# 4.4 XXX测试

## 4.4.1 XXX功能测试（GN\_ACTOR\_BT)

### 4.4.1.1 XXX功能测试（GN\_ACTOR\_BT\_MISSION\_CREATE)

#### 4.4.1.1.1.1 XXX并运行行为树测试

##### **表 XXX并运行行为树测试 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | XXX\_测试\_001 | | | | **标识** | | XXX\_TEST\_001 | |
| **对应测试环境** | | 使用了Google Test框架（GTest）作为测试框架，依赖GTest和GMock库。测试环境需要安装CMake（版本3.5.1或更高）和GTest开发包。 | | | | | | | |
| **追踪关系** | | 被测模块是YAML解析器中的EmitterState类，主要用于管理YAML发射器的状态。 | | | | | | | |
| **测试用例综述** | | 该测试用例主要测试YAML解析器中的EmitterState类的功能，包括验证节点开始时的状态变化、文档计数增加、子节点计数增加、标志位重置以及在嵌套组中 的行为等。 | | | | | | | |
| **用例初始化** | | 测试用例初始化时，每个测试用例都会创建一个EmitterState实例，并在测试结束后销毁。测试前不需要修改配置文件或预置测试数据。 | | | | | | | |
| **前提和约束** | | 执行测试用例前需要完成以下条件：1) 完成代码编译；2) 确保GTest和GMock库已安装；3) 确保yaml-cpp的相关头文件和源文件路径正确配置；4) 确保测试程序unit\_test已生成并位于构建目录中。 | | | | | | | |
| **测试步骤** | | | | | | | | | |
| **序号** | **输入及操作** | | | | **期望结果与评估标准** | | | | **测试结果** |
| 1 | 进入构建目录：cd /home/lijiao/work/TE-Agent/examples/StartedNode/bu ild | | | | 成功进入构建目录 | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:** |
| 2 | 执行测试程序：./unit\_test | | | | 终端输出测试结果，包括每个测试用例的执行状态，例如： [ RUN ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount [ OK ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount (0 ms) ... [==========] 4 tests from 1 test suite ran. (0 ms total) [ PASSED ] 4 tests. | | | | □通过  ☑不通过  其它\_\_\_\_  **截图 1:**  **截图 2:**  **截图 3:** |
| **测试用例终止条件** | | | 测试用例终止条件是所有测试用例执行完毕，或者遇到致命错误导致测试无法继续执行。 | | | | | | |
| **测试结果评估标准** | | | 测试结果评估标准是所有测试用例均显示为PASSED状态，且没有失败或错误的测试用例。终端输出应显示类似'[ PASSED ] 4 tests.'的信息。 | | | | | | |
| **测试用例执行结果** | | | 不通过 | | | | | | |
| **设计人员** | | |  | | | | | | |
| **测试时间** | | |  | **测试人员** | | |  | | |
| **操作人员** | | |  | | | | | | |

