Changes and Interface Specifications

Smarter Balanced Assessment Consortium Test Delivery System

Components: Test Authoring, Test Item Bank, Test Spec Bank

Smarter Balanced Task Orders 02 and 03

Revision History

Revision Description		Date
0.1 - First Draft	Rami Levy	Nov. 22, 2015
0.2 – Refactor document for broadened purpose.	Rami Levy	Jan. 5, 2016
1.0 – First release	Rami Levy	Feb. 5, 2016

Contents

Figures

Tables

1 Introduction

1.1 Purpose

The purpose of this document is to describe the requirements, design, and code changes associated with Smarter Balanced Task Orders 02 and 03. Since Test Authoring includes interfaces to several other components, this document will also provide an outline of the structure, as well as specifications on all relevant interfaces.

1.2 References

Ref	Document	Web Location or Source Repository (http://bitbucket.org/sbacoss/…)	Version
1.	SmarterApp Item Metadata Specification	http://www.smarterapp.org/documents/Item_Metadata_Specification.pdf	11/25/2014 and Release candidate 2/5/2016
2.	Test Item Bank (TIB) API (NEW)	testitembank_release / external_release_docs / TIB-API.pdf	2.0 2/4/2016
3.	Test Authoring (TA) API	testauthoring_release / external_release_docs / Test Auth API.pdf	9/30/2014
4.	Test Spec Bank (TSB) API	testspecbank_release / external_release_docs /API.pdf	9/30/2014
5.	Test Spec Bank (TSB) Packaging Process Diagram	testspecbank_release / designPics / Export Package Process.png	9/30/2014
6.	Test Spec Bank (TSB) Sequence Diagram	testspecbank_release / designPics / Test-Spec-Bank-sequence-diagram.png	9/30/2014

Table 1: Reference Documents

2 Task Order 2 Changes

2.1 Test Authoring

2.1.1 High Level Requirements

1. Provide support for stimuli as first-class entities
2. Support additional metadata in the item search, as defined by the Smarter specification
3. Provide ability to reorder items in item groups for adaptive item pools
4. Ensure all generated packages conform to SmarterApp package specs
5. Integrate and fix support for affinity groups
6. Ensure that test admin package can be imported into TDS
7. Import items and their scoring statistics into Test Authoring
8. Test Authoring to publish test packages containing proper item scoring and test scoring statistics (see 2.4)
9. Allow selection of item bank number
10. Allow selection of client name
11. Add support for tenant ID in item search
12. Update associated specifications documentation
13. Identify updates required for user documentation

2.1.2 Changes Made

1. Allow four-character subject abbreviations and fixed error message about max length of subject attribute.
2. Fixed structure of elements:
   i. Added *filename* attribute to itempool.passage. Was:

```
<passage>
```

After fix:

```
<itempool>
    <passage filename="stim-200-1374.xml">
```

   ii. Fixed testitem.identifier.uniqueid. Was:

```
uniqueid="item-187-1609-item-187-1609-4"/>
```

After fix:

```
uniqueid="187-1023"/>
```

   iii. Added references for each level of the path from blueprint standard element to root strand bpelement for itempool.testitem elements. Was:

```
<testitem itemtype="MC">
    <bpref>S-1-6</bpref>
    <bpref>SBAC-MA-v1:2|G|A</bpref>
```

After fix:

```
<testitem filename="item-200-102913.xml" itemtype="EBSR">
   <bpref>C1-Adaptive</bpref>
   <bpref>SBAC-ELA-v1:1-IT</bpref>
   <bpref>SBAC-ELA-v1:1-IT|12-3</bpref>
   <bpref>SBAC-ELA-v1:1-IT|12-3|3.RI.6</bpref>
```

3. Added itembankId to testitem.passageref element. Was:

```
<passageref>3709</passageref>
```

After fix:

```
<passageref>200-1374</passageref>
```

4. Deleted redundant (repeated) passage elements from itempool element. Was:

```
<passage>
   <identifier name="3743" label="Astronauts" version="1.0"
</passage>
<passage>
   <identifier name="3743" label="Astronauts" version="1.0"
</passage>
<passage>
   <identifier name="3743" label="Astronauts" version="1.0"
</passage>
```

After fix:

```
<passage>
      <identifier name=”3743” label=”Astronauts” version=”1.0”>
</passage>
```

5. Added reference for segment for testspecification.administration.itempool.testitem element. Was:

```
<testitem itemtype="MI">
   <bpref>SBAC-ELA-v3:ELA-Undesignated</bpref>
```

After fix:

```
<testitem filename="item-187-1023.xml" itemty
   <bpref>C1-Adaptive</bpref>
   <bpref>SBAC-ELA-v1:2-W</bpref>
```

6. Fixed null uniqueId for testspecification.administration.testblueprint.identifier with elementtype="test". Was:

```
<bpelement elementtype="test" ftitemcount="0" maxftitems="40" maxopi
   <identifier name="SBAC-ELA-3-ALEXADMIN-F-9" version="1.0"/>
```

After fix:

```
<bpelement elementtype="test" ftitemcount="0" maxftitems="140" maxopitems="143" minftitems="0" minopitems="0" opitemcount="0"
  <identifier name="SBAC-ELA-3-ALEX-0623-S-31" label="FirstChangesInTA-CI" version="1.0" uniqueid="5589a8a0e4b08101dc7267f3",
```

7. Made testspecification.administration.segmentid attribute equal to a GUID instead of being equal to testspecification.administration.bpelement.identifier.uniqueid. Was:

```
<bpelement elementtype="segment" ftitemcount="0" maxftitems='
   <identifier name="S1" version="1.0" uniqueid="S1-1.0"/>
<adminsegment segmentid="S1"
```

After fix:

```
<adminsegment segmentid="5589a930e4b08101dc7267f7"
```

8. Fixed attribute *uniqueid* for bpelement.identifier with type = 'test'. This attribute must be the same as testspecification.identifier.uniqueid attribute. Was:

```
uniqueid="SBAC-ELA-3-ALEX-0623-S-31-FirstChangesInTA-CI-1.0"/>
uniqueid="5589a8a0e4b08101dc7267f3"/>
```

After fix:

```
<identifier name="SBAC-ELA-3-ALEX-1002-FIXED-S-100" version="1.0" uniqueid="SBAC-ELA-3-ALEX-1002-FIXED-S-100-1.0"/>
<bpelement elementtype="test" ftitemcount="0" maxftitems="25" maxopitems="25" minftitems="0" minopitems="0" opitemcount="0">
   <identifier name="SBAC-ELA-3-ALEX-1002-FIXED-S-100" version="1.0" uniqueid="SBAC-ELA-3-ALEX-1002-FIXED-S-100-1.0"/>
```

9. Replaced restricted XML characters by entity references to identifier element in test_specification.xml.
10. Changed parentid for testspecification.administration.testblueprint.bpelement with elementtype="segment".

Was:

```
uniqueid="SBAC-ELA-3-ALEXADMIN-F-9-1.0"/>
<bpelement elementtype="segment" ftite
parentid="SBAC-ELA-3-ALEXADMIN-F-9">
```

After fix:

```
uniqueid="SBAC-ELA-3-ALEX-1002-FIXED-S-100-1.0"/>
<bpelement elementtype="segment" ftitemcount="0"
parentid="SBAC-ELA-3-ALEX-1002-FIXED-S-100-1.0">
```

11. Fixed value of the *version* attribute of the testblueprint.bpelement where type = “test”. Was:

```
<identifier name="SBAC-ELA-3-FALL-2016-S-1" version="2.0"
<bpelement elementtype="test" ftitemcount="0" maxftitems="40"
   <identifier name="SBAC-ELA-3-FALL-2016-S-1" version="1.0"/>
```

After fix:

```
<identifier name="SBAC-MATH-7-ALEX-1119-ADAPT-S-24" version="2.0"
<bpelement elementtype="test" ftitemcount="2" maxftitems="6" maxopite
   <identifier name="SBAC-MATH-7-ALEX-1119-ADAPT-S-24" version="2.0"
```

12. Fixed *uniqueid* for testblueprint.bpelement where type = “segment”.
13. Fixed testitem.bpref element on segmentid.
14. Fixed *uniqueid* for testblueprint.bpelement.identifier with type = “strand”.

