

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
文書名	<pre><xsd:simpleType name="e.文書名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="文書名" type="e.文書名-0"/></pre>	e.文書名-0	文書名	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="様式名" type="e.文書名-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="文書番号"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><様式名>燃料等(電気)使用状況届出書</様式名> モジュールの"エレメント"を参照する例 <様式識別> <文書名>燃料等(電気)使用状況届出書</文書名> <文書番号>様式第1号</文書番号> </様式識別></pre>
文書番号	<pre><xsd:simpleType name="e.文書番号-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="文書番号" type="e.文書番号-0"/></pre>	e.文書番号-0	文書番号	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="様式番号" type="e.文書番号-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="文書番号"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><様式番号>様式第1号</様式番号> モジュールの"エレメント"を参照する例 <様式識別> <文書名>燃料等(電気)使用状況届出書</文書名> <文書番号>様式第1号</文書番号> </様式識別></pre>
様式規定法令	<pre><xsd:simpleType name="e.様式規定法令-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="様式規定法令" type="e.様式規定法令-0"/></pre>	e.様式規定法令-0	様式規定法令	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="根拠法令" type="e.様式規定法令-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="様式規定法令"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><根拠法令>第4条関係</根拠法令> モジュールの"エレメント"を参照する例 <様式識別> <文書名>燃料等(電気)使用状況届出書</文書名> <様式規定法令>第4条関係</様式規定法令> </様式識別></pre>
あて先	<pre><xsd:simpleType name="e.あて先-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="あて先" type="e.あて先-0"/></pre>	e.あて先-0	あて先	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="提出先" type="e.あて先-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="あて先"/> <xsd:element ref="経由"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><提出先>国土交通大臣殿</提出先> モジュールの"エレメント"を参照する例 <様式識別> <文書名> に関する申請</文書名> <あて先>国土交通大臣殿</あて先> <経由>東京都知事殿</経由> </様式識別></pre>
経由	<pre><xsd:simpleType name="e.経由-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="経由" type="e.経由-0"/></pre>	e.経由-0	経由	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="経由したところ" type="e.経由-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="あて先"/> <xsd:element ref="経由"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><経由したところ>東京都知事殿</経由したところ> モジュールの"エレメント"を参照する例 <様式識別> <文書名> に関する申請</文書名> <あて先>国土交通大臣殿</あて先> <経由>東京都知事殿</経由> </様式識別></pre>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
申請届出根拠	<pre><xsd:simpleType name="e.申請届出根拠-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="申請届出根拠" type="e.申請届出根拠-0"/></pre>	e.申請届出根拠-0	申請届出根拠	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="届出根拠" type="e.申請届出根拠-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="申請届出根拠"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <届出根拠>エネルギー使用の合理化に関する法律第6条第2頁の規定に基づき次のとおり届け出ます。</届出根拠> モジュールの"エレメント"を参照する例 <様式識別> <文書名> に関する申請</文書名> <申請届出根拠>エネルギー使用の合理化に関する法律第6条第2頁の規定に基づき次のとおり届け出ます。</申請届出根拠> </様式識別></pre>
法令名	<pre><xsd:simpleType name="e.法令名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="法令名" type="e.法令名-0"/></pre>	e.法令名-0	法令名	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="法令の名称" type="e.法令名-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="法令名"/> <xsd:element ref="手続名"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <法令の名称>輸入の促進及び対内投資事業の円滑化に関する臨時措置法</法令の名称> モジュールの"エレメント"を参照する例 <様式識別> <文書名> に関する申請</文書名> <法令名>輸入の促進及び対内投資事業の円滑化に関する臨時措置法</法令名> <手続名>特定製品輸入事業者認定申請</手続名> </様式識別></pre>
手続名	<pre><xsd:simpleType name="e.手続名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="手続名" type="e.手続名-0"/></pre>	e.手続名-0	手続名	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="手続の名称" type="e.手続名-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="様式識別"> <xsd:sequence> <xsd:element ref="文書名"/> <xsd:element ref="法令名"/> <xsd:element ref="手続名"/> </xsd:sequence> </xsd:complexType> <xsd:element name="様式識別" type="様式識別"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <手続の名称>特定製品輸入事業者認定申請</手続の名称> モジュールの"エレメント"を参照する例 <様式識別> <文書名> に関する申請</文書名> <法令名>輸入の促進及び対内投資事業の円滑化に関する臨時措置法</法令名> <手続名>特定製品輸入事業者認定申請</手続名> </様式識別></pre>

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント	XML Schema定義	マークアップ
氏名 氏名	氏、名の定義と宣言 <pre><xsd:simpleType name="e.氏-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="氏" type="e.氏-0"/> <xsd:element name="名" type="e.名-0"/></pre> 基本パターン (1つの要素として扱う場合) <pre><xsd:simpleType name="e.氏名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="氏名" type="e.氏名-0"/></pre> その他のパターン (複数の要素に分解して扱う場合) <pre><xsd:complexType name="e.氏名-1"> <xsd:sequence> <xsd:element ref="氏"/> <xsd:element ref="名"/> </xsd:sequence> </xsd:complexType> <xsd:element name="氏名-1" type="e.氏名-1"/></pre>	基本パターン e.氏名-0 その他のパターン e.氏名-1 e.氏-0 e.名-0	基本パターン 氏名 その他のパターン 氏名-1 氏名	XML Schema定義 基本パターン (1つの要素として扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="提出者" type="e.氏名-0"/> <!--モジュールの"エレメント"を参照する例--> <xsd:complexType name="代表者氏名"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="フリガナ"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者氏名" type="代表者氏名"/></pre> その他のパターン (複数の要素に分解して扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="提出者" type="e.氏名-1"/> <!--モジュールの"エレメント"を参照する例--> <xsd:complexType name="代表者氏名"> <xsd:sequence> <xsd:element ref="氏名-1"/> <xsd:element ref="フリガナ"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者氏名" type="代表者氏名"/></pre>	マークアップ 基本パターン (1つの要素として扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <pre><提出者>山田太郎</提出者> <!--モジュールの"エレメント"を参照する例--> <代表者氏名> <氏名>山田太郎</氏名> <フリガナ>ヤマダタロウ</フリガナ> </代表者氏名></pre> その他のパターン (複数の要素に分解して扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <pre><提出者> <氏>山田</氏> <名>太郎</名> </提出者> <!--モジュールの"エレメント"を参照する例--> <代表者氏名> <氏名-1> <氏>山田</氏> <名>太郎</名> </氏名-1> <フリガナ>ヤマダタロウ</フリガナ> </代表者氏名></pre>
		法人団体名	<pre><xsd:simpleType name="e.法人団体名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="法人団体名" type="e.法人団体名-0"/></pre>	e.法人団体名-0	法人団体名

モジュール名	モジュールの定義	モジュール		XML Schema定義	様式設計者の利用例
		データ型	エレメント		
住所	<p>都道府県、市郡、区、町名等の定義と宣言</p> <pre><xsd:simpleType name="e.都道府県-0"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="北海道"/> <xsd:enumeration value="青森県"/> (中略) <xsd:enumeration value="沖縄県"/> </xsd:restriction> </xsd:simpleType></pre> <p>市郡</p> <pre><xsd:simpleType name="e.市郡-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType></pre> <p>区</p> <pre><xsd:simpleType name="e.区-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType></pre> <p>町名等</p> <pre><xsd:simpleType name="e.町名等-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType></pre> <p>基本パターン (1つの要素として扱う場合)</p> <pre><xsd:simpleType name="e.住所-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="住所" type="e.住所-0"/></pre> <p>その他のパターン (複数の要素に分解して扱う場合)</p> <pre><xsd:complexType name="e.住所-1"> <xsd:sequence> <xsd:element ref="都道府県"/> <xsd:element ref="市郡"/> <xsd:element ref="区"/> <xsd:element ref="町名等"/> </xsd:sequence> </xsd:complexType> <xsd:element name="住所-1" type="e.住所-1"/></pre>	基本パターン e.住所-0	基本パターン 住所	XML Schema定義	<p>基本パターン (1つの要素として扱う場合)</p> <pre><xsd:element name="申請者の住所" type="e.住所-0"/> <xsd:complexType name="代表者の住所"> <xsd:sequence> <xsd:element ref="郵便番号"/> <xsd:element ref="住所"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者の住所" type="代表者の住所"/></pre> <p>その他のパターン (複数の要素に分解して扱う場合)</p> <pre><xsd:element name="申請者の住所" type="e.住所-1"/> <xsd:complexType name="代表者の住所"> <xsd:sequence> <xsd:element ref="郵便番号"/> <xsd:element ref="住所-1"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者の住所" type="代表者の住所"/></pre>
		その他のパターン e.住所-1 e.都道府県-0 e.市郡-0 e.区-0 e.町名等-0	その他のパターン 住所-1 都道府県 市郡 区 町名等	マークアップ	<p>基本パターン (1つの要素として扱う場合)</p> <pre><xsd:element name="申請者の住所" type="e.住所-0"/> <申請者の住所>東京都港区三田一丁目 4番 28号</申請者の住所> <xsd:element name="代表者の住所" type="代表者の住所"> <代表者の住所> <郵便番号>100-1111</郵便番号> <住所>東京都港区三田一丁目 4番 28号</住所> </代表者の住所></pre> <p>その他のパターン (複数の要素に分解して扱う場合)</p> <pre><xsd:element name="申請者の住所" type="e.住所-1"/> <申請者の住所> <都道府県>東京都</都道府県> <市郡></市郡> <区>港区</区> <町名等>三田一丁目 4番 28号</町名等> </申請者の住所> <xsd:element name="代表者の住所" type="代表者の住所"> <代表者の住所> <郵便番号>100-1111</郵便番号> <住所-1> <都道府県>東京都</都道府県> <市郡></市郡> <区>港区</区> <町名等>三田一丁目 4番 28号</町名等> </住所-1> </代表者の住所></pre>
郵便番号	<pre><xsd:simpleType name="e.郵便番号-0"> <xsd:restriction base="xsd:string"> <xsd:pattern value="^\d{3}-\d{4}\$"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="郵便番号" type="e.郵便番号-0"/></pre>	e.郵便番号-0	郵便番号	XML Schema定義	<p>モジュールの"データ型"を参照して別名を宣言する例</p> <pre><xsd:element name="申請者の郵便番号" type="e.郵便番号-0"/> <xsd:complexType name="代表者の住所"> <xsd:sequence> <xsd:element ref="郵便番号"/> <xsd:element ref="住所"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者の住所" type="代表者の住所"/></pre>
				マークアップ	<p>モジュールの"データ型"を参照して別名を宣言する例</p> <pre><申請者の郵便番号>100-1111</申請者の郵便番号> <xsd:element name="代表者の住所" type="代表者の住所"> <代表者の住所> <郵便番号>100-1111</郵便番号> <住所>東京都港区三田一丁目 4番 28号</住所> </代表者の住所></pre>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
電話番号	<pre><xsd:simpleType name="e.電話番号-0"> <xsd:restriction base="xsd:string"> <xsd:pattern value="[0-9]*+##-##(##)]*" /> </xsd:restriction> </xsd:simpleType> <xsd:element name="電話番号" type="e.電話番号-0"/></pre>	e.電話番号-0	電話番号	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="電話" type="e.電話番号-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="連絡先"> <xsd:sequence> <xsd:element ref="電話番号" /> <xsd:element ref="FAX番号" /> <xsd:element ref="電子メールアドレス" /> </xsd:sequence> </xsd:complexType> <xsd:element name="連絡先" type="連絡先"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><電話>03-1234-5678</電話> モジュールの"エレメント"を参照する例 <連絡先> <電話番号>03-1234-5678</電話番号> <FAX番号>03-8765-4321</FAX番号> <電子メールアドレス>aaaa@bbb.co.jp</電子メールアドレス> </連絡先></pre>
