

## InfoWorks ICMExchange

### What it is?

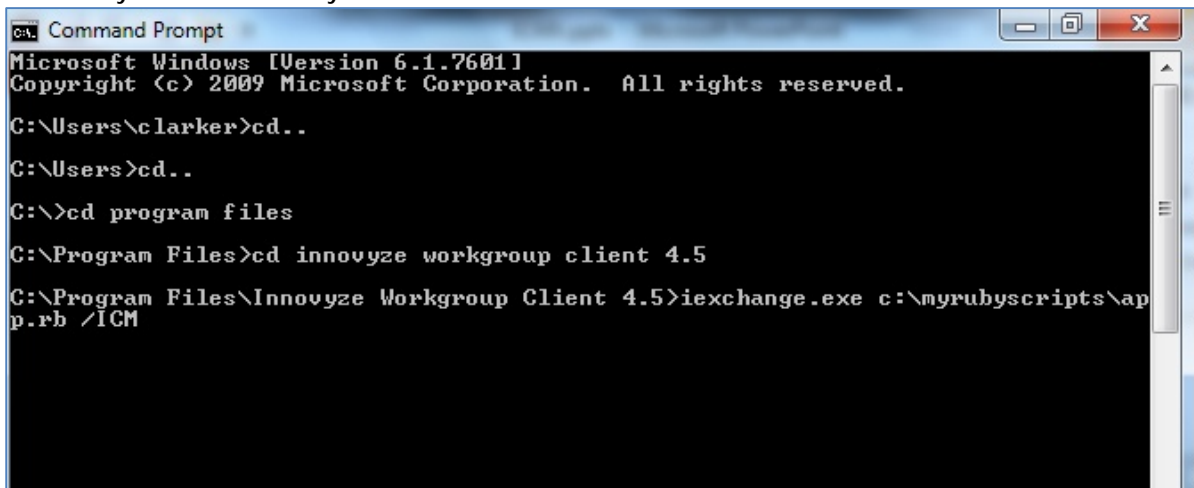
Firstly, it's really important to understand that ICMExchange is not an extra application and is not an add-on module for ICM either. An ICMExchange licence simply activates the Application Programming Interface (API) for ICM, which is hidden by default. Think of it like a back-door into ICM. Once that door is open, software code written by a 3<sup>rd</sup> party can instruct ICM to perform certain steps/operations/tasks without anyone needing to be sat in front of the ICM software itself. In other words, ICM is made to run in the background; it's driven by a 3<sup>rd</sup> party application/process and there is nothing to see from the end-users prospective.

With this final point in mind, you'll appreciate that providing a standard software demo, of the type we would typically do for applications like InfoWorks or InfoSWMM, is not a viable proposition. The whole point about ICMExchange is that there's nothing to see while it's running/working. When driven via the API, ICM is no longer a desktop application, it's just becomes a background task in a much bigger process.

### How it works / what it can do

With the API opened up, ICM is controlled by writing code in the form of Ruby Scripts. The programmer would need a high degree of familiarity with Ruby and with the terminology of object oriented programming. ICMExchange would typically be instigated by running it from the command line, or via a batch file. The Filename of a Ruby script is passed as an argument and the Script will then be run and have access to the Scripting API's classes and functions. From the command line the instruction looks something like this e.g. "C:\Program Files\Innovyze Workgroup Client 4.5\iexchange.exe" c:\myrubyscripts\icmnetwork.rb /ICM.

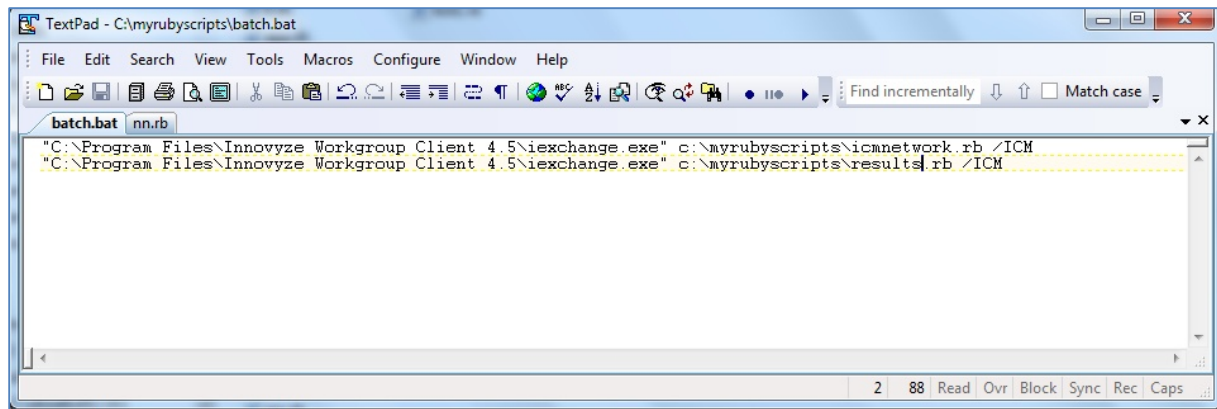
This is all you would actually see!



```
CA: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\clarker>cd..
C:\Users>cd..
C:\>cd program files
C:\Program Files>cd innovyze workgroup client 4.5
C:\Program Files\Innovyze Workgroup Client 4.5>iexchange.exe c:\myrubyscripts\ap
p.rb /ICM
```