Was:

```
<bpelement elementtype="strand" ftitemcount="0" >
    <identifier name="SBAC-MA-v1:1" version="1.0"
    uniqueid="SBAC-MA-v1:1-1.0"/>
```

After fix:

```
<bpelement elementtype="strand" ftitemco
    <identifier name="1" version="1.0"
    uniqueid="SBAC-1"/>
</bpelement>
```

15. Fixed uniqueid of the registration.registrationsegment.identifier for Registration test_specification.xml. Was:

```
<registration>
    <registrationsegment minitems="0" maxitems="15" itemselection="ADAPT
        <identifier name="S-1-6" version="1.0" uniqueid="S-1-6-1.0"/>
```

16. Fixed stimuli gathering and pool property values where property = "Language" --- deleted '[',']' brackets for "Language" values. Was:

```
<poolproperty property="Language" value="[eng, spa]" label="[eng, spa]" itemcount="40"/>
<poolproperty property="Language" value="eng" label="eng" itemcount="7"/>
```

After fix:

```
<poolproperty property="Language" value="ENU" label="ENU" itemcount="66"/>
<poolproperty property="Language" value="ENU-Braille" label="ENU-Braille" :
<poolproperty property="Language" value="ESN" label="ESN" itemcount="42"/>
```

17. Fixed passage version, and fix bug checking for associated stimulus.
18. Multiple changes per testing of Test package creation related to segment Id. Introduced notion of associated items (i.e. stimuli, tutorials and word lists) and fixed validations for number of items selected for Form partition by not counting associated items.
19. Fixed Item Selection Algorithm name when algorithm is Adaptive Algorithm.
20. Multiple bug fixes and changes to build ids compatible with loading stored procedures:
    1) Added 'itembank' attribute. Was:

```
<passage>
    <identifier name="3738" version="1.0" uniqueid="3738-1.0"/>
</passage>
```

    After fix:

```
<passage filename="stim-200-1042.xml">
    <identifier name="stim-200-1042" label="stim-200-1042" version="5" uniqueid="200-1042"/>
</passage>
```

    2) Changed signature of some methods – instead "String assessmentId" parameter added "Assessment assessment" parameter to have more information access for compatibility of with loading stored procedures: Item Selection parameters, test form parameters and so on.
    3) Fixed Form and FormPartition attributes. Was:

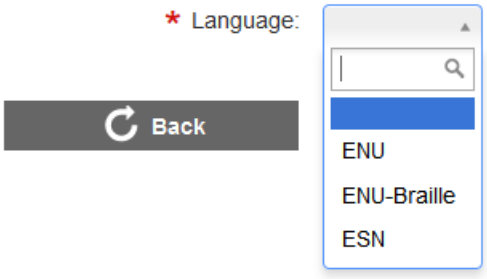
```
<testform>
  <identifier name="F1" version="1.0" uniqueid="F1-1.0"/>
  <property name="language" value="ENG" label="English"/>
  <formpartition>
    <identifier name="F-P-1" version="1.0" uniqueid="F-P-1-1.0"/>
    <itemgroup formposition="1" maxitems="1" maxresponses="1">
      <identifier name="stim-187-3739" label="stim-187-3739" version="5" uniqueid="stim-187-3739-stim-187-3739-5"/>
      <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="true" itemid="3739"
    </itemgroup>
```

After fix:

```
<testform>
  <identifier name="F1" version="1.0" uniqueid="1239451703793824"/>
  <property name="language" value="ENU" label="English"/>
  <formpartition>
    <identifier name="F1-P1" label="F1-P1" version="1.0" uniqueid="200-1239452039338145"/>
    <itemgroup formposition="1" maxitems="1" maxresponses="1">
      <identifier name="200-1239452039338145:I-200-58847" label="" version="2" uniqueid="200-1239452039338145:I-200-58847"/
      <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="false" itemid="200-58847"
    </itemgroup>
```

4) Fixed “Language” values on UI.

After fix:



5) Fixed segmentpool.itemgroup.identifier.uniqueid attribute. Was:

```
<itemgroup formposition="67">
  <identifier name="item-187-2285" label="ite
  uniqueid="item-187-2285-item-187-2285-4"/>
```

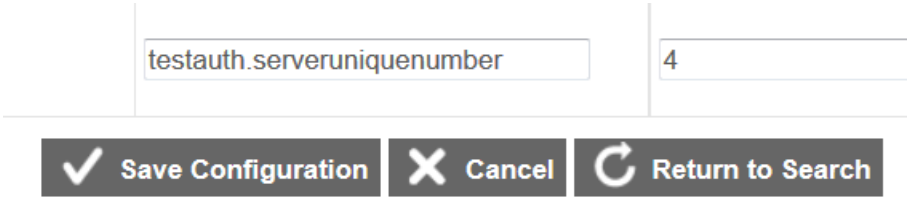
After fix:

```
<itemgroup formposition="1" maxitems="0" maxresponses
  <identifier name="200-346848116144798:G-200-409-1"
  uniqueid="200-346848116144798:G-200-409-1"/>
```

or

```
<itemgroup formposition="4" maxitems="1" maxresponses
  <identifier name="200-346848116144798:I-200-32044"
  uniqueid="200-346848116144798:I-200-32044"/>
```

21. Added support for serverUniqueNumber and building tds-compatible unique ids. Added a parameter to Program Management TestAuth component.



Each TA server in the server farm must have unique serverUniqueNumber value.

22. Fixed “Language” value for testform.poolproperty. Was:

```
<testform>
    <identifier name="F1" version="1.0" uniqueid="F1-1.0"/>
    <property name="language" value="ENG" label="English"/>
```

After fix:

```
<testform>
    <identifier name="F1" version="1.0" uniqueid="1239451703793824"/>
    <property name="language" value="ENU" label="English"/>
```

23. Fixed adminsegmentpool->itemgroup->identifier uniqueid attribute: it is now in "187-–284605500999198:G-187-1023" format suitable for loader stored procedures parsing.
24. Adding the latest version of test specification DTD file.
25. Fixed segment uniqueid in the case when test has only one segment. This test must have bpelement.identifier.uniqueid for elementtype = “segment” the same as bpelement.identifier.uniqueid for elementtype = “test”.

    After fix:

```
<testblueprint>
    <bpelement elementtype="test" ftitemcount="0" maxftitems=
        <identifier name="SBAC-ELA-3-ALEX-1104-FIXED-1S-S-145"
        uniqueid="SBAC-ELA-3-ALEX-1104-FIXED-1S-S-145-1.0"/>
    </bpelement>
    <bpelement elementtype="segment" ftitemcount="0" maxftite
        <identifier name="Segm-1" label="Segm-1" version="1.0"
        uniqueid="SBAC-ELA-3-ALEX-1104-FIXED-1S-S-145-1.0"/>
```

26. Changes to itemsByidentifierSearch API sent from TA to TIB- added itemBank as one of the search criteria.
27. Do not place itempool.passages with version = 0 into test specification.
28. Added support for the new parameter ‘itemBank’ to TIB REST API endpoints 'tibitem' and 'importfromtib'. New parameter ‘itemBank’ added to environment parameters and to records of the assessment collection of the TA MongoDB.

    After fix:

```
"publicationId": "564d23ace4b0ce859c414eb8",
"itemBank": "187",
"client": "SBAC",
```

29. Allowed Adaptive Item Selection Algorithm parameters have the same name for two different purposes (purpose = 'SCALAR' and purpose = 'Blueprint').
30. If stimulus received from TIB does not have ‘intendedGrade’ set it to ‘NA’ to avoid validation problems.
31. Add support for client:
    1) Added testauth.client parameter to Program management TestAuth environment;
    2) Using of the testauth.client value as testspecification.publisher in test_specification.xml in place of ‘tenantId’.
32. Fixed publishing pool property metadata where metadata is an array or map. Added TDS-compatible language values to domain-context.xml. Was:

```
<poolproperty property="Language" value="[eng, spa]" label="[eng, spa]"/>
```

    After fix:

```
<poolproperty property="Language" value="ENU" label="ENU"/>
<poolproperty property="Language" value="ENU-Braille" label="ENU-Braille"/>
```

