

Table of Contents

Examples of custom process components 2

[Programmers Reference](#), [Example](#), [Custom Process](#)

Examples of custom process components

The following example shows the code for the simple process component that changes the attribute of the process input.

```
public class SimpleProcess implements IProcess
{
    public SimpleProcess()
    {
    }
}
/**
 * @see com.bas.basserver.executionengine.IProcess#cancel()
 */
@Override
public boolean cancel()
{
    return true;
}
/**
 * @see
com.bas.basserver.executionengine.IProcess#execute(com.bas.basserver.executi
onengine.IExecutionEngine,java.lang.Object[])
 */
@Override
public Object execute(IExecutionEngine engine, Object[] parameters)
throws SuspendProcessException, ExecutionException,
AccessDeniedException
{
    // one entity is declared as process input
    if (parameters== null || parameters.length!=1 || !(parameters[0]
instanceof IEntity))
    {
        return null;
    }
    // get the parameter
    IEntity entity=(IEntity)parameters[0];
    Object oldValue = null;
    try
    {
        // change "attrName" attribute to "attrValue" in memory
        oldValue = entity.getAttributeValue ("attrName");
        entity.setAttributeValue("attrName","attrValue");
    }
    catch (Exception e)
    {
    }
    try
    {

```

```

        // update entity in persistence
        engine.updateEntity(this, entity,
            new String[]{"attrName"},
            new Object [] {oldValue} );
    }
    catch (ServerTimeOutException ste)
    {
        throw new SuspendProcessException(ste);
    }
    return null;
}
/**
 * @see
com.bas.basserver.executionengine.IProcess#resume(com.bas.basserver.executio
nengine.IExecutionEngine,java.lang.Object)
 */
@Override
public Object resume(IExecutionEngine engine, Object reply)
    throws SuspendProcessException, ExecutionException,
AccessDeniedException
{
    return null;
}

```

The next example shows the code of the process that performs some lengthy calculation and throws `SuspendProcessException` every now and again:

```

public class LongProcess implements IProcess
{
    private boolean m_cancelled = false;
    private long m_count;
    public LongProcess()
    {
    }
    /** @see IProcess#cancel()*/
    @Override
    public boolean cancel()
    {
        // set m_cancelled flag. This flag will be checked by execute // and
resume methods
        synchronized (this)
        {
            m_cancelled = true;
        }
        return true;
    }
    /**
     * @see IProcess#execute(IExecutionEngine, Object[])
     */
    @Override
    public Object execute(IExecutionEngine engine, Object[] parameters)

```

```
throws SuspendProcessException, ExecutionException
{
    m_count=0;
    execute(engine);
    return null;
}
private void execute(IExecutionEngine engine)
throws SuspendProcessException
{
    try
    {
        // get the context in which the process is running
        IExecutionContext myContext = engine.getExecutionContext(this);
        while (true)
        {
            Thread.sleep(500);
            // finish after 400 iterations
            if (m_count++ >=400)
                break;
            // every 4 iterations update our progress
            if (m_count % 4==0)
            {
                int progress = myContext.getProgress ();
                if (progress<100)
                    myContext.setProgress (progress + 1);
            }
            // check if we have been cancelled
            synchronized (this)
            {
                if (m_cancelled)
                    break;
            }
            // every 120 iterations suspend ourselves so that
            // transaction can be committed (though this is
            // only for illustrational purposes as we are not
            // doing any transactional operations here such as
            // changing values of attributes etc)
            if (m_count % 120 == 0)
            {
                throw new SuspendProcessException (true);
                // we will be called soon – see resume
            }
        }
    }
    catch (InterruptedException ie)
    {
    }
}
/**
 * @see IProcess#resume(IExecutionEngine, Object)
 */
```

```
@Override
public Object resume (IExecutionEngine engine, Object reply)
throws SuspendProcessException, ExecutionException
{
    // this is called every 120 iterations
    execute (engine);
    return null;
}
```

From:
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:
<http://www.awareim.com/dokuwiki/docs/3500/0300/0314?rev=1680674262>

Last update: **2023/04/05 05:57**

