

# Table of Contents

<b>Using the Integration Service.....</b>	<b>1</b>
Service Description.....	1
Checks to be done before declaring an application ready for production.....	1

# Using the Integration Service

## Service Description

The Integration Service facilitates the second step in the development process of a database application. Its goal is to provide a testing ground for developed applications before they are deployed to the production services.

The Physics Databases Services team maintains a set of Oracle RAC systems reserved for the Integration Service. For the time being these RACs are:

- **intr** (2 nodes - Integration RAC for ATLAS)
- **int2r** (2 nodes - Integration RAC for CMS)
- **int3r** (2 nodes - Integration RAC for LCG)
- **int4r** (2 nodes - Integration RAC for LHCb)
- **int5r** (5 nodes)

In order to use the service the responsible for the application under testing should contact us and request a time slot (usually week) for tests. It is necessary to provide the expected data volume (>50 GB) and the number of expected simultaneous connections (>50). Moreover, during the testing a workload should be generated that it should be simulating the production level.

## Checks to be done before declaring an application ready for production

- Before using the service it should be made sure that the checklist for a well-written Oracle application has actually been followed. Note that for applications based on CORAL [☞](#) many of the best practices are internally implemented, or guided by the design of the API; however the user must still follow the proposed guidelines, especially when the schema is defined or updated.
- When running the test applications the DBAs should be informed so that they monitor the activities on the database side. In the case of well tuned applications the DBAs should be able to notice:
  - ◆ No queries without bind variables.
  - ◆ No connections that last for too short.
  - ◆ A single physical connection to the database server per application. Oracle allows an application to start several user sessions sharing the same physical connection.
  - ◆ Not frequent CPU-intensive operations. If this is the case, the schema and the logic of the application should be reviewed.
- During the running of the tests using the Integration Service it should be demonstrated that the application should be able to handle gracefully some cases where a database node, the listener or the instance is temporarily or permanently unavailable. Note that **this requires running the tests in coordination with the DBAs of our team.**

<i>Use Case</i>	<i>Test Case Description</i>
The database server is temporarily unavailable (either the listener or the database instance is down) when an application tries to access it for the first time. The application re-tries a few times before it eventually manages to connect.	<ol style="list-style-type: none"><li>1. The DBA stops the listener (or shuts down the instance) before the test application starts.</li><li>2. The user launches the application.</li><li>3. The DBA restarts the listener (or brings up the instance) within a</li></ol>

	<p>time window that is smaller than the timeout controlled/set by the application.</p> <p>4. The application should eventually succeed in performing its intended operations.</p>
<p>The database server becomes temporarily unavailable during the running of an application in a time window, where there is no interaction with the database. The application eventually manages to continue normally in case read-only operations are involved.</p>	<ol style="list-style-type: none"> <li>1. The user launches a read-only test application.</li> <li>2. The DBA confirms that the application starts interacting with the database.</li> <li>3. The user suspends the application.</li> <li>4. The DBA brings down the instance.</li> <li>5. The user resumes the application.</li> <li>6. The DBA brings back the instance.</li> <li>7. The user verifies that the application eventually succeeds to complete all its intended operations until its normal exit.</li> </ol>
<p><i>The above are repeated but instead of retrying, once the timeout has elapsed, the application fails over to another replica.</i></p>	
<p>The listener of an Oracle database server becomes unavailable after an application has connected for the first time. The application continues to use the physical connection.</p>	<ol style="list-style-type: none"> <li>1. The user launches the test application.</li> <li>2. The DBA as soon as interaction is observed with the database brings down the listener.</li> <li>3. The user verifies that the application continues to perform the intended operations until its normal exit.</li> </ol>

- In Persistency much of this is handled transparently for the user by the ConnectionService component [↗](#) and before every release the corresponding test cases [↗](#) should be tried as well.

This topic: PSSGroup > DatabaseIntegrationService

Topic revision: r6 - 2010-06-11 - PeterJones



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback