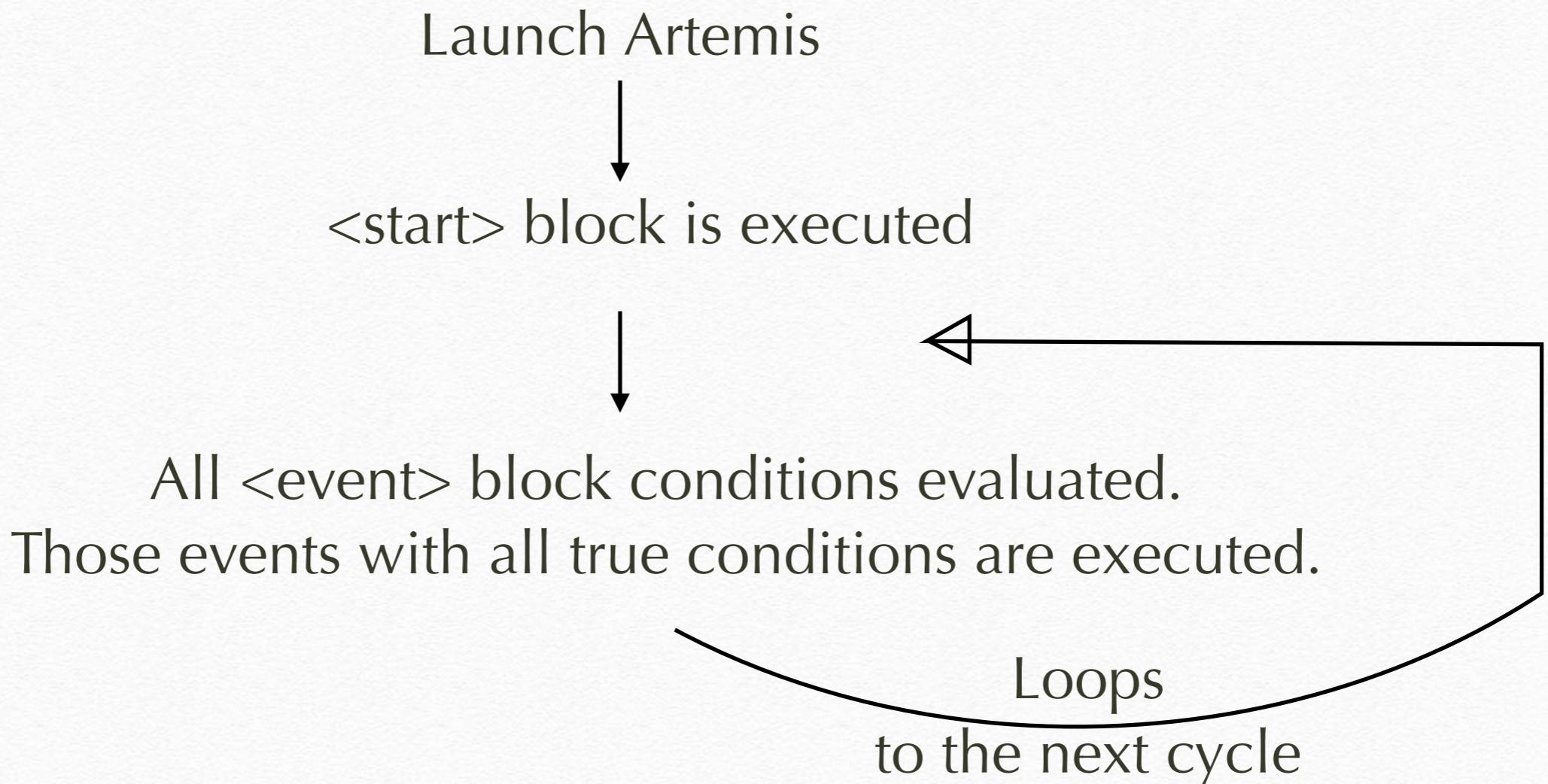


ARTEMIS
Mission Scripting
Quick Start Guide
for
Programmers

by T. L. Ford
www.Cattail.Nu

How the Artemis Engine Works with the XML



Anatomy of an <event> block:

```
<event>
<condition>
<condition>
<condition>
<command>
<command>
</event>
```

Works Like...

```
<event>
IF <condition> AND
IF <condition> AND
  IF <condition>
    DO <command> AND
    DO <command>
  </event>
```

```

<mission_data version="1.60">

    <!-- ..... Executes on Artemis launch -->
    <start>
        <create type ="player" x="50000" y="0" z="50000" name="Artemis" angle="0"/>
        <set_variable name="currentState" value="0" />
    </start>

    <!-- ..... Executes on currentState=0 -->
    <event>
        <if_variable name="currentState" comparator="EQUALS" value="0" />
        <set_variable name="currentState" value="1" />
    </event>

    <!-- ..... Executes on currentState=1 -->
    <event>
        <if_variable name="currentState" comparator="EQUALS" value="1" />
        <set_variable name="currentState" value="0" />
    </event>

</mission_data>

```

This code will flop currentState between 0 and 1.