FAX番号	<pre><xsd:simpleType name="e.FAX番号-0"> <xsd:restriction base="xsd:string"> <xsd:pattern value="[0-9]*+##-##(##)]*" /> </xsd:restriction> </xsd:simpleType> <xsd:element name="FAX番号" type="e.FAX番号-0"/></pre>	e.FAX番号-0	FAX番号	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="FAX" type="e.FAX番号-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="連絡先"> <xsd:sequence> <xsd:element ref="電話番号" /> <xsd:element ref="FAX番号" /> <xsd:element ref="電子メールアドレス" /> </xsd:sequence> </xsd:complexType> <xsd:element name="連絡先" type="連絡先"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><FAX>03-8765-4321</FAX> モジュールの"エレメント"を参照する例 <連絡先> <電話番号>03-1234-5678</電話番号> <FAX番号>03-8765-4321</FAX番号> <電子メールアドレス>aaaa@bbb.co.jp</電子メールアドレス> </連絡先></pre>
電子メールアドレス	<pre><xsd:simpleType name="e.電子メールアドレス-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="電子メールアドレス" type="e.電子メールアドレス-0"/></pre>	e.電子メールアドレス-0	電子メールアドレス	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="メール" type="e.電子メールアドレス-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="連絡先"> <xsd:sequence> <xsd:element ref="電話番号" /> <xsd:element ref="FAX番号" /> <xsd:element ref="電子メールアドレス" /> </xsd:sequence> </xsd:complexType> <xsd:element name="連絡先" type="連絡先"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><メール>aaaa@bbb.co.jp</メール> モジュールの"エレメント"を参照する例 <連絡先> <電話番号>03-1234-5678</電話番号> <FAX番号>03-8765-4321</FAX番号> <電子メールアドレス>aaaa@bbb.co.jp</電子メールアドレス> </連絡先></pre>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント	XML Schema定義	マークアップ
生年月日	文字列型日付形式を適用 <xsd:element name='生年月日' type='e.日付-0'/>	e.日付-0	文字列型日付を適用 生年月日	文字列型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='申請者の生年月日' type='e.日付-0'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者'> <xsd:sequence> <xsd:element ref='氏名' /> <xsd:element ref='生年月日' /> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者' />	
	年号型日付形式を適用 <xsd:element name='生年月日-1' type='e.日付-1'/>	e.日付-1	年号型日付を適用 生年月日-1	年号型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='申請者の生年月日' type='e.日付-1'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者'> <xsd:sequence> <xsd:element ref='氏名' /> <xsd:element ref='生年月日-1' /> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者' />	
	西暦型日付形式を適用 <xsd:element name='生年月日-2' type='e.日付-2'/>	e.日付-2	西暦型日付を適用 生年月日-2	西暦型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='申請者の生年月日' type='e.日付-2'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者'> <xsd:sequence> <xsd:element ref='氏名' /> <xsd:element ref='生年月日-2' /> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者' />	
	ISO8601型日付形式を適用 <xsd:element name='生年月日-3' type='e.日付-3'/>	e.日付-3	ISO8601型日付を適用 生年月日-3	ISO8601型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='申請者の生年月日' type='e.日付-3'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者'> <xsd:sequence> <xsd:element ref='氏名' /> <xsd:element ref='生年月日-3' /> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者' />	
				マークアップ	文字列型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <申請者の生年月日>昭和42年6月25日</申請者の生年月日> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <生年月日>昭和42年6月25日</生年月日> </代表者> 年号型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <申請者の生年月日> <年号>昭和</年号> <年>42</年> <月>6</月> <日>25</日> </申請者の生年月日> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <生年月日-1> <年号>昭和</年号> <年>42</年> <月>6</月> <日>25</日> </生年月日-1> </代表者> 西暦型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <申請者の生年月日> <年-2>1967</年-2> <月>6</月> <日>25</日> </申請者の生年月日> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <生年月日-2> <年-2>1967</年-2> <月>6</月> <日>25</日> </生年月日-2> </代表者> ISO8601型日付形式を適用 モジュールの"データ型"を参照して別名を宣言する例 <申請者の生年月日>19670625</申請者の生年月日> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <生年月日-3>19670625</生年月日-3> </代表者>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
性別	<pre><xsd:simpleType name="e.性別-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="性別" type="e.性別-0"/></pre>	e.性別-0	性別	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="申請者の性別" type="e.性別-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="管理者"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="性別"/> </xsd:sequence> </xsd:complexType> <xsd:element name="管理者" type="管理者"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <申請者の性別>男</申請者の性別> モジュールの"エレメント"を参照する例 <管理者> <氏名>山田太郎</氏名> <性別>男</性別> </管理者></pre>
役職名	<pre><xsd:simpleType name="e.役職名-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="役職名" type="e.役職名-0"/></pre>	e.役職名-0	役職名	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="代表者の役職" type="e.役職-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="代表者"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="役職名"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者" type="代表者"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <代表者の役職>代表取締役</代表者の役職> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <役職名>代表取締役</役職名> </代表者></pre>
