Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

# Simulation Network Exploitation

## *SEDReC station: user manual*

Bobby Chawla and Oliver Schoenborn



**Defence R&D Canada – Ottawa**
CONTRACT REPORT
DRDC Ottawa CR 2008-312
April 2009

Canada

# Simulation Network Exploitation

*SEDReC station: user manual*

Bobby Chawla
Oliver Schoenborn
CAE Professional Services

Prepared By:
CAE Professional Services
1135 Innovation Dr., Suite 300
Ottawa, ON  K2K 3G7
Contract Project Manager: Leo Roberts, 613-293-8993
W8475-06BM04
CSA: Nacer Abdellaoui, Defence Scientist, 613-998-4582



**Defence R&D Canada – Ottawa**

Contract Report
DRDC Ottawa CR 2008-312
April 2009

Contract Scientific Authority

*Original signed by Nacer Abdellaoui*

Nacer Abdellaoui

Defence Scientist

Approved by

*Original signed by Julie Tremblay*

Julie Tremblay

H/CARDS

Approved for release by

*Original signed by Pierre Lavoie*

Pierre Lavoie

DRP Chair

# Abstract

The Synthetic Environment Distributed Resources (SEDR) Tool Suite is a set of three applications to facilitate knowledge management, accessibility, and repeatability of simulation exercises. The SEDR Control (SEDReC) station is one of those tools, designed for automating the start-up, execution and shutdown of a complex, distributed set of simulation resources that form a simulation system. This document covers how to start, stop, and monitor repeatable experiments run on a network of computers.



# Résumé

Le kit d'outils des ressources distribuées des environnements synthétiques est un ensemble de trois applications pour faciliter la gestion des connaissances, l'accessibilité, et la répétabilité des exercices de simulation. La station de contrôle est l'un de ces outils, conçu pour automatiser le démarrage, l'exécution et l'éteinte et fermeture d'un ensemble complexe et distribué de ressources de simulation. Ce document explique comment démarrer, arrêter, contrôler et exécuter répétitivement des expériences sur un réseau distribué d'ordinateurs.

This page intentionally left blank.

# Executive summary

## Simulation network exploitation: SEDReC station: user manual

**Bobby Chawla; Oliver Schoenborn; DRDC Ottawa CR 2008-312; Defence R&D Canada – Ottawa; April 2009.**

**Introduction or background:** Environment Distributed Resources (SEDR) Tool Suite is a set of three applications to facilitate knowledge management, accessibility, and repeatability of simulation exercises. The SEDR Control (SEDReC) station is one of those tools.

**Results:** This application provides a mechanism for automating the start-up, execution and shutdown of a complex, distributed set of simulation resources that form a simulation system. This allows end-users of the simulation system, as well as non-expert users, to repeat the system execution, find what computers no longer support the system components, and to understand what steps are necessary to execute it. Moreover, the steps can be grouped into a hierarchy to help the user see how different steps, or actions, relate to one another

**Significance:** This document is a user manual for the SEDReC station. It covers how to start, stop, and monitor repeatable experiments run on a network of computers. As such it assumes familiarity with basic concepts related to networking, such as clients and servers, hostnames and IP addresses, transfer of data between hosts, etc. It also assumes some familiarity with the concepts underlying the "running" of programs, i.e. environment variables, command line parameters, command shell (console), etc.

.

# Sommaire

## Simulation network exploitation: SEDReC station: user manual

**Bobby Chawla; Oliver Schoenborn; DRDC Ottawa CR 2008-312; R & D pour la défense Canada – Ottawa; Avril 2009.**

**Introduction ou contexte:** Le kit d'outils des ressources distribuées des environnements synthétiques (SEDR) est un ensemble de trois applications développé pour faciliter la gestion des connaissances, l'accessibilité, et la répétabilité des exercices de simulation. La station contrôle (SEDReC) est l'un de ces outils.

**Résultats:** Cette application fournit un mécanisme permettant d'automatiser le démarrage, l'exécution et la fermeture d'un ensemble complexe de ressources de simulation distribuées. Cela permet aux simples utilisateurs du système, ainsi que les utilisateurs non-experts, de facilement répéter l'exécution de la simulation, d'identifier les composantes qui ne sont plus supportées par le système, et de comprendre les étapes et mesures nécessaires pour correctement exécuter ce system de simulations. En outre, ces mesures sont regroupées hiérarchiquement pour aider l'utilisateur à voir clairement les séquences et les interrelations des différentes mesures, étapes, ou actions, sont inter-liées.

**Importance:** Ce document est un manuel de l'utilisateur pour la station SEDReC. Il porte sur la façon de démarrer, arrêter, contrôler et exécuter des expériences sur un réseau distribué d'ordinateurs et de façon répétitive. À ce titre, il assume une familiarité de base concernant les concepts de réseau, tels que les clients/serveurs, les noms d'ordinateurs « hostname » et les protocoles d'adresse « IP address », le transfert de données entre les différentes machines, etc. Ce document assume également une bonne familiarité avec les concepts pré-requis pour une exécution adéquate de programmes d'ordinateur, tel les variables d'environnement, paramètres de commande, la commande shell, etc.

