

| | | | |
|---|---|---------------------|---|
|  | Descrizione commessa | pag. 1 di 30 |  |
| | Decoder 2D-Plus : Utilizzo di strutture dati vantaggiose | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Strutture dati interpretate dal Decoder 2D-Plus e vantaggiose nell'utilizzo del codice grafico bidimensionale

| Release | Data | Modifiche effettuate | Redatto (acronimo) | Approvato (acr. Pm) |
|---------|------------|--|--------------------|---------------------|
| 0.1 | 30/06/2009 | Versione iniziale; | ESP | |
| 0.2 | 07/07/2009 | Seconda revisione; | ADI | SFN |
| 0.3 | 14/07/2009 | Cambio Commessa e titolo del documento; rivisitazione struttura del testo; | SFN | n.a. |
| 1.0 | 13/08/2009 | Riscrittura "garanzia integrità ed autenticità" nel caso XML/XSL; strutture CSS ed FO; | SFN | n.a. |
| 1.1 | 24/08/2009 | Cod.Appl. FO Ibex ed FO Standard; inserimento paragrafo "generazione PDF" con CSS ed FO; | SFN | n.a. |
| 1.2 | 09/09/2009 | Note esplicative nell'uso dei file multipli (5.2.1.3); correzioni nel Formato <i>XML/XSL-FO with PKCS#7</i> (5.2.3); correzioni nel capitolo Generazione di un file PDF (5.4) | SFN | n.a. |
| 1.3 | 05/11/2009 | Nuovo InfoApp; utilizzo di xsl:param, per la generazione concorrente di file PDF a partire da file XSL-FO; riserva dei codici 207/208; spiegazione dell'uso di FO standard con renderizzatore proprietario | SFN | n.a. |
| 1.4 | 19/01/2010 | Introduzione all'uso del namespace nei file XML | ESP | SFN |
| 1.5 | 08/02/2010 | Diverse modifiche nella parte relativa XML/XSL-FO | ESP | SFN |
| 1.6 | - | - | - | - |
| 1.7 | 22/04/2010 | Modifica alla parte relativa il namespace XML | ADI | n.a. |
| 1.8 | 14/12/2010 | Aggiunta nuovo SHA256, altre modifiche e correzioni Modifica Paragrafo 5.2.1.2 | ESP | |
| 1.9 | 28/2/2011 | Modifica/Correzione Paragrafo 5.4.1.2.2 | ESP | |
| 1.10 | 9/3/2011 | Modifica/Correzione Paragrafo 5.4.1.2 e 5.4.1.2.1 | ADI | |
| 1.11 | 16/6/2011 | Correzioni parte relativa XSL-FO Paragrafo 5.4.1.2.1 | ADI | |
| 1.12 | 09/11/2011 | Modifica ed aggiuta XSL-FO Paragrafo 5.4.1.2.1 | ESP | n.a. |

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 2 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

| | | | | |
|------|------------|---|-----|--|
| 1.13 | 06/06/2012 | Correzione paragrafo relativo ad XSL-FO | ADI | |
| 2.0 | 15/09/2016 | Aggiornamento nuove tipologie 2D-Plus | MNG | |

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 3 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

SOMMARIO

| | |
|--|----------|
| 1 DEFINIZIONI E ACRONIMI..... | 3 |
| 2 DOCUMENTAZIONE DI RIFERIMENTO..... | 3 |
| 3 SCOPO DEL DOCUMENTO..... | 4 |
| 4 STRUTTURE DATI GESTITE..... | 5 |
| 4.1 CODICI PER FORMATI PROPRIETARI..... | 5 |
| 4.2 CODICI PER FORMATI STANDARD..... | 5 |
| 4.3 TABELLA RIEPILOGATIVA CODICI APPLICAZIONE..... | 6 |
| 5 DESCRIZIONE CODICI APPLICAZIONE..... | 8 |
| 5.1 FORMATI PROPRIETARI: UTILIZZO..... | 8 |
| 5.2 FORMATI STANDARD: UTILIZZO..... | 8 |
| 5.2.1 <i>Formato XML/XSL (CodAppl 194, 195, 199, 205): GENERALITÀ</i> | 9 |
| 5.2.1.1 Garanzia sull'integrità ed autenticità dei file XSL e dei file esterni..... | 9 |
| 5.2.1.2 Gestione nomi file per XML/XSL..... | 11 |
| 5.2.1.3 File multipli..... | 12 |
| 5.2.2 <i>Formato XML/XSL-CSS with PKCS#7 (Codice Applicazione 194, 195)</i> | 13 |
| 5.2.3 <i>Formato XML/XSL-FO with PKCS#7 (Codice applicazione 199, 205)</i> | 14 |
| 5.2.4 <i>Formato RTF/DOC (Codice applicazione 196, 197)</i> | 15 |
| 5.2.5 <i>HTML with PKCS#7 (Codice applicazione 200)</i> | 15 |
| 5.3 FORMATO FILESYSTEM SE (CODICE APPLICAZIONE 198) | 16 |
| 5.3.1 <i>Il linguaggio interno SE Scripting Language</i> | 17 |
| 5.3.2 <i>Programmi associati al decoder2DPlus Richiamati da SESL</i> | 18 |
| 5.3.2.1 Creazione interattiva di grafici..... | 18 |
| 5.3.2.2 Utilizzo di un semplice sistema di TextToSpeech..... | 19 |
| 5.4 GENERAZIONE DI UN FILE PDF COMPRENSIVO DI TIMBRO DIGITALE | 19 |
| 5.4.1 <i>Trasformazione XSL-FO (uso della libreria FOP)</i> | 20 |
| 5.4.1.1 XML..... | 20 |
| 5.4.1.2 XSL..... | 20 |
| 5.4.1.2.1 Posizione del Timbro Digitale..... | 20 |
| 5.4.1.2.2 Parametrizzazione del path delle immagini..... | 21 |
| 5.4.1.3 Ambiente Client [in progress]..... | 22 |
| 5.4.2 <i>Trasformazione XSL-CSS (ambiente SE XML2PDF)</i> | 22 |
| 5.4.2.1 XML..... | 22 |
| 5.4.2.2 XSL..... | 22 |
| 5.4.2.3 Formati Immagine da Utilizzare..... | 25 |
| 5.4.2.4 Ambiente Client..... | 25 |
| 5.4.3 <i>Uso del namespace http://www.timbrodigitale.org/TD_tags</i> | 25 |
| 5.4.4 <i>RSA signature (in progress)</i> | 28 |

| | | | |
|---|--|--------------|---|
|  | Descrizione commessa | pag. 4 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

1 DEFINIZIONI E ACRONIMI

| Sigla | Descrizione |
|-------|--|
| [l.r] | Last Release (questa sigla riportata a fianco di un documento ricorda che si fa riferimento all'ultima versione del documento stesso). |
| N.A. | Non Applicabile |
| | |
| | |
| | |
| | |

2 DOCUMENTAZIONE DI RIFERIMENTO

| |
|--|
| Specifiche del codice 2D Plus® Specifiche tecniche del Codice 2DPlus® [SE_T-05-0005] T I REL Specifiche 2DPlus.pdf |
| Procedure Operative Piattaforma PeS 2D-Plus® Guida alle procedure operative ed alle policy di sicurezza, nell'integrazione di una piattaforma 2D-Plus® [SE_T-07-0049] T I DST proc operative aPeS.pdf |
| Accesso all'Appliance αPeS 2D-Plus® Uso del servizio αPeS di generazione timbri digitali 2D-Plus® da parte di applicativi aziendali [SE_T-07-0049] T I DST Usage from applications.pdf |
| Decoder Plus® - User Guide Manuale utente del software Decoder 2D-Plus® [GT50-16-iTD] T I MAN Decoder 2D-Plus UserGuide.pdf |

§

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 5 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

3 SCOPO DEL DOCUMENTO

Il codice bidimensionale 2D-Plus® gestisce dati binari in modo nativo ed è quindi indifferente alla semantica dei dati stessi; tuttavia, vi sono alcune strutture dati, che possono essere utilmente interpretate dal software Decoder 2D-Plus® dedicato alla decodifica, alla gestione e presentazione dei dati contenuti nel codice 2D-Plus® stesso.

Queste strutture dati possono essere utilizzate per descrivere e definire documenti e per poterli poi presentare nel modo più consono agli utenti.

Il presente documento spiega quali sono le strutture dati interpretate dal software di decodifica, descrive l'utilizzo di alcune applicazioni opzionali del Decoder 2D-Plus® utili per la visualizzazione di grafici e la conversione di testo in parlato ed inoltre descrive quali siano i formati e le condizioni necessarie affinché l'Appliance possa generare file PDF comprensivi del timbro digitale 2D-Plus®.

§

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 6 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

4 STRUTTURE DATI GESTITE

All'atto della richiesta di creazione di un codice bidimensionale 2D-Plus[®], vengono forniti una serie di metadati i quali, opportunamente inseriti nel codice stesso, supportano la fase di decodifica, interpretazione e presentazione, operata dal software Decoder 2D-Plus[®].

In una infrastruttura basata sull'Appliance α PeS[™]; queste informazioni vengono inviate all'appliance secondo le modalità descritte nel documento:

[SE_T-07-0049] T I DST Usage from applications [l.r].pdf

Tra le varie informazioni presenti a livello configurazione, viene definito un Codice Applicazione identificativo univoco del tipo dato, con un range che va da 0 a 255.

§

4.1 CODICI PER FORMATI PROPRIETARI

I Codici Applicazione che vanno da 0 a 191 sono disponibili per identificare strutture dati/applicazione proprietarie; questi codici, sono resi univoci in quanto connessi al Codice Cliente, owner della licenza d'uso del software Core presente nell'Appliance α PeS[™]; per mezzo di questi codici, vengono identificati contenuti proprietari che possono essere sfruttati da applicazioni proprietarie, al momento della loro decodifica e presentazione.

NOTA: l'appliance è del tutto indifferente al formato ed al significato dei dati, che gli vengono passati per essere inseriti nel timbro digitale 2D-Plus[®] che viene generato; *(questa affermazione è sempre vera, ad esclusione del caso di creazione di un documento PDF da parte dell'appliance; vedi paragrafo 5.4)*.

La capacità del codice 2D-Plus[®] di non essere dipendente dai dati, permette agli utilizzatori, di realizzare applicazioni che generano timbri digitali con contenuti assolutamente proprietari; in questi casi, il software di decodifica Decoder 2D-Plus[®], sarà comunque capace di recuperare i dati codificati all'interno del timbro digitale e potrà salvarli a disposizione dell'applicazione proprietaria che dovrà poi gestirli.

§

4.2 CODICI PER FORMATI STANDARD

I codici applicazione da 192 a 255 sono riservati a formati dati standard, a strutture dati per la firma digitale ed in casi particolari per altri tipi di firma elettronica.

Ci sono quindi una serie di formati che possono essere gestiti completamente dall'appliance e dal decoder.

| | | | |
|---|--|--------------|---|
|  | Descrizione commessa | pag. 7 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

4.3 TABELLA RIEPILOGATIVA CODICI APPLICAZIONE

| Cod.Appl | Tipo dato | Descrizione |
|-------------------|-----------------------------------|---|
| <i>da 0 a 191</i> | <i>Formati proprietari</i> | <i>I dati, se pur gestiti, non verranno in nessun modo analizzati dall'Appliance aPeS™ o dal Decoder 2D-Plus®;</i> |
| 192 | Reserved | |
| 193 | Reserved | |
| 194 (C2) | XML/XLS-CSS | I dati sono in formato XML, con un riferimento (href) ad un file XSL-CSS che permette la formattazione e presentazione dei dati. |
| 195 (C3) | XML/XLS-CSS with PKCS#7 | Equivalente al codice applicazione 194; la struttura dati XML è firmata usando un formato PKCS#7 a norma legale; |
| 196 | RTF/DOC | I dati sono in formato Microsoft RTF; |
| 197 | RTF/DOC with PKCS#7 | Equivalente al codice applicazione 196; la struttura dati RTF è firmata usando un formato PKCS#7 a norma legale; |
| 198 | FS SE | struttura dati complessa; vedi paragrafo 5.3; |
| 199 (C7) | XML/XLS-FO (ext Ibex) with PKCS#7 | I dati sono in formato XML, con un riferimento (href) ad un file XSL-FO <u>con estensioni Ibex</u> , che permette la formattazione e presentazione dei dati; la struttura dati XML è firmata usando un formato PKCS#7 a norma legale; |
| 200 | HTML with PKCS#7 | Struttura HTML; la struttura dati html è firmata usando un formato PKCS#7 a norma legale; |
| 201 | <i>Data+RSA</i> | <i>work in progress</i> Struttura dati generica con firma elettronica RSA; |
| 202 | PDF with PKCS#7 | File PDF; la struttura dati pdf è firmata usando un formato PKCS#7 a norma legale; |
| 203 | FO (Ext Ibex) with PKCS#7 | I dati sono in formato FO, <u>con estensioni Ibex</u> , pronti per la formattazione e presentazione dei dati; la struttura dati FO è firmata usando un formato PKCS#7 a norma legale; |
| 204 | TXT with PKCS#7 | File di testo; la struttura dati txt è firmata usando un formato PKCS#7 a norma legale; |
| 205 (CD) | XML/XLS-FO with PKCS#7 | I dati sono in formato XML, con un riferimento (href) ad un file XSL-FO standard, che permette la formattazione e presentazione dei dati; la struttura dati XML è firmata usando un formato PKCS#7 a norma legale; |

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 8 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

| | | |
|-----|-------------------|---|
| 206 | FO with PKCS#7 | I dati sono in formato FO standard, pronti per la formattazione e presentazione dei dati; la struttura dati FO è firmata usando un formato PKCS#7 a norma legale; |
| 207 | Reserved | |
| 208 | Reserved | |
| 209 | FS SE with PKCS#7 | Equivalente al codice applicazione 198; la struttura dati è firmata usando un formato PKCS#7 a norma legale; (vedi paragrafo 5.3) |

Nota: alcuni codici applicazione non verranno più supportati; tipicamente fanno riferimento a test interni, in genere senza la presenza della firma digitale.

Nota: Al momento il sw Decoder 2D-Plus presente sul client di verifica, gestisce i codici applicazione 205 e 206, che abilitano trasformazioni XML/XSL-FO standard, tramite il software di renderizzazione che gestisce i codici 199 e 203.

In questi casi, se si devono produrre documenti FO particolarmente complessi, si richiede di prestare molta attenzione alla fase di test, in quanto il renderizzatore dei codici 199 e 203 accetta delle particolari estensioni al linguaggio FO e potrebbe non essere totalmente coerente con lo standard.