The next stage is to use a batch file. This allows ICMExchange to be run automatically as a scheduled job (for example daily). The job could include one ruby script or multiple scripts. If run as multiple scripts each will run sequentially i.e. the next one won't start until the last one has finished.



### What can be done with Ruby Scripts?

Ruby Scripts may be run from:-

1. The user interface of InfoWorks ICM and InfoNet.
2. The separately licensed ICMExchange product, which run from the command line without displaying a graphical user interface.

Whilst there is a great degree of overlap between the functionality available when running scripts from the user interface and that available within the ICMExchange product, some functionality is only available within one or the other.

The distinction between the two is broadly speaking as follows:

1. Within the user interface, scripts are run when a GeoPlan is open displaying a network, possibly with a guest network loaded. The scripts may manipulate the data in the network, perform imports and exports using the Open Data Import and Export centres, commit and revert changes, and display some simple user interface features. The scripts may not manipulate tree objects except selection lists, and may not open and close databases or set up and run simulations.
2. From the ICMExchange application, users have the extra capability to manipulate database tree objects, create databases as well as setting up and running simulations. However the user interface items available from the ICM or InfoNet User Interfaces will not be available.

In summary ICMExchange can do all the following operations as background tasks:-

- Edit, update and query database tree objects
- Create, copy and delete database tree objects
- Create network scenarios
- Commit and validate networks
- Edit network objects
- Carry out simulations
- Query results and extract data

ICMExchange does not give access to all the tools and menus items that are available in InfoWorks ICM itself and does not interact with the ICM-TSDB or Create/Export graphs and maps.

### Licence / Price Model

To use the ICMExchange API you need to purchase a licence. The door to open up the API applies to any given instance of InfoWorks ICM. If a licence of ICM does not have ICMExchange capability available, it won't accept any commands. ICMExchange is available as part of the InfoWorks ICM Executive Suite.