33. Fixed elementtype for testblueprint.bpelement with elementtype = "strand" and "contentlevel" and fixed parentid for bpelement. Was:

```
<testblueprint>
  <bpelement elementtype="test" ftitemcount="0" maxftitems="140" maxopitems="143" minftitems="0" minopitems="0
    <identifier name="SBAC-ELA-3-ALEX-0623-S-31" label="FirstChangesInTA-CI" version="1.0" uniqueid="5589a8a0
  </bpelement>
  <bpelement elementtype="segment" ftitemcount="0" maxftitems="30" maxopitems="40" minftitems="0" minopitems="
  parentid="SBAC-ELA-3-ALEX-0623-S-31">
    <identifier name="C1-Adaptive" version="1.0" uniqueid="C1-Adaptive-1.0"/>
  </bpelement>
  <bpelement elementtype="strand" ftitemcount="0" maxftitems="10" maxopitems="10" minftitems="0" minopitems="0
  parentid="SBAC-ELA-3-ALEX-0623-S-31">
    <identifier name="SBAC-ELA-v1:1-IT" version="1.0" uniqueid="SBAC-ELA-v1:1-IT-1.0"/>
  </bpelement>
  <bpelement elementtype="strand" ftitemcount="0" maxftitems="5" maxopitems="5" minftitems="0" minopitems="0"
  parentid="SBAC-ELA-3-ALEX-0623-S-31">
    <identifier name="SBAC-ELA-v1:1-IT|10-3" version="1.0" uniqueid="SBAC-ELA-v1:1-IT|10-3-1.0"/>
  </bpelement>
  <bpelement elementtype="strand" ftitemcount="0" maxftitems="2" maxopitems="2" minftitems="0" minopitems="0"
  parentid="SBAC-ELA-3-ALEX-0623-S-31">
    <identifier name="SBAC-ELA-v1:1-IT|10-3|3.L.4" version="1.0" uniqueid="SBAC-ELA-v1:1-IT|10-3|3.L.4-1.0"/>
  </bpelement>
```

After fix:

```
<bpelement elementtype="segment" parentid="SBAC-ELA-11-ALEX-0129-2S-S-42-1.0" ftitemcount="0" maxf
  <identifier name="AdaptiveS-2" label="AdaptiveS-2" version="1.0" uniqueid="56abd666e4b043f38cc8
</bpelement>
<bpelement elementtype="strand" ftitemcount="0" maxftitems="0" maxopitems="0" minftitems="0" minop
  <identifier name="1-IT" version="1.0" uniqueid="SBAC-1-IT"/>
</bpelement>
<bpelement elementtype="contentlevel" parentid="SBAC-1-IT" ftitemcount="0" maxftitems="0" maxopite
  <identifier name="1-IT|10-11" version="1.0" uniqueid="SBAC-1-IT|10-11"/>
</bpelement>
<bpelement elementtype="contentlevel" parentid="SBAC-1-IT|10-11" ftitemcount="0" maxftitems="0" ma
  <identifier name="1-IT|10-11|11-12.L.4" version="1.0" uniqueid="SBAC-1-IT|10-11|11-12.L.4"/>
</bpelement>
<bpelement elementtype="contentlevel" parentid="SBAC-1-IT|10-11" ftitemcount="0" maxftitems="0" ma
  <identifier name="1-IT|10-11|11-12.L.4a" version="1.0" uniqueid="SBAC-1-IT|10-11|11-12.L.4a"/>
</bpelement>
```

34. Fixed parentid for testblueprint.bpelement with elementtype = “contentlevel” in case CoreStandards data was incorrect. Strip publication from testblueprint.bpelement with elementtype = “contentlevel” and “strand”. Was:

```
uniqueid="SBAC-ELA-v1:1-IT|10-3-1.0"
```

After fix:

```
uniqueid="SBAC-1-IT|10-11|11-12.L.4a"
```

35. Added publication to testblueprint.bpelement with elementtype = “contentlevel” and “strand”. Was:

```
<bpelement elementtype="contentlevel"
  <identifier name="1-IT|10-10" versi
  uniqueid="1-IT|10-10"/>
```

After fix:

```
<bpelement elementtype="contentlevel" ftitemcount=
    <identifier name="SBAC-ELA-v1:1-LT|3-8|8.L.4b"
    uniqueid="SBAC-ELA-v1:1-LT|3-8|8.L.4b"/>
```

36. Changed API itemsByIdentifierSearch sent to TIB to include full item name, like 'item-200-333'.
37. Correct the unique ID attribute of bpelement:
    1) Strip the publication name from bpelement.uniqueid attribute: i.e. “SBAC-ELA-V1:” and references. Was:

    ```
    uniqueid="1-IT|10-10|10.L.4"/>
    ```

    After fix:

    ```
    uniqueid="SBAC-1-IT|10-9"/>
    ```

    2) Pretend “<publisher>-“ to the LEVEL ONE standard uniqueid attribute, i.e. SBAC-1-IT, but not to the attribute name. Was:

    ```
    ="strand" ftitemcount="0" maxftitems="
    -IT" version="1.0" uniqueid="1-IT"/>
    ```

    After fix:

    ```
    elementtype="strand" ftitemcount="0" maxftitems="5" m
    ier name="1-IT" version="1.0" uniqueid="SBAC-1-IT"/>
    ```

    3) Fixed references.

    After fix:

    ```
    <bpelement elementtype="contentlevel" f
        <identifier name="1|P" version="1.0"
        uniqueid="SBAC-1|P"/>
    ```

    And references

    ```
    <testitem filename="item-200-27601.xml" itemtype="MI">
        <bpref>SBAC-MATH-7-ALEX-1119-ADAPT-S-24-2.0</bpref>
        <bpref>SBAC-1</bpref>
        <bpref>SBAC-1|P</bpref>
    ```

38. Fixed bpelementid reference in itemselectionparameter in the case where the bpelement reference is to a segment.

    Was:

    ```
    <itemselectionparameter bpelementid="Segm-1">
    ```

    After fix:

    ```
    <itemselectionparameter bpelementid="SBAC-MATH-7-ALEX-1119-ADAPT-S-24-2.0">
    ```

39. Fixed bug when item search criteria name is used more than once, but import of found items was using only the last value for criteria that was used more than once.
40. Fixed testitem.bpref element for affinity groups.

    After fix:

    ```
    <bpref>AffGroup-1-1.0</bpref>
    ```

41. Fixed the following issues:
    1) Fixed opitemcount and ftitemcount attribute values for testblueprint.bpelement with elementtype="test" and “segment”. Values of these parameters are the real number of items from itempool in this test. Was:

```
<bpelement elementtype="test" ftitemcount="0" maxftitems="140"
maxopitems="143" minftitems="0" minopitems="0" opitemcount="0">
   <identifier name="SBAC-ELA-3-ALEX-0623-S-31" label="FirstChang
</bpelement>
<bpelement elementtype="segment" ftitemcount="0" maxftitems="30"
maxopitems="40" minftitems="0" minopitems="0" opitemcount="0"
parentid="SBAC-ELA-3-ALEX-0623-S-31">
```

After fix:

```
<bpelement elementtype="test" ftitemcount="0" maxftitems="5"
maxopitems="9" minftitems="0" minopitems="0" opitemcount="37">
   <identifier name="SBAC-ELA-11-ALEX-0129-2S-S-42-1.0" version="1.0" uniqueid="SBAC-ELA-11-AI
</bpelement>
<bpelement elementtype="segment" parentid="SBAC-ELA-11-ALEX-0129-2S-S-42-1.0"
ftitemcount="0" maxftitems="3" maxopitems="5" minftitems="0" minopitems="0" opitemcount="13">
   <identifier name="AdaptiveS-1" label="AdaptiveS-1" version="1.0" uniqueid="56abd64ae4b043f3
```

2) Fixed "Operational" status of items – added new value to Program Management TestAuth environment parameter testauth.item.op.statuses = “Active/Operational, OP, Operational”. After fix:

testauth.item.op.statuses	Active/Operational,OP,Operational

3) Fixed condition item.getPrimaryStandard ().contains (this.standardKey) for new standardKey.
4) Fixed max/minftitems and max/minopitems attribute values for testblueprint.bpelement with elementtype = "test" as summa of the same testblueprint.bpelement elements with elementtype ="segment". Fixed length of the test. (See picture to 41.1)

42. Fixed double segment uniqueid in case itemLocation contains Affinity groups.
    Was:

```
<testitem filename="item-200-29403.xml" itemtype="EBSR">
   <bpref>SBAC-ELA-3-ALEX-1117-ADAPT-1S-S-10-1.0</bpref>
   <bpref>SBAC-ELA-3-ALEX-1117-ADAPT-1S-S-10-1.0</bpref>
   <bpref>SBAC-1-LT</bpref>
   <bpref>SBAC-1-LT|5-3</bpref>
   <bpref>AffGroup-1</bpref>
```

    After fix:

```
<testitem filename="item-200-62043.xml" itemtype="SA">
   <bpref>SBAC-ELA-5-SPRING-2016-S-16-Rami-CA-5-1.0</bpref>
   <bpref>SBAC-4-CR</bpref>
   <bpref>SBAC-4-CR|3-5</bpref>
   <bpref>Affinity1-1.0</bpref>
```

43. Fixed uniqueid (blueprints, segments and affinity groups) for scoring and reporting test specification files.
44. Fixed formpartition.identifier.uniqueid. Made it different for different versions.
45. **New behavior** (for item statistics support): if StatDomain attribute is not selected as test creation search criteria, do not include any itemscoredimension element into test specification;

added support for value of StatDomain search criteria = *, which will be used only from postman etc., not from TestAuth server.

