PureConnect®

2021 R4

Generated:

04-November-2021

Content last updated:

13-June-2019

See Change Log for summary of changes.

GENESYS™

CIC VoiceXML Integration with Nuance Dialog Modules Version 5.2

Development Application Note

Abstract

This document describes updates required to allow Nuance Dialog Modules version 5.2 to work with the CIC VoiceXML Interpreter.

For the latest version of this document, see the PureConnect Documentation Library at: http://help.genesys.com/pureconnect.

Table of Contents

Introduction to CIC VoiceXML Integration with Nuance Dialog Modules Version 5.2

The *CIC VoiceXML Integration with Nuance Dialog Modules Version 5.2 Developer's Application Note* represents notes from Genesys PureConnect Voice XML developers and describes the changes that we need to make to allow Nuance's NDMs (version 5.2) to work with our VoiceXML interpreter. These changes may include changes to our VoiceXML code, changes to BladewareVXML's libraries, and changes to the grammars (and other files) that Nuance supplies with their NDMs.

VoiceXML changes

The following changes to our VoiceXML code are/were necessary allow our VoiceXML interpreter to successfully utilize the Nuance NDMs.

- Changes to allow the VoiceXML interpreter to pass any grammar URL query elements to the Reco subsystem. (CL#420381) (SCR:IC-73462)
- Changes to allow the user to set an `ignoreGrammarFileMimeType` flag. This allows the user to tell VoiceXML whether or not to use a grammar file's Mime type as the grammar type if no grammar type was specified in the <grammar> element. (CL#429359) (SCR:IC-74616)
- Change to `I3defaults.xml` file to define the `String.prototype.equalsIgnoreCase()` ecmascript method (since SpiderMonkey does not natively support it). (CL#429359) (SCR:IC-74616)
- Changes so that VoiceXML pays more attention to HTTP hints/properties that it has available to it. (CL#437688) (SCR:IC-76495)
- Changes so that VoiceXML does not try to load the same grammar file with different grammar IDs. (CL#442502 and CL#446041) (SCR:IC-77290)
- Update (in `I3runvxml.cfg`) the property `client.cache.cacheEntryMaxSizeMB` from 20 to 40. (CL#451641) (SCR:IC-78565)
- Changes so that VoiceXML will correctly handle Reco results when a <_value> slot is returned with other slots. (CL#451641) (SCR:IC-78565)

BladewareVXML changes

Testing with the NDMs has revealed some shortcomings in the BladewareVXML library.

Internal caching inconsistency

The BladewareVXML library seems to have an inconsistency in how it caches things.

Doing this:

```
<var name="dmname" expr="'test_SDyesno'"/>
<var name="osdm" expr="'yesno'"/>
<subdialog name="SDyesno"
                src="http://bbtestpc:8080/ndm-core/controller/ndm"
                namelist="dmname

osdm">
```

will cause caching "problems" (i.e., it won't detect when the difference when the next task uses a different ndm such as osdm=digits), while this:

```
<var name="dmname" expr="'test_SDyesno'"/>
<subdialog name="SDyesno"
                src="http://bbtestpc:8080/ndm-core/controller/ndm?osdm=yesno

"

                namelist="dmname">
```

behaves fine.

BladewareVXML has provided a fix for this. (CL#429683) (SCR:IC-74617)

Space at end of namelist

This will parse correctly:

```
<subdialog name="SDname"
        src="http://bbtestpc:8080/ndm52-name/controller"
        method="post"
        namelist="dmname">
```

while this won't:

```
<subdialog name="SDname"
        src="http://bbtestpc:8080/ndm52-name/controller"
        method="post"
        namelist="dmname

">
```

The only difference is the space at the end of the namelist string in the second version.

BladewareVXML has provided a fix for this. (CL#429683) (SCR:IC-74617)

<Subdialog> ignoring specified method

The <subdialog> element is ignoring the `method` attribute. All subdialogs with an HTTP URL specified in the `src` attribute are being issued as HTTP `GET` requests, regardless of whether `GET` or `POST` is specified in the `method` attribute.

BladewareVXML has provided a fix for this. (CL#452990) (SCR:IC-78840)

<data> element was handling \ characters oddly

The <data> element was handling \ characters oddly. It unescaped them twice.

We made a fix in the BladewareVXML code to fix this. (CL#681347) (SCR:IC-99575)

Nuance NDM changes

Nuance has supplied a patch that needs to be installed.

Incident ST- 201212190171 (for NDM-core 5.2.5 only)

Nuance supplied a patch in response to incident ST-201212190171 ("Using `<grammar expr=` is not supported under the VoiceXML 2.1 spec"). This contained a replacement `ndm.jar` file.

ndm-core 5.2.0 changes

The changes made to some of the files from the Nuance ndm-core 5.2.0 offering are described in the following:

- Grammars (ndm-core 5.2.0)
- Other files (ndm-core 5.2.0

Grammars (ndm-core 5.2.0)

The grammar files list in this section (from the `.\ndm-core\en-US\grammars\framework\ndm\nr9.0` directory) had to be tweaked for SRGS (SISR) compatibility.

Aug2001_dtmf_dmChooseDM.xml	inin_dtmf_currency.grxml	inin_dtmf_phone.grxml
Aug2001_dtmf_dmCommand.xml	inin_dtmf_customcontext.grxml	inin_dtmf_sadmCommand.grxml
inin_dtmf_boolean.grxml	inin_dtmf_date.grxml	inin_dtmf_socialsecurity.grxml
inin_dtmf_ccexpdate.grxml	inin_dtmf_digits.grxml	inin_dtmf_time.grxml
inin_dtmf_command.grxml	inin_dtmf_keypad.grxml	inin_dtmf_zipcode.grxml
inin_dtmf_creditcard.grxml	inin_dtmf_number.grxml	grammar_dmChooseDM.xml

Aug2001_dtmf_dmChooseDM.xml

This file is a modified version of the original `Aug2001_dtmf_dmChooseDM.xml` file.

- Change all `dm_*` variables to `out.dm_*`.

Aug2001_dtmf_dmCommand.xml

This file is a modified version of the original `Aug2001_dtmf_dmCommand.xml` file.

```
<item>
  <tag>
    dm_command='help';dm_confirmation_mode='NEVER';dm_confirm_string='help';
  </tag>
  *
</item>
<item>
  <tag>
    dm_command='goodbye';dm_confirmation_mode='NEVER';dm_confirm_string='goodbye';
  </tag>
  0
</item>
```

->

```
<item>
  <tag>
    out.dm_command='help';out.dm_confirmation_mode='NEVER';out.dm_confirm_string='help';
  </tag>
  <token>*</token>
</item>
<item>
  <tag>
    out.dm_command='goodbye';out.dm_confirmation_mode='NEVER';out.dm_confirm_string='goodbye';
  </tag>
  0
</item>
```

inin_dtmf_boolean.grxml

This file is a modified version of the original `dtmf_boolean.xml` file.

```
<ruleref uri="#YESNO" tag="
    var dtmf_yes = SWI_vars.dtmfyes ? SWI_vars.dtmfyes : 1;
    var dtmf_no = SWI_vars.dtmfno ? SWI_vars.dtmfno : 2;
    SWI_meaning=YESNO.SWI_literal.replace(/ /g, '');
    if (SWI_meaning == dtmf_yes) {
        SWI_meaning = true;
    }else if (SWI_meaning == dtmf_no){
        SWI_meaning = false;
    }else{
        SWI_disallow=1;
    }
    dm_root=SWI_meaning;
"/>
```

->

```
<ruleref uri="#YESNO"/>
<tag>
    var dtmf_yes = SWI_vars.dtmfyes ? SWI_vars.dtmfyes : 1;
    var dtmf_no = SWI_vars.dtmfno ? SWI_vars.dtmfno : 2;
    SWI_meaning=rules.YESNO.replace(/ /g, '');
    if (SWI_meaning == dtmf_yes) {
        SWI_meaning = true;
    }else if (SWI_meaning == dtmf_no){
        SWI_meaning = false;
    }else{
        SWI_disallow=1;
    }
    out.MEANING=SWI_meaning;
        </tag>
```

```
<item>*</item>
```

->

```
<item><token>*</token></item>
```

```
<item>#</item>
```

->

```
<item><token>#</token></item>
```

inin_dtmf_ccexpdate.grxml

This file is a modified version of the original `dtmf_ccexpdate.xml` file.

- `tag-format="swi-semantics/1.0"`

  ->

  `tag-format="semantics/1.0"`

- `<ruleref uri="#A" tag="`
  ```
  if (!A.C) A.C = '??';
  SWI_meaning = A.C + A.Y + A.M;

  var today = new Date();
  var default_ref = '' + today.getFullYear();
  var tmp = today.getMonth() + 1;
  if(tmp &lt; 10) {default_ref += '0';}
  ```

```
      default_ref += tmp;
      var tmp = today.getDate();
      if(tmp &lt; 10) {default_ref += '0';}
      var ref_date = SWI_vars.referencedate ? SWI_vars.referencedate : default_ref;

      var ref_ym = ref_date.substr(2,4);
      var ref_century = parseInt(ref_date.substr (0,2), 10);
      var ref_year = parseInt(ref_date.substr(2,2), 10);
      var ref_month = parseInt(ref_date.substr(4,2), 10);
      var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : 84;
      maxa = parseInt (maxa, 10);
      if (maxa &lt; 0) SWI_disallow = 1;
      var century = SWI_meaning.substr(0,2);
      var ym = SWI_meaning.substr (2,4);
      if (century == '??') {
        if (ref_ym &gt; ym) century = ref_century+1;
      else century = ref_century;
        SWI_meaning = century.toString() + SWI_meaning.substr(2,6);
      }
      if (SWI_meaning &lt; ref_date) SWI_disallow=1;
      var ref_num_month = (parseInt(ref_century, 10)-20)*1200 +
                          parseInt(ref_year, 10)*12 +
                          parseInt(ref_month, 10) - 1;
      var year = parseInt(SWI_meaning.substr(2,2), 10);
      var month = parseInt(SWI_meaning.substr(4,2), 10);

      var num_month = (century-20)*1200 + year*12 +
                      month - 1;
      if (num_month &gt; (ref_num_month + maxa) ) SWI_disallow=1;
      dm_root=SWI_meaning;
"/>
```

