi 4.2 - 0 - 4.4, sono indicati come non statici.

Tale semplice protocollo prevede che il name del campo combaci uno di questi pattern (parametri variabili tra <>):

- GLOBO\_modulo\_c<codice\_modulo>: indica che tale campo conterrà gli override relativi al comportamento del modulo principale o dell'intera istanza, a seconda degli attributi che andrà a sovrascrivere.
- GLOBO\_modulofiglio\_c<codice\_modulo\_figlio>: il contenuto di questo campo è oggetto di parsing per l'override della dipendenza tra il modulo che contiene il campo ed il modulo figlio indicato da <codice\_modulo\_figlio>
- GLOBO\_allegato\_c<codice\_allegato>: dovrà contenere le nuove peculiarità della relazione tra il modulo che contiene il campo ed il suo allegato avente codice <codice\_allegato>

Per far sì che il sistema STU interpreti correttamente le nostre sovrascritture dinamiche il value di tali campi deve consistere una serie di coppie di chiavi-valore testuali, quest'ultimo racchiuso tra 'apici singoli' o "apici doppi", separate tra di loro da caratteri di spazio e correlate tra loro dal carattere "="; esattamente come avviene la definizione degli attributi di un tag HTML. La chiave di ogni coppia deve corrispondere esattamente all'attributo che vuole sovrascrivere ed il valore appartenere al dominio specificato per tale meta-informazione (come indicato in ogni tabella dello specifico paragrafo dedicato).

La creazione di questi campi può avvenire durante la compilazione solo se il modulo è stato precedentemente indicato come dinamico; in caso contrario il sistema ignorerà il valore qualunque esso sia.

Nella versione STU3 del motore di modulistica si rendono inoltre disponibili ed eventualmente necessari i seguenti campi:

- GLOBO\_sub\_urn: indica che tale campo conterrà la classificazione specifica del modulo necessaria per un esatto calcolo dello smistamento e metodo di invio alle unità organizzative dell'ente.



## 4.10 Javascript: funzioni particolari

Aderendo alle diverse naming convention e utilizzando i nomi riservati previsti dal sistema client è possibile predisporre alcuni automatismi e richiami automatici di routine; l'implementazione di ciascuna è opzionale e dettata dalle necessità del modulo stesso (tant'è che la mancata dichiarazione non genera errore):

- **afterRecupero**: chiamare una funzione in tale modo farà sì che venga eseguita automaticamente dopo l'avvenuta ricompilazione del modulo con i dati precedentemente inseriti dall'utente; non chiamata in caso di compilazioni nuove
- **afterLoad**: tale routine viene eseguita al termine del caricamento del modulo HTML, quando tutte i comportamenti di avvio e preparazione alla compilazione sono stati completati
- **afterCheck**: ad ogni interazione con i campi viene eseguito il controllo di correttezza dei valori inseriti o modificati; subito dopo questo controllo è possibile definire un comportamento personalizzato, inserendone la logica nella funzione denominata **afterCheck**
- **beforeCheck**: viene eseguita prima del controllo finale di tutti i campi compilati e le relazioni, quando viene premuto il bottone di invio della compilazione
- **beforeSave**: a differenza della funzione di cui al punto precedente, l'esecuzione di **beforeSave** è scatenata dopo il controllo finale dei dati e solo se nessuna regola di validazione è stata infranta

I valori di ritorno sono ignorati e non vengono catturate eventuali eccezioni scatenate da tali porzioni di codice.

### 4.10.1 Javascript in STU 3

In STU3 la logica di funzionamento di un modulo all'interno del portale funziona diversamente infatti, attraverso una serie di elaborazioni fatte al caricamento del modulo, viene presa in considerazione solamente tutto ciò che si trova all'interno del tag `<body>`; appunto per questo tutti gli `<script>` e `<style>` descritti all'interno del tag `<head>` viene ignorato.

NB. Non solo in STU3 tutte le funzioni Javascript sono gestite diversamente (inserite direttamente in validator), ma le varie funzioni *after/before* sono disabilitate in favore dell'utilizzo degli attributi di validator.

### 4.10.2 Funzioni Personalizzate

Delle molte funzioni esistenti all'interno del progetto validator e che vengono utilizzate esclusivamente lato sportello, di seguito un elenco delle function maggiormente utilizzate all'interno della redazione di moduli.

