White Paper

General Public Use

G/On

Enable Applications

Version 5

Make Connectivity Simple

Preface

This document describes the G/On Admin application, and how it enables the administrator of G/On to let different users run different programs, both local and remotely using G/On. It explains the concepts of Users, Groups, Menus, Zones, Actions and Application strings as they are used in G/On Admin.

Briefly, G/On Admin is the tool for administering G/On. It enables administrators to control which applications, with which parameters, can be launched with the G/On Client, by who and where. It's easy to use, yet offers great flexibility on both the administrator and hotline level.

It offers advanced administrators the possibility to make very flexible applications strings, which will then enable system users to publish applications using a simple point'n'click interface.

The combination and flexibility of the Group/Zone/Menu system makes it possible to fine tune, who gets which applications where, on both an organizational and per user level.

Users and groups

The basic user in G/On is just a login name, and information on how to validate the user's login. The user can either be validated against an Active Directory or against a local password stored in G/On. Users can either be created in the G/OnAdmin application or synchronized into the system from an Active Directory using the Usync program.

User groups are a simple collection of users - either created during the synchronization process or in the G/OnAdmin program.

It's important to notice that users and groups are either synchronized with the AD or managed in G/On Admin. It's possible to administer users from AD in G/OnAdmin, but we advise against doing this, as changes could get lost at the next Usync run.

A group is basically a list of members, and a possible default menu (with possible zone limitations) associated with that group.