 ->

```
<ruleref uri="#A"/>
<tag>
  var tmp = rules.A.C;
  if (!tmp) tmp = '??'; else {tmp;}
  SWI_meaning = tmp + rules.A.Y + rules.A.M;

  var today = new Date();
  var default_ref = '' + today.getFullYear();
  var tmp = today.getMonth() + 1;
  if(tmp &lt; 10) {default_ref += '0';} else {tmp;}
  default_ref += tmp;
  var tmp = today.getDate();
  if(tmp &lt; 10) {default_ref += '0';} else {tmp;}
  var ref_date = SWI_vars.referencedate ? SWI_vars.referencedate : default_ref;

  var ref_ym = ref_date.substr(2,4);
  var ref_century = parseInt(ref_date.substr (0,2), 10);
  var ref_year = parseInt(ref_date.substr(2,2), 10);
  var ref_month = parseInt(ref_date.substr(4,2), 10);
  var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : 84;
  maxa = parseInt (maxa, 10);
  if (maxa &lt; 0) SWI_disallow = 1; else {tmp;}
  var century = SWI_meaning.substr(0,2);
  var ym = SWI_meaning.substr (2,4);
  if (century == '??') {
  if (ref_ym &gt; ym) century = ref_century+1;
  else century = ref_century;
  SWI_meaning = century.toString() + SWI_meaning.substr(2,6);
  } else {tmp;}
  if (SWI_meaning &lt; ref_date) SWI_disallow=1; else {tmp;}
  var ref_num_month = (parseInt(ref_century, 10)-20)*1200 +
                      parseInt(ref_year, 10)*12 +
                      parseInt(ref_month, 10) - 1;
  var year = parseInt(SWI_meaning.substr(2,2), 10);
  var month = parseInt(SWI_meaning.substr(4,2), 10);
```

```
    var num_month = (century-20)*1200 + year*12 + month - 1;
    if (num_month &gt; (ref_num_month + maxa) ) SWI_disallow=1; else {tmp;}
    out.MEANING=SWI_meaning;
  </tag>
```

- <count number="optional"> à

<item repeat="0-1">

- </count> à

</item>

---

inin_dtmf_command.grxml

This file is a modified version of the original `dtmf_command.xml` file.

```
<item tag="dm_command='operator';dm_confirmation_mode='NEVER';dm_confirm_string='operator';">
  0
</item>
<item tag="dm_command='help';dm_confirmation_mode='NEVER';dm_confirm_string='help';">
  *
</item>
<item tag="dm_command='goodbye';dm_confirmation_mode='ALWAYS';dm_confirm_string='goodbye';">
  #
</item>
```

->

```
<item>
  0
  <tag>
    out.dm_command='operator';
    out.dm_confirmation_mode='NEVER';
    out.dm_confirm_string='operator';
  </tag>
</item>
<item>
  <token>*</token>
  <tag>
    out.dm_command='help';
    out.dm_confirmation_mode='NEVER';
    out.dm_confirm_string='help';
  </tag>
</item>
<item>
  <token>#</token>
  <tag>
    out.dm_command='goodbye';
    out.dm_confirmation_mode='ALWAYS';
    out.dm_confirm_string='goodbye';
  </tag>
</item>
```

---

inin_dtmf_creditcard.grxml

This file is a modified version of the original `dtmf_creditcard.xml` file.

- `tag-format="swi-semantics/1.0"`
  ->
  `tag-format="semantics/1.0"`

- `<ruleref uri="#CREDITCARD" tag="`
  `SWI_meaning = CREDITCARD.SWI_literal.replace(/[ ]+/g, '');`
  `var mask = [ 0, 2, 4, 6, 8, 1, 3, 5, 7, 9 ];`
  `var sum = 0;`

```
    var double_the_value = 0; /* start with check digit */
    /* move from end to start */
    var i;
    var add;
    var allowed;
    for (i=SWI_meaning.length - 1;i &gt;= 0; i--) {
      add = parseInt(SWI_meaning[i], 10); /* get value at index i */
      sum += double_the_value ? parseInt(mask[add], 10) : add; /* increase sum */
      double_the_value = !double_the_value;
    }
    if (sum % 10 != 0) SWI_disallow = 1;
    CARDTYPE=CREDITCARD.CARDTYPE;
    if (CARDTYPE == 'private' &amp;&amp;
      (SWI_meaning.match (/^4/) ||
       SWI_meaning.match (/^5/) ||
       SWI_meaning.match (/^34/) ||
       SWI_meaning.match (/^37/) ||
       SWI_meaning.match (/^30/) ||
       SWI_meaning.match (/^38/) ||
       SWI_meaning.match (/^36/) ||
       SWI_meaning.match (/^6011/))) SWI_disallow=1;

      allowed = SWI_vars.typesallowed ?
      SWI_vars.typesallowed.toLowerCase() :
      'visa+mastercard+amex+dinersclub+discover+private';
    if (!allowed ||
        allowed.indexOf (CARDTYPE) == -1) SWI_disallow=1;
    dm_root=SWI_meaning;
    "/>
```

->

```
<ruleref uri="#CREDITCARD"/>
  <tag>
    SWI_meaning = rules.CREDITCARD.V.replace(/[ ]+/g, '');
    var mask = [ 0, 2, 4, 6, 8, 1, 3, 5, 7, 9 ];
    var sum = 0;
    var double_the_value = 0; /* start with check digit */
    /* move from end to start */
    var i;
    var add;
    var allowed;
    for (i=SWI_meaning.length - 1;i &gt;= 0; i -= 1) {
      add = parseInt(SWI_meaning[i], 10); /* get value at index i */
      sum += double_the_value ? parseInt(mask[add], 10) : add; /* increase sum */
      double_the_value = !double_the_value;
    }
    if (sum % 10 != 0) SWI_disallow = 1;
    out.CARDTYPE=rules.CREDITCARD.CARDTYPE;
    if (out.CARDTYPE == 'private' &amp;&amp;
      (SWI_meaning.match (/^4/) ||
       SWI_meaning.match (/^5/) ||
       SWI_meaning.match (/^34/) ||
       SWI_meaning.match (/^37/) ||
       SWI_meaning.match (/^30/) ||
       SWI_meaning.match (/^38/) ||
       SWI_meaning.match (/^36/) ||
       SWI_meaning.match (/^6011/))) SWI_disallow=1;
    allowed = SWI_vars.typesallowed ?
    SWI_vars.typesallowed.toLowerCase() :
      'visa+mastercard+amex+dinersclub+discover+private';
    if (!allowed || allowed.indexOf (out.CARDTYPE) == -1) SWI_disallow=1;
    out.MEANING=SWI_meaning;
  </tag>
```

- **(several times):**
```
<count number="optional">
```
->
```
<item repeat="0-1">
```

- (several times):

```
</count>
->
</item>
```

- (several times):

```
</count>
->
</item>
```

inin_dtmf_currency.grxml

This file is a modified version of the original `dtmf_currency.xml` file.

- `tag-format="swi-semantics/1.0"`

  `->`

  `tag-format="semantics/1.0"`

- ```
  <ruleref uri="#ROOT" tag="
    SWI_meaning = ROOT.SWI_literal.replace(/[ ]+/g, '');
    SWI_meaning = SWI_meaning.replace(/\*/g, '.');
    var both = SWI_meaning.split ('.');
    if (both.length &lt; 2) SWI_meaning += '.00';
    else if (both.length == 2 &amp;&amp; both[1].length == 1) SWI_meaning += '0';
    if (SWI_meaning.substr(0,1) == &quot;.&quot; ||
        SWI_meaning.length == 0) SWI_meaning = '0' + SWI_meaning;
    var numeric = parseFloat (SWI_meaning);
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '999999.99';
    mina = parseFloat(mina);
    maxa = parseFloat(maxa);
    if (mina &gt; maxa) SWI_disallow = 1;
    var grana = parseFloat(SWI_vars.granularityallowed);
    if (numeric &lt; mina || numeric &gt; maxa ||
        (grana &amp;&amp; Math.round(100*numeric) % Math.round(100 * grana) != 0) )
        SWI_disallow=1;
    dm_root=SWI_meaning;
  "/>
  ```

  `->`

  ```
  <ruleref uri="#ROOT"/>
  <tag>
    SWI_meaning = rules.ROOT.replace(/[ ]+/g, '');
    SWI_meaning = SWI_meaning.replace(/\*/g, '.');
    var both = SWI_meaning.split ('.');
    var decimal = both[1];
    if (both.length &lt; 2) SWI_meaning += '.00';
    else if (both.length == 2 &amp;&amp; decimal.length == 1) SWI_meaning += '0';
    if (SWI_meaning.substr(0,1) == &quot;.&quot; ||
        SWI_meaning.length == 0) SWI_meaning = '0' + SWI_meaning;
    var numeric = parseFloat (SWI_meaning);
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '999999.99';
    mina = parseFloat(mina);
    maxa = parseFloat(maxa);
    if (mina &gt; maxa) SWI_disallow = 1;
    var grana = parseFloat(SWI_vars.granularityallowed);
    if (numeric &lt; mina || numeric &gt; maxa ||
        (grana &amp;&amp; Math.round(100*numeric) % Math.round(100 * grana) != 0) )
    SWI_disallow=1;
    out.MEANING=SWI_meaning;
  </tag>
  ```

- (many times):

  `<count number="optional">`

  `->`

  `<item repeat="0-1">`

- (many times):

  `</count>`

  `->`

  `</item>`

- (several times):

  `*`

  `->`

  `<token>*</token>`

inin_dtmf_customcontext.grxml

This file is a modified version of the original `dtmf_customcontext.xml` file.

```
<item tag="dm_root='payment';
         dm_confirmation_mode='NEVER';
         dm_confirm_string='make a payment';">
  1
</item>
<item tag="dm_root ='accounts';
         dm_confirmation_mode='NEVER';
         dm_confirm_string='list accounts';">
  2
</item>
<item tag="dm_root ='balance';
         dm_confirmation_mode='ALWAYS';
         dm_confirm_string='get my balance';">
  3
</item>
```

->

```
<item>
  1
  <tag>
    out.dm_ root='payment';
    out.dm_confirmation_mode='NEVER';
    out.dm_confirm_string='make a payment';
  </tag>
</item>
<item>
  2
  <tag>
    out.dm_ root ='accounts';
    out.dm_confirmation_mode='NEVER';
    out.dm_confirm_string='list accounts';
  </tag>
</item>
<item>
  3
  <tag>
    out.dm_ root ='balance';
    out.dm_confirmation_mode='ALWAYS';
    out.dm_confirm_string='get my balance';
  </tag>
</item>
```

inin_dtmf_date.grxml

This file is a modified version of the original `dtmf_date.xml` file.

We don't believe that this is currently used.

- `tag-format="swi-semantics/1.0"`

  ->

  `tag-format="semantics/1.0"`

- ```
  <ruleref uri="#A" tag="
    if (!A.C) A.C = '??';
    if (!A.Y) A.Y = '??';
    if (!A.M) A.M = '??';
    CENTURY = A.C; TWO_DIGIT_YEAR=A.Y; MONTH=A.M; DAY=A.D;
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '19000101';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '21991231';
    SWI_disallow = 0;
    if (mina &gt; maxa) SWI_disallow = 1;
    var minac = mina.substr (0,2);
    var minay = mina.substr (2,2);
  ```

```
var minam = mina.substr (4,2);
var minad = mina.substr (6,2);

var maxac = maxa.substr (0,2);
var maxay = maxa.substr (2,2);
var maxam = maxa.substr (4,2);
var maxad = maxa.substr (6,2);

SWI_disallow = 0;
SWI_meaning = CENTURY + TWO_DIGIT_YEAR + MONTH + DAY;
if (CENTURY != '??' && TWO_DIGIT_YEAR != '??'
  && MONTH != '??'
  && (SWI_meaning < mina ||
      SWI_meaning > maxa )) {
  SWI_disallow=1;
}
else if (CENTURY == '??' || TWO_DIGIT_YEAR == '??' || MONTH == '??') {
  var miny, maxy, minc, maxc, minm, maxm, inc;
  if (CENTURY == '??') {
    minc = minac; maxc = maxac; inc = 'CENTURY';
  }
  else minc = maxc = CENTURY;

  if (TWO_DIGIT_YEAR == '??') {
    miny = minay; maxy = maxay; inc = 'YEAR';
  }
  else miny = maxy = TWO_DIGIT_YEAR;

  if (MONTH == '??') {
    minm = minam;
    maxm = maxam; inc = 'MONTH';
  }
  else minm = maxm = MONTH;

  var min_full = minc + miny + minm + DAY;
  var max_full = maxc + maxy + maxm + DAY;
  var n_match = 0;
  var full = min_full;
  var fc = full.substr(0 ,2);
  var fy = full.substr( 2 ,2);
  var fm = full.substr (4 ,2);
  var ambig_c, ambig_y, ambig_m;
  var mc, my, mm;
  ambig_c = ambig_y = ambig_m = 0;
  while (full <= max_full) {
    if (full >= mina && full <= maxa) {
      n_match++;
      if (n_match == 1) {
      mc = fc; my = fy; mm = fm;
      }
      else {
        if (!ambig_m && fm != mm) {
        ambig_m = 1; inc = 'YEAR';
        }
        if (!ambig_y && fy != my) {
        ambig_y = 1; inc = 'CENTURY';
        }
        if (!ambig_c && fc != mc ) {
        ambig_c = 1; break;
        }
      }
    }
    if (inc == 'CENTURY') {
      fc = parseInt (fc, 10) + 1;
      fc = fc.toString();
    }
    else if (inc == 'YEAR') {
      fy = parseInt (fy, 10) + 1;
      if (fy == 100) {
        fy = '00';
```

```
        fc = parseInt (fc, 10) + 1;
        fc = fc.toString()
      }
      else {
        fy = fy.toString();
      }
    }
    else if (inc == 'MONTH') {
      fm = parseInt (fm, 10) + 1;
      if (fm == 13) {
        fm = '01';
        fy = parseInt (fy, 10) + 1;
        if (fy == 100) {
          fy = '00';
          fc = parseInt (fc, 10) + 1;
          fc = fc.toString()
        }
        else fy = fy.toString();
      }
      else fm = fm.toString();
    }
    if (fc.length == 1) fc = '0' + fc;
    if (fy.length == 1) fy = '0' + fy;
    if (fm.length == 1) fm = '0' + fm;
    full = fc + fy + fm + DAY;
  }
  if (n_match == 0) SWI_disallow=1;
  else {
    CENTURY = ambig_c ? '??' : mc;
    TWO_DIGIT_YEAR = ambig_y ? '??' : my;
    MONTH = ambig_m ? '??' : mm;
  }
}
YEAR = CENTURY + TWO_DIGIT_YEAR;
SWI_meaning = CENTURY + TWO_DIGIT_YEAR + MONTH + DAY;
dm_root=SWI_meaning;
"/>
```

->

```
<ruleref uri="#A"/>
<tag>
  if (!rules.A.C) out.CENTURY = '??'; else out.CENTURY = rules.A.C;
  if (!rules.A.Y) out.TWO_DIGIT_YEAR = '??'; else out.TWO_DIGIT_YEAR = rules.A.Y;
  if (!rules.A.M) out.MONTH = '??'; else out.MONTH = rules.A.M; out.DAY=rules.A.D;
  var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '19000101';
  var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '21991231';
  SWI_disallow = 0;
  if (mina &gt; maxa) SWI_disallow = 1;
  var minac = mina.substr (0,2);
  var minay = mina.substr (2,2);
  var minam = mina.substr (4,2);
  var minad = mina.substr (6,2);

  var maxac = maxa.substr (0,2);
  var maxay = maxa.substr (2,2);
  var maxam = maxa.substr (4,2);
  var maxad = maxa.substr (6,2);

  SWI_disallow = 0;
  SWI_meaning = out.CENTURY + out.TWO_DIGIT_YEAR + out.MONTH + out.DAY;
  if (out.CENTURY != '??' &amp;&amp; out.TWO_DIGIT_YEAR != '??'
    &amp;&amp; out.MONTH != '??'
    &amp;&amp; (SWI_meaning &lt; mina ||
        SWI_meaning &gt; maxa )) {
    SWI_disallow=1;
  }
  else if (out.CENTURY == '??' || out.TWO_DIGIT_YEAR == '??' || out.MONTH == '??') {
    var miny, maxy, minc, maxc, minm, maxm, inc;
```

```
if (out.CENTURY == '??') {
  minc = minac; maxc = maxac; inc = 'CENTURY';
}
else minc = maxc = out.CENTURY;

if (out.TWO_DIGIT_YEAR == '??') {
  miny = minay; maxy = maxay; inc = 'YEAR';
}
else miny = maxy = out.TWO_DIGIT_YEAR;

if (out.MONTH == '??') {
  minm = minam; maxm = maxam; inc = 'MONTH';
}
else minm = maxm = out.MONTH;

var min_full = minc + miny + minm + out.DAY;
var max_full = maxc + maxy + maxm + out.DAY;

var n_match = 0;
var full = min_full;
var fc = full.substr(0 ,2);
var fy = full.substr( 2 ,2);
var fm = full.substr (4 ,2);
var ambig_c, ambig_y, ambig_m;
var mc, my, mm;
ambig_c = ambig_y = ambig_m = 0;
while (full <= max_full) {
  if (full >= mina && full <= maxa) {
    n_match+=1;
    if (n_match == 1) {
      mc = fc; my = fy; mm = fm;
    }
    else {
      if (!ambig_m && fm != mm) {
        ambig_m = 1; inc = 'YEAR';
      }
      if (!ambig_y && fy != my) {
        ambig_y = 1; inc = 'CENTURY';
      }
      if (!ambig_c && fc != mc ) {
        ambig_c = 1; break;
      }
    }
  }
  if (inc == 'CENTURY') {
    fc = parseInt (fc, 10) + 1;
    fc = fc.toString();
  }
  else if (inc == 'YEAR') {
    fy = parseInt (fy, 10) + 1;
    if (fy == 100) {
      fy = '00';
      fc = parseInt (fc, 10) + 1;
      fc = fc.toString()
    }
    else {
      fy = fy.toString();
    }
  }
  else if (inc == 'MONTH') {
    fm = parseInt (fm, 10) + 1;
    if (fm == 13) {
      fm = '01';
      fy = parseInt (fy, 10) + 1;
      if (fy == 100) {
        fy = '00';
        fc = parseInt (fc, 10) + 1;
        fc = fc.toString()
      }
      else fy = fy.toString();
```

```
        }
      else fm = fm.toString();
      }
    if (fc.length == 1) fc = '0' + fc;
    if (fy.length == 1) fy = '0' + fy;
    if (fm.length == 1) fm = '0' + fm;
    full = fc + fy + fm + out.DAY;
  } // end while
  if (n_match == 0) SWI_disallow=1;
  else {
    out.CENTURY = ambig_c ? '??' : mc;
    out.TWO_DIGIT_YEAR = ambig_y ? '??' : my;
    out.MONTH = ambig_m ? '??' : mm;
  }
}
out.YEAR = out.CENTURY + out.TWO_DIGIT_YEAR;
SWI_meaning = out.CENTURY + out.TWO_DIGIT_YEAR + out.MONTH + out.DAY;
out.MEANING=SWI_meaning;
</tag>
```

- **(several times):**
  ```
  <count number="optional">
  ```
  ->
  ```
  <item repeat="0-1">
  ```
- **(several times):**
  ```
  </count>
  ```
  ->
  ```
  </item>
  ```

inin_dtmf_digits.grxml

This file is a modified version of the original `dtmf_digits.xml` file.

- ```
  tag-format="swi-semantics/1.0"
  ```
  ->
  ```
  tag-format="semantics/1.0"
  ```

- ```
  <ruleref uri="#DIGIT_STRING" tag = "
    SWI_meaning=DIGIT_STRING.SWI_literal.replace(/ /g, '');
    var min_len = SWI_vars.minlength ? parseInt(SWI_vars.minlength, 10) : 1;
    var max_len = SWI_vars.maxlength ? parseInt(SWI_vars.maxlength, 10) : 20;
    var len = SWI_meaning.length;
    if (len &gt; max_len || len &lt; min_len) SWI_disallow=1;
    if (SWI_vars.length) {
      var disallow = 1;
      var allowed_arr = SWI_vars.length.split ('+');
      for (i=0; i &lt; allowed_arr.length; i++) {
        if (len == parseInt(allowed_arr[i], 10)) {
          disallow = 0;
          break;
        }
      }
      SWI_disallow = disallow;
    }
    dm_root=SWI_meaning;
  "/>
  ```

  ->

  ```
  <ruleref uri="#DIGIT_STRING" />
  <tag>
    SWI_meaning=rules.DIGIT_STRING.replace(/ /g, '');
    var min_len = SWI_vars.minlength ? parseInt(SWI_vars.minlength, 10) : 1;
    var max_len = SWI_vars.maxlength ? parseInt(SWI_vars.maxlength, 10) : 20;
    var len = SWI_meaning.length;
    if (len &gt; max_len || len &lt; min_len) SWI_disallow=1;
    if (SWI_vars.length) {
      var disallow = 1;
      var allowed_arr = SWI_vars.length.split ('+');
      for (i=0; i &lt; allowed_arr.length; i+=1) {
        if (len == parseInt(allowed_arr[i], 10)) {
          disallow = 0;
          break;
        }
      } //end for
      SWI_disallow = disallow;
    }
    out.MEANING=SWI_meaning;
  </tag>
  ```

- ```
  <count number="1+">
  ```
  ->
  ```
  <item repeat="1-">
  ```
- ```
  </count>
  ```
  ->
  ```
  </item>
  ```

inin_dtmf_keypad.grxml

This file is a modified version of the original `dtmf_keypad.xml` file.

- ```
  <ruleref uri="#KEYPAD_STRING" tag = "
    SWI_meaning=KEYPAD_STRING.SWI_literal.replace(/ /g, '');
    dm_root=SWI_meaning;"/>
  ```