46. Fixed uniqueid attribute value for enemylist.identifier element and objectid attribute value for enemylist.enemy element.
47. Fixed the following:
    1) Added version to name attribute in testspecification.identifier element, in order to resolve issues with new test version notion thru TestAuth, ART, TDS test_specification.xml loading stored procures and Student/Proctor server. Was:

    ```
    <identifier name="SBAC-ELA-3-ALEX-1104-FIXED-1S-S-145"
    ```

    After fix:

    ```
    <testspecification publishdate="2016-01-29T16:28:04.894·
       <identifier name="SBAC-ELA-11-ALEX-0129-2S-S-42-1.0"
    ```

    2) Switched scorename and scorelabel attributes values in reportingmeasure.scoretype element for Reporting test package to match legacy Reporting package.
    3) Switched name and label attribute values in computationrule.identifier element in Scoring test package to match legacy system.
    4) Removed label and set name to the name of name of view/edit Scoring rule screen for computationrule.computationparameter.identifier element in Scoring test package. Purpose - to match legacy system.
    5) Set computationrule.identifier.uniqueid attribute and computationrule.computationparameter.identifier.uniqueid attribute to random GUID for Scoring test package, in order to match legacy system.
    6) Updated publish-binding-metadata.xml to have attribute 'plevel' rather than 'level' in Scoring for performancelevels.performancelevel element and in Reporting for performancelevels.performancelevel element, in to match testspecification.official.dtd doc.
48. Added property attribute to computationruleparameter element in Scoring test_specification.xml file to identify index type of Dictionary-type parameters.
49. Added spreadsheet that describes Adaptive Algorithm parameters for Item Selection in Test Authoring.
50. Changed item name parameter from short to full item name, like 'item-200-333' for POST /itemsByIdentifierSearch.

2.1.2.1 Program Management Configuration Changes

1. Three parameters were added to Test Authoring's configuration file in Program Management:
    1) testauth.serveruniquenumber
    2) testauth.itembank
    3) testauth.client
2. An additional status value of "Operational" was added to the testauth.item.op.statuses parameter.

2.2 Test Item Bank

2.2.1 Requirements

1. Add support for item scoring statistics (see 2.4).
2. Add ability to update existing TIB items.
3. Make items and stimuli unique to a tenant.
4. Names of the item and stimuli must follow the convention `{item|stim}-{item bank ID}-{item number}`. For example, item-**187**-8271.
5. Each item and stimuli must be associated to an item bank via inclusion of the item bank number in the filename. For example, item-**187**-8271 references item bank 187.
6. Item bank number should be one of the available search criteria.

2.2.2 Changes Made

1. Added ability to have different validation for item vs. stimuli.
2. Allow loading of item validation to proceed in case Braille resource is missing
3. Introduced notion of itemBank associated with each item: ability to pass it to REST API (sftpFileImport, uploadFile, stagedFile), plus ability to store item Mongo collections with itemBank.
4. Added ability to search items by criteria that include itemBank (itemsByIdentifierSearch, item). ItemBank must be one of the search parameters. 'Item' REST API will only search for items that belong to a given itemBank.
5. Added ability to itemsByIdentifierSearch API to use list of items that include full item identifier as body of the POST, for example {'item-200-111', 'stim-200-123'} rather than {'111', '123'}.
6. Added ability to convert item language to TDS-compatible format and added Braille language if appropriate. Fixed item metadata appropriately. For example, if the old language attribute was "eng", "spa", after these changes those would become "ENU", "SPN", "ENU-Braille"
7. Added programmatic support for IrtStatDomain, including ability to do item searches with criteria that include statdomain.

2.3 Test Delivery System

This section details the changes made to TDS stored procedures

2.3.1 Item Selection Algorithm

1. Fixed null pointer exceptions for Fixed Form algorithm.
2. Allow to upload <testitem> without <itemscoredimension> element.
3. Remove restriction on item group identifiers being only "I-" or "G-".

2.3.2 load_adminstrands.sql stored procedure

1. Changed definition of v_issegmented (line 45)
2. Removed commented lines

2.3.3 load_adminsubjects.sql stored procedure

1. Changed definition of v_issegmented (line 101)

2. Changed selection condition from "tmp._key = sisp.bpelementid" to "tmp.testid = sisp.bpelementid" (line 190)
3. Changed selection condition from "tp.bpelementid = tmp._key" to "tp.bpelementid = tmp.testid" (line 198)

2.3.4 loader_testformgroupitems.sql stored procedure

1. Added INOUT argument to stored procedure ", INOUT v_sequentiallyformposition int" (line 19)
2. extractvalue(...attribute::formposition) replaced on v_sequentiallyformposition (line 35)
3. Added recalculation of v_sequentiallyformposition (line 60)

2.3.5 loader_testformitemgroup.sql stored procedure

1. Added declaration of v_sequentiallyformposition "declare v_sequentiallyformposition int default 1;" (line 28)
2. Added new argument in calling of loader_testformgroupitems stored procedure "call loader_testformgroupitems ..." (line 51)
3. The above two changes were made to fix the problem that the legacy test specification file had two places where formposition attributes were used, while the new test authoring specification has only one place where formposition is used.
    a. Legacy file attributes: **itemgroup**.formposition and **groupitem**.formposition (see below):

```
<identifier uniqueid="187-695" name="ELA Training 3-5::SP15" version="8226" />
<itemgroup formposition="1" maxitems="ALL" maxresponses="ALL">
  <identifier uniqueid="187-695:G-187-3709-0" name="187-695:G-187-3709-0" version="8226" />
  <passageref>187-3709</passageref>
  <groupitem itemid="187-1692" formposition="1" groupposition="1" adminrequired="false" responserequired="false" isactive="tru
  <groupitem itemid="187-1700" formposition="2" groupposition="2" adminrequired="false" responserequired="false" isactive="tru
</itemgroup>
<itemgroup formposition="2" maxitems="ALL" maxresponses="0">
  <identifier uniqueid="187-695:I-187-1701" name="187-695:I-187-1701" version="8226" />
  <groupitem itemid="187-1701" formposition="3" groupposition="1" adminrequired="true" responserequired="true" isactive="true"
</itemgroup>
```

        i. The itemgroup.formposition attribute is the order number of the group in the test. The groupitem.formposition attribute is the global order number of the item in the test. TDS MySQL database and Fixed Form Item Selection Algorithm use groupitem.formposition. Note that the groupitem.groupposition attribute denotes the order of the item within the group.
        ii. Note the Test Authoring user interface allows the user to change the order of the groups and the order of the items in the group.

    b. New test authoring test_specification attributes:

```
<itemgroup formposition="5" maxitems="0" maxresponses="0">
  <identifier name="187-315260422300679:G-187-3735-1" label="" version="1.0" uniqueid="187-315260422300679:G-187-3735-1"
  <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="true" itemid="187-2466" resp
  <groupitem adminrequired="true" blockid="A" groupposition="2" isactive="true" isfieldtest="true" itemid="187-2468" resp
  <groupitem adminrequired="true" blockid="A" groupposition="3" isactive="true" isfieldtest="true" itemid="187-2525" resp
  <passageref passageid="187-3735"/>
</itemgroup>
<itemgroup formposition="11" maxitems="0" maxresponses="0">
  <identifier name="187-315260422300679:G-187-3743-1" label="" version="1.0" uniqueid="187-315260422300679:G-187-3743-1"
  <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="true" itemid="187-2558" resp
  <groupitem adminrequired="true" blockid="A" groupposition="2" isactive="true" isfieldtest="true" itemid="187-2643" resp
  <groupitem adminrequired="true" blockid="A" groupposition="3" isactive="true" isfieldtest="true" itemid="187-2644" resp
  <groupitem adminrequired="true" blockid="A" groupposition="4" isactive="true" isfieldtest="true" itemid="187-2645" resp
  <passageref passageid="187-3743"/>
</itemgroup>
```

    Or

```
<itemgroup formposition="9" maxitems="0" maxresponses="0">
   <identifier name="200-646143594508670:G-200-1920-1" label="" version="1.0" uniqueid="200-646143594508670:G-200-1920-1",
   <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="false" itemid="200-57392" re
   <groupitem adminrequired="true" blockid="A" groupposition="2" isactive="true" isfieldtest="false" itemid="200-70046" re
   <passageref passageid="200-1920"/>
</itemgroup>
<itemgroup formposition="13" maxitems="0" maxresponses="0">
   <identifier name="200-646143594508670:G-200-1934-1" label="" version="1.0" uniqueid="200-646143594508670:G-200-1934-1",
   <groupitem adminrequired="true" blockid="A" groupposition="1" isactive="true" isfieldtest="false" itemid="200-57511" re
   <passageref passageid="200-1934"/>
</itemgroup>
```