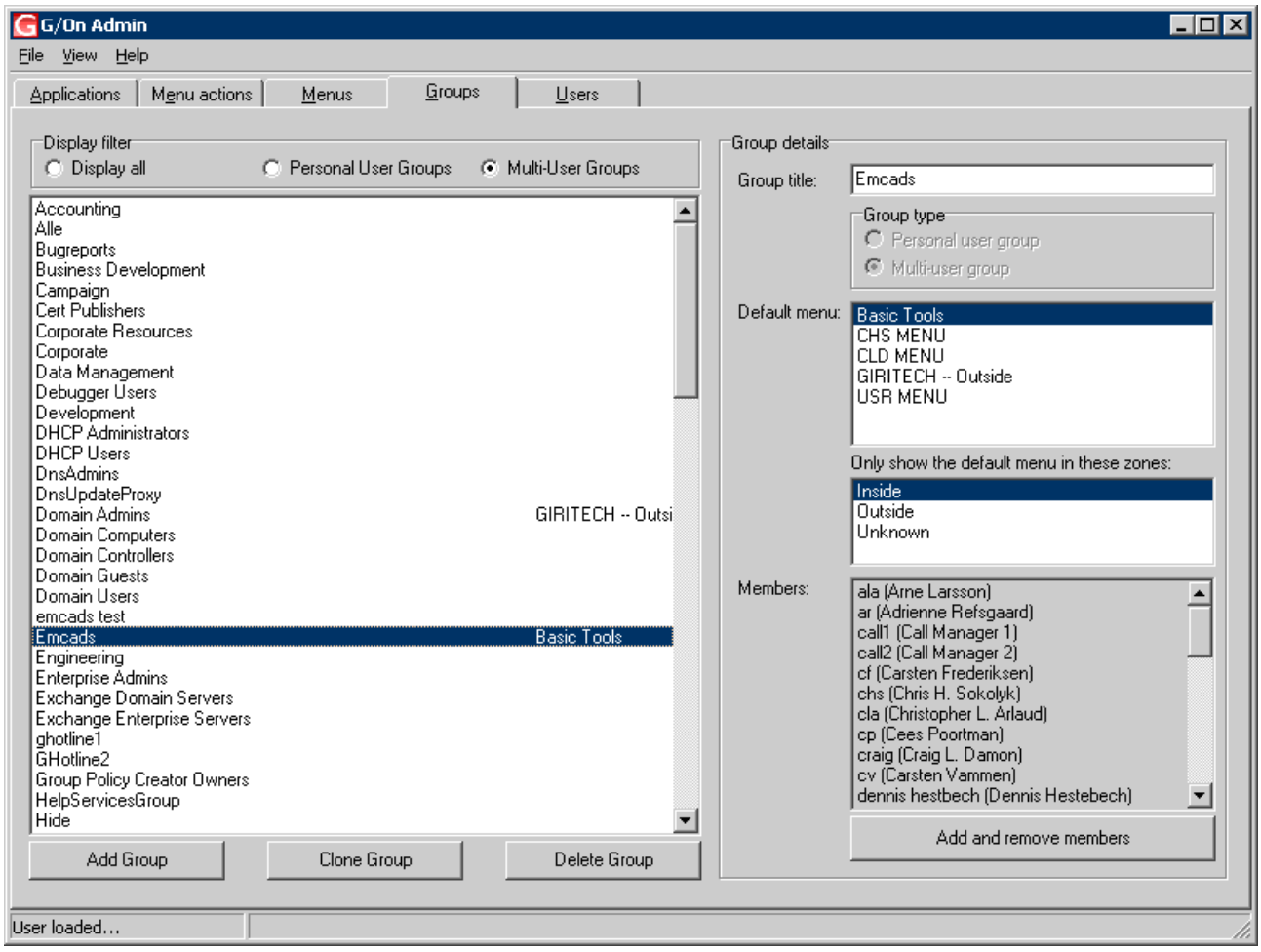


Illustration 1: Basic group setup as shown in G/On Admin

Menus

A menu is a collection of one or more actions to run on the G/On client. Depending on group membership, a user can be presented with a menu that's build from one or more sub-menus. If a user is a member of several groups, each of them having a default menu (for the current Zone) the user will get all the menu items for all of the menus. Menu items will be fitted into sub-menus, if so designed.

Zones

A Zone is a tool for modifying which menus are given to users, depending on just about any client criteria the administrator wants to filter on. The default configuration of G/On suggest using: Inside, Outside and Unknown zones. The simplest rule for enabling zones could be defined something like “if client-ip-address in range 192.168.0.* enable inside zone”. This would allow the administrator to make a special menu with applications that you are only allowed to run when you are inside the firewall. The rules enabling zones can be based on just about any imaginable information from the client, from OS version, ip address, EDC serial, G/OnClient version, etc.

Actions

Actions are what you use to build menus. An action launches an application with a given set of parameters. Application String + Action Parameters = Menu Item.

Application strings

Application strings are at the core of the application launching abilities of G/On. Everything starts with an application string. The basic application string simply tells the client to launch something. An example of this could be an application string that launches Notepad.exe on the client. An advanced Application string could launch a multi port gateway, and a Remote Citrix application depending on which action parameters the administrator has specified for that menu action. Or just about anything else that communicates over tcp/ip and/or UDP.

An example of a really simple application string:

```
9;c:\notepad.exe;c:\readme.txt
```

This is just about the simplest application string you can make. It's a type 9, which simple means launch this application with these parameters. Note that the application string sections are separated by semicolons.

Type	Connector	No. of Parameters	Description
4	Legacy Terminal Services connector	9 parameters	This is the Giritech wrapper for Microsoft Terminal Services. It starts GRDPClient.exe on the G/On client PC and connects it to the server you specify
5	Legacy Citrix connector	9 parameters	This is the Giritech wrapper for Citrix Services. It starts GICAClient.exe on the G/On client PC and connects it to the server you specify. Note: Only Citrix version 3, feature release 3 is supported.
8	Generic TClient connector	11 parameters	Basic application launcher that will set up a port forwarding service and launch an application that uses that port/those ports. Can also be used for Terminal Services and Citrix.
9	Application launcher	2 parameters	Launches applications with corresponding parameters. Meant to be used after launching a gateway
10	Application Connector– multiple ports	?	Same as 8, but with multiple ports.

Lets take a looks at a type 10 with a simple parameters in it.

```
10;%CITRIX_SERVER%;1494:tcp#1604:udp#2598:tcp;;;ICA TCP/UDP Gateway;ICA
TCP/UDP Gateway;;;;True;False
```

Notice the %CITRIX_SERVER% parameter. Anything surrounded by % characters is treated as a parameter to be replaced with information somewhere in the application launching process. When you build an action item you'll be prompted to enter “CITRIX SERVER” information. So this string can be used again and again to build gateways to different Citrix servers, as you can specify the actual server to use at the menu Action level.

What this string does is set up a gateway that forwards the ports that Citrix needs to run. In the example above, it will forward the tcp/ip 1494, udp 1604 and tcp/ip 2598 to the server specified for the %CITRIX_SERVER% parameter in it's action. The rest of the parameters are naming and different security settings. Note that this string doesn't launch anything in itself. It simply sets up a gateway.

With a gateway up and running (it is possible to get it to auto run, when the client/user gets a menu) we could launch a remote Citrix application, with a new menu action based on an application type 9.

```
9;%VENDORPATH,noedit%\wfica\wfcrun32.exe;"%VENDORPATH,noedit%\wfica\%ICA_NAM
E,forceselect[outlook|powerpoint|word|excel]%.ica"
```

```
/username:%USERNAME,noedit% /password:%PASSWORD,noedit%
```

This type 9 is just like the first notepad launching application string we looked at. It contains a series of different parameters that will get replaced during the launching process.