Value at End of Cycle	currentState
Start	0
First Cycle	1
Second Cycle	0
Third Cycle	1
Fourth Cycle	0
Fifth Cycle	1

*Wastefully sucking down CPU cycles
as it should be doing something else...
or terminating...*

Use a Timer.

```
<mission_data version="1.60">

<start>
  <set_variable name="created" value="0">
</start>

<event>
  <if_variable name="created" comparator="EQUALS" value="0" />
  <create type="enemy" x="45846.0" y="0.0" z="94421.0" angle="294.8" raceKeys="Kralien enemy"
hullKeys="large medium small" fleetnumber="1" name="badlyNamedShip" />
  <set_variable name="created" value="1">
    <set_timer name="addShipAI" seconds="2" />
</event>

<event>
  <if_timer_finished name="addShipAI" />
  <clear_ai name="badlyNamedShip" />
  <add_ai type="POINT_THROTTLE" value1="50000.0" value2="0.0" value3="50000.0" value4="1.0"
name="badlyNamedShip" />
</event>

</mission_data>
```

Some objects want to exist before properties
(like the Brain Stack) can be added.

The forums recommend a timer to give the engine
a chance to fully create the object.

Expressions and randomness.

```
<mission_data version="1.60">

<start>
  <set_variable name="someNumber" value="0">
  <set_variable name="anotherNumber" value="5">
  <set_variable name="increment" value="1">
</start>

<event>
  <if_variable name="someNumber" comparator="LESS_EQUAL" value="anotherNumber+increment" />
  <set_variable name="randomNumber" randomIntLow="1" randomIntHigh="10" />
  <set_variable name="someNumber" value="increment+randomNumber-3">
  <addto_object_property value="someNumber" property="positionY" name="Artemis" />
</event>

</mission_data>
```

*Variables can be used in expressions (+ - / *).*

*Don't write code like this...
This is just to give you syntax.
Use random numbers and expressions meaningfully.*

To Test/Run your Mission:

1. Create a folder named **MISS_yourMissionName**
2. Name your XML **MISS_yourMissionName.xml**
3. Put the XML in your folder
4. Copy your folder to **%artemisInstallationFolder%\dat\Missions**
5. Start Artemis
6. Select 'Start Server'
7. Select 'Scripted Game'
8. Choose your Mission
9. Select 'Start Game'

Note: Custom artwork has to be installed on each station.

Debugging.

Try out the <log text="Debug message">.

Try pressing F7 multiple times
on the server when Artemis is running.

Next: Check out mission-file-docs.txt that came with your installation.

Install and try out the Mission Editor.

Read some good code examples that actually do Real Things. Many fine missions can be downloaded.

Read the forums.

Another Example.

```
<mission_data version="1.60">

<start>
  <create type ="player" x="50000" y="0" z="50000" name="Artemis" angle="0"/>
  <set_variable name="currentState" value="0" />
</start>

<!-- ..... Executes on currentState=0 -->
<event>
  <if_variable name="currentState" comparator="EQUALS" value="0" />
  <set_variable name="currentState" value="1" />
</event>

<!-- ..... Executes on currentState=1 -->
<event>
  <if_variable name="currentState" comparator="EQUALS" value="1" />
  <set_variable name="currentState" value="2" />
</event>

</mission_data>
```

This code will execute both events sequentially and stop.

Value at End of Cycle	currentState
Start	0
First Cycle	1
Second Cycle	2
Third Cycle	2
Fourth Cycle	2
Fifth Cycle	2

Another Example.

```
<mission_data version="1.60">

<start>
  <create type ="player" x="50000" y="0" z="50000" name="Artemis" angle="0"/>

  <!-- ..... Enable: Set enabled to 1 -->
  <!-- ..... Disable: Set enabled to 0 -->
  <set_variable name="enabled" value="1" />
  <set_variable name="currentState" value="0" />

</start>

<!-- ..... Executes on enabled=1 AND currentState=0 -->
<event>
  <if_variable name="enabled" comparator="EQUALS" value="1" />
  <if_variable name="currentState" comparator="EQUALS" value="0" />
  <set_variable name="currentState" value="1" />
</event>

<!-- ..... Executes on enabled=1 AND currentState=1 -->
<event>
  <if_variable name="enabled" comparator="EQUALS" value="1" />
  <if_variable name="currentState" comparator="EQUALS" value="1" />
  <set_variable name="currentState" value="2" />
</event>

<!-- ..... Executes on enabled=1 AND currentState=1 -->
<event>
  <if_variable name="enabled" comparator="EQUALS" value="1" />
  <if_variable name="currentState" comparator="EQUALS" value="2" />
  <set_variable name="enabled" value="0" />
  <set_variable name="currentState" value="0">
</event>

</mission_data>
```

	Value at End of	enabled	currentState
Start		1	0
First Cycle		1	1
Second Cycle		1	2
Third Cycle		0	0
Fourth Cycle		0	0
Fifth Cycle		0	0

This code will execute each event sequentially and stop, primed for something to make enabled=1 again.

You could still use one variable if you wanted...