  ->

  ```
  <ruleref uri="#KEYPAD_STRING"/>
  <tag>
    SWI_meaning=rules.KEYPAD_STRING.replace(/ /g, '');
    out.MEANING=SWI_meaning;
  </tag>
  ```

- ```
  <count number="1+">
  ```
  ->
  ```
  <item repeat="1-">
  ```
- ```
  </count>
  ```
  ->
  ```
  </item>
  ```
- ```
  <item>*</item>
  ```
  ->
  ```
  <item><token>*</token></item>
  ```
- ```
  <item>#</item>
  ```
  ->
  ```
  <item><token>#</token></item>
  ```

inin_dtmf_number.grxml

This file is a modified version of the original `dtmf_number.xml` file.

- ```
  tag-format="swi-semantics/1.0"
  ```
  ->
  ```
  tag-format="semantics/1.0"
  ```

- ```
  <ruleref uri="#ROOT" tag="
    SWI_meaning = ROOT.SWI_literal.replace(/[ ]+/g, '');
    SWI_meaning = SWI_meaning.replace(/\*/g, '.');
    if (SWI_meaning.substr(0,1) == &quot;.&quot; ||
        SWI_meaning.length == 0) SWI_meaning = '0' + SWI_meaning;
    var numeric = parseFloat (SWI_meaning);
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '999999.99';
    var max_dec = SWI_vars.maxdecimal ? SWI_vars.maxdecimal : '2';
    mina = parseFloat(mina);
    maxa = parseFloat(maxa);
    var grana = parseInt(SWI_vars.granularityallowed, 10);
    max_dec = parseInt (max_dec, 10);
    var num_and_dec = SWI_meaning.split('.');
    if (num_and_dec.length == 2 &amp;&amp;
        num_and_dec[1].length > max_dec) SWI_disallow=1;
    if (numeric &lt; mina || numeric &gt; maxa ||
        (grana &amp;&amp; numeric % grana != 0))
    SWI_disallow=1;
    dm_root=SWI_meaning;
  "/>
  ```

  ->

  ```
  <ruleref uri="#ROOT"/>
  <tag>
    SWI_meaning = rules.ROOT.replace(/[ ]+/g, '');
    SWI_meaning = SWI_meaning.replace(/\*/g, '.');
    if (SWI_meaning.substr(0,1) == &quot;.&quot; ||
        SWI_meaning.length == 0) SWI_meaning = '0' + SWI_meaning;
    var numeric = parseFloat (SWI_meaning);
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '999999.99';
    var max_dec = SWI_vars.maxdecimal ? SWI_vars.maxdecimal : '2';
    mina = parseFloat(mina);
    maxa = parseFloat(maxa);
    var grana = parseInt(SWI_vars.granularityallowed, 10);
    max_dec = parseInt (max_dec, 10);
    var num_and_dec = SWI_meaning.split('.');
    if (num_and_dec.length == 2 &amp;&amp;
        num_and_dec[1].length > max_dec) SWI_disallow=1;
    if (numeric &lt; mina || numeric &gt; maxa ||
        (grana &amp;&amp; numeric % grana != 0))
    SWI_disallow=1;
    out.MEANING=SWI_meaning;
  </tag>
  ```

- (many times):
  ```
  <count number="optional">
  ```
  ->
  ```
  <item repeat="0-1">
  ```
- (many times):
  ```
  </count>
  ```
  ->
  ```
  </item>
  ```
- (several times):
  ```
  * -> <token>*</token>
  ```

inin_dtmf_phone.grxml

This file is a modified version of the original `dtmf_phone.xml` file.

- `tag-format="swi-semantics/1.0"`

  ->

  `tag-format="semantics/1.0"`

- ```
  <ruleref uri="#DIGIT_STRING" tag="
    SWI_meaning=DIGIT_STRING.SWI_literal.replace(/ /g, '');
    SWI_meaning=SWI_meaning.replace (/\*/, 'x');
    SWI_meaning=SWI_meaning.replace (/^1/, '');
    var num_and_ext=SWI_meaning.split ('x');
    if (num_and_ext.length == 2) {
      var ext = parseInt (num_and_ext[1], 10);
      var ext_min = SWI_vars.minextension ? SWI_vars.minextension : 1;
      var ext_max = SWI_vars.maxextension ? SWI_vars.maxextension : 9999;
      if (ext &amp;&amp; (ext &lt; ext_min || ext &gt; ext_max))
        SWI_disallow=1;
    }
    dm_root=SWI_meaning;
  "/>
  ```

  ->

  ```
  <ruleref uri="#DIGIT_STRING"/>
  <tag>
    SWI_meaning=rules.DIGIT_STRING.replace(/ /g, '');
    SWI_meaning=SWI_meaning.replace('#',''); // Line addded for testing
    SWI_meaning=SWI_meaning.replace (/\*/, 'x');
    SWI_meaning=SWI_meaning.replace (/^1/, '');
    var num_and_ext=SWI_meaning.split ('x');
    if (num_and_ext.length == 2) {
      var ext = parseInt (num_and_ext[1], 10);
      var ext_min = SWI_vars.minextension ? SWI_vars.minextension : 1;
      var ext_max = SWI_vars.maxextension ? SWI_vars.maxextension : 9999;
      if (ext &amp;&amp; (ext &lt; ext_min || ext &gt; ext_max))
        SWI_disallow=1;
    }
    out.MEANING=SWI_meaning;
  </tag>
  ```

- change:

  ```
  <count number="optional">1</count>
  <count number="optional">
  ```

  to:

  ```
  <item repeat="0-1">1</item>
  <item repeat="0-1">
  ```

- `</count>`

  ->

  `</item>`

- change:

  ```
  <count number="optional">
    <item>*</item>
    <ruleref uri="#DIGIT"/>
    <count number="optional">
      <ruleref uri="#DIGIT"/>
      <count number="optional">
        <ruleref uri="#DIGIT"/>
        <count number="optional">
  ```

```
        <ruleref uri="#DIGIT"/>
      </count>
    </count>
  </count>
</count>
```

  to:

```
<item repeat="0-1">
  <item><token>*</token></item>
  <ruleref uri="#DIGIT"/>
  <item repeat="0-3">
    <ruleref uri="#DIGIT"/>
  </item>
</item>
```

inin_dtmf_sadmCommand.grxml

This file is a modified version of the original `dtmf_sadmCommand.xml` file.

```
<item>
  <tag> dm_command='help';dm_confirmation_mode='NEVER';dm_confirm_string='help';
  </tag>
  *
</item>
<item>
  <tag> dm_command='goodbye';dm_confirmation_mode='NEVER';dm_confirm_string='goodbye';
  </tag>
  0
</item>
```

->

```
<item>
  <tag> out.dm_command='help';out.dm_confirmation_mode='NEVER';out.dm_confirm_string='help';
  </tag>
  <token>*</token>
</item>
<item>
  <tag> out.dm_command='goodbye';out.dm_confirmation_mode='NEVER';out.dm_confirm_string='goodbye';
  </tag>
  0
</item>
```

inin_dtmf_socialsecurity.grxml

This file is a modified version of the original `dtmf_socialsecurity.xml` file.

- ```
  tag-format="swi-semantics/1.0"
  ```
  ->
  ```
  tag-format="semantics/1.0"
  ```

- ```
  <ruleref uri="#SSN" tag="
    SWI_meaning = SSN.SWI_literal.replace(/[ ]+/g, '');
    /* Based on some SSA documentation */
    /* area (first three) range from 001 to 768 with several gaps. */
    /* No guarantee that these gaps will not be filled or */
    /* that numbers won't continue to be added */
    /* group cannot be 00 */
    /* serial cannot be 0000 */
    var area = parseInt (SWI_meaning.substr(0,3), 10);
    var group = SWI_meaning.substr (3,2);
    var serial = SWI_meaning.substr (5,4);
    if (group == '00') SWI_disallow=1;
    if (serial == '0000') SWI_disallow=1;
    if (area == 0 || area &gt;= 800) SWI_disallow = 1;
    dm_root=SWI_meaning;
  "/>
  ```

  ->

  ```
  <ruleref uri="#SSN"/>
  <tag>
    SWI_meaning = rules.SSN.replace(/[ ]+/g, '');
    /* Based on some SSA documentation */
    /* area (first three) range from 001 to 768 with several gaps. */
    /* No guarantee that these gaps will not be filled or */
    /* that numbers won't continue to be added */
    /* group cannot be 00 */
    /* serial cannot be 0000 */
    var area = parseInt (SWI_meaning.substr(0,3), 10);
    var group = SWI_meaning.substr (3,2);
    var serial = SWI_meaning.substr (5,4);
    if (group == '00') SWI_disallow=1;
    if (serial == '0000') SWI_disallow=1;
    if (area == 0 || area &gt;= 800) SWI_disallow = 1;
    out.MEANING=SWI_meaning;
  </tag>
  ```

---

inin_dtmf_time.grxml

This file is a modified version of the original `dtmf_time.xml` file.

- ```
  tag-format="swi-semantics/1.0"
  ```
  ->
  ```
  tag-format="semantics/1.0"
  ```

- ```
  <ruleref uri="#TIME_ABS" tag="
    HOUR=TIME_ABS.H;
    MINUTE=TIME_ABS.M;
    AMPM=(HOUR > 12 || HOUR == 0) ? 'h' : '?';
    QUALIFIER='exact';
    if (HOUR==24) HOUR = '00';

    SWI_disallow = 0;
    var hm = parseInt (HOUR+MINUTE, 10);
    var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0000';
    var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '2359';
    mina = parseInt(mina, 10);
    maxa = parseInt(maxa, 10);
    var hmpm;
  ```

```
    if (AMPM == '?') {
      hmpm = hm + 1200;
      if (hmpm >= 2400) hmpm -= 2400;
      if ( (hm < mina || hm > maxa) &&
           (hmpm < mina || hmpm > maxa)) SWI_disallow=1;

      /* if am/pm ambiguous but only fits one allowed range */
      /* make it unambiguous */
      /* It is not in AM range but is in PM range */
      if ( (hm < mina || hm > maxa) &&
        !(hmpm < mina || hmpm > maxa))
        AMPM = 'p';      /* vice-versa */
      else if ( !(hm < mina || hm > maxa) &&
        (hmpm < mina || hmpm > maxa))
        AMPM = 'a';
    }
    else if (AMPM == 'a' || AMPM == 'h') {
      if (hm < mina || hm > maxa) SWI_disallow=1;
    }
    else if (AMPM == 'p') {
      hmpm = hm + 1200;
      if (hmpm >= 2400) hmpm -= 2400;
      if (hmpm < mina || hmpm > maxa) SWI_disallow=1;
    }
    var minutes = 60 * parseInt (HOUR, 10) + parseInt (MINUTE);
    var grana = SWI_vars.granularityallowed;
    if (grana && (minutes % grana != 0)) SWI_disallow=1;
    SWI_meaning = HOUR+MINUTE+AMPM;   dm_root=SWI_meaning;
"/>
```

->

```
<ruleref uri="#TIME_ABS"/>
<tag>
  out.HOUR=rules.TIME_ABS.H;
  out.MINUTE=rules.TIME_ABS.M;
  out.AMPM=(out.HOUR > 12 || out.HOUR == 0) ? 'h' : '?';  out.QUALIFIER='exact';
  if (out.HOUR==24) out.HOUR = '00';
  SWI_disallow = 0;
  var hm = parseInt (out.HOUR+out.MINUTE, 10);
  var mina = SWI_vars.minallowed ? SWI_vars.minallowed : '0000';
  var maxa = SWI_vars.maxallowed ? SWI_vars.maxallowed : '2359';
  mina = parseInt(mina, 10);
  maxa = parseInt(maxa, 10);
  var hmpm;
  if (out.AMPM == '?') {
    hmpm = hm + 1200;
    if (hmpm >= 2400) hmpm -= 2400;
    if ( (hm < mina || hm > maxa) &&
      (hmpm < mina || hmpm > maxa)) SWI_disallow=1;

    /* if am/pm ambiguous but only fits one allowed range */
    /* make it unambiguous */
    /* It is not in AM range but is in PM range */
    if ( (hm < mina || hm > maxa) &&
       !(hmpm < mina || hmpm > maxa))
       out.AMPM = 'p';
    /* vice-versa */
    else if ( !(hm < mina || hm > maxa) &&
          (hmpm < mina || hmpm > maxa))
          out.AMPM = 'a';
  }
  else if (out.AMPM == 'a' || out.AMPM == 'h') {
    if (hm < mina || hm > maxa) SWI_disallow=1;
  }
  else if (out.AMPM == 'p') {
    hmpm = hm + 1200;
    if (hmpm >= 2400) hmpm -= 2400;
    if (hmpm < mina || hmpm > maxa) SWI_disallow=1;
  }
```

```
    var minutes = 60 * parseInt (out.HOUR, 10) + parseInt (out.MINUTE);
    var grana = SWI_vars.granularityallowed;
    if (grana &amp;&amp; (minutes % grana != 0)) SWI_disallow=1;
    SWI_meaning = out.HOUR+out.MINUTE+out.AMPM;
    out.MEANING=SWI_meaning;
  </tag>
```

inin_dtmf_zipcode.grxml

This file is a modified version of the original dtmf_zipcode.xml file.