職業	<pre><xsd:simpleType name="e.職業-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="職業" type="e.職業-0"/></pre>	e.職業-0	職業	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="職種" type="e.職業-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="受験者"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="職業"/> </xsd:sequence> </xsd:complexType> <xsd:element name="受験者" type="受験者"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <職種>会社員</職種> モジュールの"エレメント"を参照する例 <受験者> <氏名>山田太郎</氏名> <職業>会社員</職業> </受験者></pre>
国籍	<pre><xsd:simpleType name="e.国籍-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="国籍" type="e.国籍-0"/></pre>	e.国籍-0	国籍	XML Schema定義	<pre>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="申請者の国籍" type="e.国籍-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="船長"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="国籍"/> </xsd:sequence> </xsd:complexType> <xsd:element name="船長" type="船長"/></pre>
				マークアップ	<pre>モジュールの"データ型"を参照して別名を宣言する例 <申請者の国籍>日本</申請者の国籍> モジュールの"エレメント"を参照する例 <船長> <氏名>山田太郎</氏名> <国籍>日本</国籍> </船長></pre>

モジュール名	モジュールの定義	モジュール		様式設計者の利用例		
		データ型	エレメント	XML Schema定義	マークアップ	
本籍	住所の基本パターンを適用 <xsd:element name='本籍' type='e.住所-0'/>	基本パターン e.住所-0	基本パターン 本籍	XML Schema定義 基本パターン(1つの要素として扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='本籍地' type='e.住所-0'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者' <xsd:sequence <xsd:element ref='氏名'/> <xsd:element ref='本籍'/> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者'/>		
	住所のその他のパターンを適用 <xsd:element name='本籍-1' type='e.住所-1'/>	その他のパターン e.住所-1	その他のパターン 本籍-1	XML Schema定義 基本パターン(1つの要素として扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='本籍地' type='e.住所-1'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='代表者' <xsd:sequence <xsd:element ref='氏名'/> <xsd:element ref='本籍-1'/> </xsd:sequence> </xsd:complexType> <xsd:element name='代表者' type='代表者'/>		
続柄	<xsd:simpleType name='e.続柄-0' <xsd:restriction base='xsd:string' </xsd:restriction> </xsd:simpleType> <xsd:element name='続柄' type='e.続柄-0'/>	e.続柄-0	続柄	XML Schema定義 基本パターン(1つの要素として扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <本籍地>東京都港区三田一丁目4番28号</本籍地> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <本籍>東京都港区三田一丁目4番28号</本籍> </代表者>		
				XML Schema定義 その他のパターン(複数の要素に分解して扱う場合) モジュールの"データ型"を参照して別名を宣言する例 <本籍地> <都道府県>東京都</都道府県> <市郡></市郡> <区>港区</区> <町名等>三田一丁目4番28号</町名等> </本籍地> モジュールの"エレメント"を参照する例 <代表者> <氏名>山田太郎</氏名> <本籍-1> <都道府県>東京都</都道府県> <市郡></市郡> <区>港区</区> <町名等>三田一丁目4番28号</町名等> </本籍-1> </代表者>		
業種	<xsd:simpleType name='e.業種-0' <xsd:restriction base='xsd:string' </xsd:restriction> </xsd:simpleType> <xsd:element name='業種' type='e.業種-0'/>	e.業種-0	業種	XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name='事業内容' type='e.業種-0'/> モジュールの"エレメント"を参照する例 <xsd:complexType name='企業情報' <xsd:sequence <xsd:element ref='名称'/> <xsd:element ref='業種'/> </xsd:sequence> </xsd:complexType> <xsd:element name='企業情報' type='企業情報'/>		
				XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <あなたとの続柄>妻</あなたとの続柄> モジュールの"エレメント"を参照する例 <扶養者> <氏名>山田花子</氏名> <続柄>妻</続柄> </扶養者>		
業種	<xsd:simpleType name='e.業種-0' <xsd:restriction base='xsd:string' </xsd:restriction> </xsd:simpleType> <xsd:element name='業種' type='e.業種-0'/>	e.業種-0	業種	XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <事業内容>製造業</事業内容> モジュールの"エレメント"を参照する例 <企業情報> <名称> 株式会社</名称> <業種>製造業</業種> </企業情報>		
				XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <あなたとの続柄>妻</あなたとの続柄> モジュールの"エレメント"を参照する例 <扶養者> <氏名>山田花子</氏名> <続柄>妻</続柄> </扶養者>		

モジュール名	モジュールの定義	モジュール		様式設計者の利用例
		データ型	エレメント	
日付 年号 年月日	<pre><年号、年、月、日の定義と宣言 <xsd:simpleType name="e.年号-0"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="西暦"/> <xsd:enumeration value="明治"/> <xsd:enumeration value="大正"/> <xsd:enumeration value="昭和"/> <xsd:enumeration value="平成"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.年-0"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="9999"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.年-1"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="999"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.年-2"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="9999"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.月-0"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="12"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.日-0"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="31"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="年号" type="e.年号-0"/> <xsd:element name="年" type="e.年-0"/> <xsd:element name="年-1" type="e.年-1"/> <xsd:element name="年-2" type="e.年-2"/> <xsd:element name="月" type="e.月-0"/> <xsd:element name="日" type="e.日-0"/> -文字列型日付形式 <xsd:simpleType name="e.日付-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> 年号型日付形式 <xsd:complexType name="e.