# Table of contents

Abstract …….. ..... i
Executive summary ..... iii
Table of contents ..... v
List of figures ..... vi
1....Overview of Document..... 1
1.1 Concept of Operation ..... 1
1.1.1 Installation and Setup..... 1
1.2 Intended Audience..... 2
1.2.1 Assumed background..... 2
2....Startup..... 3
2.1 Launched via SEDRA ..... 3
2.2 Command line ..... 3
2.2.1 Command line arguments ..... 4
2.3 Windows Graphical Interface – set up a Shortcut. ..... 4
3....Use .... 8
3.1 Graphical User Interface Overview..... 8
3.2 Browse actions: ..... 9
3.3 View available simulation hosts..... 11
3.4 Execute a "recipe" ..... 13
3.5 Stopping a Recipe..... 14
3.6 Failure Modes..... 14
4....Configuring a Recipe .xml File..... 15
Annex A .. Sample recipe .xml file..... 17
List of symbols/abbreviations/acronyms/initialisms ..... 19

# List of figures

Figure 1: Activation of a Launch Procedure from SEDRA. ..... 3
Figure 2: Creating a shortcut in Windows Operating Systems. ..... 5
Figure 3: Accessing Properties in Windows Operating Systems. ..... 6
Figure 4: Creating a shortcut on the desktop. ..... 6
Figure 5: Editing a recipe shortcut in Windows Operating Systems. ..... 7
Figure 6: SEDReC Graphical User Interface. ..... 8
Figure 7: Launcher Action tree panel. ..... 9
Figure 8: Launcher Action Details panel. ..... 10
Figure 9: Simulation Hosts panel ..... 12
Figure 10: Execution status colours in the Simulation Hosts panel. ..... 13
Figure 11: Launcher Operations panel. ..... 13
Figure 12: Launcher Failure panel. ..... 14

# 1 Overview of Document

The Synthetic Environment Distributed Resources (SEDR) Tool Suite is a set of three applications to facilitate knowledge management, accessibility, and repeatability of simulation exercises. The SEDR Control (SEDReC) station is one of those tools. This application provides a mechanism for automating the start-up, execution and shutdown of a complex, distributed set of simulation resources that form a simulation system. This allows end-users of the simulation system, as well as non-expert users, to repeat the system execution, find what computers no longer support the system components, and to understand what steps are necessary to execute it. Moreover, the steps can be grouped into a hierarchy to help the user see how different steps, or actions, relate to one another.

This document is a user manual for the SEDReC station. It covers how to start, stop, and monitor repeatable experiments run on a network of computers. As such it assumes familiarity with basic concepts related to networking, such as clients and servers, hostnames and IP addresses, transfer of data between hosts, etc. It also assumes some familiarity with the concepts underlying the "running" of programs, i.e. environment variables, command line parameters, command shell (console), etc.

## 1.1 Concept of Operation

SEDReC allows a user to describe a sequence of actions necessary to execute a system of networked programs that interoperate in some fashion. This sequence of actions is referred to as a "recipe", or a "mission". SEDReC has no concept of "simulation" or of how a system's programs communicate; it knows only how to perform actions on various host machines in a defined order. It also knows that these actions can span a finite amount of time and therefore require monitoring of the task for completion (or another condition) and success. Current actions available include transferring files, and starting and stopping a process, though several more are planned. Details of the available actions are described in section 3.

SEDReC is typically started from the SEDRA portal. However it can be run standalone as well, as described in section 2.2. Running standalone would be advisable while developing scenarios, to reduce the overhead of going through the portal GUI for each iteration.

### 1.1.1 Installation and Setup

The SEDReC station is typically installed along with the SEDRA portal. It is designed to run on multiple Operation System platforms (e.g. Windows 2000, XP, and various flavours of Linux), and therefore should possible to install almost anywhere.

The detailed installation procedure is described Synthetic Environment Distributed Resources (SEDR) System Manual.

## 1.2 Intended Audience

This manual is aimed primarily at people who understand distributed simulation, but don't have the detailed experience of a specific simulation.

Having read this manual you should be able to:

1. Start up SEDReC with a given recipe,

2. Identify resource problems before starting a simulation

3. Start, run, and then

4. Stop a simulation.

Certain sections of this manual will also be of interest to recipe developers, such as Annex A which discusses the format of a SEDReC recipe.

### 1.2.1 Assumed background

The readers of this manual should understand the basic concepts of TCP/IP networking, such as hostnames, IP addresses; also some basic information on how to power on computers, how to login, and start processes in a windowing environment.

# 2 Startup

SEDReC is usually launched from the SEDRA portal. When using the SEDReC frequently to design and run an experiment it is handy to be able to run the SEDReC outside SEDRA. This can be done via the command line (section 2.2); though a setting up a shortcut will often be more convenient (section 2.3)

## 2.1 Launched via SEDRA

Typically, the launch configuration is started through the SERDA portal.

When the end-user finds a “active item” simulation resource of interest in the SEDRA portal, they can read it as a formalized description of the steps to take and the environment to use to start-up the associated simulation resource; If this description corresponds to their intentions, then they can let SEDRA transmit it to SEDReC by right-clicking on the simulation resource and selecting “Run Launcher” menu item (Figure 1).