1. In the MongoDB database (see figure below), There are two sort indexes: “level1SortIndex” is itemgroup.formposition and “level2SortIndex” is groupitem.groupposition. Note that groupitem.formposition is not local parameter. Therefore groupitem.formposition cannot be calculated from itemgroup.formposition (order number of group, level1SortIndex) and groupitem.groupposition (order number of item in the group, level2SortIndex) because we need to know number of items in previous groups. For this reason, loader_testformgroupitems.sql and loader_testformitemgroup.sql stored procedures were changed to calculate the global order number of the item in the test (groupitem.formposition). These changes don't alter the behavior of these stored procedures for legacy test specification files.

```
Update | Delete | New Field | Duplicate | Refresh | Text | Collapse

{
   "_id": ObjectId("566738ea438d19efb7f28789"),
   "_class": "org.opentestsystem.authoring.testauth.domain.Item",
   "itemLocation": {
     "0": {
       "segmentId": "5665e7431cb3e7c44b6c3476",
       "level1SortIndex": NumberInt(9),
       "level2SortIndex": NumberInt(2),
       "itemIdentifier": "item-200-70046",
       "itemGroupId": "566738e9438d19efb7f28770",
       "adminRequired": true,
       "responseRequired": true,
       "active": true,
       "fieldTestItem": false,
       "blockId": "A",
       "associatedItem": false,
       "_class": "org.opentestsystem.authoring.testauth.domain.AdaptiveItemLocation"
     }
   },
   "tibIdentifier": "item-200-70046",
```

2.3.6 updatetdsconfigs.sql stored procedure

1. Added population of fields ", origin, source" in configs.client_testwindow table (line 271) by null values (line 274)
2. Added population of field "dateentered" in configs.client_testtooltype table (line 441) by now(3) value (line 442)
3. Added population of field "dateentered" in configs.client_testtooltype table (line 470) by now(3) value (line 473)

2.3.7 load_adminitemmasurementpar.sql

Changed definition of v_issegmented (line 30)

2.4 Item Statistics Support

Item statistics are a necessary part of the administration of adaptive tests. Prior to this task order, there was no allowance for incorporating them either in TIB, Test Authoring, or TSB – nor in the current Item Metadata Specification.

2.4.1 Change Overview

1. Update the Item Metadata Specification to support multiple sets of item statistics.
2. Update TIB to support storage and export of this data.
3. Update Test Authoring to support selection and retrieval of items based on this data.

2.4.2 Requirements

1. TIB shall be able to import item metadata including multiple sets of item statistics.
2. TIB shall be able to export item metadata to Test Authoring.
3. Test Authoring shall be able to select item from TIB also based on item statistics set names.

2.4.3 Design

2.4.3.1 Item Content Import to TIB

Item content packages (which include content and metadata) must be exported from the Item Authoring system using an updated version (as of 05-Feb-2016) of the Item Metadata Specification [1]. Specifically, the <IrtDimension> node would change from the example in Figure 2 to that in Figure 3 (addition of an optional IrtStatDomain node is shown in **red bold**):

```
<IrtDimension>
   <IrtDimensionPurpose>Computation</IrtDimensionPurpose>
   <IrtModelType>IRT3pl</IrtModelType>
   <IrtScore>1</IrtScore>
   <IrtWeight>1.0</IrtWeight>
   <IrtParameter>
      <Name>IRT_A</Name>
      <Value>1.0</Value>
   </IrtParameter>
   <IrtParameter>
      <Name>IRT_B</Name>
      <Value>-0.4</Value>
   </IrtParameter>
   <IrtParameter>
      <Name>IRT_C</Name>
      <Value>0.33</Value>
   </IrtParameter>

</IrtDimension>
```

Figure 1: Existing IrtDimension Node Example

```
<IrtDimension>
   <IrtDimensionPurpose>Computation</IrtDimensionPurpose>
   <IrtModelType>IRT3pl</IrtModelType>
   <IrtStatDomain>Online</IrtStatDomain>
   <IrtScore>1</IrtScore>
   <IrtWeight>1.0</IrtWeight>
   <IrtParameter>
      <Name>IRT_A</Name>
      <Value>1.0</Value>
   </IrtParameter>
   <IrtParameter>
      <Name>IRT_B</Name>
      <Value>-0.4</Value>
   </IrtParameter>
   <IrtParameter>
      <Name>IRT_C</Name>
      <Value>0.33</Value>
   </IrtParameter>

</IrtDimension>
```

Figure 2: New IrtDimension Node Example

2.4.3.2 Storage in TIB

TIB's Mongo DB has been altered to support the additional item statistics data.

2.4.3.3 TIB Item Update

TIB code has been modified to support updates to existing items. Since item statistics, by their nature, are available after the item has been in the field (and are updated periodically), TIB must support updates to existing items.

2.4.3.4 Export from TIB

TIB export (via REST interface) shall include any associated Item Statistics information. This applies to any call which returns item metadata (including from test Authoring and from TSB). For example, **GET /item/{identifier}** REST call would return JSON including any and all available item statistics. The rules governing which set of scoring statistics are returned are described below.

irtStatDomain parameter status	Result	Comment
not included in API call	Items are returned stripped of their scoring statistics sets	This is available via API call as well as the Test Authoring user interface.
*	Items are returned with the full complement of all scoring statistics sets in all stat domains they may have.	This is only available via API call, not via the Test Authoring user interface.
stat_domain_label	Only items whose stat domain equals the value of	This is available via API call as well as the Test Authoring user

	stat_domain_label are included. This includes labels such as • `--NOVALUE--` • `Paper` • `Online`	interface. `--NOVALUE--` is a special value which is stored in TIB DB for items with statistics that are not associated with any other named statistics.

Table 2: IrtStatDomain Parameter Usage

2.4.4 Import into Test Authoring

Test Authoring would provide an additional filter term to the TIB REST call to allow selection of the statistics referenced by a particular IrtStatDomain. For example, **GET /item?{key}={value(s)}…**

2.4.4.1 Test Authoring Export to TSB

Test Authoring must include item statistics into the Test Administration XML. The packaging DTD does not need to change.

3 Task Order 3 Changes

3.1 Test Spec Bank

The Test Spec Bank (TSB) API [4] and export package process [5] are essentially unchanged due to this work. The only exception is a change to the way the TIB API is being called.

3.1.1 Requirements

1. Ensure TSB-generated content packages are compatible with Content Uploader and follow the SmarterApp Content Packaging Specification.
2. Update the specifications as needed.

3.1.2 Changes Made

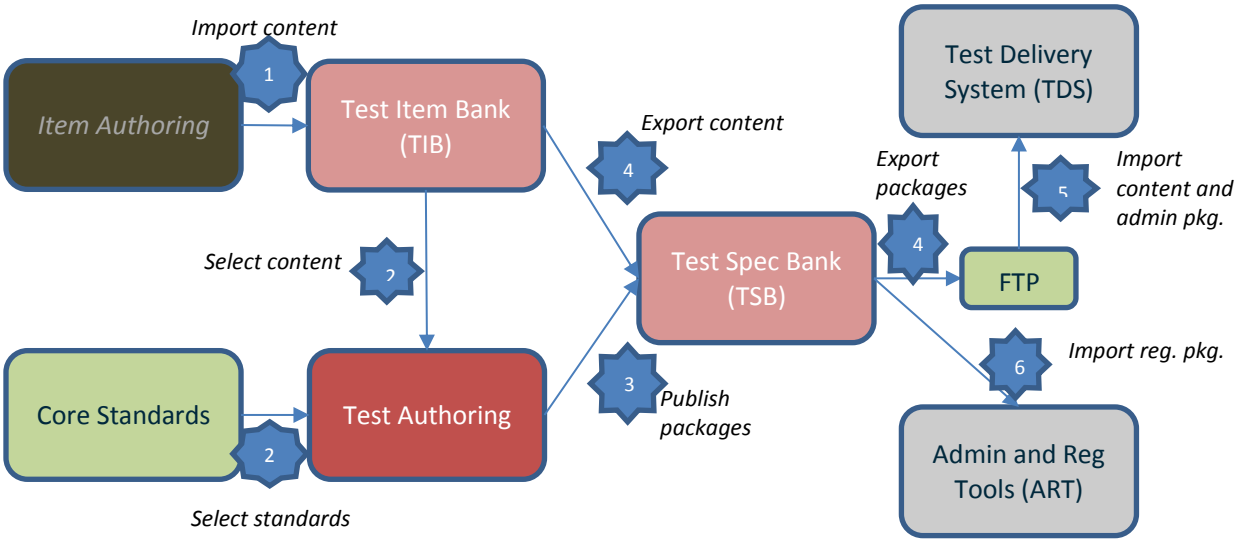
1. Added passage file names to /exportSet REST API sent to TIB. It now sends a list of item identifiers as well as associated items such as stimuli, word lists (WIT), and tutorials (TUT), rather than a list of only item identifiers for items presented to students.