- `tag-format="swi-semantics/1.0"`

  ->

  `tag-format="semantics/1.0"`

- ```
  <ruleref uri="#ZIP" tag="
    SWI_meaning = ZIP.SWI_literal.replace(/[ ]+/g, '');
    if (SWI_meaning.substr(0,5) == '00000') SWI_disallow=1;
    dm_root=SWI_meaning;
  "/>
  ```

  ->

  ```
  <ruleref uri="#ZIP"/>
  <tag>
    SWI_meaning = rules.ZIP.replace(/[ ]+/g, '');
    if (SWI_meaning.substr(0,5) == '00000') SWI_disallow=1;
    out.MEANING=SWI_meaning;
  </tag>
  ```

- ```
  <ruleref uri="#DIGIT"/>
  <ruleref uri="#DIGIT" />
  <ruleref uri="#DIGIT" />
  <ruleref uri="#DIGIT" />
  <ruleref uri="#DIGIT" />
  <count number="optional">
    <ruleref uri="#DIGIT" />
    <ruleref uri="#DIGIT" />
    <ruleref uri="#DIGIT" />
    <ruleref uri="#DIGIT" />
  </count>
  ```

  ->

  ```
  <item repeat="5">
    <ruleref uri="#DIGIT"/>
  </item>
  <item repeat="0-1">
    <ruleref uri="#DIGIT"/>
    <ruleref uri="#DIGIT"/>
    <ruleref uri="#DIGIT"/>
    <ruleref uri="#DIGIT"/>
  </item>
  ```

grammar_dmChooseDM.xml

This file is a modified version of the original `grammar_dmChooseDM.xml` file.

```
<grammar xml:lang="en-US" version="1.0" root="___ROOT___">
  <rule id="___ROOT___" scope="public">
    <item>
      <ruleref tag="dm_root=CustomContext.dm_root;
                    dm_confirmation_mode=CustomContext.dm_confirmation_mode;
                    dm_confirm_string=CustomContext.dm_confirm_string;"
              uri="#CustomContext"/>
    </item>
  </rule>
```

```
<rule id="CustomContext">
  <one-of>
    <item tag="dm_root='alphanum';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='alphanum';">
      alphanum
    </item>
    <item tag="dm_root='secureansweralphanum';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='secure answer alphanum';">
      secure answer alphanum
    </item>
    <item tag="dm_root='customcontext';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='customcontext';">
      custom context
    </item>
    <item tag="dm_root='date';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='date';">
      date
    </item>
    <item tag="dm_root='secureanswerdate';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='secure answer date';">
      secure answer date
    </item>
    <item tag="dm_root='digits';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='digits';">
      digits
    </item>
    <item tag="dm_root='secureanswerdigits';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='secure answer digits';">
      secure answer digits
    </item>
    <item tag="dm_root='secureanswerphrase';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='secure answer phrase';">
      secure answer phrase
    </item>
    <item tag="dm_root='phone';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='phone';">
      phone
    </item>
    <item tag="dm_root='secureanswerphone';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='secure answer phone';">
      secure answer phone
    </item>
    <item tag="dm_root='time';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='time';">
      time
    </item>
    <item tag="dm_root='yesno';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='yesno';">
      yesno
    </item>
    <item tag="dm_root='ccexpdate';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='credit card expiry date';">
      credit card expiry date
    </item>
    <item tag="dm_root='creditcard';
               dm_confirmation_mode='IF_NECESSARY';
               dm_confirm_string='credit card';">
```

```
        credit card
      </item>
      <item tag="dm_root='currency';
                 dm_confirmation_mode='IF_NECESSARY';
                 dm_confirm_string='currency';">
        currency
      </item>
      <item tag="dm_root='naturalnumbers';
                 dm_confirmation_mode='IF_NECESSARY';
                 dm_confirm_string='natural numbers';">
        natural numbers
      </item>
      <item tag="dm_root='socialsecurity';
                 dm_confirmation_mode='IF_NECESSARY';
                 dm_confirm_string='social security number';">
        social security <count number='optional'>number</count>
      </item>
      <item tag="dm_root='zipcode';
                 dm_confirmation_mode='IF_NECESSARY';
                 dm_confirm_string='zipcode';">
        zipcode
      </item>
      <item tag="dm_root='record';
                 dm_confirmation_mode='IF_NECESSARY';
                 dm_confirm_string='record';">
        record
      </item>
    </one-of>
  </rule>
</grammar>
```

->

```
<grammar xml:lang="en-US" version="1.0" root="___ROOT___">
  <rule id="___ROOT___" scope="public">
    <item>
      <ruleref uri="#CustomContext"/>
        <tag>
          out.dm_root=rules.CustomContext.dm_root;
          out.dm_confirmation_mode=rules.CustomContext.dm_confirmation_mode;
          out.dm_confirm_string=rules.CustomContext.dm_confirm_string;
        </tag>
    </item>
  </rule>
  <rule id="CustomContext">
    <one-of>
      <item>
        <tag>out.dm_root='alphanum';
             out.dm_confirmation_mode='IF_NECESSARY';
             out.dm_confirm_string='alphanum';
        </tag>
        alphanum
      </item>
      <item>
        <tag>
          out.dm_root='secureansweralphanum';
          out.dm_confirmation_mode='IF_NECESSARY';
          out.dm_confirm_string='secure
        answer alphanum';
        </tag>
        secure answer alphanum
      </item>
      <item>
        <tag>
          out.dm_root='customcontext';
          out.dm_confirmation_mode='IF_NECESSARY';
          out.dm_confirm_string='customcontext';
        </tag>
        custom context
```

```
    </item>
    <item>
      <tag>out.dm_root='date';
           out.dm_confirmation_mode='IF_NECESSARY';
           out.dm_confirm_string='date';
      </tag>
      date
    </item>
    <item>
      <tag>
        out.dm_root='secureanswerdate';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='secure
      answer date';
      </tag>
      secure answer date
    </item>
    <item>
      <tag>out.dm_root='digits';
           out.dm_confirmation_mode='IF_NECESSARY';
           out.dm_confirm_string='digits';
      </tag>
      digits
    </item>
    <item>
      <tag>
        out.dm_root='secureanswerdigits';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='secure answer digits';
      </tag>
      secure answer digits
    </item>
    <item>
      <tag>
        out.dm_root='secureanswerphrase';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='secure answer phrase';
      </tag>
      secure answer phrase
    </item>
    <item>
      <tag>out.dm_root='phone';
           out.dm_confirmation_mode='IF_NECESSARY';
           out.dm_confirm_string='phone';
      </tag>
      phone
    </item>
    <item>
      <tag>
        out.dm_root='secureanswerphone';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='secure answer phone';
      </tag>
      secure answer phone
    </item>
    <item>
      <tag>out.dm_root='time';
           out.dm_confirmation_mode='IF_NECESSARY';
           out.dm_confirm_string='time';
      </tag>
      time
    </item>
    <item>
      <tag>out.dm_root='yesno';
           out.dm_confirmation_mode='IF_NECESSARY';
           out.dm_confirm_string='yesno';
      </tag>
      yesno
    </item>
    <item>
```

```
      <tag>
        out.dm_root='ccexpdate';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='credit card expiry date';
      </tag>
      credit card expiry date
    </item>
    <item>
      <tag>
        out.dm_root='creditcard';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='credit card';
      </tag>
      credit card
    </item>
    <item>
      <tag>
        out.dm_root='currency';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='currency';
      </tag>
      currency
    </item>
    <item>
      <tag>
        out.dm_root='naturalnumbers';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='natural numbers';
      </tag>
      natural numbers
    </item>
    <item>
      <tag>
        out.dm_root='socialsecurity';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='social security number';
      </tag>
      social security
      <item repeat ='0-1'>
        number
      </item>
    </item>
    <item>
      <tag>
        out.dm_root='zipcode';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='zipcode';
      </tag>
      zipcode
    </item>
    <item>
      <tag>
        out.dm_root='record';
        out.dm_confirmation_mode='IF_NECESSARY';
        out.dm_confirm_string='record';
      </tag>
      record
    </item>
  </one-of>
  </rule>
</grammar>
```

Other files (ndm-core 5.2.0)

The following files also had to be tweaked:

- Relative files in grammars
- *.properties (obsolete)
- web.xml
- *.xml

Relative files in grammars

The following grammar files (from the `.\ndm-core\en-US\grammars\framework\ndm\nr9.0` directory) reference relative files from inside the grammar. This can be a problem for two reasons:

- Performance can suffer because we don't cache files references inside of grammar files.
- The referenced files may not be found because we tell the Reco subsystem the location of the cached grammars, not the original location.

alphanum.xml

```
<meta name="swirec_user_dict_name" content="alphanum.userdict"/>
```

->

```
<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>
```

number.xml

```
<meta name="swirec_user_dict_name" content="number.userdict"/>
```

->

```
<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/number.userdict"/>
```

saphrase.jsp

- ```
  <%@ page language="java" contentType="text/xml"%>
  ```
  ->
  ```
  <%@ page language="java" contentType="application/x-swi-grammar-xml"%>
  ```
- ```
  <grammar xml:lang="en-US" version="1.0" root="_secureanswerphrase" tag-format="swi-semantics/1.0">
  ```
  ->
  ```
  <grammar xml:lang="en-US" version="1.0" root="_secureanswerphrase" tag-format="swi-semantics/1.0" xml:base="${pageContext.request.requestURL}">
  ```

> **Note:**
> The first two listed changes are necessary so that the MIME type or the tag-format will force our RecoSubsystem to pass the grammars on to the Nuance engine, rather than try to interpret the grammars itself.

time.xml

```
<meta name="swirec_user_dict_name" content="time.userdict"/>
```

->

```
<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/time.userdict"/>
```

*.properties (obsolete)

The various `*.properties` files in the `ndm-core\properties` subdirectory need to be changed to refer to the new DTMF grammar files names. This usually just means changing the DTMF grammar files' extensions from `.xml` to `.grxml`.