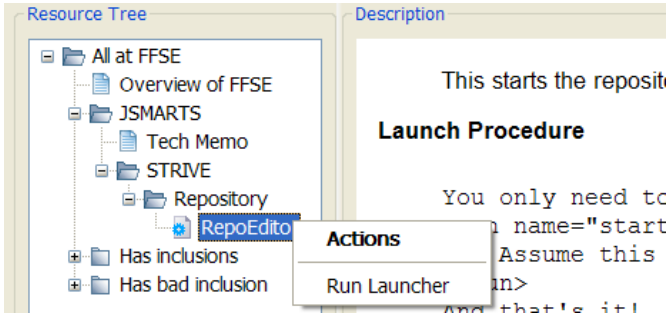


*Figure 1: Activation of a Launch Procedure from SEDRA.*

In this process, SEDRA make a temporary copy of the recipe on the local host, and then launches SEDReC. Note that any modifications of the recipe must be done on the Database-URL where the recipe is taken from.

## 2.2 Command line

It is possible to launch SEDReC without setting up, or starting SEDRA. This could be useful in the initial stages of creating an experiment where the final configuration is changing frequently, and not ready to be published, and also if an experiment is run frequently.

The SEDReC script is called *recipe_station.py* and exists in the root installation directory of your system (e.g. C:\ARP\src). The executable is an interpreted python script, and therefore needs to be invoked via Python. For example:

python recipe_station.py –i your_recipe.xml

### 2.2.1 Command line arguments

- [-i] Specify the Launch Configuration application file. The control station reads a "launch procedure" file on start-up. This file is also called an "action recipe" file. This is an XML file whose elements contain a mixture of text, attributes and action sub-elements. The text forms part of the documentation (together with elements and attributes), but the elements and attributes per se form the executable portion of the document.

- [-itmp] Specify the Launch Configuration application file, and flag it as a temporary file. This is similar to the "-i" option, with the additional action of deleting the file when SEDReC exits. This is the mode used by the SEDRA application whenever SEDReC is launched.

- [-n] Specify the name for this recipe. This name is displayed as the root of the action tree. It allows the user to know what resource from SEDRA is being viewed in SEDReC.

## 2.3 Windows Graphical Interface – set up a Shortcut.

A handy feature of Windows operating systems, such as Windows 2000, or Windows XP, is the ability to create shortcuts to applications, and then to customize the shortcut as desired for your personal use. To do so for the SEDReC utility, find the *Recipe_station.py* file in your installation (e.g. C:\ARP\src) using your browser, and then right-click on the Recipe_station.py, and select "Create Shortcut" (Figure 2).

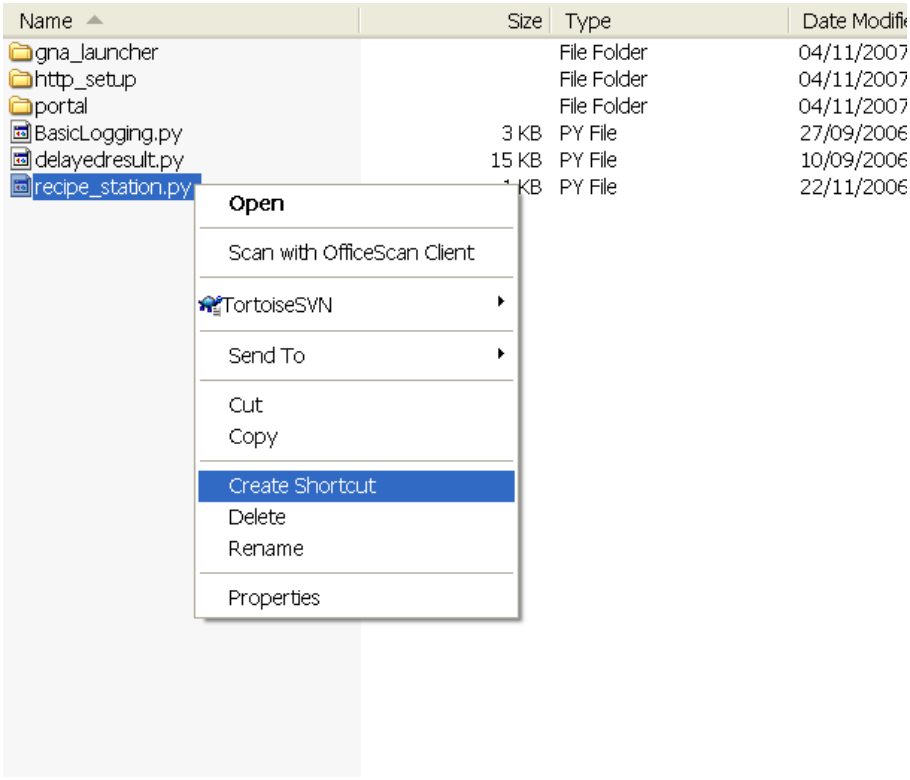


*Figure 2: Creating a shortcut in Windows Operating Systems.*

A shortcut is created with the default name of "Shortcut to recipe_station.py". This can be renamed with a more meaningful name (e.g. launch_Malo_Scenario_017). Now that the shortcut exists, it can be modified to run a custom script. To do so, right click on the properties of the shortcut (Figure 3), and then edit the Target field (Figure 5) from the filename e.g.:

C:\ARP\src\recipe_station.py

to the desired command line, e.g.:

python C:\ARP\src\recipe_station.py –i C:\ARP\src\your_recipe.xml

as described in section 2.2.