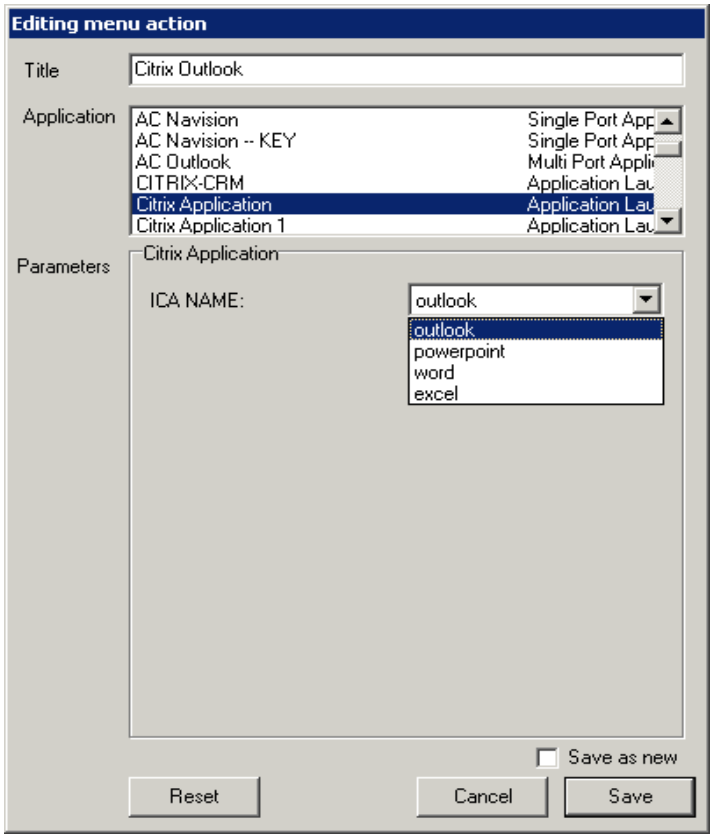
The first two parameters are %VENDORPATH,noedit% - the vendorpath parameter is a special one that gets replaced in the client with the path to the vendor area on either the USB key or the G/On Desktop installation. The “noedit” property of the parameter tells the GOn/Admin program that this is a special parameter and that the administrator shouldn't be asked for this value during the building of the Menu Action.

So the exe will be launched with something like:

> C:\Program Files\GOn Desktop\Applications\wfica\wfcrun32.exe for the desktop client and G:\wfica\wfcrun32.exe for the USB (if the vendor partition of the USB got mapped to G:\ )

The third parameter, %ICA_NAME,forceselect[outlook|powerpoint|word|excel]% is used to replace part of the parameters string that the wfcrun32.exe program is started with. We want to launch an .ica file, but we want the application string to be flexible, so we add a parameter that must be filled out during the Menu Action creation process.

The menu Action dialogbox for the above string will look something like this:



Notice how a dropdown list has been created with the options specified in the application string parameter, forcing the administrator to select one of the options (because of the “ForceSelect” property of the parameter).

This allows us to create a series of Menu Actions with the same application string. In this case the application string is limited to the basic Microsoft Office applications, as these are, in this case, the ones we want to publish and they are the ones we distribute .ica files for.

The last two parameters %USERNAME% and %PASSWORD% are, just like %VENDORPATH%, special client-side parameters, that will get replaced with the information the user used to login to the G/On Client. In this case it's used for Citrix Single Sign-on, so that the user doesn't have to specify user name and password every time they launch a Citrix Application.

There really is no limit to how simple or complex you can make your application strings. You can make them with no parameters, which will lock them to one possible action, you can lock them to a simple series of options like the ones in the given Citrix example, or you can open them up and build them totally of parameters, all of them configurable on the Menu Action level. Or any level in between, depending on how much power you want to give to different kind of users of G/On Admin.

User levels

G/On Admin is designed to accommodate users on different levels – from first level support to advanced administrators who wants to design new application strings. G/On Admin has four built-in user levels:

1. **Read Only User**: Can search for users and see their group membersships and menus. This will often be enough to answer basic question from users, like why they don't have certain menu items (they may be in the wrong group or zone).
2. **Hotline 1** – (user group:ghotline1): This user can add and remove users from groups and edit basic user information, including enabling and disabling EDCs (Client Identifiers/USB Keys), can add and remove predefined menu actions to/from menus.
3. **Hotline 2** – (user group:ghotline2): Same as Hotline1, plus the rights to create new Menus Actions, using pre-defined Application strings.
4. **Admin** (user name and password specified in GonBuilder): Everything including making new application strings.