This is for the first version of ndm-core (5.0.x) that we were using, and no longer applies.

web.xml

The following parameters need to be set in the `web.xml` file (from the `.\ndm-core\WEB-INF` directory):

- ```
  <!-- This control the browser on which you are running the NDMs -->
  <context-param>
    <param-name>browser</param-name>
    <param-value>osb</param-value>
  </context-param>
  ```

  ->

  ```
  <context-param>
    <param-name>browser</param-name>
    <param-value>standard</param-value>
  </context-param>
  ```

- ```
  <mime-mapping>
    <extension>xml</extension>
    <mime-type>application/srgs+xml </mime-type>
  </mime-mapping>
  ```

  ->

  ```
  <mime-mapping>
    <extension>xml</extension>
    <mime-type></mime-type>
  </mime-mapping>
  ```

> **Note:**
> - The first change is so that the NDM modules are aware of the type of voice browser being used.
> - The second change is so that grammar files with an `.xml` extension won't have the `application/srgs+xml` MIME type thereby triggering our RecoSubsystem into interpreting those grammars. Since we've changed recommended VoiceXML settings, this change may no longer be necessary.

*.xml

The various `*.xml` files in the `.\ndm-core\en-US\configuration\framework\ndm` subdirectory need to be changed to refer to the new grammar files as described in [Grammars (ndm-core 5.2.0)](#).

All.xml	NaturalNumber.xml	SocialSecurity.xml
CCExpiryDate.xml	Phone.xml	Time.xml
CreditCard.xml	SecureAnswerDate.xml	YesNo.xml
Currency.xml	SecureAnswerDigits.xml	ZipCode.xml
Date.xml	SecureAnswerPhone.xml	

All.xml

- `dtmfkeypadgrammar="dtmf_keypad.xml"` **->** `dtmfkeypadgrammar="inin_dtmf_keypad.grxml"`
- `<dtmfgrammars filename="dtmf_boolean.xml" count="1"/>` **->** `<dtmfgrammars filename="inin_dtmf_boolean.grxml" count="1"/>`

CCExpiryDate.xml

```
<dtmfgrammars filename="dtmf_ccexpdate.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_ccexpdate.grxml" count="1"/>
```

CreditCard.xml

```
<dtmfgrammars filename="dtmf_creditcard.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_creditcard.grxml" count="1"/>
```

Currency.xml

```
<dtmfgrammars filename="dtmf_currency.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_currency.grxml" count="1"/>
```

Date.xml

```
<dtmfgrammars filename="dtmf_digits.xml?minlength=2;maxlength=8;" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_digits.grxml?minlength=2;maxlength=8;" count="1"/>
```

NaturalNumber.xml

```
<dtmfgrammars filename="dtmf_number.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_number.grxml" count="1"/>
```

Phone.xml

```
<dtmfgrammars filename="dtmf_phone.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_phone.grxml" count="1"/>
```

SecureAnswerDate.xml

```
<dtmfgrammars filename="dtmf_digits.xml?minlength=2;maxlength=8;" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_digits.grxml?minlength=2;maxlength=8;" count="1"/>
```

SecureAnswerDigits.xml

```
<dtmfgrammars filename="dtmf_digits.xml" count="1"/>
->
<dtmfgrammars filename="inin_dtmf_digits.grxml" count="1"/>
```

SecureAnswerPhone.xml

```
<dtmfgrammars filename="dtmf_phone.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_phone.grxml" count="1"/>
```

SocialSecurity.xml

```
<dtmfgrammars filename="dtmf_socialsecurity.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_socialsecurity.grxml" count="1"/>
```

Time.xml

```
<dtmfgrammars filename="dtmf_time.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_time.grxml" count="1"/>
```

YesNo.xml

```
<dtmfgrammars filename="dtmf_boolean.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_boolean.grxml" count="1"/>
```

ZipCode.xml

```
<dtmfgrammars filename="dtmf_zipcode.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_zipcode.grxml" count="1"/>
```

ndm-core 5.2.5 changes (*.xml)

Following are additional changes (in addition to the ones noted for NDM-core 5.2.0) made to some of the *.xml* files from the Nuance ndm-core 5.2.5 offering.

For NDM-core 5.2.5, you do not need to do the changes noted in *.xml of the NDM-core 5.2.0 changes. Rather, the various `*.xml` files in the `.\ndm-core\en-US\configuration\framework\ndm` subdirectory need to be changed to refer to the new grammar files names.

All.xml	SecureAnswerDigits.xml
CCExpiryDate.xml	SecureAnswerPhone.xml
CreditCard.xml	SocialSecurity.xml
Date.xml	YesNo.xml
SecureAnswerDate.xml	ZipCode.xml

All.xml

- `dtmfkeypadgrammar="dtmf_keypad.xml"` **->** `dtmfkeypadgrammar="inin_dtmf_keypad.grxml"`
- `<dtmfgrammars filename="dtmf_boolean.xml" count="1"/>` **->** `<dtmfgrammars filename="inin_dtmf_boolean.grxml" count="1"/>`

CCExpiryDate.xml

```
<dtmfgrammars filename="builtin:dtmf/ccexpdate" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_ccexpdate.grxml" count="1"/>
```

CreditCard.xml

```
<dtmfgrammars filename="builtin:dtmf/creditcard" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_creditcard.grxml" count="1"/>
```

Date.xml

```
<dtmfgrammars filename="dtmf_digits.xml?minlength=2;maxlength=8;" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_digits.grxml?minlength=2;maxlength=8;" count="1"/>
```

SecureAnswerDate.xml

```
<dtmfgrammars filename="dtmf_digits.xml?minlength=2;maxlength=8;" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_digits.grxml?minlength=2;maxlength=8;" count="1"/>
```

SecureAnswerDigits.xml

```
<dtmfgrammars filename="dtmf_digits.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_digits.grxml" count="1"/>
```

SecureAnswerPhone.xml

```
<dtmfgrammars filename="dtmf_phone.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_phone.grxml" count="1"/>
```

SocialSecurity.xml

```
<dtmfgrammars filename="builtin:dtmf/socialsecurity" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_socialsecurity.grxml" count="1"/>
```

YesNo.xml

```
<dtmfgrammars filename="dtmf_boolean.xml" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_boolean.grxml" count="1"/>
```

ZipCode.xml

```
<dtmfgrammars filename="builtin:dtmf/zipcode" count="1"/>
```

->

```
<dtmfgrammars filename="inin_dtmf_zipcode.grxml" count="1"/>
```

ndm52-name changes (DTMF grammars)

Following are the changes made to some of the DTMF grammar files from the Nuance ndm52-name offering. The files had to be tweaked for SRGS (SISR) compatibility as noted.

These files originally had an `.xml` filename extension, but were renamed to have a `.grxml` filename extension.

.\ndm52-name\en-US\grammars\nr9.0 directory

inin_dtmf_boolean.grxml

This file is a modified version of the original `dtmf_boolean.xml` file.

The same changes as where made to the `inin_dtmf_boolean.grxml` file in the ndm-core section.

inin_dtmf_command.grxml

This file is a modified version of the original `dtmf_command.xml` file.

The same changes as where made to the `inin_dtmf_command.grxml` file in the ndm-core section.

inin_dtmf_keypad.grxml

This file is a modified version of the original `dtmf_keypad.xml` file.

The same changes as where made to the `inin_dtmf_keypad.grxml` file in the ndm-core section.

name.properties, all.properties

These files (from the `.\ndm52-name\properties` directory) needs to be changed to reflect the file name changes noted above.

- `dtmf_boolean.xml` -> `inin_dtmf_boolean.grxml`
- `dtmf_command.xml` -> `inin_dtmf_command.grxml`
- `dtmf_keypad.xml` -> `inin_dtmf_keypad.grxml`

web.xml

The following parameters need to be set in the `web.xml` file (from the `.\ndm52-name\WEB-INF` directory):

- `com.speechworks.osdm.enableServerSideLogging`
  ->
  `false`

-
  ```
  <mime-mapping>
   <extension>
     xml
   </extension>
   <mime-type>
     application/srgs+xml
   </mime-type>
  </mime-mapping>
  ```

  ->

```
<mime-mapping>
  <extension>
    xml
  </extension>
  <mime-type>
  </mime-type>
</mime-mapping>
```

> **Note:**
> - The first change is because having server side logging was causing a problem, but we don't remember what it was.
> - The second change is so that grammar files with an .xml extension won't have the `application/srgs+xml` MIME type thereby triggering our RecoSubsystem into interpreting those grammars. Since we've changed recommended VoiceXML settings, this change may no longer be necessary.

ndm52-address changes (DTMF grammars)

Following are the changes made to some of the DTMF grammar files from the Nuance ndm52-address offering. The files had to be tweaked for SRGS (SISR) compatibility as noted.

These files originally had a `.xml` filename extension, but were renamed to have a `.grxml` filename extension.

.\ndm52-address\en-US\grammars\nr9.0 directory

inin_dtmf_boolean.grxml

This file is a modified version of the original `dtmf_boolean.xml` file.

The same changes as where made to the `inin_dtmf_boolean.grxml` file in the ndm-core section.

inin_dtmf_command.grxml

This file is a modified version of the original `dtmf_command.xml` file.

The same changes as where made to the `inin_dtmf_command.grxml` file in the ndm-core section.

inin_dtmf_digits.grxml

This file is a modified version of the original `dtmf_digits.xml` file.

The same changes as where made to the `inin_dtmf_digits.grxml` file in the ndm-core section.

inin_dtmf_keypad.grxml

This file is a modified version of the original `dtmf_keypad.xml` file.

The same changes as where made to the `inin_dtmf_keypad.grxml` file in the ndm-core section.

inin_dtmf_zipcode.grxml

This file is a modified version of the original `dtmf_zipcode.xml` file.

The same changes as where made to the `inin_dtmf_zipcode.grxml` file in the ndm-core section.