3.1.3 Interface Change Description

See Section 4.2.

4 Interfaces and Change Descriptions

The detailed interface descriptions among these components are available in the SBAC-11 Design Documentation. Provided below are high-level overviews and/or references related to interactions between Test Authoring, TSB, and TIB. Of all of the API interactions pictured in Figure 3, only the TIB interfaces were changed. It should also be noted that not all interfaces are pictured here; for example, Single Sign-On (SSO), Program Management (PM), and Permissions interfaces.



Not pictured: Interfaces to SSO, Program Management, and Permissions

Figure 3: Test Authoring Components and Interfaces

4.1 Test Authoring to TIB

Test Authoring calls TIB to obtain item information (see path #2 in Figure 3). These APIs are fully described in the TIB API specification [2]. Changes made to this specification for Task Order 2 are described below.

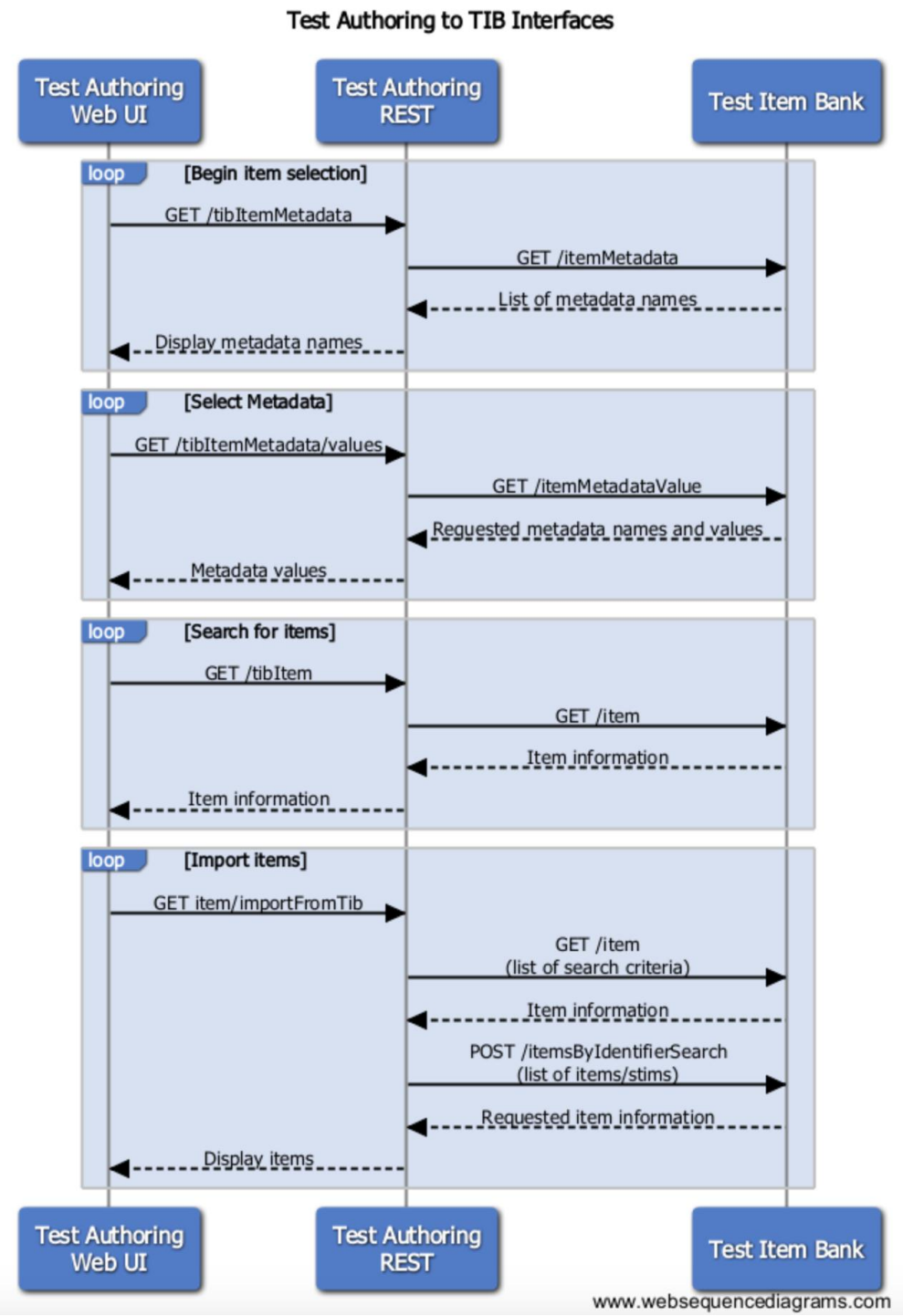


Figure 4: Test Authoring to TIB APIs

4.1.1 Begin Item Selection

This is initiated when the user browses to Segment Pools or Forms and selects “TIB Search” in the Test Authoring UI under Tools. This action calls the Test Authoring REST interface, which makes a call to TIB to obtain the list of available item metadata. The user will select from this list as he narrows down his search criteria during the item selection process (see Figure 5).

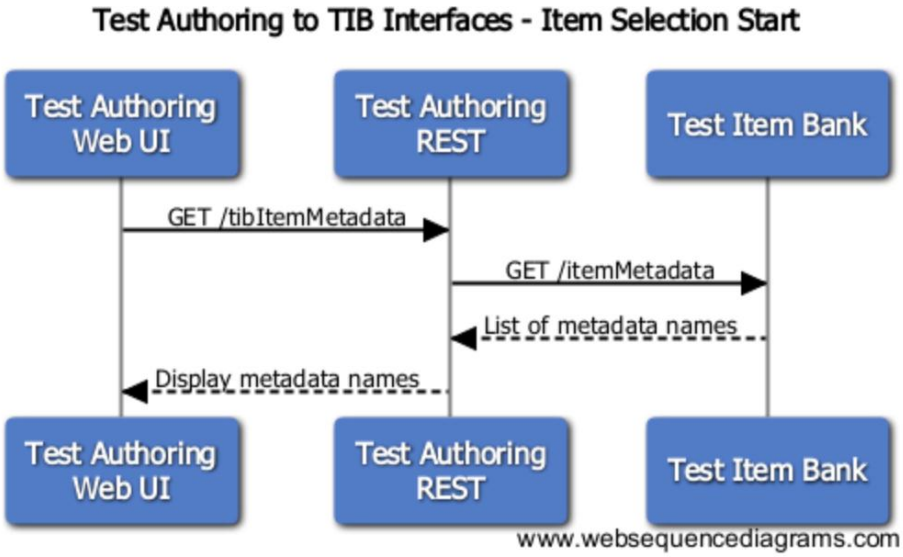


Figure 5: TA to TIB - Begin Item Selection

The following API calls are unchanged:

- `GET /itemMetadata` - called when Test Authoring receives `/tibItemMetadata` API call

4.1.2 Select Metadata Values

Once the test author selects a metadata name to filter on, Test Authoring UI must get a list of the available values for that metadata attribute. This is done each time a new filter is selected (see Figure 6).

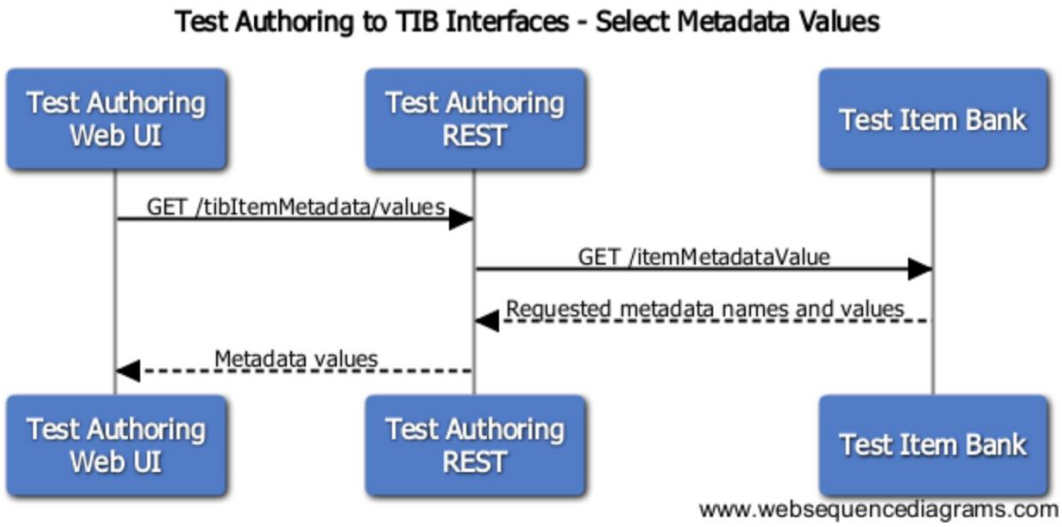


Figure 6: Select Metadata Values

The following API calls are unchanged:

- `GET /itemMetadataValue` – called when Test Authoring receives `/tibItemMetadata/values` API call.