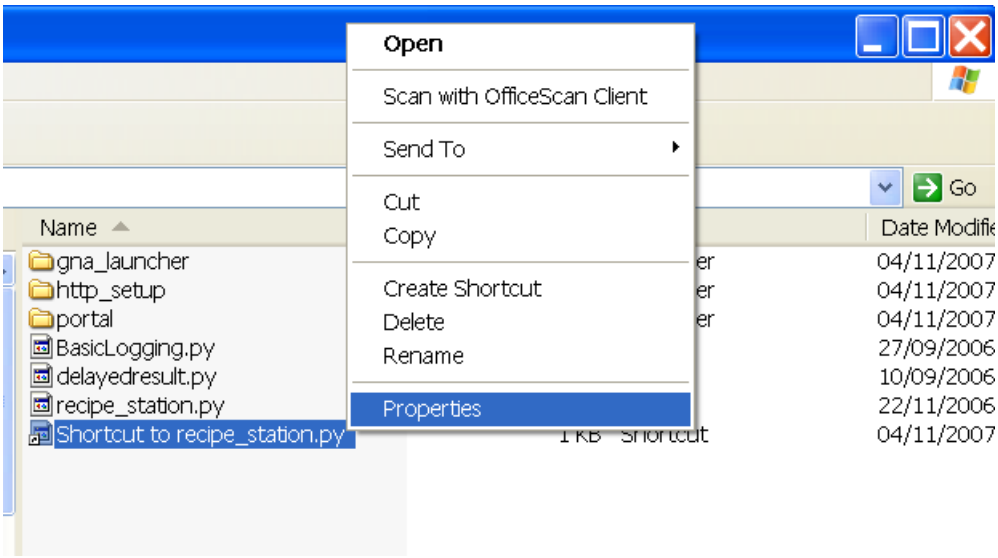


*Figure 3: Accessing Properties in Windows Operating Systems.*

The shortcut can eventually be installed on the user's desktop (Figure 4):

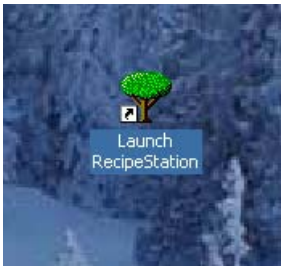


*Figure 4: Creating a shortcut on the desktop.*

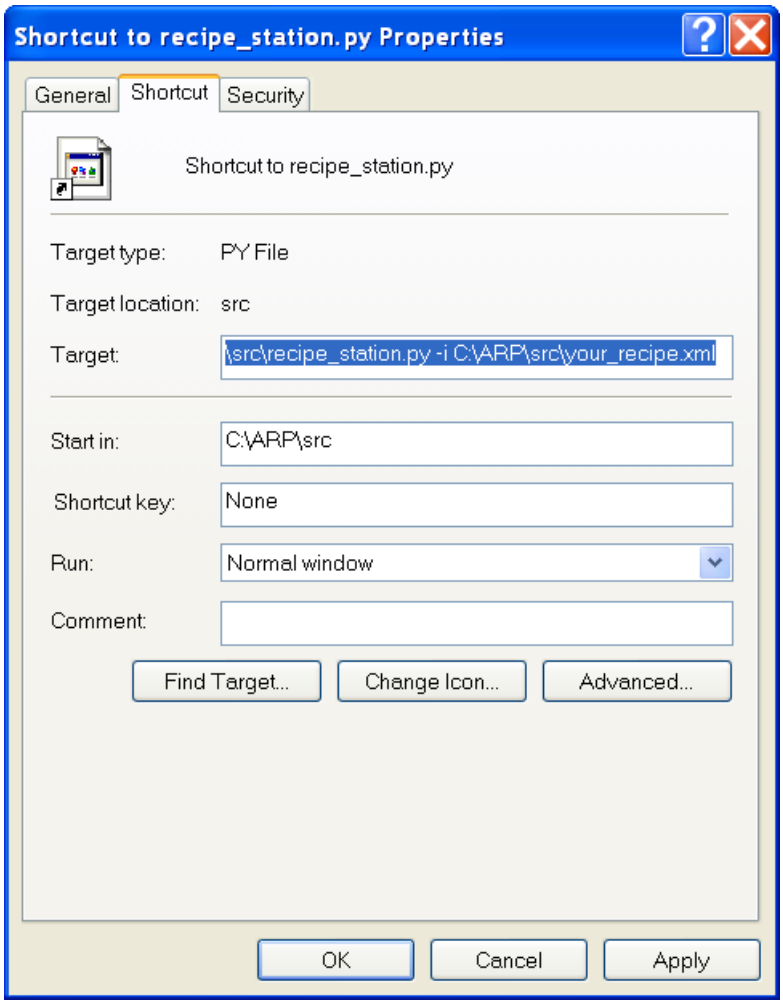


*Figure 5: Editing a recipe shortcut in Windows Operating Systems.*

This allows the user to create an icon, potentially on the desktop, that can be double-clicked to run a recipe.

