# Overview

The Third Party testing package includes the following items:

1. The AUSKey Keystore Manager **(AKM) Version 2**

This must be obtained by registering directly with Standard Business Report (SBR).

* <http://www.sbr.gov.au/software-developers/what-can-i-expect/registration-form>

Please ensure that you enter ‘**Department of Education and Training – USI’**in the agency field to identify you as a USI SWD

1. **AUSkey Device key store and Organisation codes for the test environment**
   1. Keystore.xml : this file contains your test environment AUSkey
   2. AUSkeyInfo (docx or pdf): this document contains details of the keys contained within the store and organisation codes that can be used when calling the USI web services.
2. **Vanguard S007 Security Token Service Technical Service Contract**Documentation for the service that must be used to retrieve a security token that is required to connect the USI web services.
3. **USI Web Services Technical Services Contract**Documentation for the USI web services.
4. **Sample code for the USI web services**

Demonstration code that calls the USI Web Services, in Microsoft C# .Net and separately in Java. You will need to have the AKM before you can compile the same.

1. **USI check character algorithm**

The USI check character is calculated using a Luhn Mod N algorithm.  The character weighting and algorithm pseudo code for calculating it is shown in the attached document.

1. **USI Web Services SMS presentation**

A presentation on web services covering information about authentication, authorisation and the methods available to organisations.

# Web Services Connection Tips

1. The details for connecting to VANguard STS are covered in the VANguard documentation, including sample code for .NET and Java.
2. The details for connecting to USI are included in the Technical Services Contract. There is sample code for .NET and USI is working on an equivalent sample code for Java.
3. From experience to date with some parties already in third party test, use libraries wherever possible:  
   - WCF for .NET (the latest version possible - .NET 3.5+ - works “out-of-the-box”)  
   - for Java, there is a library called Metro authored by Sun and Microsoft.

# Support

Support is available to assist in connecting to the VANguard and USI web services.

Please read the provided documentation and sample code carefully and attempt to troubleshoot problems before calling support

If you do request support, be sure to send:

* + Fiddler (or equivalent) trace, SSL decrypted
  + Call to STS and response received
  + Call to USI and response received

# Integration Steps for .NET

In the target project, perform the following steps to connect to USI:

1. Ensure the following 5 dependencies from AUSkey are added as references:
   1. Abr.AuskeyManager.dll
   2. BouncyCastle.CryptoExt.dll
   3. Common.Logging.dll
   4. Common.Logging.Log4Net.dll
   5. log4net.dll
2. Ensure the project has a reference to System.ServiceModel
3. If the project has been integrated with earlier sample code, remove it:
   1. Delete the USI “Service Reference” from the solution.
   2. Open the config file and delete remaining USI and VANguard clients and bindings from the <system.serviceModel> section.
   3. Delete any code imported from earlier sample code. NOTE: Retain code that uses the IUSIService interface and any classes it accepts/returns.
4. Add a new service reference using one of the following addresses, as appropriate:
   1. Prod = [https://portal.usi.gov.au/Service/**v2**/UsiService.svc](https://portal.usi.gov.au/Service/v2/UsiService.svc)
   2. 3PT = [https://3pt.portal.usi.gov.au/Service/**v2**/UsiService.svc](https://3pt.portal.usi.gov.au/Service/v2/UsiService.svc)
5. Copy the class ServiceChannel.cs from the sample code.
   1. Check that the three const fields match your keystore.
6. Update/write your calling code that uses IUSIService so that it gets an instance from ServiceChannel.OpenWithAUSkey().
   1. When finished, be sure to close the client and channel as shown in the sample code.

Notes

* In future, an update to the published service WSDL will only require selecting “Update Service Reference” on the USI Service Reference. None of the steps above will need to be redone except as a troubleshooting step.
* If it is necessary to do multiple calls in quick succession, you can use an instance of IUSIService multiple times before closing it.

# Integration Steps for Java

1. Add a META-INF folder and include the following:
   1. wsdl folder containing:
      1. WSDL file downloaded from target service
      2. cxf\_bindings.config file from sample code - used with Apache CXF wsdl2java utility to make it wrap all operation arguments in singular request/response objects.
   2. wsit-client.xml file, which targets the WSDL file above
2. Add references to:
   1. abrakm.jar (from ATO)
   2. auskey-dep-1.1.jar (from ATO)
   3. webservices-api.jar (from Metro 2.3)
   4. webservices-extra.jar (from Metro 2.3)
   5. webservices-rt.jar (from Metro 2.3)
3. Generate the client contract objects (requests and responses)
   1. Download Apache CXF and use wsdl2java to generate new class files. http://cxf.apache.org/download.html
   2. Include a bindings file to ensure the request arguments are wrapped in singular request objects.
   3. Command for 3pt = D:\Java\apache-cxf-3.0.1\bin\wsdl2java.bat -b "META-INF\wsdl\cxf\_bindings.config" -client -wsdlLocation "src\\META-INF\\wsdl\\BatchCreateService\_3pt.wsdl" "META-INF\wsdl\BatchCreateService\_3pt.wsdl"
      1. The project src folder includes a file RegenerateClientJava.bat that does this.
      2. Ensure that the two WSDL references match the name of the WSDL file; the production environment does not have “\_3pt” suffixed to its WSDL file name.
4. Import the resulting "au" folder of client contract objects into \src
5. Import the file UsiServiceChannel.java from sample code and make the following changes:
   1. Either copy your keystore.xml to the keystore folder, or update AUSKEY\_KEYSTORE to match the keystore.xml location.
   2. Update AUSKEY\_ALIAS to match your alias.
   3. Update AUSKEY\_PASSWORD to match your password.
   4. Update SetupRequestContext() so its STS\_ENDPOINT and STS\_WSDL\_LOCATION setup items match the appropriate VANguard STS environment.
6. In your application code, call UsiServiceChannel.GetNewClient() and use the resulting object to perform operations on the USI service.

# Vendor Test Pack - Change History

|  |  |  |
| --- | --- | --- |
| Date | Editor | Detail |
|  |  | Initial Drafts |
| 24/7/2014 | Trent Kerin | Removed console interaction from ServiceChannel.cs  Updated sample code document to match code and 3PT endpoint.  Added .NET integration steps to ReadMeFirst |
| 13/08/2014 | Trent Kerin | Added Java sample code and integration steps.  Updated .NET sample code document to remove SecurityTokenManager. |