4.1.3 Search for Items

Once search criteria are established by the test author, the UI calls the REST interface, which calls TIB to get the items that match the criteria provided in the call (see Figure 7).

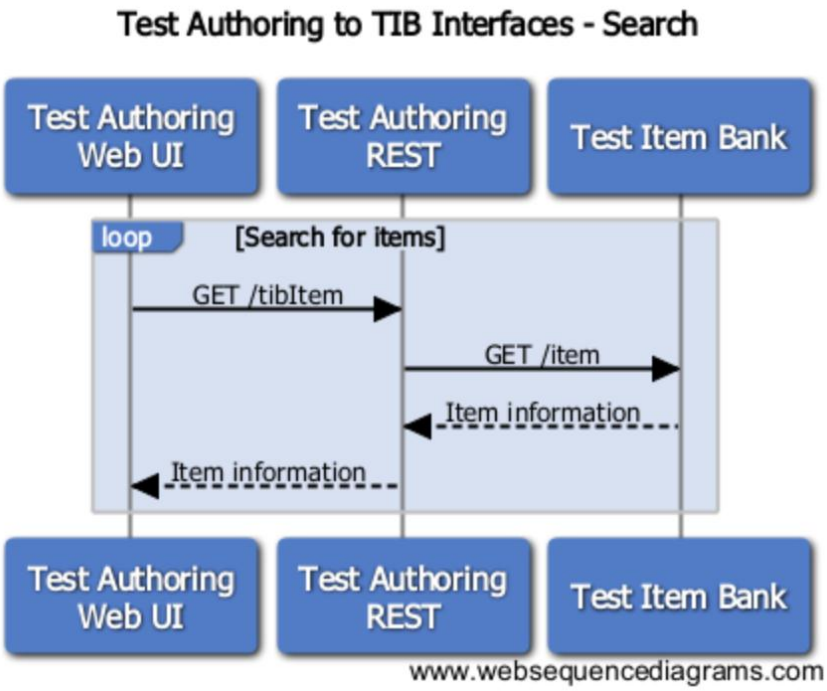


Figure 7: Search for Items

The following API call has been modified:

- `GET /item` - called when TestAuth receives `item/importFromTib` API call OR `/tibitem` API
    - NEW: additional parameter **itemBank**
    - NEW: added special processing (see section 2.4.3.4) of retrieved data in case search parameter includes IrtDimension.IrtStatDomain. As described above, there are three cases: statDomain criteria is not present, statDomain is present and is set to '*' and statDomain is present and is set to any other value.

4.1.4 Import Items

Once the test author is satisfied with the selected items, he selects "import". This pulls item information (not the actual full content) from TIB (see Figure 8).

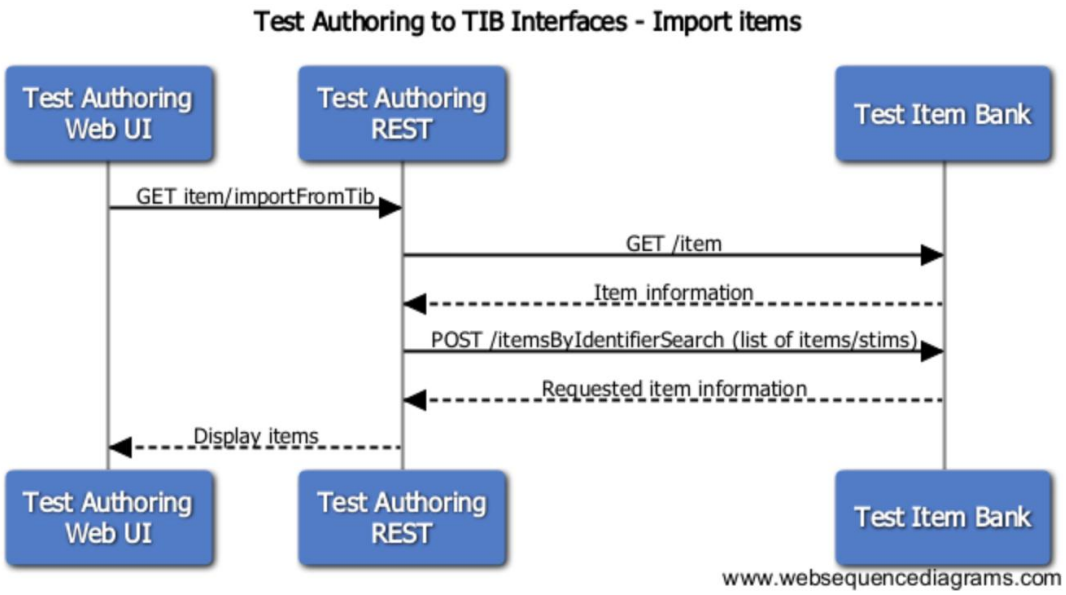


Figure 8: Import Items

The following API calls have been modified:

- `GET /item` - called when TestAuth receives `item/importFromTib` API call OR `/tibitem` API
    - NEW: additional parameter **itemBank**
    - NEW: added special processing (see section 2.4.3.4) of retrieved data in case search parameter includes IrtDimension.IrtStatDomain. As described above, there are three cases: statDomain criteria is not present, statDomain is present and is set to '*' and statDomain is present and is set to any other value.
- `POST /itemsByIdentifierSearch` - called by Test Authoring while processing `/item/importFromTib` API.
    - NEW: additional parameter **itemBank**
    - CHANGED: identifiersList in request body have been changed from '1234' format to 'item-200-1234' or 'stim-200-1234' format

4.2 TSB to TIB

Test Spec Bank (TSB) calls TIB to obtain an "export set", or item content zip file (see path #4 in Figure 3). These APIs are described in the TIB API specification [2]. TSB sequence diagrams, designs, and other specifications are available at [5] and [6]. Changes made to this specification for Task Order 2 are described below. The APIs changed in this task order are:

- `POST /exportSet`
- `GET /exportSet/{setid}`

4.2.1 Export Administration Package

The process of exporting an Administration package is a time consuming process (due to the large amount of content) involving TSB, TIB, and an SFTP site. The process can be described below and pictured in Figure 9.

1. TSB uses the Administration XML published by Test Authoring to determines which items, stims, and other associated content it will need to satisfy the package requirements.
2. TSB calls TIB with this detailed list. (POST /exportSet)
3. TIB receives this information, generates a unique “export set ID”, and returns that ID to TSB for status tracking purposes.
4. In the meantime, TIB gathers this content, arranges it in a format consistent with the Smarter Balanced item packaging specification, and delivers the zip file to the configured target SFTP location and updates status of this export set to ‘Complete’ in TIB’s Mongo DB.
5. TSB periodically polls TIB for the current status of this time-consuming, asynchronous process (GET /exportSet/{setid})
6. When GET /exportSet/{setid} reply indicates that status is ‘Complete’ (i.e. the zip is complete and available on the SFTP site), TSB transfers the zip locally and creates a new zip file containing the administration XML and the content zip, and transfers it back to the SFTP server.