Similar options exist in Linux operating systems, allowing the user to create an icon that runs a customized command.

# 3 Use

SEDReC's Graphical User Interface (GUI) is illustrated in Figure 6.

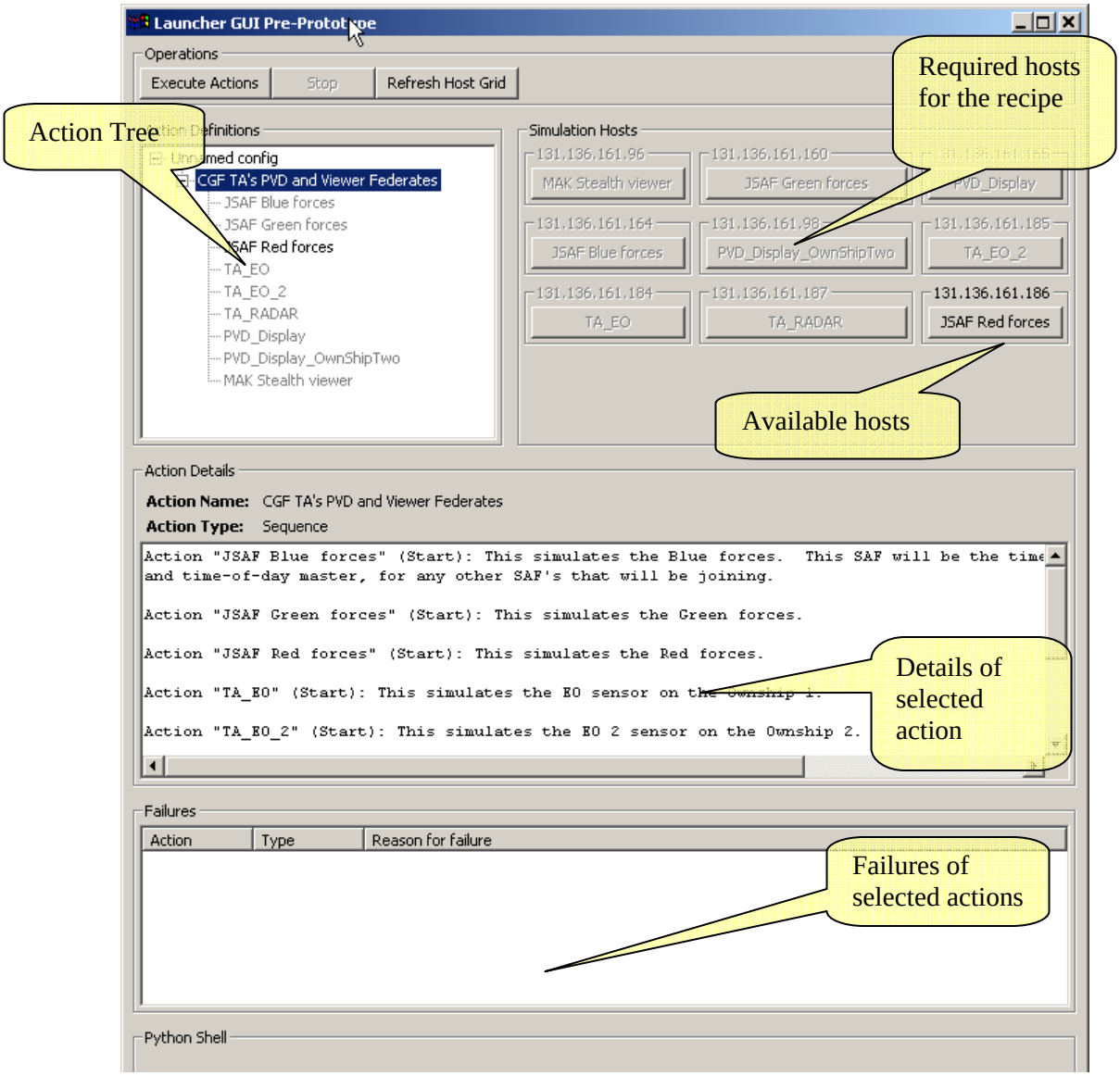


*Figure 6: SEDReC Graphical User Interface.*

## 3.1 Graphical User Interface Overview

The GUI is divided into different sections, as described below:

- a tree view of the recipe;
- a simulation hosts view of all the simulation hosts mentioned in the recipe;
- a detailed description view, where the tree view's selected action's details are shown; and
- a failures view, which shows the actions that failed to execute, and the cause of failure.

## 3.2 Browse actions:

By clicking on an action in the action definition's tree view (Figure 7),

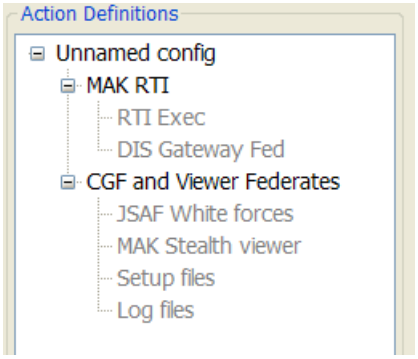


*Figure 7: Launcher Action tree panel.*

it is possible to see the details of the action (Figure 8), including:

- a documented description of the action,
- which host the action is destined for,
- action type,
- attributes used to execute the action.