NB. Queste funzioni, per funzionare correttamente, devono essere richiamate all'interno degli attributi `dynamic_value` / `dynamic_check` / `mandatory` / `enabled`

#### 4.10.2.1 `getDataFromString(date)`

Dato in input una stringa viene restituita una data in formato `dd/mm/yyyy`



#### 4.10.2.2 copiaDatiDa(prefisso\_dest, campo\_origine, elenco\_campi)

Dato in input un prefisso destinatario, un prefisso di origine ed un array di campi, in questi ultimi vengono copiati tutti i valori presenti nei medesimi campi presenti nel campo di origine.

Esempio: copiaDatiDa("Progettista", "Richiedente", new array("\_Nome", "\_Cognome", etc...))

##### 4.10.2.2.1 dateDifference(date1, date2, expected\_value)

Dato in input una data di inizio ed una data di fine viene restituita la differenza (in giorni mesi o anni) tra le due date.

Le date devono essere in formato dd/mm/yyyy

L'ultimo parametro (expected\_value) accetta i seguenti valori:

1. **y** se si vuole la differenza in anni
2. **m** se si vuole la differenza in mesi
3. **d** se si vuole la differenza in giorni

Vengono considerate tutte le eventuali casistiche (come la presenza di un anno bisestile)

## 4.11 Interagire con la libreria PD4ML

Ogni compilazione ed ogni modulo vengono trasformati automaticamente in una stampa pdf strutturalmente, esteticamente e contenutisticamente fedele grazie all'elaborazione eseguita tramite la libreria [PD4ML](#) distribuita da Globo con ogni installazione di STU.

Può essere utile al redattore poter pilotare alcune piccole peculiarità che la stampa PDF deve o non deve presentare rispetto alla visualizzazione telematica: questo capitolo spiega gli accorgimenti necessari ed il codice da scrivere per ottenere salti di pagina forzati, fondo pagina e altro.

### 4.11.1 Il tag `globo_salto_pagina`

La generazione del documento in formato Adobe è eseguita con attiva l'opzione – `smarttablesplit`, che permette alla libreria di non spezzare le tabelle HTML tra due pagine ma anticipare eventualmente il passaggio al successivo foglio.

Se si avesse la necessità di forzare tale salto di pagina è sufficiente inserire, nel punto esatto in cui si vuole ottenere tale modifica, il seguente tag:

```
<globo_salto_pagina />
```

### 4.11.2 Il tag `globo_fondo_pagina`

Particolarmente utilizzati nei cartacei per contenere note, riferimenti o sezioni dedicate sono i fondo pagina.

Anche nei moduli telematici di Globo s.r.l. è possibile definire del codice HTML che, inserito all'interno del tag `globo_fondo_pagina`, comparirà ancorato al lato inferiore di uno o più fogli della stampa PDF; sono richiesti obbligatoriamente due attributi per questo tag:

- `areahight`: da valorizzare di default a "150", indica l'altezza in punti del fondo pagina; aumentare o diminuire a seconda del contenuto
- `scope`: indica la pagina o l'insieme di pagine in cui comparirà il footer;

N.B.: ulteriori opzioni inseribili in un fondo pagina sono disponibili a [questo indirizzo](#); la routine di validazione del modulo lascerà inalterati tutti gli attributi.

Inoltre è necessario che il DOM contenuto nel `globo_fondo_pagina` segua le seguenti direttive:

- Per non generare il contenuto attaccato al bordo del foglio è necessario inserire nello *style* dell'elemento contenitore un attributo CSS `margin` fisso a 28pt;
- Per non mostrare la porzione di footer anche nella visualizzazione tramite browser è necessario che il CSS dell'elemento contenitore contenga la seguente definizione:

```
display: none; pd4ml-display: inline;
```

#### 4.11.3 Nascondere o mostrare elementi nel PDF

Nel corpo del documento è possibile dettare regole specifiche sulla visibilità di determinate porzioni o singoli campi, definendo un diverso comportamento e/o stile tra la visualizzazione telematica e la rappresentazione del formato Adobe.