##### **表 XXX并运行行为树测试 2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | XXX\_测试\_002 | | | | **标识** | | XXX\_TEST\_002 | |
| **对应测试环境** | | 使用了Google Test框架（GTest）作为测试框架，依赖GTest和GMock库。测试环境需要安装CMake（版本3.5.1或更高）和GTest库。 | | | | | | | |
| **追踪关系** | | 被测模块是YAML解析器中的EmitterState类，主要用于管理YAML文档的发射状态，包括节点开始、组管理、标志位设置等。 | | | | | | | |
| **测试用例综述** | | 该测试用例主要用于测试YAML解析器中的EmitterState类的功能，包括验证节点开始时的文档计数增加、子节点计数增加、标志位重置以及在嵌套组中的行为 等。 | | | | | | | |
| **用例初始化** | | 测试用例初始化时，通过SetUp方法创建一个EmitterState实例，并在TearDown方法中释放该实例。每个测试用例执行前都会重新初始化状态。 | | | | | | | |
| **前提和约束** | | 执行测试用例前需要完成以下条件：1) 完成代码编译；2) 确保GTest和GMock库已安装；3) 确保yaml-cpp的头文件和源文件路径正确配置；4) 确保测试程序unit\_test生成在构建目录下。 | | | | | | | |
| **测试步骤** | | | | | | | | | |
| **序号** | **输入及操作** | | | | **期望结果与评估标准** | | | | **测试结果** |
| 1 | 进入构建目录：cd /home/lijiao/work/TE-Agent/examples/StartedNode/bu ild | | | | 成功进入构建目录 | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:**  **截图 2:** |
| 2 | 执行测试程序：./unit\_test | | | | 终端输出类似以下内容： [==========] Running 4 tests from 1 test suite. [----------] Global test environment set-up. [----------] 4 tests from EmitterStateTest [ RUN ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount [ OK ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount (0 ms) [ RUN ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount [ OK ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount (0 ms) [ RUN ] EmitterStateTest.StartedNo deResetsAllFlags [ OK ] EmitterStateTest.StartedNo deResetsAllFlags (0 ms) [ RUN ] EmitterStateTest.StartedNo deInNestedGroups [ OK ] EmitterStateTest.StartedNo deInNestedGroups (0 ms) [----------] 4 tests from EmitterStateTest (0 ms total) [==========] 4 tests from 1 test suite ran. (0 ms total) [ PASSED ] 4 tests. | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:**  **截图 2:**  **截图 3:**  **截图 4:** |
| **测试用例终止条件** | | | 测试用例终止条件是所有测试步骤执行完成，或者遇到致命错误导致测试无法继续执行。 | | | | | | |
| **测试结果评估标准** | | | 测试结果评估标准是终端输出显示所有测试用例均通过（[ PASSED ] 4 tests），并且没有失败或错误的测试用例。 | | | | | | |
| **测试用例执行结果** | | | 通过 | | | | | | |
| **设计人员** | | |  | | | | | | |
| **测试时间** | | |  | **测试人员** | | |  | | |
| **操作人员** | | |  | | | | | | |

##### **表 XXX并运行行为树测试 3**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | XXX\_测试\_003 | | | | **标识** | | XXX\_TEST\_003 | |
| **对应测试环境** | | 使用Google Test框架(gtest/gmock)进行单元测试，依赖yaml-cpp项目中的emitterstate模块。 | | | | | | | |
| **追踪关系** | | 测试YAML库中的EmitterState模块，该模块负责管理YAML发射器的状态，包括文档计数、标志位管理和组结构处理。 | | | | | | | |
| **测试用例综述** | | 该测试用例测试YAML::EmitterState类的功能，主要包括：1) 开始节点时文档计数和标志位的正确更新；2) 在非空组中开始节点时子节点计数的正确更新；3) 开始节点时所有标志位的正确重置；4) 在嵌套组结构中开始节点时的行为。 | | | | | | | |
| **用例初始化** | | 1) 每个测试用例开始前会创建一个新的EmitterState实例；2) 测试结束后会销毁该实例；3) 部分测试会预先设置标志位或创建组结构。 | | | | | | | |
| **前提和约束** | | 1) 必须完成项目编译；2) 需要安装gtest/gmock库；3) 需要yaml-cpp的头文件和源文件(特别是emitterstate.h和emitterstate.cpp)；4) 可执行程序会被生成在构建目录下。 | | | | | | | |
| **测试步骤** | | | | | | | | | |
| **序号** | **输入及操作** | | | | **期望结果与评估标准** | | | | **测试结果** |
| 1 | 进入构建目录：cd /home/lijiao/work/TE-Agent/examples/StartedNode/bu ild | | | | 成功进入包含测试程序的目录 | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:** |
| 2 | 执行测试程序: ./unit\_test | | | | 终端输出包含4个测试用例的执行结果[ RUN ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount[ OK ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount[ RUN ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount[ OK ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount[ RUN ] EmitterStateTest.StartedNo deResetsAllFlags[ OK ] EmitterStateTest.StartedNo deResetsAllFlags[ RUN ] EmitterStateTest.StartedNo deInNestedGroups[ OK ] EmitterStateTest.StartedNo deInNestedGroups[========= =] 4 tests from EmitterStateTest ran[ PASSED ] 4 tests | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:**  **截图 2:**  **截图 3:**  **截图 4:** |
| **测试用例终止条件** | | | 当所有4个测试用例都执行完毕，并且测试程序正常退出时，认为用例执行完成。 | | | | | | |
| **测试结果评估标准** | | | 如果终端输出显示所有4个测试用例都标记为[OK]或[PASSED]，且没有失败或错误的测试用例，则认为测试通过。 | | | | | | |
| **测试用例执行结果** | | | 通过 | | | | | | |
| **设计人员** | | |  | | | | | | |
| **测试时间** | | |  | **测试人员** | | |  | | |
| **操作人员** | | |  | | | | | | |