.\ndm52-address\en-US\grammars\nr9.0\address directory

inin_dtmf_apartment.grxml

This file is a modified version of the original `dtmf_apartment.xml` file.

```
•  <ruleref uri="#DIGIT_STRING"
    tag = "
      SWI_meaning=DIGIT_STRING.SWI_literal.replace(//g,

   '');
      var min_len = SWI_vars.minlength ? parseInt(SWI_vars.minlength,

   10) : 1;
      var max_len = SWI_vars.maxlength ? parseInt(SWI_vars.maxlength,

   10) : 20;
      var len = SWI_meaning.length;
      if (len &gt; max_len || len &lt; min_len)
```

```
      SWI_disallow=1;
        if (SWI_vars.length) {
          var disallow = 1;
          var allowed_arr = SWI_vars.length.split

    ('+');
        for (i=0; i &lt; allowed_arr.length;

    i++) {
            if (len == parseInt(allowed_arr[i],

    10)) {
              disallow =

    0;
              break;
            }
          }
          SWI_disallow = disallow;
        }
        APTNUM=SWI_meaning;
        APTTYPE='APARTMENT';
        APTTYPEABBR='APT';
        FULLAPT = APTTYPE + ' ' + APTNUM;
        FULLAPTABBR = APTTYPEABBR + ' ' + APTNUM;
        SWI_meaning = FULLAPTABBR;
        MEANING=SWI_meaning;
"/>
```

->

```
 <ruleref uri="# DIGIT_STRING "/>
<tag>
  SWI_meaning=DIGIT_STRING.SWI_literal.replace(/ /g, '');
  var min_len = SWI_vars.minlength ? parseInt(SWI_vars.minlength,

  10) : 1;
  var max_len = SWI_vars.maxlength ? parseInt(SWI_vars.maxlength,

  10) : 20;
  var len = SWI_meaning.length;
  if (len &gt; max_len || len &lt; min_len) SWI_disallow=1;
  if (SWI_vars.length) {
    var disallow = 1;
    var allowed_arr = SWI_vars.length.split ('+');
    for (i=0; i &lt; allowed_arr.length; i += 1)

  {
      if (len == parseInt(allowed_arr[i],

  10)) {
        disallow = 0;
        break;
      }
    }
    SWI_disallow = disallow;
  }
  out.APTNUM=SWI_meaning;
  out.APTTYPE='APARTMENT';
  out.APTTYPEABBR='APT';
  out.FULLAPT = out.APTTYPE + ' ' + out.APTNUM;
  out.FULLAPTABBR = out.APTTYPEABBR + ' ' + out.APTNUM;
  SWI_meaning = out.FULLAPTABBR;
  out.MEANING=SWI_meaning;
</tag>
```

- `<count number="1+">`
  ->
  `<item repeat="1-">`
- `</count>`
  ->
  `</item>`

address.properties, all.properties

These files (from the `.\ndm52-address\properties` directory) need to be changed to reflect the file name changes noted above (the `.grxml` changes).

- `dtmf_apartments.xml` **->** `inin_dtmf_apartment.grxml`
- `dtmf_boolean.xml` **->** `inin_dtmf_boolean.grxml`
- `dtmf_command.xml` **->** `inin_dtmf_command.grxml`
- `dtmf_digits.xml` **->** `inin_dtmf_digits.grxml`
- `dtmf_keypad.xml` **->** `inin_dtmf_keypad.grxml`
- `dtmf_zipcode.xml` **->** `inin_dtmf_zipcode.grxml`
- `dtmf_digits.xml` **->** `inin_dtmf_digits.grxml`

web.xml

The following parameters need to be set in the `web.xml` file (from the `.\ndm52-address\WEB-INF` directory):

- `com.speechworks.osdm.enableServerSideLogging`
  ->
  `false`

- ```
  <mime-mapping>
   <extension>
     xml
   </extension>
   <mime-type>
     application/srgs+xml
   </mime-type>
  </mime-mapping>
  ```

  ->

  ```
  <mime-mapping>
   <extension>
     xml
   </extension>
  </mime-mapping>
  ```

> **Note:**
> - The first change is because having server side logging was causing errors on the server side.
> - The second change is so that grammar files with an `.xml` extension won't have the `application/srgs+xml` MIME type thereby triggering our RecoSubsystem into interpreting those grammars.

spelling-dm changes

Following are the changes made to some of the files from the Nuance spelling-dm offering:

- spelling-dm changes DTMF grammars
- Other files (spelling-dm)

spelling-dm changes DTMF grammars

The following DTMF grammar fileshad to be tweaked for SRGS (SISR) compatibility as noted.

.\spelling-dm\en-US\grammars\framework\ndm\nr9.0 directory

inin_dtmf_boolean.grxml

This file is a modified version of the original `dtmf_boolean.xml` file.

The same changes as where made to the `inin_dtmf_boolean.grxml` file in the ndm-core section.

inin_dtmf_command.grxml

This file is a modified version of the original `dtmf_command.xml` file.

The same changes as where made to the `inin_dtmf_command.grxml` file in the ndm-core section.

inin_dtmf_keypad.grxml

This file is a modified version of the original `dtmf_keypad.xml` file.

The same changes as where made to the `inin_dtmf_keypad.grxml` file in the ndm-core section.

Other files (spelling-dm)

The following files also had to be tweaked:

- Relative files in grammars
- Dialog files
- model.es
- All.xml, Spelling.xml, YesNo.xml
- web.xml

Relative files in grammars

The following grammar files (from the `.\spelling-dm\en-US\grammars\framework\ndm\nr9.0` directory) reference relative files from inside the grammar. This can be a problem for two reasons:

- Performance can suffer because we don't cache files references inside of grammar files.
- The referenced files may not be found because we tell the Reco subsystem the location of the cached grammars, not the original location.

inin_alphanum_.xml

This file is a modified version of the original `alphanum_.xml` file.

- `<meta name="swirec_user_dict_name" content="alphanum.userdict"/>`

  `->`

  `<meta name="swirec_user_dict_name" content=" http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>`

- `<ruleref uri=" full-domains.xml#Extensions"/>`

```
->
<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-
domains.xml#Extensions"/>
```

inin_content.xml

This file is a modified version of the original `content.xml` file.

- `<meta name="swirec_user_dict_name" content="alphanum.userdict"/>`

  `->`

  `<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>`

- `<ruleref uri=" first-names.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/first-names.xml#Main"/>`

- `<ruleref uri=" last-names.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/last-names.xml#Main"/>`

- `<ruleref uri=" email-delimiter.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-delimiter.xml#Main"/>`

- `<ruleref uri=" email-specials.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-specials.xml#Main"/>`

- `<ruleref uri=" full-domains.xml#Domain"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#Domain"/>`

- `<ruleref uri=" email-specials.xml#UsernameParts"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-specials.xml#UsernameParts"/>`

- `<ruleref uri=" full-domains.xml#DomainParts"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#DomainParts"/>`

- `<ruleref uri=" full-domains.xml#Extra"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#Extra"/>`

inin_correct.jsp

This file is a modified version of the original `correct.jsp` file.

- `<meta name="swirec_user_dict_name" content="alphanum.userdict"/>`

  `->`

  `<meta name="swirec_user_dict_name" content=" http://bbtestpc:8080/spelling-dm/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>`

- Change all `.SWI_meaning` to `.zSWI_meaning`

> **Note:**
> I'm not 100% sure why the second change was done. We believe that SWI_meaning was frosting our RecoSubsystem.

Dialog files

Files in the `.\ spelling-dm\dialogs` directory need to be changed so as *NOT* to use the following line:

```
<!DOCTYPE vxml PUBLIC "-//Nuance/DTD VoiceXML 2.0//EN"
"http://voicexml.nuance.com/dtd/nuancevoicexml-2-0.dtd">
```

(just commented it out).

> **Note:**
> This change is necessary in IC 4.0, though we don't believe that it is in IC 3.0. This isn't really because of us, but because the newer version of msxml became more strict.

alphanum_controller.jsp

In addition to the change noted above, this file needs the following changes:

- `<grammar type="application/srgs+xml" src="${currentstate.grammar}"/>`
  `->`
  `<grammar type="application/x-swi-grammar-xml" src="${currentstate.grammar}"/>`
- `<grammar type="application/srgs+xml" src="${currentstate.commandGrammar}"/>`
  `->`
  `<grammar src="${currentstate.commandGrammar}"/>`
- `<grammar srcexpr="'${currentstate.dtmfCommandGrammar}'"/>`
  `->`
  `<grammar mode="dtmf" srcexpr="'${currentstate.dtmfCommandGrammar}'"/>`
- Change all `.SWI_meaning` to `.zSWI_meaning`

> **Note:**
> - The first two changes are necessary so that the MIME type will force our RecoSubsystem to pass the grammars on to the Nuance engine, rather than try to interpret the grammars itself.
> - The third change is necessary so that the RecoSubsystem knows that the grammar is a `dtmf` grammar.
> - I'm not 100% sure why the fourth change was done. We believe that SWI_meaning was interfering with the RecoSubsystem.

model.es

This file (from the `.\spelling-dm\en-US\scripts\framework\ndm` directory) needs to be tweaked for what seem to be *programming* errors in the script.

```
} else if (obj instanceof Object || typeof(obj) == 'object') {
  if (obj._type == 'ApplicationState') {
    xml += recognitions2XML(obj);
  }
```

->

```
} else if (obj instanceof Object || typeof(obj) == 'object') {
  if ( ! obj._type ) {
    obj._type = 'zztop';
  }
  if (obj._type == 'ApplicationState') {
    xml += recognitions2XML(obj);
  }
```

> **Note:**
> This change is necessary to make our VoiceXML interpreter's strict SpiderMonkey happy.

All.xml, Spelling.xml, YesNo.xml

These files (from the `.\ spelling-dm\en-US\configuration\framework\ndm` directory) need to be changed to reflect the grammar file name changes noted above.

- `alphanum.jsp` -> `inin_alphanum.jsp`
- `alphanum.xml` -> `inin_alphanum.xml`
- `correct.jsp` -> `inin_correct.jsp`
- `dtmf_boolean.xml` -> `inin_dtmf_ boolean.grxml`
- `dtmf_command.xml` -> `inin_dtmf_command.grxml`
- `dtmf_keypad.xml` -> `inin_dtmf_keypad.grxml`

web.xml

The following parameters need to be set in the `web.xml` file (from the `.\spelling-dm\WEB-INF` directory):

```
<mime-mapping>
  <extension>xml</extension>
  <mime-type>application/srgs+xml</mime-type>
</mime-mapping>
```

->

```
<mime-mapping>
  <extension>xml</extension>
  <mime-type></mime-type>
</mime-mapping>
```

> **Note:**
> This change is so that grammar files with an `.xml` extension won't have the `application/srgs+xml` MIME type thereby triggering our RecoSubsystem into interpreting those grammars. Since we've changed recommended VoiceXML settings, this change may no longer be necessary.

email-dm changes

Following are the changes made to some of the files from the Nuance email-dm offering:

- email-dm changes (DTMF grammars)
- Other files (DTMF grammars)

email-dm changes (DTMF grammars)

The following DTMF grammar files had to be tweaked for SRGS (SISR) compatibility as noted.

.\email-dm\en-US\grammars\framework\ndm\nr9.0 directory

inin_dtmf_boolean.grxml

This file is a modified version of the original `dtmf_boolean.xml` file.

The same changes as where made to the `inin_dtmf_boolean.grxml` file in the ndm-core section.

inin_dtmf_command.grxml

This file is a modified version of the original `dtmf_command.xml` file.

The same changes as where made to the `inin_dtmf_command.grxml` file in the ndm-core section.

inin_dtmf_keypad.grxml

This file is a modified version of the original `dtmf_keypad.xml` file.