- pd4ml-display: emula la corrispondente proprietà display del CSS standard sull'elemento a cui è attribuito ed i suoi figli, ma solamente nella stampa PDF.
- pd4ml-visibility: emula la corrispondente proprietà visibility del CSS standard sull'elemento a cui è attribuito ed i suoi figli, ma solamente nella stampa PDF.

La *reference* completa per il supporto CSS da parte della libreria PD4ML è disponibile [qui](#).

## 4.12 Integrazione con SOLO1

Per convenzione, tutti gli input definiti all'interno del modulo devono rispettare una determinata struttura per essere correttamente integrati all'interno dell'applicativo Solo1.

### 4.12.1 La tassonomia

La tassonomia fa riferimento alla scheda **Classificazione** di Solo1 dove vengono valorizzate le opzioni multiple o singole (checkbox e radio) di una pratica per poterne diversificare il contenuto (per esempio capire se un permesso di costruire è in sanatoria)

Tutti i radio e checkbox devono essere dichiarati con il prefisso **Tassonomia\_**

Entrambi gli elementi necessitano di un div (che viene utilizzato come raggruppatore) che deve essere dichiarato una volta soltanto.

All'interno di questo contenitore verranno specificate eventuali regole di obbligatorietà o di abilitazione dei campi (mandatory ed enabled)

#### 4.12.1.1 Dichiarazione di radio button all'interno del modulo HTML:

```
<div id="Tassonomia_Radio" name="Tassonomia_Radio" label="Tipologia" [mandatory/enabled]> </div>
<input id="Tassonomia_Radio1" name="Tassonomia_Radio" type="radio" value="1" [dynamic_check]/>
<input id="Tassonomia_Radio2" name="Tassonomia_Radio" type="radio" value="2" [dynamic_check]/>
...
<input id="Tassonomia_Radio[n]" name="Tassonomia_Radio" type="radio" value="[n]"
[dynamic_check]/>
```

#### 4.12.1.2 Dichiarazione di checkbox all'interno del modulo HTML:

```
<div id="Tassonomia_Checkbox" name="Tassonomia_Checkbox" label="Tipologia"
[mandatory/enabled]> </div>
<input group="Tassonomia_Checkbox" id="Tassonomia_Checkbox1" name="Tassonomia_Checkbox1"
type="checkbox" value="SI" [dynamic_check]/>
<input group="Tassonomia_Checkbox" id="Tassonomia_Checkbox2" name="Tassonomia_Checkbox2"
type="checkbox" value="SI" [dynamic_check]/>
...
<input group="Tassonomia_Checkbox" id="Tassonomia_Checkbox[n]" name="Tassonomia_Checkbox[n]"
type="checkbox" value="SI" [dynamic_check]/>
```

### 4.12.2 I parametri

I parametri fanno riferimento alla scheda **Parametri** di Solo1 dove vengono importate tutte le informazioni aggiuntive di una pratica.

#### 4.12.2.1 Input

```
<input class="inputNormal" id="Parametro_Input" name="Parametro_Input" type="text"
label="Protocollo" [mandatory/enabled/dynamic_value/format]
```

- Se ad un input esiste una data di riferimento, essa dovrà essere dichiarata con il seguente name: Parametro\_Input\_Data
- Se ad un input esiste una nota di riferimento, essa dovrà essere dichiarata con il seguente name: Parametro\_Input\_Note

#### **4.12.2.1.2 Select**

```
<select class="inputNormal" id="Parametri_Select" name="Parametri_Select" style="width: 100%;" label=""  
[mandatory/enabled]>  
  <option value=""> </option>  
  <option value="valore 1"> valore 1 </option>  
  <option value="valore 2"> valore 2 </option>  
  ...  
  <option value="valore [n]"> valore [n] </option>  
</select>
```

#### **4.12.2.1.3 Textarea**

```
<textarea class="inputNormal" id="Parametri_Textarea" name="Parametri_Textarea" style="width: 100%;  
height: 5cm;" label="Textarea" [mandatory/enabled/dynamic_value]> [testo] </textarea>
```

#### **4.12.3 Esempi**

Documentazione di riferimento: [Esempi di redazione\Moduli\ Formazione modulistica.html](#)

## 5 APPENDICE B: I FIRMATARI DEGLI ALLEGATI

Qualora, durante la redazione di un procedimento o modulo, risulti necessario far firmare un allegato ad un soggetto diverso, o a soggetti diversi, da quello della compilazione da cui l'allegato stesso dipende, si rende necessario l'utilizzo dell'attributo 'firmatari' del meta-tag *GLOBO\_allegato*.