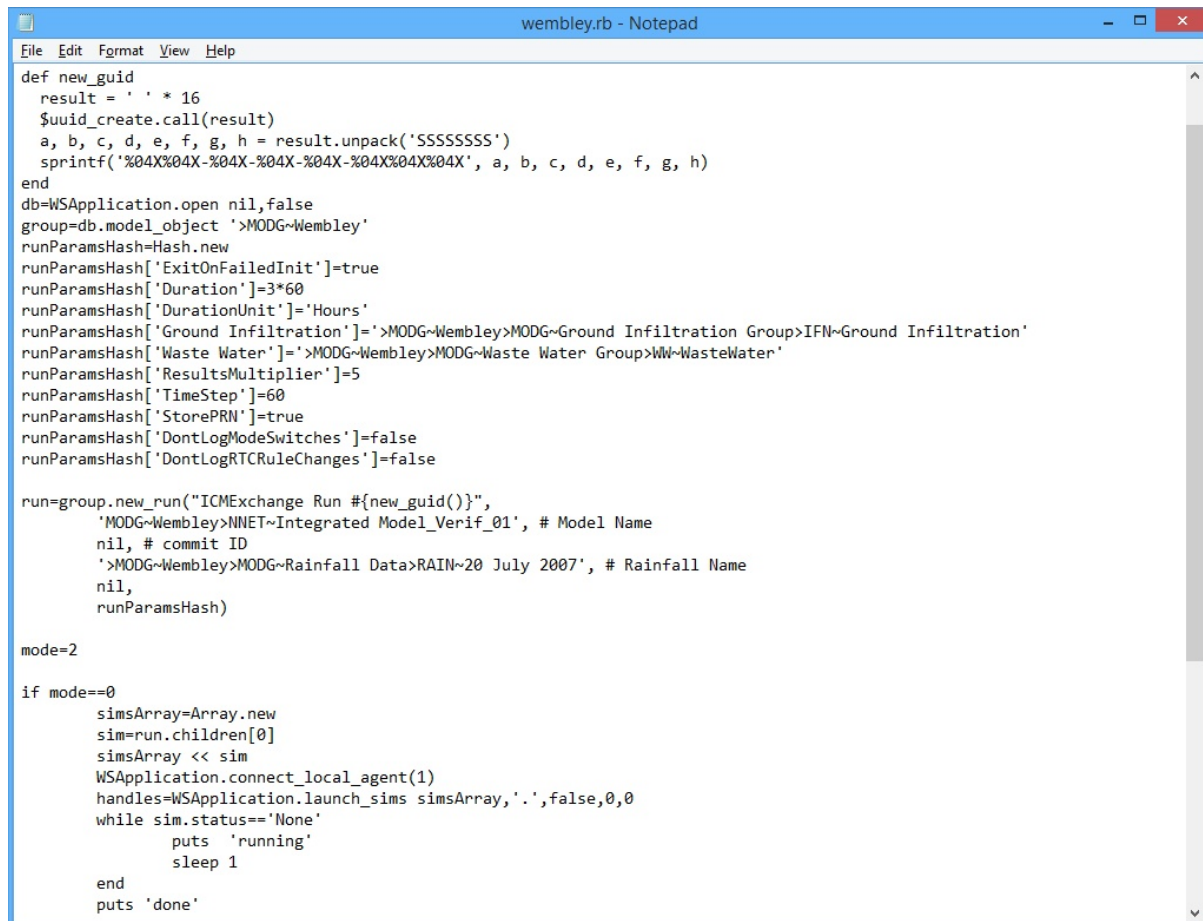
## Help / Advice / Support from within Innovyze

The Ruby interface for InfoWorks ICM and InfoNet is common to both applications, but licenced independently. Clients needing help with creating the scripts to drive InfoWorks/InfoNet can call upon the Innovyze Support Team for a fee. Behind the scenes, various members of the Development Team can provide additional help and advice on the object oriented programming terminology (e.g. 'class', 'method', etc., etc.) that are key to creating successful scripts.

Users must note that the flexible nature of the Ruby scripting language means that the language is almost infinitely flexible, the embedding of the language within the InfoWorks ICM and InfoNet applications is intended primarily for the manipulation of data via the product's APIs. It may not be possible to get other Ruby libraries working within InfoWorks or InfoNet and it will not be possible to provide support for such attempts.

## Example Ruby Script – Running a Simulation

The script below is a simple example showing how to select a network and various time-vary files from the database and define the parameters for a Simulation.



```
wembley.rb - Notepad
File Edit Format View Help
def new_guid
  result = ' ' * 16
  $uuid_create.call(result)
  a, b, c, d, e, f, g, h = result.unpack('SSSSSSSS')
  sprintf('%04X%04X-%04X-%04X-%04X-%04X%04X%04X', a, b, c, d, e, f, g, h)
end
db=WSApplication.open nil,false
group=db.model_object '>MODG~Wembley'
runParamsHash=Hash.new
runParamsHash['ExitOnFailedInit']=true
runParamsHash['Duration']=3*60
runParamsHash['DurationUnit']='Hours'
runParamsHash['Ground Infiltration']='>MODG~Wembley>MODG~Ground Infiltration Group>IFN~Ground Infiltration'
runParamsHash['Waste Water']='>MODG~Wembley>MODG~Waste Water Group>WW~WasteWater'
runParamsHash['ResultsMultiplier']=5
runParamsHash['TimeStep']=60
runParamsHash['StorePRN']=true
runParamsHash['DontLogModeSwitches']=false
runParamsHash['DontLogRTCRuleChanges']=false

run=group.new_run("ICMExchange Run #{new_guid()}",
  'MODG~Wembley>NNET~Integrated Model_Verif_01', # Model Name
  nil, # commit ID
  '>MODG~Wembley>MODG~Rainfall Data>RAIN~20 July 2007', # Rainfall Name
  nil,
  runParamsHash)

mode=2

if mode==0
  simsArray=Array.new
  sim=run.children[0]
  simsArray << sim
  WSApplication.connect_local_agent(1)
  handles=WSApplication.launch_sims simsArray, '.',false,0,0
  while sim.status=='None'
    puts 'running'
    sleep 1
  end
  puts 'done'
```