The action attributes depend on the type of action. E.g. for the « run a program » action, some of the attributes are working directory, environment variables, and name of executable.

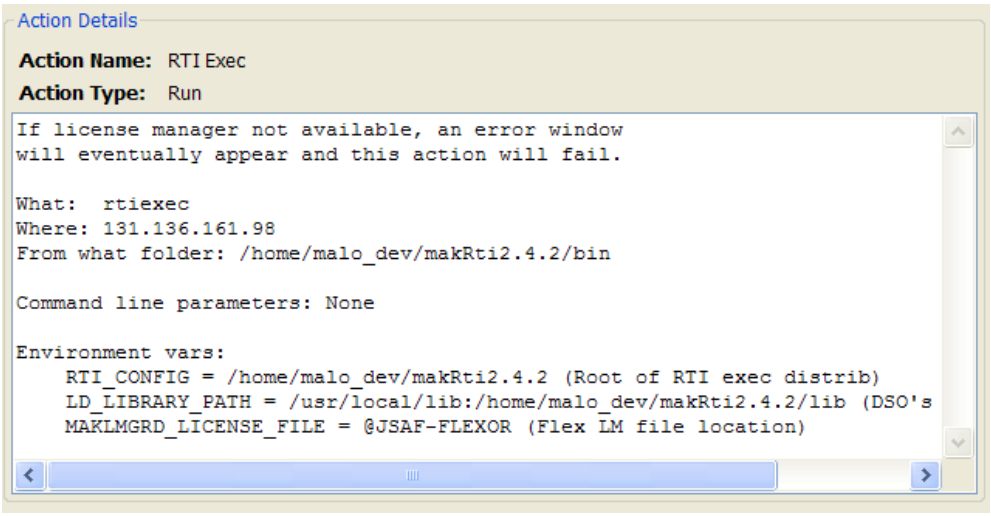


*Figure 8: Launcher Action Details panel.*

After a recipe has been started, any errors associated with the selected action are displayed in the Failures panel (Figure 12 or Figure 6). Failures are discussed further in section 3.6.

The types of actions currently supported by SEDReC include a sequential "run" and parallel "start" action. There are plans to consider adding additional actions: The recipe file supports the following action types:

- *run,* The run action's role is to start a simulation component and wait till it completes. The launcher remains responsive while it is waiting, but in a sequence, this means that the next action is not started until the run has completed. This is different from "start", where the next action starts as soon as the component has started.
- *start,* The start action's role is to start a simulation component. It then monitors its existence and notifies user when the component has terminated, and whether the termination was "normal" or "an error" (e.g. a crash). Depending on the simulation component, distinguishing the two may not be possible, in which case the default is to assume normal.
- *transfer*, The transfer action is to transfer a file between two hosts. The upload and download actions are a special case of transfer, where one of the two hosts is the local host.
- *upload,* The upload action is a special case of transfer, where destination is the local host.
- *download,* The download action is a special case of transfer, where the source is the local host.

The following will be available in future versions of SEDReC:

- *wait on condition,* The wait action just waits for a condition to become true. The condition could be Python code (or other, if the launcher has a different interpreter), or a small custom

language, such that the condition can be fairly sophisticated (involving filtering, listening on sockets etc). A triggered action is essentially a wait plus an execution, where the condition is simply an event having been generated.

- *exec local code,* The exec code action is like a run action, but is interpreted code (Python most likely) that is to be run from the interpreter on the local or a remote host. This includes referring to stand-alone code modules available only on the local host, in which case they get first transferred to the remote host, then executed there. The exec code action is equivalent to writing the code to a file, uploading to a remote host (if necessary), and executing the run action on it, but is easier to manage for small portions of code specific to the recipe. Code that is generic in nature should probably be executed as a script upload + run, or simply installed on each simulation hosts via Python eggs or other method.
- *sequence group*, The sequence group is a sequence of actions, whereas a..
- *parallel group*, The parallel group is a group of actions to execute simultaneously
- *triggered sequence group*, The triggered sequence group is a group of actions, similar to the sequence group, that can be triggered by an event emitted by another action causing a contition to become true.
- *triggered parallel group*. The triggered parallel group is similar to the triggered sequence group, but the actions are executed simultaneously in that group.

As mentioned above, so far, only the “run” and “start” have been implemented.

## 3.3 View available simulation hosts

The “simulation hosts” view displays all of the hosts required to run a “recipe”. No each host box, a series of buttons is displayed corresponding to an action required for the recipe. If a host is unavailable, this is indicated by greying-out the unavailable remote execution buttons (Figure 9).