日付-1"> <xsd:sequence> <xsd:element ref="年号"/> <xsd:element ref="年"/> <xsd:element ref="月"/> <xsd:element ref="日"/> </xsd:sequence> </xsd:complexType> <xsd:element name="日付-1" type="e.日付-1"/> 西暦型日付形式 <xsd:simpleType name="e.日付-2"> <xsd:restriction base="xsd:string"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="999"/> </xsd:restriction> </xsd:simpleType> ISO8601型日付形式 <xsd:simpleType name="e.日付-3"> <xsd:restriction base="xsd:string"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="9999"/> </xsd:restriction> </xsd:simpleType></pre>	<pre>文字列型日付形式 e.日付-0 年号型日付形式 e.日付-1 年号 e.年-0 年 e.年-1 年-1 e.年-2 年-2 e.月-0 月 e.日-0 日 西暦型日付形式 e.日付-2 年-2 月 e.月-0 日 ISO8601型日付形式 e.日付-3</pre>	<pre>XML Schema定義 文字列型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="許認日" type="e.日付-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="許認"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="日付"/> </xsd:sequence> </xsd:complexType> <xsd:element name="許認" type="許認"/> 年号型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="許認日" type="e.日付-1"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="許認"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="日付-1"/> </xsd:sequence> </xsd:complexType> <xsd:element name="許認" type="許認"/> 西暦型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="許認日" type="e.日付-2"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="許認"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="日付-2"/> </xsd:sequence> </xsd:complexType> <xsd:element name="許認" type="許認"/> ISO8601型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="許認日" type="e.日付-3"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="許認"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="日付-3"/> </xsd:sequence> </xsd:complexType> <xsd:element name="許認" type="許認"/></pre>	<pre>マークアップ 文字列型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <認許日>12年1月30日</認許日> モジュールの"エレメント"を参照する例 <許認> <氏名>山田太郎</氏名> <日付>12年1月30日</日付> </許認> 年号型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <認許日> <年号>平成</年号> <年>12</年> <月>1</月> <日>30</日> </認許日> モジュールの"エレメント"を参照する例 <許認> <氏名>山田太郎</氏名> <日付-1> <年号>平成</年号> <年>12</年> <月>1</月> <日>30</日> </日付-1> </許認> 西暦型日付形式 モジュールの"データ型"を参照して別名を宣言する例 <認許日> <年-2>2000</年-2> <月>1</月> <日>30</日> </認許日> モジュールの"エレメント"を参照する例 <許認> <氏名>山田太郎</氏名> <日付-2> <年-2>2000</年-2> <月>1</月> <日>30</日> </日付-2> </許認></pre>

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
	西暦型日付形式 <xsd:complexType name="e.日付-2"> <xsd:sequence> <xsd:element ref="年-2"/> <xsd:element ref="月"/> <xsd:element ref="日"/> </xsd:sequence> </xsd:complexType> <xsd:element name="日付-2" type="e.日付-2"/> ISO8601型日付形式 <xsd:simpleType name="e.日付-3"> <xsd:restriction base="xsd:string"> <xsd:pattern value="*d{8}"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="日付-3" type="e.日付-3"/>				ISO8601型日付形式 ・モジュールの"データ型"を参照して別名を宣言する例 <許認日>20000130</許認日> ・モジュールの"エレメント"を参照する例 <許認> <氏名>山田太郎</氏名> <日付-3>20000130</日付-3> </許認>
時刻 時 分	時、分の定義の宣言 <xsd:simpleType name="e.時-0"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="0"/> <xsd:maxInclusive value="23"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.分-0"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="0"/> <xsd:maxInclusive value="59"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="時" type="e.時-0"/> <xsd:element name="分" type="e.分-0"/> 文字列型時刻形式 <xsd:simpleType name="e.時刻-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="時刻" type="e.時刻-0"/> 時分型時刻形式 <xsd:complexType name="e.時刻-1"> <xsd:sequence> <xsd:element ref="時"/> <xsd:element ref="分"/> </xsd:sequence> </xsd:complexType> <xsd:element name="時刻-1" type="e.時刻-1"/> ISO8601型時刻形式 <xsd:simpleType name="e.時刻-2"> <xsd:restriction base="xsd:string"> <xsd:pattern value="*d{4}"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="時刻-2" type="e.時刻-2"/>	文字列型時刻形式 e.時刻-0 時分型時刻形式 e.時刻-1 e.時-0 e.分-0 ISO8601型時刻形式 e.時刻-2	文字列型時刻形式 時刻 時分型時刻形式 時刻-1 時 分 ISO8601型時刻形式 時刻-2	XML Schema定義 文字列型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="提出時刻" type="e.時刻-0"/> ・モジュールの"エレメント"を参照する例 <xsd:complexType name="提出"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="時刻"/> </xsd:sequence> </xsd:complexType> <xsd:element name="提出" type="提出"/> 時分型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="提出時刻" type="e.時刻-1"/> ・モジュールの"エレメント"を参照する例 <xsd:complexType name="提出"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="時刻-1"/> </xsd:sequence> </xsd:complexType> <xsd:element name="提出" type="提出"/> ISO8601型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="提出時刻" type="e.時刻-2"/> ・モジュールの"エレメント"を参照する例 <xsd:complexType name="提出"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="時刻-2"/> </xsd:sequence> </xsd:complexType> <xsd:element name="提出" type="提出"/>	