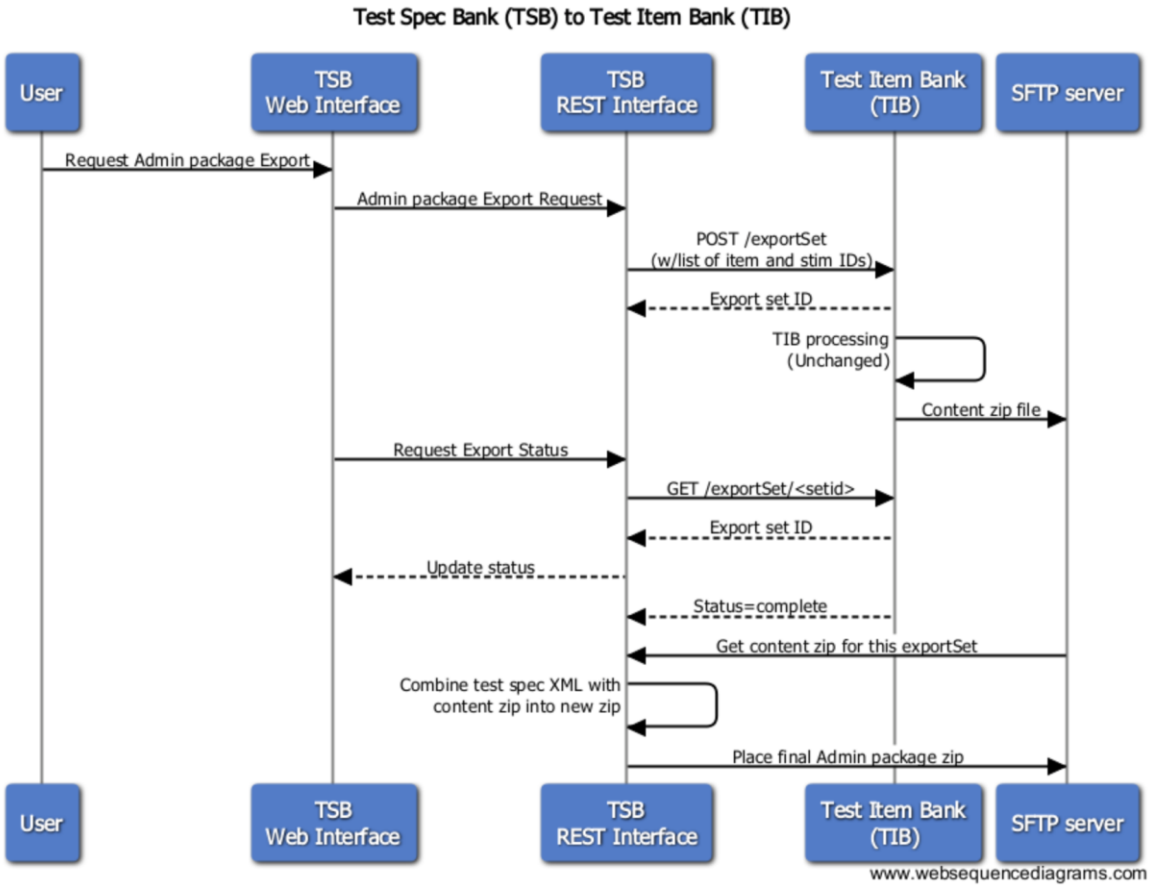


Figure 9: TSB to TIB

4.2.2 Export Scoring Package

To export a Scoring package, TSB uses the Test Scoring XML published by Test Authoring and exports it to the SFTP server. This does not involve TIB.

4.2.3 Export Reporting Package

To export a Reporting package, TSB uses the Test Scoring XML published by Test Authoring and exports it to the SFTP server. This does not involve TIB.

4.2.4 Export Registration Package

To export a Registration package, TSB uses the Registration XML published by Test Authoring and exports it to the SFTP server. This XML package is also obtained by ART via a REST API call to TSB. No changes were made to this API. This does not involve TIB.

4.2.5 Export Complete Package

To export a Complete package, TSB performs all of the actions in 4.2.4 through 4.2.3 and exports them as a single zip file to the SFTP server.

4.3 load_item_package.pl script to TIB

The script load_item_package.pl is used to load items (zip files of QTI-formatted, Smarter/IMS packaged items) into TIB. This `script` makes use of the following TIB APIs and has been modified as follows:

- `/sftpFileImport` (TIB API)
  - NEW: additional parameter **itemBank**
  - NEW: include itemBank element for each item
  - NEW: implemented algorithm of converting SmarterBalanced language value (eng, spa) to TDS expected values (ENU, ESN)
  - NEW: added algorithm of adding 'ENU-Braille' as another language element value based on value of provided language and value of BrailleType element
  - CHANGED: Support for IrtDimension was enhanced with IrtStatDomain element support

4.4 Other TIB API Changes

Other import-related TIB APIs have been modified for consistency with the new itemBank parameter:

- `/uploadFile`
- `/stagedFile`

5 TIB Database Reference

The TIB Mongo database has nine collections. Below are some example documents for the collections that may be relevant to Smarter Balanced TIB architecture changes. Note that documents in the itemMetadataValue are not linked to any other collection.

Collection Name	Example Document
exportSet	{ "_id": ObjectId("564e1672e4b09a46640b1b71"), "_class": "org.opentestsystem.authoring.testitembank.domain.ExportSet", "tenantId": "53cd604ee4b0127edfb96ccf", "status": "EXPORT_COMPLETE", "items": [{ "identifier": "item-200-29403", "version": "7" }, { "identifier": "item-200-30989", "version": "6" }, ….
fs.chunks	(filename info)
fs.files	(Content)
importSet	(for import files)
item	{ "_id": ObjectId("563a70a4e4b0a807ec1d6468"), "_class": "org.opentestsystem.authoring.testitembank.domain.Item", "identifier": "item-200-5567", "interactionType": "EQ", "tenantId": "53cd604ee4b0127edfb96ccf", "itemBank": "200", "version": "22", "status": "Operational", "subject": "MATH", "intendedGrade": "11", "languages": ["ENU", "ESN"], "itemScoreDimensionList": [{ "irtModelType": "IRT3pl", "irtScore": "2", "irtDimensionPurpose": "C",

	"irtWeight": "1",
	"irtStatDomain": "ForPaper",
	"irtParameter": [
	{
	"name": "a",
	"value": "1.96"
	},
	{
	"name": "b",
	"value": "2.05"
	},
	{
	"name": "c",
	"value": "-2.05"
	}
	]
	},
	… and remainder of item metadata in JSON format

itemHistory	This collection enables TIB to store all versions of an item. When TSB asks for a list of items (POST /exportSet), the request includes the version of each item it needs. Items are selected from this collection rather than from the item collection.
itemMetadata	{ "_id": ObjectId("56323afbe4b056179ed3ffd0"), "_class": "org.opentestsystem.authoring.testitembank.domain.ItemMetadata", "tenantId": "53cd604ee4b0127edfb96ccf", "metadataKeys": ["AccessibilityTagsASLLanguage", "AchievementQuintile", "AdministrationDate", "AlgebraFunctionDescriptor", "AllowCalculator", "AssociatedStimulus", "AssociatedTutorial", "AssociatedWordlist", "BrailleType", "Claim2Category", "Claim2RevisionCategory", "ContentSubTaskModel", "ContentTaskModel", "CopyrightAndOtherRestrictions", "DepthOfKnowledge", "EducationalDifficulty", "EnemyItem", "EvidenceStatement", "Identifier", "IntendedGrade", "InteractionType", "IrtDimension.IrtDimensionPurpose", "IrtDimension.IrtModelType",

```
    "IrtDimension.IrtParameter.Name",
    "IrtDimension.IrtParameter.Value",
    "IrtDimension.IrtScore",
    "IrtDimension.IrtStatDomain",
    "IrtDimension.IrtWeight",
    "ItemAuthorIdentifier",
    "ItemSpecFormat",
    "Language",
    "LastModifiedBy",
    "MathematicalPractice",
    "MaximumGrade",
    "MaximumNumberOfPoints",
    "MinimumGrade",
    "Notes",
    "PerformanceTaskComponentItem",
    "PresentationFormat",
    "PtClassroomActivity",
    "PtEnemy",
    "PtSequence",
    "PtWritingType",
    "ScorePoints",
    "ScoringEngine",
    "SecurityStatus",
    "SmarterAppItemDescriptor",
    "SpecificationsVersion",
    "StandardPublication.PrimaryStandard",
    "StandardPublication.Publication",
    "StandardPublication.SecondaryStandard",
    "Status",
    "StimulusCopyright",
    "StimulusFormat",
    "StimulusGenre",
    "StimulusGraphic",
    "StimulusLength",
    "StimulusName",
    "StimulusReadabilityFK",
    "StimulusReadabilityLexile",
    "StimulusTextComplexity1",
    "StimulusTextComplexity2",
    "StimulusTextComplexity4",
    "StimulusTextComplexity5",
    "StimulusTextComplexity6",
    "StimulusTextComplexity7",
    "StimulusTextComplexity8",
    "StimulusType",
    "Subject",
    "SufficientEvidenceOfClaim",
    "TargetAssessmentType",
    "Version",
    "xmlns"
  ]
}
```

itemMetadataValue	{
	"_id": ObjectId("5638c959e4b0a6b9d9291db3"),
	"_class":
	"org.opentestsystem.authoring.testitembank.domain.ItemMetadataValue",
	"tenantId": "53cd604ee4b0127edfb96ccf",
	"metadataKey": "IrtDimension.IrtParameter.Value",
	"metadataValue": "-0.05"
	}

system.indexes	(internal use)

Table 3: TIB DB Structure