La sintassi ed i casi d'uso sono dichiarabili nelle seguenti modalità:

- L'attributo viene omesso: nel caso in cui non venga proprio dichiarato all'interno dell'HTML il sistema considererà in automatico firmatari dell'istanza validi anche per i file allegati. Questo comportamento è determinabile anche dalla parola chiave *EREDITA*, particolarmente utile in caso di override: se infatti la definizione originale dell'allegato prevede firmatari particolari
- Il firmatario è indicato nel modulo principale: per ogni singolo firmatario, il cui codice fiscale è scritto in un campo noto del modulo da cui l'allegato dipende direttamente, è necessario riportare il *name* del campo all'interno dell'attributo (stando attenti a considerare *MAIUSCOLE/minuscole* correttamente). In caso di più firmatari è sufficiente separare le diverse porzioni con uno spazio.
- Il firmatario è indicato in una compilazione secondaria: quando il codice fiscale del firmatario è inserito in un modulo accessorio a quello da cui l'allegato dipende direttamente è possibile indicarlo valorizzando *firmatari* con un dato stringa così composto:
  - La prima porzione indica il codice identificativo del modulo che conterrà l'informazione. Tale modulo deve essere presente nel set di dipendenze del compilabile principale.
  - Il carattere '.' separa le due porzioni logiche del valore.
  - La seconda parte deve corrispondere al *name* del campo, obbligatoriamente contenuto nel modulo secondario dichiarato, che conterrà il codice fiscale del firmatario.

Se il modulo secondario indicato è stato compilato un numero multiplo di volte saranno richiesti come firmatari tutti i soggetti indicati in ogni singola compilazione. Anche in questo caso è possibile separare le diverse porzioni con un singolo spazio; è possibile usando inoltre soluzioni miste tra modulo principale e secondari.

Se non è stata ancora compilata neanche una copia di tale modulo, all'utente sarà inibita la possibilità di caricare il file e presentato un messaggio contenente indicazioni sui moduli mancanti per completare l'insieme dei firmatari dell'allegato.

Alcuni esempi d'uso:

- "AventeTitolo\_CodiceFiscale": un solo firmatario, contenuto nel modulo da cui dipende l'allegato nel campo 'AventeTitolo\_CodiceFiscale'
- "AventeTitolo\_CodiceFiscale,Progettista\_CodiceFiscale": due firmatari, entrambi nel modulo stesso relativo all'allegato, i cui codici fiscali sono contenuti nei campi indicati

- "SCIA-2.Titolare\_CodiceFiscale": il firmatario è contenuto nel modulo 'SCIA-2', all'interno del campo 'Titolare\_CodiceFiscale'; un firmatario per ogni compilazione del modulo
- "SCIA-2.Titolare\_CodiceFiscale,SCIA-3.Titolare\_CodiceFiscale": i firmatari sono i soggetti titolari delle compilazioni dei moduli secondari 'SCIA-2' e 'SCIA-3'; numero di firmatari direttamente dipendente dal numero di compilazioni
- "Titolare\_CodiceFiscale,SCIA-3.Titolare\_CodiceFiscale": soluzione mista; il primo firmatario è preso dal modulo stesso relativo all'allegato, mentre l'altra porzione indica il campo 'Titolare\_CodiceFiscale' del modulo secondario 'SCIA-3' (un firmatario per ogni compilazione)

## 6 APPENDICE C: LO SCRIPT DI ATTIVAZIONE MAPPA

```
<script>
var metas = document.getElementsByTagName('meta');
var prefisso = "";
var codice_mod_req = "";
for (i=0; i<metas.length; i++) {
    if (metas[i].getAttribute("name") == "GLOBO_mappa") {
        prefisso = metas[i].getAttribute("prefisso");
    }
    if (metas[i].getAttribute("name") == "GLOBO_modulo") {
        codice_mod_req = metas[i].getAttribute("codice_modulo");
    }
}
var url_ser = "";
var xmlhttp;
if (window.XMLHttpRequest) {
    xmlhttp=new XMLHttpRequest();
} else {
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function() {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
        url_ser = xmlhttp.responseText;
    }
}
xmlhttp.open("GET","/sportello_telematico/get_url_mappa/"+codice_mod_req+"/"+prefisso,false);
xmlhttp.send();
dojo.require("esri.map");
dojo.require("esri.layers.agstiled");
dojo.require("esri.toolbars.draw");

var app = {
    map:null,
    toolbar:null,
    markerSymbol:null,
    lineSymbol:null,
    fillSymbol:null,
    url: url_ser,
    savedGeometry: null,
    savedExtent: null
}
dojo.ready(init);
</script>
```



## 7 APPENDICE C: GESTIONE DELLE VERSIONI E LA POSIZIONE DI GLOBO

Dalla versione 2.6 dell'applicativo l'attributo 'versione' del meta tag principale viene utilizzato per gestire le informazioni relative al versionamento delle modifiche strutturali al modulo.

Il campo segue la presente struttura generale:

<numero\_versione> | <data\_rilascio>

Il **numero** è **obbligatorio**, almeno di 1 carattere – senza vincoli di alcun tipo sul formato e numero di livelli di numerazione; se la data viene omessa il portale vi attribuisce quella della pubblicazione in formato gg/mm/aaaa.

Tali informazioni poi verranno ricopiate all'interno del modulo nelle posizioni occupate dai seguenti segnaposto:

- {globo\_placeholder\_versione\_numero}: verrà sostituito con il numero di versione (il valore prima della ' | ')
- {globo\_placeholder\_versione\_data}: verrà sostituito con la data del rilascio (il valore dopo il carattere ' | ')

N.B.: Sulle modalità di gestione di tale informazione non vi sono indicazioni specifiche, la descrizione che segue è relativa alle modalità tenute da Globo s.r.l.

### 7.1 Gestione della versione da parte di Globo s.r.l.

Globo s.r.l. ha definito alcune politiche interne per la gestione del cambio di versione di un modulo, che tuttavia non impone né richiede dal momento in cui un soggetto di terze parti volesse sviluppare della modulistica STU-compatibile:

- Cambio di versioni minor (1.x):
  - Un cambio legislativo modifica la logica di controllo dei dati
  - Un cambio legislativo modifica la comunicazione ed i riferimenti di legge
  - Una evoluzione tecnica necessita di essere tracciata e distinguere quali compilazioni sono avvenute prima e dopo la sua introduzioni
- Cambio di versioni major (x.0):
  - Un cambio legislativo modifica la struttura del documento
  - Un'evoluzione tecnica porta all'introduzione di nuovi meta-tag

## 8 APPENDICE D: LA SINTASSI DELLE SUBSTITUTION

Verranno qui approfondite le peculiarità relative alla sintassi dei placeholder/substitutions utilizzati nei report di Sportello Telematico.

Lo scopo di tale elemento è predisporre il documento HTML a contenere, in tale posizione, un **valore** che verrà **calcolato e sostituito** per la stampa PDF dal motore dell'applicazione. Praticamente viene posta un'uguaglianza tra una porzione formale del placeholder, ovvero cosa dovrà essere cercato nel documento, e una funzionale, in pratica con che valore andrà effettivamente sostituito in fase di stampa.

Genericamente ha una sintassi che rispetta il seguente pattern:

`!<nome_placeholder>[=<namespace_placeholder>:<specifiche>]`

Il quale viene interpretato in questo modo:

- `!`: necessario per identificare un placeholder; qualsiasi direttiva che non inizia con tale carattere genera un errore in upload del report
- `nome_placeholder`: stringa identificativa della substitution nel documento
- `=`: separa la porzione di dichiarazione formale (a sinistra) da quella funzionale (a destra); opzionale
- `Namespace_placeholder`: indica come andranno trattate ed elaborate le effettive specifiche; opzionale – verranno analizzati singolarmente all'interno dei loro sottoparagrafi
- `Specifiche`: indicazioni su come calcolare il valore del placeholder, in base al namespace

Se la porzione opzionale – racchiusa nel pattern tra parentesi quadre – risulta omessa, tale placeholder viene trattato nel seguente modo:

`!<nome_placeholder>=valore:<nome_placeholder>]`

### 8.1 Namespace “valore”

La keyword identifica che il valore verrà estratto direttamente dai valori compilati dall'utente per l'istanza, cercando un valore campo denominato esattamente come nelle specifiche – se omesso, nominato esattamente come il placeholder privo di ‘!’ iniziale. Viene rispettata la gerarchia dell'istanza: il valore viene prima cercato nei campi dell'istanza principale. Il valore non subisce alcuna trasformazione rispetto a quanto digitato dall'utente.