				マークアップ	文字列型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <提出時刻>1020</提出時刻> ・モジュールの"エレメント"を参照する例 <提出> <氏名>山田太郎</氏名> <時刻>10時20分</時刻> </提出> 時分型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <提出時刻> <時>10</時> <分>20</分> </提出時刻> ・モジュールの"エレメント"を参照する例 <提出> <氏名>山田太郎</氏名> <時刻-1> <時>10</時> <分>20</分> </時刻-1> </提出> ISO8601型時刻形式 ・モジュールの"データ型"を参照して別名を宣言する例 <提出時刻> 1020</提出時刻> ・モジュールの"エレメント"を参照する例 <提出> <氏名>山田太郎</氏名> <時刻-2>1020</時刻-2> </提出>

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	要素	XML Schema定義	
金額	金額、資本金、売上高、単位付きの金額の定義と宣言 <xsd:simpleType name="e.金額-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.円-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.千円-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.百万円-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.ﾄﾞﾚ-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.千ﾄﾞﾚ-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.百万ﾄﾞﾚ-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="金額" type="e.金額-0"/> <xsd:element name="円" type="e.円-0"/> <xsd:element name="千円" type="e.千円-0"/> <xsd:element name="百万円" type="e.百万円-0"/> <xsd:element name="ﾄﾞﾚ" type="e.ﾄﾞﾚ-0"/> <xsd:element name="千ﾄﾞﾚ" type="e.千ﾄﾞﾚ-0"/> <xsd:element name="百万ﾄﾞﾚ" type="e.百万ﾄﾞﾚ-0"/>	e.円-0 e.千円-0 e.百万円-0 e.ﾄﾞﾚ-0 e.千ﾄﾞﾚ-0 e.百万ﾄﾞﾚ-0 e.金額-0	円 千円 百万円 ﾄﾞﾚ 千ﾄﾞﾚ 百万ﾄﾞﾚ 金額	XML Schema定義 単位なしの場合 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="給与" type="e.金額-0"/> モジュールの"要素"を参照する例 <xsd:complexType name="給与"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="金額"/> </xsd:sequence> </xsd:complexType> <xsd:element name="給与" type="給与"/> 単位ありの場合 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="交通費-円" type="e.円-0"/> モジュールの"要素"を参照する例 <xsd:complexType name="交通費"> <xsd:sequence> <xsd:element ref="費目"/> <xsd:element ref="円"/> </xsd:sequence> </xsd:complexType> <xsd:element name="交通費" type="交通費"/>	
		マークアップ 単位なしの場合 モジュールの"データ型"を参照して別名を宣言する例 <給与>300000</給与> モジュールの"要素"を参照する例 <給与> <氏名>山田太郎</氏名> <金額>300000</金額> </給与> 単位ありの場合 モジュールの"データ型"を参照し、別名を宣言する例 <交通費-円>300</交通費-円> モジュールの"要素"を参照する例 <交通費> <費目>タクシー代</費目> <円>860</円> </交通費>			
資本金	<xsd:simpleType name="e.資本金-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="資本金" type="e.資本金-0"/>	e.資本金-0	資本金	XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="資本" type="e.資本金-0"/> モジュールの"要素"を参照する例 <xsd:complexType name="企業情報"> <xsd:sequence> <xsd:element ref="法人団体名"/> <xsd:element ref="資本金"/> <xsd:element ref="売上高"/> </xsd:sequence> </xsd:complexType> <xsd:element name="企業情報" type="企業情報"/>	
				マークアップ モジュールの"データ型"を参照して別名を宣言する例 <資本>10000000</資本> パターン2 <企業情報> <法人団体名>A株式会社</法人団体名> <資本金>10000000</資本金> <売上高>50000000</売上高> </企業情報>	
売上高	<xsd:simpleType name="e.売上高-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="売上高" type="e.売上高-0"/>	e.売上高-0	売上高	XML Schema定義 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="売上" type="e.売上高-0"/> モジュールの"要素"を参照する例 <xsd:complexType name="企業情報"> <xsd:sequence> <xsd:element ref="法人団体名"/> <xsd:element ref="資本金"/> <xsd:element ref="売上高"/> </xsd:sequence> </xsd:complexType> <xsd:element name="企業情報" type="企業情報"/>	
				マークアップ モジュールの"データ型"を参照して別名を宣言する例 <売上>50000000</売上> パターン2 <企業情報> <法人団体名>A株式会社</法人団体名> <資本金>10000000</資本金> <売上高>50000000</売上高> </企業情報>	

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
数量型期間	数量型期間、単位付きの数量型期間の定義と宣言 <xsd:simpleType name="e.数量型期間-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.年数-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.月数-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.日数-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.時間-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="数量型期間" type="e.数量型期間-0"/> <xsd:element name="年数" type="e.年数-0"/> <xsd:element name="月数" type="e.月数-0"/> <xsd:element name="日数" type="e.日数-0"/> <xsd:element name="時間" type="e.時間-0"/>	e.数量型期間-0 e.年数-0 e.月数-0 e.日数-0 e.時間-0	数量型期間 年数 月数 日数 時間	XML Schema定義	単位なしの場合 ・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="実施期間" type="e.数量型期間-0"/> ・モジュールの"エレメント"を参照する例 <xsd:complexType name="実施"> <xsd:sequence> <xsd:element ref="項目"/> <xsd:element ref="数量型期間"/> </xsd:sequence> </xsd:complexType> <xsd:element name="実施" type="実施"/> 単位ありの場合 ・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="実施期間" type="e.年数-0"/> ・モジュールの"エレメント"を参照する例 <xsd:complexType name="実施"> <xsd:sequence> <xsd:element ref="項目"/> <xsd:element ref="年数"/> </xsd:sequence> </xsd:complexType> <xsd:element name="実施" type="実施"/>
				マークアップ	単位なしの場合 ・モジュールの"データ型"を参照して別名を宣言する例 <実施期間>12</実施期間> ・モジュールの"エレメント"を参照する例 <実施> <項目>様式調査</項目> <数量型期間>20</数量型期間> </実施> 単位ありの場合 ・モジュールの"データ型"を参照して別名を宣言する例 <実施期間>5</実施期間> ・モジュールの"エレメント"を参照する例 <実施> <項目>ソフトウェア開発</項目> <年数>3</年数> </実施>
年齢	<xsd:simpleType name="e.年齢-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="年齢" type="e.年齢-0"/>	e.年齢-0	年齢	XML Schema定義	・モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="管理者の年齢" type="e.年齢-0"/> ・モジュールの"エレメント"を参照する例 <ELEMENT 管理者 (氏名,年齢)> <xsd:complexType name="管理者"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="年齢"/> </xsd:sequence> </xsd:complexType> <xsd:element name="管理者" type="管理者"/>