##### **表 XXX并运行行为树测试 4**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | XXX\_测试\_004 | | | | **标识** | | XXX\_TEST\_004 | |
| **对应测试环境** | | 使用了Google Test框架(gtest)作为测试框架，依赖yaml-cpp库的头文件和源文件，以及gmock库。 | | | | | | | |
| **追踪关系** | | 被测模块是YAML解析器中的EmitterState类，用于管理YAML发射器的状态。 | | | | | | | |
| **测试用例综述** | | 该测试用例主要测试YAML解析器中的EmitterState类的功能，包括验证节点开始时的状态变化、组内子节点计数、标志位重置以及在嵌套组中的行为。 | | | | | | | |
| **用例初始化** | | 1. 在SetUp方法中创建EmitterState实例；2. 在TearDown方法中释放实例；3. 测试前会设置各种状态标志位来验证StartedNode的行为。 | | | | | | | |
| **前提和约束** | | 1. 必须完成项目编译；2. 可执行程序unit\_test会被生成在构建目录下；3. 需要安装gtest和gmock库；4. 需要yaml-cpp的头文件和源文件在指定路径下可用。 | | | | | | | |
| **测试步骤** | | | | | | | | | |
| **序号** | **输入及操作** | | | | **期望结果与评估标准** | | | | **测试结果** |
| 1 | 进入构建目录：cd /home/lijiao/work/TE-Agent/examples/StartedNode/bu ild | | | | 成功进入包含unit\_test可执行文件的目录 | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:** |
| 2 | 运行测试程序: ./unit\_test | | | | 终端会打印类似以下的测试结果： [==========] Running 4 tests from 1 test suite. [----------] Global test environment set-up. [----------] 4 tests from EmitterStateTest [ RUN ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount [ OK ] EmitterStateTest.StartedNo deWithEmptyGroupsIncrement sDocCount (0 ms) [ RUN ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount [ OK ] EmitterStateTest.StartedNo deWithNonEmptyGroupsIncrem entsChildCount (0 ms) [ RUN ] EmitterStateTest.StartedNo deResetsAllFlags [ OK ] EmitterStateTest.StartedNo deResetsAllFlags (0 ms) [ RUN ] EmitterStateTest.StartedNo deInNestedGroups [ OK ] EmitterStateTest.StartedNo deInNestedGroups (0 ms) [----------] 4 tests from EmitterStateTest (0 ms total) [----------] Global test environment tear-down [==========] 4 tests from 1 test suite ran. (0 ms total) [ PASSED ] 4 tests. | | | | ☑通过  □不通过  其它\_\_\_\_  **截图 1:**  **截图 2:**  **截图 3:**  **截图 4:** |
| **测试用例终止条件** | | | 当所有测试用例执行完毕，无论成功或失败，测试程序都会自动终止。 | | | | | | |
| **测试结果评估标准** | | | 如果终端输出显示所有4个测试都通过（[ PASSED ] 4 tests），则认为测试用例通过。 | | | | | | |
| **测试用例执行结果** | | | 通过 | | | | | | |
| **设计人员** | | |  | | | | | | |
| **测试时间** | | |  | **测试人员** | | |  | | |
| **操作人员** | | |  | | | | | | |