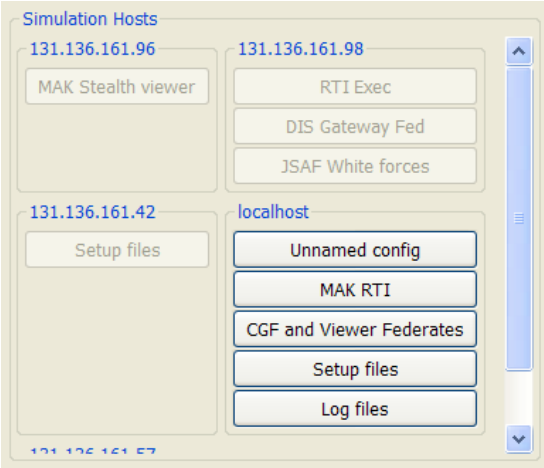


*Figure 9: Simulation Hosts panel*

For a remote execution buttons to be available (i.e. not greyed-out) the remote machine must be available on the network, and also have a Python RPyC (pronounced *are-pie-see*), or **R**emote **Py**thon **C**all implementation server, communicating over a secure transport layer security (TLS) server. Usually the remote machines can be configured to start-up RPyC when booting, and is therefore not necessary to physically access each host to start RPyC server, nor for a recipe.

The list of simulation hosts can be refreshed at any time, except while a recipe is executing. This is accomplished by pressing the "Refresh Host Grid" button in the operations panel (see Figure 6 or Figure 11)

The "simulation hosts" view will eventually allow each host to have a username and password associated, such that this information need only be entered once, and only for those hosts actually used in the execution of a recipe (or part of a recipe). This functionality is not impelemented yet.

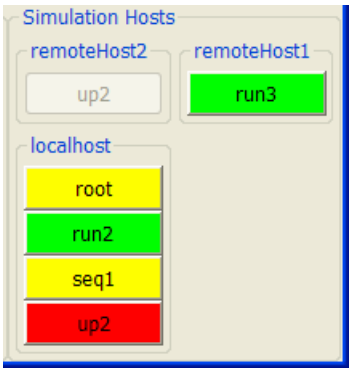


*Figure 10: Execution status colours in the Simulation Hosts panel.*

The simulation hosts panel uses colors to indicate the status of execution of actions (Figure 10):

- *Grey (i.e. background colour):* Action not started yet.
- *Throbbing yellow:* Action started, waiting to see if it will succeed
- *Throbbing green:* Action succeeded, and is still alive
- *Green:* Action succeeded and completed normally
- *Red:* Action failed

## 3.4 Execute a "recipe"

At the top, the Operations panel shows the operations available for the recipe (Figure 11).

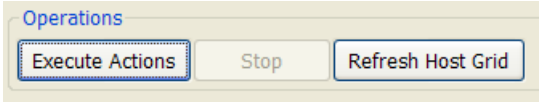


*Figure 11: Launcher Operations panel.*

Typically the Execute Actions button would not be available until all the required hosts are available, in which case it would be necessary to properly startup the missing remote hosts, and then select “Refrech Host Grid”. Nor would it be available if the recipe is already executing, in which case it would be necessary to stop the execution before attempting another one.

## 3.5 Stopping a Recipe

When a recipe is executing, pressing on the Stop button in the Operations panel will terminate the tasks on all the remote hosts in an orderly fashion.

Note that SEDReC has no control over how well a process reacts to being stopped, as there is no standard method of commanding an exit other than brute-force termination. Properly written programs will trap some signals and release resources appropriately. Other programs (e.g. JSAF) have been observed to crash, though this is rare.

## 3.6 Failure Modes

As mentioned in section 3.2, any failures that occur while launching or running a recipe are displayed in the Failures panel (Figure 12).

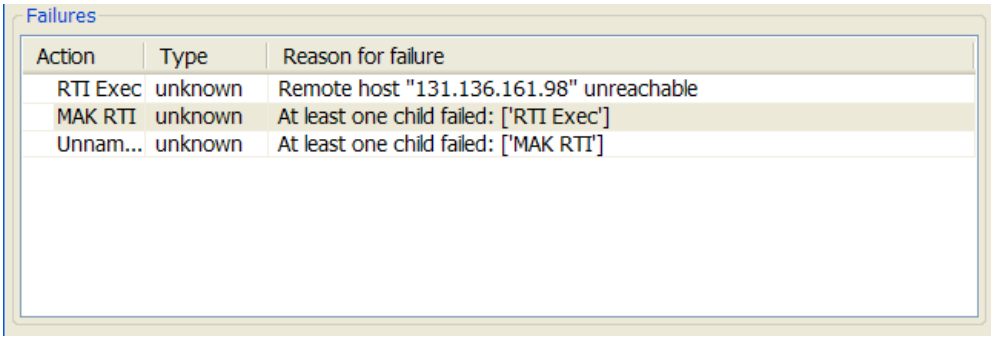


*Figure 12: Launcher Failure panel.*

The figure displays two of several possible types of failures. The main types are:

1. Failure of an action to start; e.g. not being able to communicate with the host on which the action was to take place;

2. Failure of an action because one of its children could not start-up, e.g. the “MAK RTI” action failed because one of its children actions, “RTI Exec”, failed;

3. Failure of an action because the execution of the action failed; e.g. Abnormal termination of an application run as part of the action.

# 4 Configuring a Recipe .xml File.

A recipe is contained in an .xml file which is described via the RelaxNG compact notation (RCN) schema. A sample .xml file is attached in Annex A. The file consists of a tree of <actionSeq>s that two of the supported action types; namely <start> and <run>. The example should be used as a basis for other recipes.