				マークアップ	・モジュールの"データ型"を参照して別名を宣言する例 <管理者の年齢>37</管理者の年齢> ・モジュールの"エレメント"を参照する例 <管理者> <氏名>山田太郎</氏名> <年齢>37</年齢> </管理者>

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント	XML Schema定義	マークアップ
従業員数	<pre><xsd:simpleType name="e.従業員数-0"> <xsd:restriction base="xsd:integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="従業員数" type="e.従業員数-0"/> <xsd:simpleType name="e.人-0"> <xsd:restriction base="integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="人" type="e.人-0"/></pre>	<pre>e.従業員数-0 e.人-0</pre>	<pre>従業員数 人</pre>	<p>XML Schema定義</p> <p>単位なしの場合 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="職員数" type="e.従業員数-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="企業情報"> <xsd:sequence> <xsd:element ref="名称"/> <xsd:element ref="従業員数"/> </xsd:sequence> </xsd:complexType> <xsd:element name="企業情報" type="企業情報"/></p> <p>単位ありの場合 モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="職員数" type="e.人-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="事業所"> <xsd:sequence> <xsd:element ref="名称"/> <xsd:element ref="人"/> </xsd:sequence> </xsd:complexType> <xsd:element name="事業所" type="事業所"/></p>	<p>マークアップ</p> <p>単位なしの場合 モジュールの"データ型"を参照して別名を宣言する例 <職員数>250</職員数> モジュールの"エレメント"を参照する例 <企業情報> <名称> 株式会社</名称> <従業員数>1000</従業員数> </企業情報></p> <p>単位ありの場合 モジュールの"データ型"を参照して別名を宣言する例 <職員数>250</職員数> モジュールの"エレメント"を参照する例 <事業所> <名称> 営業所</名称> <人>15</人> </事業所></p>
				<p>XML Schema定義</p> <p>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="所有株式数" type="e.株式数-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="企業情報"> <xsd:sequence> <xsd:element ref="名称"/> <xsd:element ref="株式数"/> </xsd:sequence> </xsd:complexType> <xsd:element name="企業情報" type="企業情報"/></p>	<p>マークアップ</p> <p>モジュールの"データ型"を参照して別名を宣言する例 <所有株式数>100</所有株式数> モジュールの"エレメント"を参照する例 <企業情報> <名称> 株式会社</名称> <株式数>100</株式数> </企業情報></p>
株式数	<pre><xsd:simpleType name="e.株式数-0"> <xsd:restriction base="integer"> </xsd:restriction> </xsd:simpleType> <xsd:element name="株式数" type="e.株式数-0"/></pre>	<pre>e.株式数-0</pre>	<pre>株式数</pre>	<p>XML Schema定義</p> <p>モジュールの"データ型"を参照して別名を宣言する例 <xsd:element name="所有株式数" type="e.株式数-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="企業情報"> <xsd:sequence> <xsd:element ref="名称"/> <xsd:element ref="株式数"/> </xsd:sequence> </xsd:complexType> <xsd:element name="企業情報" type="企業情報"/></p>	<p>マークアップ</p> <p>モジュールの"データ型"を参照して別名を宣言する例 <所有株式数>100</所有株式数> モジュールの"エレメント"を参照する例 <企業情報> <名称> 株式会社</名称> <株式数>100</株式数> </企業情報></p>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
範囲型期間 開始日付 終了日付 開始時刻 終了時刻	・開始日付、終了日付、開始時刻、終了時刻の宣言 <pre><xsd:element name='開始日付' type='e.日付-0'/> <xsd:element name='終了日付' type='e.日付-0'/> <xsd:element name='開始時刻' type='e.時刻-0'/> <xsd:element name='終了時刻' type='e.時刻-0'/></pre> ・文字列範囲型期間 <pre><xsd:simpleType name='e.範囲型期間-0'> <xsd:restriction base='xsd:string'> </xsd:restriction> </xsd:simpleType> <xsd:element name='範囲型期間' type='e.範囲型期間-0'/></pre> ・日付範囲型期間 <pre><xsd:complexType name='e.範囲型期間-1'> <xsd:sequence> <xsd:element ref='開始日付'/> <xsd:element ref='終了日付'/> </xsd:sequence> </xsd:complexType> <xsd:element name='範囲型期間-1' type='e.範囲型期間-1'/></pre> ・時刻範囲型期間 <pre><xsd:complexType name='e.範囲型期間-2'> <xsd:sequence> <xsd:element ref='開始時刻'/> <xsd:element ref='終了時刻'/> </xsd:sequence> </xsd:complexType> <xsd:element name='範囲型期間-2' type='e.範囲型期間-2'/></pre>	・文字列範囲型期間 e.範囲型期間-0	・文字列範囲型期間 範囲型期間	XML Schema定義	・文字列範囲型期間 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name='工事期間' type='e.範囲型期間-0'/></pre> ・モジュールの"エレメント"を参照する例 <pre><xsd:complexType name='工事'> <xsd:sequence> <xsd:element ref='事業者'/> <xsd:element ref='範囲型期間'/> </xsd:sequence> </xsd:complexType> <xsd:element name='工事' type='工事'/></pre> ・日付範囲型期間 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name='工事期間' type='e.範囲型期間-1'/></pre> ・モジュールの"エレメント"を参照する例 <pre><xsd:complexType name='工事'> <xsd:sequence> <xsd:element ref='事業者'/> <xsd:element ref='範囲型期間-1'/> </xsd:sequence> </xsd:complexType> <xsd:element name='工事' type='工事'/></pre> ・時刻範囲型期間の場合 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name='工事期間' type='e.範囲型期間-2'/></pre> ・モジュールの"エレメント"を参照する例 <pre><xsd:complexType name='工事'> <xsd:sequence> <xsd:element ref='事業者'/> <xsd:element ref='範囲型期間-2'/> </xsd:sequence> </xsd:complexType> <xsd:element name='工事' type='工事'/></pre>
				マークアップ ・文字列範囲型期間 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><工事期間>自平成13年10月12日～至平成13年10月23日</工事期間></pre> ・モジュールの"エレメント"を参照する例 <pre><工事> <事業者> A株式会社</事業者> <範囲型期間>自平成13年10月12日～至平成13年10月23日</範囲型期間> </工事></pre> ・日付範囲型期間 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><工事期間> <開始日付>平成13年10月12日</開始日付> <終了日付>平成13年10月23日</終了日付> </工事期間></pre> ・モジュールの"エレメント"を参照する例 <pre><工事> <事業者> A株式会社</事業者> <範囲型期間-1> <開始日付>平成13年10月12日</開始日付> <終了日付>平成13年10月23日</終了日付> </範囲型期間-1> </工事></pre> ・時刻範囲型期間 ・モジュールの"データ型"を参照して別名を宣言する例 <pre><工事期間> <開始時刻>11時30分</開始時刻> <終了時刻>16時00分</終了時刻> </工事期間></pre> ・モジュールの"エレメント"を参照する例 <pre><工事> <事業者> A株式会社</事業者> <範囲型期間-2> <開始時刻>11時30分</開始時刻> <終了時刻>16時00分</終了時刻> </範囲型期間-2> </工事></pre>	

