**Prepare for Assembly Test**

**Module Description**

This module will deal with the next step in testing, that is, prepare for the Assembly Test. You will create automated test scripts, integrate the created code, build the application, and perform an assembly readiness test.

**Scenario**

The team will review all the Unit Test TCERs that are appropriate candidates to verify the readiness of the application for the assembly test. The TCERs to be identified will effectively test the sending of messages (i.e., method calls) from class to class across layers of the Application code as directed by the Test Approach document.

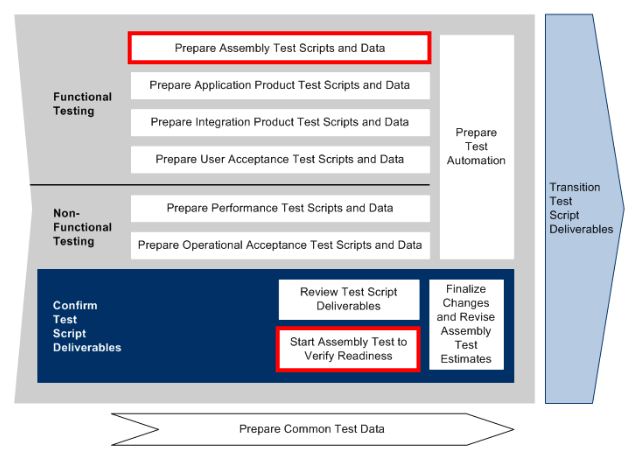
The Team Lead (with help from the team) will create an Assembly Readiness Test Cycle Control Sheet (TCCS). The TCCS lists the order in which the test scripts will be executed and reflects the order groups of classes/components that will be introduced into the test environment.

The team will define steps to test assemblies of growing complexity by ordering the introduction of classes in the assembly until it contains all the classes and:

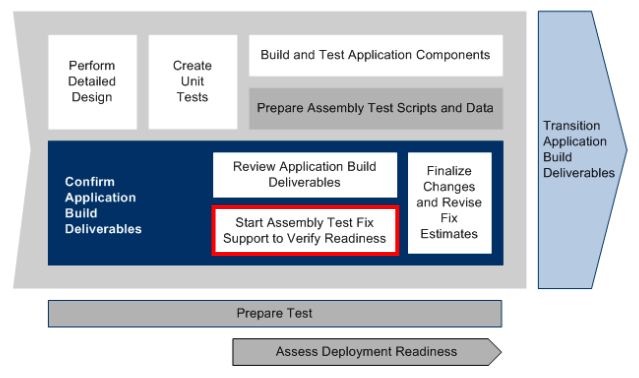
* Determine which TCER outputs should be expected at each level of the assembled application code.
* Make updates, if required, to TCERs to reflect when stub code is, and isn’t, used.
* Create JUnit classes for Assembly Test Readiness testing by leveraging the existing Unit Test JUnits.

As previously stated in the orientation, team status meetings are held periodically and this time is allocated for you to have a meeting with the rest of the team. The Team Lead will host this meeting and gather the status from you and other members of the team. The status is sent to the management on a regular basis and helps in communicating how well (or poorly) the project is progressing and can help the management determine if intervention is needed.

**ADM Activity Context Diagram**



Application > 4100 Build Application > 4193 Start Assembly Test Fix Support to Verify Readiness



**Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **ADM Task** | **ADM Responsibility** | **Description** |
| Project Manager |  | Management Oversight | Provides guidance on the [Team Status Report](https://accenture.desire2learn.com/content/enforced/9406-Pending/NCCC_Project_Work_Products/1_PM/Release_1/StatusRpt_FERS_R1_REF.doc?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8). |
| Team Lead | 4710 - Prepare Assembly Test Scripts and Data  4193 - Start Assembly Test Fix Support to Verify Readiness  4793 - Start Assembly Test to Verify Readiness | Additional Performer (for ADM task 4710)  Primary Performer (for ADM task 4193)  Additional Performer (for ADM task 4793) | Creates the Assembly Readiness TCCS.  Leads the Status meeting and compiles feedback.  Addresses and corrects any issues where possible. Escalates issues when appropriate. |
| Team (All) | 4710 - Prepare Assembly Test Scripts and Data  4193 - Start Assembly Test Fix Support to Verify Readiness  4793 - Start Assembly Test to Verify Readiness | Primary Performer (for ADM task 4710)  Primary Performer (for ADM task 4193)  Primary Performer (for ADM task 4793) | Chooses the Unit Test Scenarios appropriate for Assembly Test Readiness.  Creates automated Assembly Readiness Test scripts. |

**Participant Instructions**

*NOTE: Some of the steps here are typically done by testers, but for smaller projects and projects behind schedule the Developer Team can perform them.*

1. First, open the [Assembly Test Scenarios](https://accenture.desire2learn.com/content/enforced/9406-Pending/W4_Participant_Materials/Module_36/Assm_TestScen_FERS_R1_REF.xlsx?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8) work product. To assist you, you may wish to refer to the [Component Class Detailed Design](https://accenture.desire2learn.com/content/enforced/9406-Pending/NCCC_Project_Work_Products/3b_App_Design/Release_1/CompClsDes_FERS_R1_WPC_REF.docx?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8) and the [Application Development Standards](https://accenture.desire2learn.com/content/enforced/9406-Pending/NCCC_Project_Work_Products/4_Tech_Arch/Release_1/AppDevStd_FERS_R1_REF.docx?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8).  
2. Save the file so it can be used as your team’s TE583 Assembly Readiness Test Scenarios work product.  
3. Review the Assembly Test Scenarios (and associated [Test Conditions and Expected Results](https://accenture.desire2learn.com/content/enforced/9406-Pending/W4_Participant_Materials/Module_36/Assm_TestCond_FERS_R1_REF.xlsx?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8)) and mark those scenarios that will not be used for Assembly Readiness Testing. Once the review is complete, delete all the marked test scenarios so those that remain will form part of Assembly Readiness Testing.  
4. The Team Lead (with help from the team) will create the team’s TE585 Assembly Readiness Test Cycle Control Sheet (TCCS) based on the [Assembly Test Cycle Control Sheet](https://accenture.desire2learn.com/content/enforced/9406-Pending/W4_Participant_Materials/Module_36/Assm_TestCycle_FERS_R1_REF.xlsx?_&d2lSessionVal=YnoL4daZGC8ne2RyGf0cWMrk8).  
5. The team will then:

a. Determine what Test Scenario outputs should be expected at each level of assembled application code.

b. Define the test data (ie, input values/parameters) to get the desired test scenario outputs (expected results).

c. Create new JUnit classes for assembly readiness testing by leveraging the existing Unit Test JUnits.

i. Create new JUnit test script code using the name defined in the Test Cycle Control Sheet.

ii. Open the original JUnit Classes for Unit Test.

iii. Copy the code from the Unit Test scripts into the Assembly Readiness Test scripts and make any required updates.

6. Run the assembly readiness test scripts as defined in the Assembly Readiness TCCS.  
7. Correct code, if needed, based on the results of the test execution.  
8. **Team Lead Only**: As team members are completing their work products, perform the Team Lead Review in Module 37 – Review Application Build Deliverables.

**Hints and Tips**

* Be careful when changing and using your input values in your Test Scenarios!
* The TCCS will list the order in which the test scripts will be executed and reflect the order groups of classes/components that will be introduced into the test environment.
* Due to the simplicity of the code, perform Vertical testing only (the sending of messages or method calls) from class to class across layers of the Web Application code and consider one of two options:
  + One iteration of testing: the entire assembly Controllers down to the database.
  + **Advanced activity** - two iterations of testing: DAOs to database (Integration Tier to Resource Tier), then Controllers to database (Business Tier to Resource Tier).
* Reflect on what you learned in your Professional Skills training and in the Master Practice documents to make sure you all have a productive meeting.