Note that a recipe is intended to be stored in a repository accessible from SEDRA. Some types of Databases also support including the script into the database itself (i.e. file, or html databases). In such cases, the recipe is obtained from the database may be wrapped in other XML required by SEDRA to interact with the database. For example SERDA may require the recipe's <launchProcedure> to be wrapped in a <launchConfig>:

```
<launchConfig>

    <launchProcedure>

        …..actions…

    </launchProcedure>

</launchConfig>
```

SEDRA extracts the recipe from this wrapping before giving it to SEDReC. If you obtain an XML file witch is to be used directly with SEDReC, remove all xml (except encoding) outside of the <launchProcedure>.

This page intentionally left blank.

# Annex A Sample recipe .xml file

```
<xinclusion>
<launchConfig name="Start xterms on various MALO machines">
        <summary><field>
            This is an example launch of some of the MALO
            components. </field>
        </summary>
<launchProcedure>
    This launches a sample application (xterm) on various machines
    in the MALO system.  The first two are <start>ed, and the final
    xterm in the sequence is <run> to illustrate the different options
    in the xml file.

    <actionSeq name="Start some xterms">
        Start xterms on various machines, to see if they launch.

        <start name="Bobby" path="xlogo" onHost="131.136.161.186"
            wdir="/usr/X11R6/bin/">

            <envVar name="RTI_CONFIG" value="/home/malo_dev/makRti3.1.1">
                       Root of RTI exec distrib
                       </envVar>
            <envVar name="MAKLMGRD_LICENSE_FILE" value="@JSAF-FLEXOR">
                       Flex LM file location
                       </envVar>
        </start>

        <start name="Radar" path="xlogo" onHost="131.136.161.187"
            wdir="/usr/X11R6/bin/">

            <envVar name="MAKLMGRD_LICENSE_FILE" value="@JSAF-FLEXOR">
                       Flex LM file location
                       </envVar>
        </start>

        <run name="Paul" path="xlogo" onHost="131.136.161.184"
            wdir="/usr/X11R6/bin/">

            <envVar name="RTI_CONFIG" value="/home/malo_dev/makRti3.1.1">
                       Root of RTI exec distrib
                       </envVar>
            <envVar name="MAKLMGRD_LICENSE_FILE" value="@JSAF-FLEXOR">
                       Flex LM file location
                      </envVar>
            <envVar name="LD_LIBRARY_PATH"
                       value="/usr/local/lib:/home/malo_dev/makRti3.1.1/lib">
                       DSO's for RTI exec
                       </envVar>
        </run>
    </actionSeq>


</launchProcedure>

</launchConfig>
</xinclusion>
```

This page intentionally left blank.

# List of symbols/abbreviations/acronyms/initialisms

| | |
|---|---|
| ARP | Advanced Research Program |
| DND | Department of National Defence |
| DRDC | Defence Research & Development Canada |
| DRDKIM | Director Research and Development Knowledge and Information Management |
| FFSE | Future Forces Synthetic Environments |
| MALO | Maritime Air-Litoral Operations (a project at FFSE) |
| RPyC | Remote Python Calls (package written for interacting with remote computers via Python) |
| SE | Synthetic Environments |
| SNE | Simulation Network Exploitation |
| SEDRA | Synthetic Environment Distributed Resources Access |
| SEDReC | Synthetic Environment Distributed Resources Control |
| SEDReDE | Synthetic Environment Distributed Resources Data Entry |
| XML | Extensible Markup Language (a way of structuring information) |
| R&D | Research & Development |

This page is intentionally left blank

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

CAE Professional Services
1135 Innovation Dr., Suite 300
Ottawa, ON K2K 3G7

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Simulation network exploitation: SEDReC station: user manual

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Chawla, B.; Schoenborn,O.

5. DATE OF PUBLICATION (Month and year of publication of document.)

April 2009

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

30

6b. NO. OF REFS (Total cited in document.)

0

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Ottawa
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

13jb

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

W8475-06BM04

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

DRDC Ottawa CR 2008-312

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The Synthetic Environment Distributed Resources (SEDR) Tool Suite is a set of three applications to facilitate knowledge management, accessibility, and repeatability of simulation exercises. The SEDR Control (SEDReC) station is one of those tools, designed for automating the start-up, execution and shutdown of a complex, distributed set of simulation resources that form a simulation system. This document covers how to start, stop, and monitor repeatable experiments run on a network of computers

Le kit d'outils des ressources distribuées des environnements synthétiques est un ensemble de trois applications pour faciliter la gestion des connaissances, l'accessibilité, et la répétabilité des exercices de simulation. La station de contrôle est l'un de ces outils, conçu pour automatiser le démarrage, l'exécution et l'éteinte et fermeture d'un ensemble complexe et distribué de ressources de simulation. Ce document explique comment démarrer, arrêter, contrôler et exécuter répétitivement des expériences sur un réseau distribué d'ordinateurs.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Modeling & Simulation, Synthetic Environments, Distributed Simulation, Data Management, Knowledge Management, Remote Access, Software

**Defence R&D Canada**

Canada's leader in Defence
and National Security
Science and Technology

**R & D pour la défense Canada**

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale


**www.drdc-rddc.gc.ca**