#### 8.1.1 Esempi:

##### 8.1.1.1 Dichiarazione di una Substitution

```
substitutions="!denominazione=istanza:ente_compilazione  
!identificativo=istanza:identificativo_suap  
!codice_pratica=funzione:getCodice$w  
!comune=istanza:ente_compilazione  
!provincia=valore:Titolare_ProvinciaSede"
```

### 8.1.1.2 Richiamare una variabile dichiarata nella Substituion

`<div [attributes]><span [attributes]>Sportello Unico Attività produttive di !denominazione </span></div>`

La keyword !denominazione verrà sostituita con il valore ente\_compilazione presente nell'istanza.

La stessa logica viene utilizzata per tutte le altre keywords

## 8.2 Namespace "istanza"

Per risolvere questi placeholder vengono usati dei meta-dati relativi all'istanza che non sono riportati in alcun campo e per i quali non è conveniente dover scaricare sul redattore ogni volta l'onere di sviluppare una funzione specifica. La tabella elenca le possibili specifiche implementate

PLACEHOLDER	VALORE	ESEMPIO
istanza:destinatario_ufficio	Codice identificativo dell'ufficio destinatario dell'istanza	UOSUAP
istanza:destinatario_amministrazione	Codice dell'amministrazione destinataria dell'istanza	c_a794
istanza:ente_compilazione	Nome esteso dell'amministrazione destinataria	BERGAMO
istanza:identificativo_suap	Identificativo a livello nazionale del SUAP competente per la pratica	1719
istanza:data_compilazione	La data di ultima modifica della compilazione	03/09/2015
istanza:cf_compilatore	Codice fiscale del soggetto che ha compilato la pratica	CVDNDR92C08A794W
Istanza:codice_modulo	Il codice del modulo a cui è allegato il report	SCIA-A
Istanza:descrizione_modulo	La descrizione estesa del modulo a cui è allegato il report	Segnalazione certificata inizio/modifica attività
Istanza:firmatari	Elenco, separato da virgola, dei codici fiscali richiesti come firmatari della compilazione a cui è allegato il report	CVDNDR92C08A794W, PBLSFN85R06A794E
Istanza:firmatari_label	Elenco, separato da virgola, dei codici fiscali- completi di label descrittiva - richiesti come firmatari della compilazione a cui è allegato il report	Il progettista (CVDNDR92C08A794W), il dirigente (PBLSFN85R06A794E)

### 8.3 Namespace “funzione”

Quando un valore non è compreso tra i valori di compilazione dell'istanza o nei relativi metadati, è necessario sviluppare una specifica funzione che generi il codice HTML da sostituire a tale segnaposto. La funzione deve essere scritta in sintassi PHP

L'elaborazione avviene in questo modo:

- Al caricamento:
  - Al caricamento del report – o degli override del modulo – vengono estratti i placeholder di tipo “funzione” e viene letto, dalla loro porzione di specifiche, il nome della funzione che genererà il loro valore
  - Viene quindi cercato un meta tag di tipo 'Globo\_function' il cui attributo 'function' sia valorizzato con lo stesso identificativo letto al punto precedente: quello sarà il codice da eseguire in fase di elaborazione
- All'elaborazione:
  - Viene eseguita la funzione tramite l'istruzione 'eval'
  - Viene preso il valore di ritorno dell'esecuzione e sostituito al placeholder, senza alcun controllo; eventuali stream verranno ignorati

Alcune specifiche per scrivere il codice:

- Deve avere sintassi PHP, senza particolari requisiti sulla versione di compatibilità
- **Non** servono i tag di apertura o chiusura del codice <?php ?>
- È possibile accedere alla variabile \$istanza, di tipo Compilazione\_globo
- **Non** dichiarare una function ma riportare semplicemente il codice da eseguire, garantendo sempre un valore di return.