The same changes as where made to the `inin_dtmf_keypad.grxml` file in the ndm-core section.

Other files (DTMF grammars)

The following files also had to be tweaked:

- Relative files in grammars
- Dialog files
- EmailDMSubdialog.jsp
- model.es
- All.xml, Email.xml, YesNo.xml

Relative files in grammars

The following grammar files (from the `.\email-dm\en-US\grammars\framework\ndm\nr9.0` directory) reference relative files from inside the grammar. This can be a problem for two reasons:

- Performance can suffer because we don't cache files references inside of grammar files.
- The referenced files may not be found because we tell the Reco subsystem the location of the cached grammars, not the original location.

inin_content.xml

This file is a modified version of the original `content.xml` file.

- `<meta name="swirec_user_dict_name" content="alphanum.userdict"/>`

  `->`

  `<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>`
- `<ruleref uri=" first-names.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/first-names.xml#Main"/>`
- `<ruleref uri=" last-names.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/last-names.xml#Main"/>`
- `<ruleref uri=" email-delimiter.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-delimiter.xml#Main"/>`
- `<ruleref uri=" email-specials.xml#Main"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-specials.xml#Main"/>`
- `<ruleref uri=" full-domains.xml#Domain"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#Domain"/>`
- `<ruleref uri=" email-specials.xml#UsernameParts"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/email-specials.xml#UsernameParts"/>`
- `<ruleref uri=" full-domains.xml#DomainParts"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#DomainParts"/>`
- `<ruleref uri=" full-domains.xml#Extra"/>`

  `->`

  `<ruleref uri="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/full-domains.xml#Extra"/>`

inin_correct.jsp

This file is a modified version of the original correct.jsp file.

`<meta name="swirec_user_dict_name" content="alphanum.userdict"/>`

`->`

`<meta name="swirec_user_dict_name" content="http://bbtestpc:8080/email-dm/en-US/grammars/framework/ndm/nr9.0/alphanum.userdict"/>`

Dialog files

Files in the `.\email-dm\dialogs` directory need to be changed so as *NOT* to use the following line:

```
<!DOCTYPE vxml PUBLIC "-//Nuance/DTD VoiceXML 2.0//EN"
"http://voicexml.nuance.com/dtd/nuancevoicexml-2-0.dtd">
```

(just commented it out).

> **Note:**
> This change is necessary in CIC 4.0, though we don't believe that it is in CIC 3.0. This isn't really because of us, but because the newer version of msxml became more strict.

alphanum_controller.jsp

In addition to the change noted above, this file needs the following changes:

- `<grammar type="application/srgs+xml" src="${currentstate.grammar}"/>`
  `->`
  `<grammar type="application/x-swi-grammar-xml" src="${currentstate.grammar}"/>`
- `<grammar type="application/srgs+xml" src="${currentstate.commandGrammar}"/>`
  `->`
  `<grammar src="${currentstate.commandGrammar}"/>`
- `<grammar srcexpr="'${currentstate.dtmfCommandGrammar}'"/>`
  `->`
  `<grammar mode="dtmf" srcexpr="'${currentstate.dtmfCommandGrammar}'"/>`
- Change all `.SWI_meaning` to `.dm_root`

> **Note:**
> - The first two changes are necessary so that the MIME type will force our RecoSubsystem to pass the grammars on to the Nuance engine, rather than try to interpret the grammars itself.
> - The third change is necessary so that our RecoSubsystem knows that the grammar is a `dtmf` grammar.
> - I'm not 100% sure why the fourth change was done. We believe that SWI_meaning was frosting our RecoSubsystem.

EmailDMSubdialog.jsp

This file (from the `.\email-dm\dialogs\framework\ndm` directory) needs to be changed to use new grammar file names.

```
<!--var name="collection_grammar1"
expr="'<%=appName%>/grammars/content.xml;application/srgs+xml'"/-->

->

<!--var name="collection_grammar1" expr="'<%=appName%>/grammars/inin_content.xml'"/-->
```

model.es

This file (from the `.\email-dm\en-US\scripts\framework\ndm` directory) needs to be tweaked for what seem to be *programming* errors in the script.

-
```
var object2XML = function(obj, name) {
var name = isNaN(name) ? name : '_' + name;
if(! /^[a-z_][\w$]*$/i.test(name)){
  return '';
}
var xml = '<' + name;
```

  ->

```
var object2XML = function(obj, name) {
var name1 = isNaN(name) ? name : '_' + name;
if(! /^[a-z_][\w$]*$/i.test(name1)){
  return '';
}
var xml = '<' + name1;
```

-
```
} else if (obj instanceof Object || typeof(obj) == 'object') {
  if (obj._type == 'ApplicationState') {
    xml += recognitions2XML(obj);
  }
```

  ->

```
} else if (obj instanceof Object || typeof(obj) == 'object') {
  if (obj._type) {
  } else {
    obj._type = 'zztop';
  }
  if (obj._type == 'ApplicationState') {
    xml += recognitions2XML(obj);
  }
```

-
```
xml += '</' + name + '>';
```
  ->
```
xml += '</' + name1 + '>';
```

> **Note:**
> These changes are necessary to make our VoiceXML interpreter's strict SpiderMonkey happy.

All.xml, Email.xml, YesNo.xml

These files (from the `.\ email-dm\en-US\configuration\framework\ndm` directory) need to be changed to reflect the grammar file name changes noted above.

- `alphanum_.xml` -> `inin_alphanum_.xml`
- `content.xml` -> `inin_content.xml`
- `correct.jsp` -> `inin_correct.jsp`
- `dtmf_boolean.xml` -> `inin_dtmf_boolean.grxml`
- `dtmf_command.xml` -> `inin_dtmf_command.grxml`
- `dtmf_keypad.xml` -> `inin_dtmf_keypad.grxml`

Recommended VoiceXML parameter settings

The following VoiceXML configuration parameter settings are recommended:

- `defaultGrammarMimeType` - **leave blank**
- `ignoreGrammarFileMimeType` - **false**
- `recoValueSlotName` - MEANING

NDMs tested from core

alphanum

The basics seem to work, but voice recognition isn't the best.

ccexpdate

Basics seem to work.

Possible problem: Voice grammars don't seem to pay attention to maxallowed (DTMF grammars do).

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_ccexpdate.grxml?
SWI_vars.maxallowed=+84;SWI_vars.maxexpected=+60

creditcard

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_creditcard.grxml?
SWI_vars.typesallowed=visa+mastercard+amex+dinersclub+discover+private

currency

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_currency.grxml?SWI_vars.granularityallowed=0.01;
SWI_vars.granularityexpected=0.01;SWI_vars.minallowed=0.0;SWI_vars.maxallowed=999999.99;
SWI_vars.minexpected=999999.99;SWI_vars.maxexpected=999999.99;SWI_vars.disambiguationmode=ASSUME_SMALLER

date

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:
- voice:
  http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/date.xml?language=en-
  US;SWI_vars.minallowed=19000101;SWI_vars.minexpected=19900101;SWI_vars.maxallowed=21991231;SWI_vars.maxexpected=20191231
- dtmf:
  http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_digits.grxml?minlength=2;maxlength=8

digits

Basics seem to work.

Possible problem: The `length` parameter doesn't seem to work with more than one value (i.e., `5+6`) in voice or DTMF mode.

The URL for the DTMF grammar used when this is called is of the form:

`builtin:dtmf/digits?minlength=1;maxlength=20`

naturalnumbers

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_number.grxml?
SWI_vars.granularityallowed=1;SWI_vars.granularityexpected=1;

SWI_vars.minallowed=0;SWI_vars.maxallowed=999999.99;SWI_vars.maxdecimal=2;SWI_vars.minexpected=0;SWI_vars.maxexpected=999999.99

phone

For the most part seems to work.

Possible problem: Not all the configuration parameters seem to work. The parameter `handlesextension`, for example, is ignored. This does not appear to be related to our stuff, as the parameters are not even being attached to the grammar URI.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_phone.grxml?
SWI_vars.minextension=1;SWI_vars.maxextension=9999

record

Basics seem to work.

secureansweralphanum

Basics seem to work, voice recognition really sucks though.

secureanswerdate

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:
- voice:
  http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/date.xml?language=en-US;SWI_vars.minallowed=19000101;SWI_vars.minexpected=19900101;SWI_vars.maxallowed=21991231;SWI_vars.maxexpected=20191231
- date:
  http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_digits.grxml?minlength=2;maxlength=8

secureanswerdigits

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_digits.grxml?SWI_vars.minlength=1;SWI_vars.maxlength=20

secureanswerphone

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_phone.grxml?
SWI_vars.minextension=1;SWI_vars.maxextension=9999

secureanswerphrase

Basics seem to work.

We never get a `SUCCESS` when we say "Don't know", nor do we ever receive a *Decoy* response back.

socialsecurity

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_socialsecurity.grxml

time

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_time.grxml?SWI_vars.minallowed=0;SWI_vars.minexpected=0; SWI_vars.maxallowed=2359;SWI_vars.maxexpected=2359;SWI_vars.granularityallowed=1;SWI_vars.granularityexpected=1

yesno

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_boolean.grxml?SWI_vars.dtmfyes=1;SWI_vars.dtmfno=3

zipcode

Basics seem to work.

The URL for the DTMF grammar used when this is called is of the form:

http://bbtestpc:8080/ndm-core/en-US/grammars/framework/ndm/nr9.0/dtmf_zipcode.grxml

Change Log

The following table lists the changes to the *CIC VoiceXML Integration with Nuance Dialog Modules Version 5.2 Developer's Application Note* since its initial release.

Date	Changes
21-January-2011	Version 1.0 Document created
22-December-2011	Version 1.0.1 Added spelling-dm and e-mail-dm sections
18-April-2012	Version 1.0.2 Updated title
01-October-2012	Version 1.0.3 Referred to spelling-dm directory in email-dm section
October 8, 2012	Version 1.0.4 Added inin_content.xml to section 8.2.5, All.xml, Email.xml, YesNo.xml
12-December-2012	Version 1.0.5 Notes added to explain changes
02-January-2013	Version 1.0.6 Additional updates
03-February-2013	Version 4.0 Updated for 4.0 SU3 release
05-February-2013	Added SCR information to Section 2 VoiceXML changes
05-February-2013	Added section for NDM 5.2.5
01-August-2014	Updated documentation to reflect changes required in the transition from version 4.0 SU# to CIC 2015 R1, such as updates to product version numbers, system requirements, installation procedures, references to Interactive Intelligence Product Information site URLs, and copyright and trademark information.
11-June-2015	- Updated cover page to reflect new color scheme and logo. - Updated copyright and trademark information.
09-October-2015	Updated documentation to reflect 2016 R1 Release
04-February-2016	Updated documentation to reflect 2016 R2 Release
08-May-2018	Rebranded from Interactive Intelligence to Genesys
13-June-2019	Reorganized the content only, which included combining some topics and deleting others that just had an introductory sentence such as, "In this section...".