Non appena sarà disponibile nel Decoder 2D-Plus, la procedura guidata per l'installazione dell'ambiente Java, la gestione dei codici 205 e 206 verrà effettuata tramite libreria FOP, che è dedicata allo standard W3C.

§

| | | | |
|---|--|---------------------|---|
|  | Descrizione commessa | pag. 9 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

5 DESCRIZIONE CODICI APPLICAZIONE

Segue una descrizione dei Codici Applicazione, il formato interno ed una breve descrizione:

5.1 FORMATI PROPRIETARI: UTILIZZO

I dati inviati all'appliance, che devono essere elaborati secondo una configurazione che contiene un Codice Applicazione proprietario (valori da 0 a 191), sono soggetti esclusivamente alle seguenti operazioni: firma digitale, compressione, trasformazione in codice grafico 2D-Plus®.

Il software Decoder 2D-Plus®, una volta decodificato il codice grafico, eliminerà la compressione, ricostruendo così i dati originari; presentare le informazioni relative al firmatario all'utente che sta verificando il timbro digitale; chiederà quindi a quest'ultimo, dove salvare i dati estratti e quale nome dare al file.

Decoder 2D-Plus® preparerà una finestra di dialogo che permetterà di salvare i dati con un suffisso "CCCA", dove CC e CA ognuno di due digit esadecimali, rappresentano il Codice Cliente che ha creato i dati ed il Codice Applicazione che li può gestire.

Ad esempio, il timbro digitale creato da una azienda con Codice Cliente=44, che ha dei dati in formato proprietario con Codice applicazione=13, una volta decodificato genererà un file di nome "nome-di-file.2C0D.

§

5.2 FORMATI STANDARD: UTILIZZO

Il timbro digitale 2D-Plus® apposto sui documenti, viene analizzato dal sw Decoder 2D-Plus®; questo software ha la capacità di:

- ♯ ricostruire le strutture dati contenute dal codice grafico;
- ♯ verificare l'integrità di una firma digitale (se presente), presentando i dati del firmatario e della Certification Authority che ha emesso il certificato X.509;
- ♯ presentare nella forma opportuna i dati presenti;

NOTA: quest'ultimo punto naturalmente dipende dal tipo di formato dati contenuto nel codice; di seguito vengono date indicazioni sui diversi formati.

§

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 10 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

5.2.1 FORMATO XML/XSL (CODAPPL 194, 195, 199, 205): GENERALITÀ

In questo caso i dati sono strutturati come file XML.

Anche se la struttura che stiamo per descrivere non è obbligatoria, tipicamente il file XML farà riferimento (href) ad una struttura XSL, che permette una trasformazione in un formato più facilmente interpretabile dei dati presenti nell'XML stesso: in genere verrà ricreato l'aspetto esteriore di un documento cartaceo.

È compito di chi realizza i file XML/XSL e si interfaccia con l'apppliance, configurare correttamente il tutto per una corretta visualizzazione da parte del Decoder 2D-Plus®.

Si consideri quindi che in questo caso, il modello di presentazione dei documenti prevede l'utilizzo di un file XML a cui viene associato un XSL appositamente definito. A quest'ultimo possono essere poi collegate delle immagini o loghi utili alla visualizzazione del documento finale.

L'XML presente all'interno di un codice 2D-Plus®, sarà quindi un documento XML standard con l'unica differenza che inizialmente, avrà il riferimento esplicito all'XSL al quale deve venire associato per la creazione del documento finito:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="nomexsl.xsl"?>
<root_node>
...
</root_node>
```

Il file XSL, come vediamo dall'esempio di cui sopra, viene esplicitamente associato attraverso il tag *xml-stylesheet*.

In questo caso il nome dell'XSL in questione è *nomexsl.xsl*.

NOTA 1: E' necessario prestare particolare attenzione all'eventuale presenza di caratteri spuri all'interno dei file XML/XSL in quanto è stato riscontrato che numerosi editor sembrano inserirne soprattutto all'esterno dei Tag ed all'inizio del file.

NOTA 2: alla data attuale, la versione XSLT utilizzato è la 1.0

§

5.2.1.1 GARANZIA SULL'INTEGRITÀ ED AUTENTICITÀ DEI FILE XSL E DEI FILE ESTERNI

Il semplice riferirsi ad una struttura di stylesheet come quella XSL, presente all'esterno del file XML firmato, può comportare un basso livello di garanzia della fase di presentazione.

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 11 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Il file XSL; potrebbe infatti essere diverso (corrotto o modificato) da quello pensato durante la creazione dell'XML, la qual cosa comporterebbe la creazione di un documento con un aspetto diverso da quello veramente voluto.

Allo scopo di attribuire all'intero processo di interpretazione dell'XML e di sua presentazione, gli stessi attributi che la firma digitale associa all'XML firmato --Integrità Autenticità e non ripudio-- viene raccomandata la modalità di richiamo href che segue:

```
<?xml-stylesheet type="text/xsl" href="nome_file.xml?SHA256= B8D150 [...] 61DB5766E"?>
```

Questa modalità di richiamo href, è più che raccomandata, nell'uso di una struttura dati XML/XSL per un timbro digitale.

Diversamente dalla sintassi classica, si può notare che al *nome-di-file-XSL*, viene aggiunta la keyword “?SHA256” a cui viene associato il valore dell'hash del file XSL stesso.

Questa sintassi è coerente con quella dell'XML e non presenta problemi di interpretazione e permette la verifica dell'integrità del file XSL (vedi oltre), che verrà utilizzato per operare la trasformazione per la presentazione dei dati.

Questo approccio è fondamentale per dare la certezza che il documento finale presentato all'utente, sia esattamente quello voluto dal creatore del timbro digitale.

La presenza del parametro SHA256, obbliga il software Decoder 2D-Plus® ad operare una verifica di integrità sul file a cui questo digest si riferisce; naturalmente questa verifica avviene prima che il file XSL stesso possa essere utilizzato.

In caso di difformità del valore dell'hash calcolato, il software Decoder 2D-Plus® presenterà all'utente un messaggio di warning, avvisandolo della non integrità del file XSL stesso e non procederà alla costruzione del documento finale.

L'uso dei codici di hash per il file esterno dall'XML, consente di accorgersi di modifiche, volontarie o meno, a cui possono essere stati sottoposti questi file.

Data la necessità molto frequente di associare al file XSL immagini ed eventuali file CSS od altro si è pensato di comprimere in un unico file compresso ZIP l'insieme dei diversi file. Il valore di HASH sarà dunque calcolato sull'intero file compresso in formato ZIP al cui interno deve necessariamente essere presente il layout XSL ed ogni file utile ad una corretta rappresentazione del documento. Questa opzione è stata resa disponibile per ovviare in maniera semplice alla complessa verifica a catena dei diversi file che compongono il documento sostituiti da un unico file ZIP il cui nome ed estensione deve seguire precise regole come riportato nel paragrafo successivo.

I codici hash presenti all'interno di un file XML firmato digitalmente e di cui si sia verificata l'integrità ed autenticità, sono di conseguenza verificati nella loro integrità ed autenticità; quindi se l'hash di un file XSL è integro ed autentico, sarà possibile verificare integrità ed autenticità del file XSL stesso.

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 12 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

NOTA: Se pur sopportato anche dalle più recenti versioni del Decoder 2DPlus®, l'utilizzo dell'algoritmo SHA1 è deprecato, in quanto non più in linea con la normativa sulla firma digitale in vigore in Italia.

§

5.2.1.2 GESTIONE NOMI FILE PER XML/XSL

GT50 supporta una specifica ottimale, descritta nel presente paragrafo, per costruire e gestire i nomi dei file XSL.

Il layout XSL dei documenti deve essere disponibile per molto tempo; è quindi possibile che versioni diverse dello stesso layout debbano essere disponibili contemporaneamente.

Le applicazioni che utilizzano la soluzione XML/XSL per la costruzione di un timbro digitale implicano che il file XML contenga al proprio interno il nome dell'XSL, il percorso dove trovare questo file e l'HASH del file XSL stesso.

Così facendo la prima volta che un utente prova a decodificare un nuovo timbro digitale, il cui contenuto XML faccia riferimento ad un XSL, il software Decoder 2D-Plus® cercherà questo file in locale (lo trova se lo ha già scaricato in precedenza) ovvero in rete.

I dettagli sull'operatività del software Decoder 2D-Plus® relativamente ai file XSL, sono presenti nel documento:

Decoder 2D-Plus® User Guide

disponibile all'indirizzo http://www.timbrodigitale.com/Appliance_PeS/doc/

Di seguito presentiamo il nostro standard per la costruzione dei nomi degli XML/XSL:

CcCa_CiCs_nomedelfile_M.m.XSL con il seguente significato:

| | |
|------|---|
| Cc | Codice Cliente; 2 caratteri esadecimale; valore assegnato al momento del rilascio di una licenza del software Core sull'Appliance α PeS™; Durante le fasi di test, è possibile utilizzare il codice 01 |
| Ca | Codice Applicazione; 2 caratteri esadecimale; attualmente può rientrare nei casi: C2 XML/XLS-CSS C3 XML/XLS-CSS with PKCS#7 C7 XML/XLS-FO (Ext Ibex) with PKCS#7 CD XML/XLS-FO (standard) with PKCS#7 |
| CiCs | Codice univoco che identifica l'ente emittente il documento (se necessario) |

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 13 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

(*opzionale*) Per gli enti si propone il Codice ISTAT od il Codice Catastale che identifica puntualmente l'ente emittente

nomedelfile è il nome del file che può contenere esclusivamente caratteri alfanumerici e il punto, il trattino o il "trattino basso", ovvero '.' (dot), '-' (minus), '_' (underscore) e deve contenere una sola volta ".xml"; il carattere spazio (blank) non è ammesso.

M.m major.minor number; espressi in decimale; è la versione del file

In questo modo sarà facile gestire XSL di documenti simili anche con differenti versioni in contemporanea.

Naturalmente questo standard può essere convenientemente applicato anche ad altri tipi di file referenziati, come immagini, loghi etc...

Il codice catastale o codice ISTAT del comune è uno di quelli indicati in questo sito:

<http://www.comuni-italiani.it/058/091/>

Ad esempio, il file XSL di un certificato di stato di famiglia del Comune di Casalecchio di Reno, potrebbe essere nominato come segue:

56C3_B880_Certificato-Stato-di-famiglia_1.0.XSL

Seguendo questa logica un file XSL destinato a descrivere lo stesso certificato, magari del Comune di Imola, verrebbe invece identificato così:

56C3_E289_Certificato-Stato-di-famiglia_1.0.XSL

§

5.2.1.3 FILE MULTIPLI

Si ricorda che un file XSL può avere la necessità di altri file a cui fa riferimento: ad esempio le immagini.

In questo caso, non c'è più solo un file da gestire, ma un gruppo di file.

La soluzione consigliata e gestita dal sw Decoder 2D-Plus® in modo totalmente trasparente per l'utilizzatore, è quella di creare un file compresso tipo ZIP, all'interno del quale vengono inseriti sia il file XSL che tutti i file ad esso necessari.

Questo file zip, deve essere prodotto in modo tale che:

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 14 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

- scompattandolo, il file XSL principale -quello referenziato dal tag href, all'interno del file XML- deve essere nella root del filesystem del file .zip;
- il nome del file .XSL principale deve coincidere con il nome indicato nel tag `<?xml-stylesheet type=[...] href= ...` contenuto nel file XML presente nel timbro digitale;
- eventuali file esterni referenziati dal file XSL, possono trovarsi nella root del filesystem del file .zip, ovvero in una delle sottodirectory presenti;
- il nome del file ZIP, deve essere lo stesso nome del file XSL referenziato nel tag href= presente nella struttura XML
- questo file deve avere il suffisso composito “.XSL.ZIP”

In presenza di un file .XSL.ZIP, il parametro SHA1 presente nel tag href, sarà relativo a questo file: sarà compito del sw Decoder 2D-Plus[®], l'onere di interpretare in modo corretto questo parametro.

Riprendendo la sintassi precedente, la struttura dati ZIP contenente il file XSL ed altri file ad esso necessari, sarà un file con nome:

CcCa_CiCs_nomedelfile_M.m.XSL.ZIP

Come detto, il file XML continuerà a fare riferimento ad un file “.XSL”, in questo modo:

```
<?xml-stylesheet type="text/xsl" href="CCCA_CiCs_nomedelfile_M.m.xsl?SHA1=984C[...]87EF"?>
```

mentre la verifica dello SHA1, sarà fatta utilizzando il file:

"CcCa_CiCs_nomedelfile_M.m.xsl.zip"

La gestione del file “.ZIP” è completamente trasparente all'utente ed all'ambiente XML e demandata totalmente al software Decoder 2D-Plus[®].

I dettagli sull'operatività del software Decoder 2D-Plus[®] relativamente ai file XSL, sono presenti nel documento:

Decoder 2D-Plus[®] User Guide

disponibile all'indirizzo http://www.timbrodigitale.com/Appliance_PeS/doc/

§

5.2.2 FORMATO XML/XSL-CSS WITH PKCS#7 (CODICE APPLICAZIONE 194, 195)

La coppia XML/XSL-CSS permette di creare documenti utilizzando, per la renderizzazione del documento stesso, la sintassi XHTML/CSS.

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 15 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

L’XSL, scritto in modalità XHTML/CSS, oltre a tutta la sintassi XSLT, potrà includere per la renderizzazione del documento tutto quello che riguarda XHTML e CSS.

Il software Decoder 2D-Plus® consente di presentare il contenuto del timbro digitale, elaborando una trasformazione XSLT e generando un file HTML il quale viene presentato all’utente tramite la l’utilizzo del viewer di Internet Explorer, all’interno di una finestra gestita dal software.

NOTA: il codice applicazione 194 (XML/XSL-CSS) non comporta la presenza di dati firmati. Pur essendo gestito dall’appliance e dal software di decodifica, questo tipo di struttura dati non permette i controlli sull’integrità ed autenticità di cui necessitano le applicazioni che richiedono un timbro digitale.

NOTA: il Cascading Style Sheets (CSS) è orientato alla presentazione di documenti scritti con markup language, tipicamente HTML; questo consente una certa facilità ed elasticità di processo a scapito però, del completo controllo tipografico e di formattazione.

Allo scopo di superare queste limitazioni può essere utilizzato il linguaggio FO (vedi par. 5.2.3).

§

5.2.3 FORMATO XML/XSL-FO WITH PKCS#7 (CODICE APPLICAZIONE 199, 205)

I codici 199 e 205, differiscono per il fatto che fanno riferimento a due trasformazioni FO differenti; l’applicazione 199 permette di gestire FO con estensioni Ibex, mentre l’applicazione 205 è aderente allo standard W3C.

La gestione delle estensioni Ibex permette la creazione di documenti complessi, che gli attuali renderizzatori standard FO (FOP) non sempre riescono a supportare; viene operata una trasformazione che genera un FO non completamente standard ed utilizzata una libreria proprietaria Ibex per la creazione dell’output finale.

Una trasformazioni FO standard, oltre ad alcune limitazioni nella struttura dei documenti finali, prevede tipicamente l’utilizzo del software “FOP” il quale anche lato client opera e necessita di un ambiente Java.

In ogni caso, la struttura dati che viene passata all’appliance ed inserita nel timbro digitale è un file XML (all’interno di una busta crittografica PKCS#7), che fa riferimento, tramite *href*, ad un XSL-FO.

La coppia XML/XSL-FO permette la creazione (tramite trasformazione XSLT) di strutture dati in linguaggio FO; a sua volta questo linguaggio permette la generazione di documenti in svariati formati, con il completo controllo dell’impaginazione e la formattazione, anche di strutture documentali particolarmente complesse.

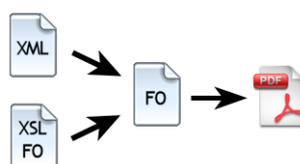
Da un file FO è quindi possibile generare documenti nei formati PDF, JPG, TIFF, PNG, rtf, postscript, SVG, a partire dai dati (XML) e da un descrittore di Layout / Contenuti (XSL-FO).

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 16 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

NOTA: per l'utilizzo del codice applicazione 205 XML/XSL-FO, vedi la nota in fondo alla Tabella presente al par. 4.3 .

La generazione del documento finale avviene tramite 2 passaggi ben definiti: da XML / XSL-FO a FO e da FO al formato prescelto.

Ad esempio, il processo XSLT (la trasformazione) inserisce i dati estrapolati dall'XML ed interpreta la parte XSL, mettendo il risultato finale in un file FO. Il secondo step interpreta il file FO per renderizzarlo come PDF.



Il software Decoder 2D-Plus® consente di presentare il contenuto del timbro digitale, elaborando una trasformazione XMLT e generando dal file FO risultante, un file PDF; questo a sua volta viene presentato all'utente tramite un Adobe Acrobat Reader, all'interno di una finestra gestita dal software.

§

5.2.4 FORMATO RTF/DOC (CODICE APPLICAZIONE 196, 197)

La struttura dati che viene passata all'appliance ed inserita nel timbro digitale viene interpretata dal decoder come Rich Text Format, il formato dati Microsoft, orientato alla condivisione multiplatforma di documenti.

Al contrario del formato DOC, per sua natura l'RTF non può contenere *Presentation Corruption Agent*, come ad esempio le macroistruzioni; questo rende il suo utilizzo coerente con i requirement legali.

Il software Decoder 2D-Plus® può nativamente presentare il file RTF, all'interno di una finestra, in modalità *non modificabile*.

NOTA: il codice applicazione 196 (RTF semplice) non comporta la presenza di dati firmati. Pur essendo gestito dall'appliance e dal software di decodifica, questo tipo di struttura dati non permette i controlli sull'integrità ed autenticità di cui necessitano le applicazioni che richiedono un timbro digitale.

§

5.2.5 HTML WITH PKCS#7 (CODICE APPLICAZIONE 200)

La struttura dati che viene passata all'appliance ed inserita nel timbro digitale è un file in formato HTML sempre firmato digitalmente (all'interno di una busta crittografica PKCS#7).

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 17 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Il software Decoder 2D-Plus® può nativamente presentare questo file tramite la visualizzazione di Internet Explorer, all'interno di una finestra gestita dal software.

§

5.3 FORMATO FILESYSTEM SE (CODICE APPLICAZIONE 198)

Il Codice Applicazione 198 è stato pensato per permettere una elaborazione complessa dei dati contenuti nel Timbro Digitale.

In questo caso infatti il contenuto del timbro digitale 2D-Plus® è un file compresso di tipo ZIP, il quale a sua volta contiene almeno un file di nome [start.sesl](#) ed uno o più file, oggetto della elaborazione successiva.

Dopo la decodifica del timbro digitale, il software Decoder 2D-Plus® estrarrà temporaneamente tutti i file contenuti al suo interno.

Una volta fatto questo, Decoder 2D-Plus® aprirà il file [start.sesl](#), che contiene in formato ASCII una serie di comandi che il software può interpretare.

Il suffisso .sesl, sta per “SE-Scripting Language”; di seguito l’elenco dei comandi disponibili e la loro sintassi:

| | |
|---------------------------|--|
| ; <commento> | Commento che non viene interpretato |
| Mkdir <nome_directory> | Crea una directory di nome <nome_directory> |
| Copy <from> <to> | Comando di copia file |
| Del <nome file> | Cancella il file <nome file> ATTENZIONE non viene fatto nessun controllo sul file da cancellare |
| Message <msg> | Mostra a video un messaggio |
| Speech <file.txt> | Vocalizza <file.txt> invoca <i>VoiceTextAPISimple.exe</i>, parte del package Decoder 2D-Plus®, controllando la sua integrità tramite SHA1: se il controllo dà esito negativo, il programma non viene eseguito. Nota: devono essere installate le Microsoft SAPI 4.0 ed il relativo engine in italiano; |
| Graph <filegraph.xml> | Invoca <i>textgraph.exe</i>, parte del package Decoder 2D-Plus®, controllando la sua integrità tramite SHA1: se il controllo dà esito negativo, il programma non viene eseguito. Vedi la sintassi di filegraph.xml nel paragrafo 5.3.2.1 |
| Show <file.xml> | Mostra a video un file xml, interpretandolo come se fosse stato decodificato |

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 18 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

| | |
|--|--------------------------------|
| | a partire dal timbro digitale. |
|--|--------------------------------|

§

Un esempio di file start.sesl:

```

; analisi vendite trimestrali
; -----
mkdir 1stqrt
copy relazionetrimestre.txt 1stqrt\
copy dativendite.txt 1stqrt
message fatto!
speech "1stqrt\relazionetrimestre.txt"
graph "1stqrt\dativendite.txt"
; finito

```

Una volta completata l'esecuzione del file start.sesl, Decoder 2D-Plus® elimina tutti i file precedentemente estratti del file ZIP.

§

5.3.1 IL LINGUAGGIO INTERNO SE SCRIPTING LANGUAGE

Ogni file system SE deve contenere un file denominato start.sesl in cui sono presenti comandi del SE Scripting Language. E' consigliabile terminare con un comando DoShow in modo da mostrare a video i documenti estratti dal FSSE. Di seguito la lista dei comandi disponibili:

Lista dei comandi disponibile e loro sintassi:

Commento che non viene interpretato

; <commento>

Comando creazione directory

mkdir <directory>

Comando copia file

copy <from> <to>

Comando cancellazione file (Attenzione questo comando non esegue alcun controllo sul file da cancellare!)

del <nome file>

Mostra a video un messaggio

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 19 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

message <message text>

Vocalizza il file (devono essere installate le Microsoft SAPI 4.0 ed il relativo engine in italiano)

speech <file name .txt>

Questo comando invoca il file *VoiceTextAPISimple.exe* controllando tramite SHA1 se il file è stato cambiato, in quel caso non viene eseguito nulla.

Mostra a videoun grafico utilizzando i seguenti dati e settaggi

graph <file name xml>

Questo comando invoca il file *textgraph.exe* controllando tramite SHA1 se il file è stato cambiato, in quel caso non viene eseguito nulla. Il file xml in input deve rispettare la sintassi del paragrafo 5.3.2.1.

Mostra a video il file interpretandolo

show <file name xml> (per ora gestisce solo file xml, ma nelle prossime versioni permetterà il riconoscimento di diversi formati secondo quanto prevede il Decoder 2D-Plus[®])

§

5.3.2 PROGRAMMI ASSOCIATI AL DECODER2DPLUS RICHIAMATI DA SESL)

Di seguito una breve descrizione degli applicativi opzionali del Decoder 2D-Plus[®].

Questi applicativi sono utilizzati dal decoder, solo dopo che questo si sia accertato della loro integrità ed autenticità.

5.3.2.1 CREAZIONE INTERATTIVA DI GRAFICI

La creazione di grafici contenuti nel tag 2DPlus è ottenuta utilizzando un programma apposito denominato *textgraph.exe* realizzato per la piattaforma Windows che sfrutta un file XML come dati di input.

Per cambiare il tipo di grafico, il nodo da modificare nell'xml è <tipo_graf>tipo</tipo_graf> .

Attualmente vengono gestiti i seguenti tipi di grafico:

- Bar = Istogramma
- FastLine = Grafico a linea veloce
- Area = Grafico ad area

Per configurare tutti i valori del grafico utilizzare il tag principale <grafico>dati</grafico>.

Il tag <punto>tag x ed y</punto> identifica il singolo punto da mostrare coordinate x,y

Il tag <x>valore</x> è il valore numerico della X.

Il tag <y>valore</y> è il valore numerico della Y.

Il tag <desc_x>stringa</desc_x> l'etichetta per l'asseX.

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 20 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Il tag `<desc_y>stringa</desc_y>` l'etichetta per l'asse Y.

Di seguito un esempio:

```
<?xml version="1.0"?>
<grafico>
  <titolo>Prova di un grafico letto da XML</titolo>
  <desc_x>Secondi</desc_x>
  <desc_y>Dati</desc_y>
  <tipo_graf>FastLine</tipo_graf>
  <punto>
    <x>10</x>
    <y>50</y>
  </punto>
</grafico>
```

§

5.3.2.2 UTILIZZO DI UN SEMPLICE SISTEMA DI TEXTTOSPEECH

VoiceTextAPISimple.exe è l'applicazione invocata dal comando *speech* del SESL. Tale programma, sviluppato per la piattaforma Windows, è in grado di vocalizzare un file di testo tramite le Microsoft Speech API 4.0 (tale pacchetto deve obbligatoriamente essere correttamente installato pena il mal funzionamento dell'applicazione VoiceTextAPISimple.exe)

§

5.4 GENERAZIONE DI UN FILE PDF COMPRENSIVO DI TIMBRO DIGITALE

Il software presente nell'appliance, in genere non entra in merito alla presenza di un riferimento a file XSL, da parte di una struttura dati XML, ma c'è una eccezione: quando la configurazione invocata richiede, oltre firma e la creazione di un timbro digitale 2D-Plus[®], anche la creazione di un file PDF, che sarà l'unico oggetto che l'appliance restituirà all'applicazione chiamante.

In questo caso, il software presente nell'appliance, dopo aver generato l'immagine che rappresenta il timbro digitale, governerà una trasformazione XSL (CSS ovvero FO), per generare il file PDF del documento finale.

In questo file PDF, viene inoltre inserito il timbro digitale precedentemente generato; la posizione di questa immagine all'interno del file PDF è una informazione presente nella configurazione utilizzata; a sua volta questa l'ha prelevata dal file XSL, quando questo è stato originariamente caricato nell'appliance stesso.

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 21 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

L'appliance necessita quindi di avere il file XSL all'interno della sua area operativa, sia nel caso di una trasformazione CSS che nel caso FO; in caso di mancanza del file XSL, viene restituito un codice di errore.

I dettagli relativi alla gestione delle configurazioni da parte del PAdm, sono presenti nei documenti:

Procedure Operative Piattaforma PeS 2D-Plus®

Manuale dell'Amministratore Appliance aPeS™ - PAdm

disponibile all'indirizzo http://www.timbrodigitale.com/Appliance_PeS/doc/

§

5.4.1 TRASFORMAZIONE XSL-FO (USO DELLA LIBRERIA FOP)

5.4.1.1 XML

L'XML deve essere un documento XML standard well formed, con il consueto riferimento interno al foglio di stile (href) che in questo caso deve essere un XSL-FO standard.

L'appliance viene richiamato sempre con il codice applicazione 205 (XML/XSL-FO with PKCS#7); la differenza viene fornita dalla configurazione utilizzata, nella quale deve essere presente la richiesta di attivazione della funzione di generazione di file PDF.

In questa condizione l'appliance, invece di un timbro digitale 2D-Plus, restituisce il documento finale in formato PDF, al cui interno è presente il timbro digitale.

§

5.4.1.2 XSL

Quando nella configurazione è presente la richiesta di generazione di file PDF, un file XSL coerente con la trasformazione desiderata, deve essere precaricato nell'Appliance stesso.

Anche in questa situazione, quando insieme al file XSL devono essere presenti altri file di supporto, deve essere utilizzato un file.XSL.ZIP, con le regole già indicate nel paragrafo 5.2.1.3.

Notare che possono essere presenti anche più file XSL adibiti ad un'unica trasformazione e, quindi, presenti nello stesso XSL.ZIP, ma:

- L'XSL principale deve essere presente nella root del file XSL.ZIP.
- Tutti gli elementi grafici `<fo:external-graphic />` devono essere contenuti solo ed esclusivamente nell'XSL principale. Per elementi grafici si intende, ovviamente, anche il Timbro Digitale.
- L'XSL (o gli XSL) non possono contenere codice Java al loro interno

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 22 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Inoltre, al momento, è possibile utilizzare solamente la sintassi XSLT 1.0 e non la 2.0.

5.4.1.2.1 POSIZIONE DEL TIMBRO DIGITALE

Naturalmente, nel file XSL, deve essere presente la specifica FO della posizione grafica, in cui dovrà essere inserito il timbro digitale creato, ma c'è un'altra informazione importante da inserire obbligatoriamente: il *rettangolo di contenimento* dell'immagine del Timbro Digitale.

Infatti, mentre il processo che elabora un file FO e crea il documento finale, ha la capacità di elaborare i metadati presenti in una immagine PNG e ricavarne, ad esempio, le dimensioni esatte ed il numero di punti per pollice, il swCore dell'appliance, ha la necessità di conoscere in anticipo, la grandezza massima (rettangolo di contenimento) dell'immagine del Timbro Digitale, che andrà poi inserita nel documento.

La posizione del timbro digitale deve essere espressa esattamente in questa forma:

```
<fo:external-graphic id="PeS_tag_0">
  <xsl:attribute name="src" width="XXXmm" height="YYYmm">
    <xsl:text>url('</xsl:text><xsl:value-of select="$PeS_tag_baseloc"/>
    <xsl:text>TimbroDigitale.jpg')</xsl:text>
  </xsl:attribute>
</fo:external-graphic>
```

dove, al posto di XXX e YYY andranno inseriti i valori relativi alla dimensione (in mm), per esempio:

```
<fo:external-graphic id="PeS_tag_0">
  <xsl:attribute name="src" width="120mm" height="35mm">
    <xsl:text>url('</xsl:text><xsl:value-of select="$PeS_tag_baseloc"/>
    <xsl:text>TimbroDigitale.jpg')</xsl:text>
  <xsl:attribute>
</fo:external-graphic>
```

Poiché le informazioni sulla dimensione sono le uniche opzionali, è ammessa anche la forma:

```
<fo:external-graphic id="PeS_tag_0">
  <xsl:attribute name="src">
    <xsl:text>url('</xsl:text><xsl:value-of select="$PeS_tag_baseloc"/>
    <xsl:text>TimbroDigitale.jpg')</xsl:text>
  </xsl:attribute>
```

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 23 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

</fo:external-graphic>

o, infine, si può inserire una sola delle due dimensioni sempre nel modo indicato. Per le dimensioni non specificate nel file XSL, queste saranno determinate dalla configurazione Paper e-Sign che utilizza il foglio di stile; si rende quindi necessario controllarla per evitare incongruenze nella formattazione. Il fatto che le dimensioni siano da interpretarsi come "massime" (l'immagine sarà grande solo quanto necessario) o "assolute" (l'immagine avrà esattamente, comunque, le dimensioni indicate) è sempre determinato dalla configurazione.

L'utilizzo del "\$PeS_tag_baseloc" è illustrato nel paragrafo successivo.

NOTA: è fondamentale che il nome del file immagine, sia scritto esattamente come negli esempi:

TimbroDigitale.jpg

5.4.1.2.2 PARAMETRIZZAZIONE DEL PATH DELLE IMMAGINI

L'appliance opera in parallelo, su più richieste di timbro digitale contemporanee e questo comporta alcune concorrenze interne, su risorse condivise.

Allo scopo di mantenere correttamente gestite una serie di file ed immagini, è necessario che all'interno del file XSL di trasformazione, esista la dichiarazione che segue:

```
<xsl:param name="PeS_tag_baseloc" select=""></xsl:param>
```

Il tag *xsl:param* è un *top-level element* e va inserito quindi negli elementi generali ad inizio del file XSL:

In quanto *top-level element* rappresenta una dichiarazione di variabile globale, che è visibile dovunque nella struttura XSL.

Il tag *xsl:param* deve essere utilizzato ogni volta che si riferenzia un elemento esterno al file XSL; questo vale in special modo per le eventuali immagini referenziate; ad es., considerando la dichiarazione precedente:

```
<fo:external-graphic>
  <xsl:attribute name='src'>
    url(<xsl:value-of select='$PeS_tag_baseloc'/><xsl:value-of select='logo-del-comune.png'/>)
  </xsl:attribute>
</fo:external-graphic>
```

La dichiarazione iniziale del parametro "**PeS_tag_baseloc**", come parametro "vuoto" e la modalità utilizzata per costruire i reference all'immagine del logo, permettono di gestire contemporaneamente, sia l'ambiente appliance nella fase di costruzione del PDF iniziale, sia l'ambiente decoder presente sul client, nel momento in cui venga operata la trasformazione XML/XSL-FO a partire del codice 2D-Plus® interpretato.

Il parametro "**PeS_tag_baseloc**" deve essere utilizzato anche per l'immagine digitale "TimbroDigitale.jpg". Di seguito un esempio esplicativo:

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 24 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

```
<fo:external-graphic id="PeS_tag_0" width="30mm" height="130mm">
  <xsl:attribute name="src">
    <xsl:text>url('</xsl:text><xsl:value-of select="$PeS_tag_baseloc"/><xsl:text>TimbroDigitale.jpg')</xsl:text>
  </xsl:attribute>
</fo:external-graphic>
```

Dove il Timbro, in questo caso, ha un'altezza di 3cm e una larghezza di 13cm. Notare il "\$PeS_tag_baseloc " che, ovviamente, deve essere presente in ogni riferimento a files esterni all'interno dell'XSL-FO.

§

5.4.1.3 AMBIENTE CLIENT [IN PROGRESS]

L'utilizzo di un formato FO standard, prevede la presenza sul Client, di un ambiente Java, necessario ad operare con il software FOP.

Attualmente, pur non essendo implementata questa libreria, la presentazione di un documento a partire da strutture dati XML/XSL-FO, viene realizzata con l'uso di un renderizzatore proprietario.

NOTA: per l'utilizzo del codice applicazione 205 XML/XSL-FO nella generazione di un PDF, vedi la nota in fondo alla Tabella presente al par. 4.3 .

§

5.4.2 TRASFORMAZIONE XSL-CSS (AMBIENTE SE XML2PDF)

5.4.2.1 XML

L'XML deve essere un documento XML standard well formed, con il consueto riferimento interno al foglio di stile (href) che in questo caso deve essere un XSL-CSS con alcune estensioni necessarie alla formattazione di un documento anche multipagina, che verranno discusse di seguito.

L'appliance viene richiamato sempre con il codice applicazione 195 (XML/XSL-CSS with PKCS#7); la differenza viene fornita dalla configurazione utilizzata, la quale deve attivare la funzione di generazione di file PDF.

In questa condizione l'appliance, invece di un timbro digitale 2D-Plus, restituisce il documento finale in formato PDF, al cui interno è presente il timbro digitale.

§

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 25 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

5.4.2.2 XSL

Quando nella configurazione è presente la richiesta di generazione di file PDF, un file XSL coerente con la trasformazione desiderata, deve essere precaricato nell'Appliance stesso.

Anche in questa situazione, quando insieme al file XSL devono essere presenti altri file di supporto, deve essere utilizzato un file.XSL.ZIP, con le regole già indicate nel paragrafo 5.2.1.3.

Il file XSL richiamato dall'XML permette una trasformazione CSS; deve essere scritto utilizzando lo standard XHTML/CSS con l'aggiunta di alcuni tag particolari specifici dell'ambiente SE XML2PDF.

Tag speciali dell'ambiente GT50 XML2PDF:

in testa ai CSS va aggiunta sempre la seguente proprietà:

```
@page {
    margin:1cm;
}
```

Questa indica a XML2PDF la grandezza dei margini di pagina, e non influisce nella creazione dell'HTML renderizzato dal browser.

Le proprietà CSS del tag "Body" dovranno contenere la seguente:

```
body {
    ...
    line-height:90%;
}
```

Questo per garantire una corretta spaziatura fra le righe di testo durante la generazione del PDF.

La posizione grafica, in cui dovrà essere inserito il timbro digitale creato, segue le regole della sintassi utilizzata da CSS, ma c'è una informazione importante da inserire obbligatoriamente: il *rettangolo di contenimento* dell'immagine del Timbro Digitale.

Il swCore dell'appliance, ha infatti la necessità di conoscere in anticipo, la grandezza massima (rettangolo di contenimento) dell'immagine del Timbro Digitale, che andrà poi inserita nel documento.

Sarà quindi necessario specificare nell'XSL-CSS, queste informazioni; si dovrà quindi indicare le proprietà della "img class" secondo la sintassi CSS:

```
.Pes_code_img {
    width:XXXpx;
    height:YYYpx;
    zoom:NN%;
}
```

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 26 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

Quindi, nella posizione ritenuta opportuna per l’inserimento del Timbro Digitale, andrà referenziata l’immagine del codice bidimensionale 2D-Plus[®], come segue:

```
.
```

NOTA1 è fondamentale che i nomi utilizzati per “*img class*” e per “*src*” siano scritti esattamente come nell’esempio sopra indicato: **Pes_code_img** e **TimbroDigitale.gif**

NOTA2 I valori numerici da utilizzare nella “*img class*” indicano:

- **XXX**: la larghezza massima in pixel del bollo (non più piccolo di 404)
- **YYY**: l’altezza massima in pixel del bollo (non più piccolo di 30)
- **NN**: lo zoom, che deve valere
 - 66.7% per una stampa a 150DPI;
 - 33.4% per una stampa a 300DPI

§

5.4.2.3 FORMATI IMMAGINE DA UTILIZZARE

I formati immagine testati e funzionanti (quindi da adottare per evitare eventuali problemi) sono i seguenti:

- JPG
- GIF (senza trasparenza)

§

5.4.2.4 AMBIENTE CLIENT

L’utilizzo di un formato CSS, permette al software Decoder 2D-Plus[®] di presentare il contenuto del timbro digitale, elaborando una trasformazione XMLT e generando un file XHTML il quale viene presentato all’utente tramite la visualizzazione di Internet Explorer, all’interno di una finestra gestita dal software.

§

| | | | |
|---|--|---------------|---|
|  | Descrizione commessa | pag. 27 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

5.4.3 USO DEL NAMESPACE HTTP://WWW.TIMBRODIGITALE.ORG/TD_TAGS

L'utilizzo del namespace `http://www.timbrodigitale.org/TD_tags` permette al client de decodifica, Decoder 2D-Plus, una gestione più sofisticata e completa dei dati inseriti nella struttura XML; il suo uso non è tuttavia obbligatorio.

La struttura XML del namespace `http://www.timbrodigitale.org/TD_tags` è la seguente:

```
<?xml version="1.0" encoding="utf-8"?>
<TD:Global xmlns:TD="http://www.timbrodigitale.org/TD_tags">
  <!-- informazioni relativa al file XML -->

  <TD:XSL_Def>
    <!-- indirizzo download -->
    <TD:orig>http://www.timbrodigitale.com/Decoder2DPlus/xsl//</TD:orig>
    <!-- tipo trasformazione -->
    <TD:type>CSS</TD:type>
  </TD:XSL_Def>

  <!-- informazioni relativa al codice 2D-Plus -->
  <TD:C2D-Plus>

    <!-- Codice Applicazione -->
    <TD:AppCode>199</TD:AppCode>

    <!-- Codice Cliente -->
    <TD:CstmeCode>1</TD:CstmeCode>

    <!-- Codice Licenza -->
    <TD:LicCode>018-09</TD:LicCode>

  </TD:C2D-Plus>
</TD:Global>
```

L'utilizzo più immediato del namespace `http://www.timbrodigitale.org/TD_tags`, è quello relativo alla definizione dell'indirizzo di download del file XSL (ovvero .XSL.ZIP) associato tramite l'elemento `<TD:orig>`.

Se presente l'elemento `<TD:orig>`, la configurazione di default del Decoder-2DPlus relativa all'indirizzo da utilizzare per il download dei file XSL viene ignorata; al suo posto viene utilizzato il percorso indicato come attributo di questo elemento: tipicamente una URL.

Nell'esempio:

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 28 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

`<TD:orig>http://www.timbrodigitale.com/xsl/</TD:orig>`

Questa caratteristica, permette una grande flessibilità sulla pubblicazione e collocazione dei file XSL, consentendo anche di specializzare per ogni tipo di XML, la URL da cui prelevare i file di supporto necessari.

Qualora il nodo `<TD:orig>` non sia presente all'interno della struttura XML, il Decoder-2DPlus si baserà sulle configurazioni di default presenti nel file *Decoder2DPlus.ini* e visualizzabili a video dal relativo Menu "Options".

Quando si utilizzano codici applicazione standard, gli altri nodi presenti nel namespace TD sono ridondanti, in quanto le informazioni che possono essere da loro dichiarate, fanno già parte della semantica del codice applicazione stesso; questi nodi, sono quindi dedicati a supportare una migliore gestione di codici applicativi proprietari.

Per utilizzare gli elementi di questo namespace in un documento XML, è naturalmente necessario dichiarare il namespace stesso: `http://www.timbrodigitale.org/TD_tags`.

Esempio:

```
<?xml version="1.0" encoding="utf-8"?>
<Certificato>
  <TipoCert>Comune</TipoCert>
  <NumCert>00045</NumCert>
  <DatiPersonalili>
    <Nome>Mario</Nome>
    <Cognome>Rossi</Cognome>
    <Cap>00100</Cap>
    <Telefono>012345678</Telefono>
  </DatiPersonalili>
  <TD:Global xmlns:TD="http://www.timbrodigitale.org/TD_tags">
    <TD:XSL_Def>
      <TD:orig>http://www.comunedioizo.com/xsl/</TD:orig>
    </TD:XSL_Def>
    <TD:C2D-Plus>
      <TD:AppCode>201</TD:AppCode>
    </TD:C2D-Plus>
  </TD:Global>
  <FooterNote>Documento Generato il XX-XX-XXXX alle XX:XX</FooterNote>
```

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 29 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

</Certificato>

Questo XML fa uso del namespace http://www.timbrodigitale.org/TD_tags associato al prefisso **TD**, come visibile nella parte in rosso del documento; inoltre vengono definiti gli attributi degli elementi:

"TD:Global/TD:XSL_Def/TD:orig"

"TD:Global/TD:C2D-Plus/TD:AppCode".

NOTA: alla data di pubblicazione del presente documento, si richiede che il namespace http://www.timbrodigitale.org/TD_tags, **sia sempre dichiarato** associato al prefisso **TD**. Deve inoltre essere definito un URI del repository dei file xsl.zip che finisca necessariamente con una barra "/". Es. <http://www.timbrodigitale.com/xsl/>

§

| | | | |
|---|--|----------------------|---|
|  | Descrizione commessa | pag. 30 di 30 |  |
| | Documenta per la realizzazione di documenti XML/XSL e programmi correlati | | |
| | Nome del file di riferimento | | |
| | [GT50-16-iTch] T I DST useful data structures [2.0].doc | | |

5.4.4 RSA SIGNATURE (IN PROGRESS)

Questo codice applicazione, prevede di realizzare una timbro digitale, contenente un file di dati compresso e firmato tramite algoritmo asimmetrico RSA.

La firma risultante, non è a norma legale; pur mantenendo le caratteristiche tecniche di Integrità, Autenticità e Non Ripudio, la sua struttura non risponde ne alla normativa italiana, ne a strutture dati standard quali PKCS#7.

La struttura dati scelta consente di ridurre al minimo i dati utilizzati; molto semplicemente:

<payload>_<ID-Cert>_<Signature>

<payload>: dati firmati

<ID-Cert>: identificativo (20 byte) del certificato di chiave pubblica del firmatario;

<Signature>: 1024 byte di firma RSA

Nota: il simbolo “_” (underscore) è utilizzato come separatore dei tre elementi;