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント	XML Schema定義	
区間 始点 終点	<p>始点・終点の定義と宣言</p> <pre><xsd:simpleType name="e.始点-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.終点-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="始点" type="e.始点-0"/> <xsd:element name="終点" type="e.終点-0"/></pre> <p>基本パターン (1つの要素として扱う場合)</p> <pre><xsd:simpleType name="e.区間-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="区間" type="e.区間-0"/></pre> <p>その他のパターン (複数の要素に分解して扱う場合)</p> <pre><xsd:complexType name="e.区間-1"> <xsd:sequence> <xsd:element ref="始点"/> <xsd:element ref="終点"/> </xsd:sequence> </xsd:complexType> <xsd:element name="区間-1" type="e.区間-1"/></pre>	基本パターン e.区間-0	基本パターン 区間	XML Schema定義	<p>基本パターン</p> <pre><xsd:element name="乗車区間" type="e.区間-0"/> <xsd:complexType name="乗車"> <xsd:sequence> <xsd:element ref="区間"/> <xsd:element ref="路線"/> </xsd:sequence> </xsd:complexType> <xsd:element name="乗車" type="乗車"/></pre> <p>その他のパターン</p> <pre><xsd:element name="乗車区間" type="e.区間-1"/> <xsd:complexType name="乗車"> <xsd:sequence> <xsd:element ref="区間-1"/> <xsd:element ref="路線"/> </xsd:sequence> </xsd:complexType> <xsd:element name="乗車" type="乗車"/></pre>
				マークアップ	<p>基本パターン (1つの要素として扱う場合)</p> <pre><乗車区間>東京から戸塚まで</乗車区間> <xsd:element name="乗車" type="乗車"> <区間>東京から戸塚まで</区間> <路線>東海道線</路線> </乗車></pre> <p>その他のパターン (複数の要素に分解して扱う場合)</p> <pre><乗車区間> <始点>東京</始点> <終点>戸塚</終点> </乗車区間> <xsd:element name="乗車" type="乗車"> <区間-1> <始点>東京</始点> <終点>戸塚</終点> </区間-1> <路線>東海道線</路線> </乗車></pre>
フリガナ ふりがな	<pre><xsd:simpleType name="e.フリガナ-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="e.ふりがな-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="フリガナ" type="e.フリガナ-0"/> <xsd:element name="ふりがな" type="e.ふりがな-0"/></pre> <p>(通常は、氏名モジュール、法人団体名モジュールと合わせて用いられる)</p>	e.フリガナ-0 e.ふりがな-0	フリガナ ふりがな	XML Schema定義	<pre><xsd:element name="代表者のフリガナ" type="e.フリガナ-0"/> <xsd:complexType name="代表者氏名"> <xsd:sequence> <xsd:element ref="氏名"/> <xsd:element ref="フリガナ"/> </xsd:sequence> </xsd:complexType> <xsd:element name="代表者氏名" type="代表者氏名"/></pre>
				マークアップ	<pre><代表者のフリガナ>ヤマダタロウ</代表者のフリガナ> <xsd:element name="代表者氏名" type="代表者氏名"> <氏名>山田太郎</氏名> <フリガナ>ヤマダタロウ</フリガナ> </代表者氏名></pre>
場所	<pre><xsd:simpleType name="e.場所-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="場所" type="e.場所-0"/></pre>	e.場所-0	場所	XML Schema定義	<pre><xsd:element name="測定場所" type="e.場所-0"/> <xsd:complexType name="測定"> <xsd:sequence> <xsd:element ref="業者名"/> <xsd:element ref="場所"/> </xsd:sequence> </xsd:complexType> <xsd:element name="測定" type="測定"/></pre>
				マークアップ	<pre><測定場所> 施設</測定場所> <xsd:element name="測定" type="測定"> <業者名> ×株式会社</業者名> <場所> ×地区</場所> </測定></pre>

付録H 申請書共通ボキャブラリ・リファレンス【XML Schema版】

モジュール名	モジュールの定義	モジュール		様式設計者の利用例	
		データ型	エレメント		
理由	<pre><xsd:simpleType name="e.理由-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="理由" type="e.理由-0"/></pre>	e.理由-0	理由	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="開始理由" type="e.理由-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="開始"> <xsd:sequence> <xsd:element ref="日付"/> <xsd:element ref="理由"/> </xsd:sequence> </xsd:complexType> <xsd:element name="開始" type="開始"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><開始理由> により/開始理由> モジュールの"エレメント"を参照する例 <開始> <日付>平成13年7月18日</日付> <理由> により</理由> </開始></pre>
符号	<pre><xsd:simpleType name="e.符号-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="符号" type="e.符号-0"/></pre>	e.符号-0	符号	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="輸入者符号" type="e.符号-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="輸入"> <xsd:sequence> <xsd:element ref="符号"/> <xsd:element ref="法人団体名"/> </xsd:sequence> </xsd:complexType> <xsd:element name="輸入" type="輸入"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><輸入者符号>A001</輸入者符号> モジュールの"エレメント"を参照する例 <輸入> <符号>A001</符号> <法人団体名>A株式会社</法人団体名> </輸入></pre>
記載説明	<pre><xsd:simpleType name="e.記載説明-0"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:element name="記載説明" type="e.記載説明-0"/></pre>	e.記載説明-0	記載説明	XML Schema定義	モジュールの"データ型"を参照して別名を宣言する例 <pre><xsd:element name="記載要領" type="e.記載説明-0"/> モジュールの"エレメント"を参照する例 <xsd:complexType name="備考欄"> <xsd:sequence> <xsd:element ref="x x"/> <xsd:element ref="記載説明"/> <xsd:element ref=""/> </xsd:sequence> </xsd:complexType> <xsd:element name="備考欄" type="備考欄"/></pre>
				マークアップ	モジュールの"データ型"を参照して別名を宣言する例 <pre><記載要領>次年度以降において燃料等(電気)の使用量が 令第2条第1項(第2項)に該当しないことが明らかである場合 は、その旨及びその理由を備考の欄に記入すること</記載 要領> モジュールの"エレメント"を参照する例 <備考欄> <x x>・・・</x x> <記載説明>次年度以降において燃料等(電気)の使用量が 令第2条第1項(第2項)に該当しないことが明らかである場合 は、その旨及びその理由を備考の欄に記入すること</記載 説明> < >・・・</ > </備考欄></pre>