Pre-defining flexible and consistent application strings is essential for enabling hotline users and lowlevel admins to publish and control remote applications to users, based on group membership and zones.

Appendix A
Technical walk through of internal G/On Application string usage

An application string is, not surprisingly, just a string. It can contain any ANSI character, with only a few characters treaded specially. All strings are separated into sections with semi-colons (;). The number of sections depends on the action the string is associated with (see the G/On Admin Manual for further information on this). No special code page treatment takes place, as the string is mostly treated as is. This also means that no escape characters are defined and it's impossible to use semicolons inside the strings sections.

This is what happens to the strings from its raw form until it's used to actually launch some kind of application:

Server	1. Raw string 2. Action Data gets added 3. User Data gets added 4. Action parameters gets used to build the string into the menu (move to root, auto launch, etc) 5. Gets sent to the client as a menu action string
Client	1. Local parameters gets replaced with local variables (%SERVER%, %PASSWORD%, %GONPATH%, etc) 2. Registry parameters get replaced.

The simples form of application string is a simple string with everything hardcoded into it. This means that nothing really happens in the above steps, as no parameters have been used and all paths and addresses are already specified in the string. It's certainly possible to run your G/On system like this, but it's only recommended for the smallest of installations.

Application string parameters

Application string parameters are the way to extend application strings and make their usage flexible.

Application strings parameters come in three different flavors: Action-, User- and Client-data parameters.

The Action Data Parameter is used to build the action editor dialog box. Adding a %SERVERNAME% parameter to a string will ensure that the Action Editor dialogbox will have a field called “servername” where the user can enter a servername before the action is assigned to a menu.

The User Data Parameter, is not for editing, and will normally be specified with a “noEdit” property, i.e. %user_fullname,noedit%, as this automatically gets replaced by the system, and should not be displayed in the Action Editor dialog box.

The Client Data Parameter is also not for editing in the Action Editor dialog box, as the data

automatically gets replaced at the client. An example could be: %gopath,noedit%.

The User Data Parameter and the Client Data Parameter are fairly easy to use and should just be specified with the % characters. If you want to make sure that the parameters isn't edited in the Action Editor dialogbox, you should add a ,noEdit propertie, ei: %goPath,noEdit%.

Action Data Parameters are used to build the Action Editor dialog box, and can be made fairly complex. You can find more information on how to build an Action Data parameter in the G/On Admin Manual.

How Application String Parameters are parsed

The Application string parsing function is described in the follow bit of pseudo code. This function is used both to create the Action Editor dialog-box in G/On Admin, and to find parameters to replace with Action Data on the server when a menu is built for a user.

1. Find all parameters strings. A parameter is simply anything between two %.
2. For each parameter do:
    1. Replace parameter in Application String with string that's easier to replace later (%APPPARAMNO:guid%, where guid is a random number).
    2. Parser the parameter string.
        1. Remove the value list part by looking for a [and a]. If values exist:
            1. Split values with bar | character.
                1. For each value, split it on ',' and
                2. see if the second value is "default", if that's the case sets default value to this one
        2. Set properties to default values (false)
        3. Split parameter properties with ',', for each property do:
            1. if it's the first property, it's the name, set the parameter name equal to it
            2. if property = "ForceSelect", set forceselect to true
            3. if property = "NoEdit", set forceselect to true
            4. if property = "MustEdit", set mustedit to true
            5. if property = "NoBlank", set NoBlank to true
    3. Set the new Value
        1. Set newValue equal to "%" + name + "%"
        2. If MustEdit or ForceSelect is true
            1. if there's a list of values
                1. if there's a default value use that
                2. otherwise use the first one.
            2. If there's no list of values
                1. if NoBlank is true, set new values to "*** N/A Value ***"
                2. otherwise set it to blank.

This leaves us with a list of parameter objects and an application string with places holders for putting the new values back into the string.

If we are using G/OnAdmin the list of parameters objects is used to populate the Action Editor dialog-box. The dialog box builder function simply runs through the list of parameter objects and creates an onscreen edit field for all objects where NoEdit is false. When the Action Editor dialog-box data is saved, the value in the edit field is saved under the parameter name.

When the application string is parsed as part of the creating of a menu-string on the server, the parameter action values are loaded into a list that is matched against the list of parameters from the application string. The first parameter found in the application string, with the same name as the action parameter name, gets the value associated with the action parameter.