

Data Administration

Provisioning IceWarp Server Using the API

An IceWarp User's Guide

October 2008

www.icewarp.com

Understanding and Using the IceWarp API via Tool.exe

Administrators have the ability to make bulk changes to users, domains and system settings using the integrated IceWarp API via tool.exe. They can also use it to change display settings, to upgrade the server, or to programmatically access any server module.

Any module, option, or field in the IceWarp server can be modified, enabled, or disabled using tool.exe. Tool.exe provides more flexibility in user management. Whereas most API calls are seen as options in the GUI and can be selected one at a time via distinct tabs, tool.exe gives administrative control over all options via a single command.

Administrators can use the tool to export the full list of current settings used in a server. This is useful if server migration ever becomes necessary; tool.exe makes certain that settings remain the same.

In addition, the API crosses platforms as Windows and Linux use the same API calls.

How to use tool.exe

The administrator must first open a CMD prompt in Windows, then navigate to the IceWarp installation folder on the server. (Linux: No prompt is needed.)

To use it, administrators must type “tool” within IceWarp's installation folder.

Variables will appear along with objects that can be used in conjunction with the tool when building the command.

```

Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Merak>tool
tool v3.7 for IceWarp Merak Mail Server
Copyright (c) 1999-2008 IceWarp Ltd. All rights reserved.
E-mail: info@icewarp.com

Commands (help: tool <command>):
create/new/add <object> - Creates new object
delete <object> - Deletes an object
modify/set <object> - Modifies object variables
display/get/list <object> - Displays object variables
check/validate <object> - Validates object data
export <object> - Exports object variables in CSV format
import <object> <filename> - Imports objects in CSV format from file
upgrade <version> - Calls upgrade functions from previous version
search <keyword> - Search API for string

Objects:
system - <variables>
domain - <domain> <variables>
account - <account> <variables>
remoteaccount - <account> <variables>
service - <service> <variables>
userstatistics - <fromdate> <todate> <filterstring>
tables - <service> <connectionstring>

```

Administrators can run the tutorial by entering the following command:

Tool – tutorial

Building the Command Syntax and Usage

Tool.exe command syntax must contain a Command and an Object.

For instance:

Tool (Command) create (Object) account API Call ‘variable’

In this example, “create” is the Command and “account” is the Object. This format will be needed whenever the tool syntax is built. Below is a list of all Commands and Objects.

Commands	Objects	Object Usage
Create/New/Add	System	For any <i>system</i> option
Delete	Domain	For any <i>domain</i> option
Modify/Set	Account	For any <i>account</i> option
Display/Get/List	Storage	<i>Storage</i> related options

Check/Validate	UserStatistics	<i>Statistics</i> related options
Import	Tables	<i>Database</i> related options
Export	Backup	Backup options
Search	RemoteAccount	Remote account options
Upgrade	Service	Service related options

In order to utilize API calls to the server, an administrator can match the Commands and Objects from this table. For instance, administrators that want to change the mailbox limits for all users under a specific domain, can use the following:

Tool modify account *@domain.com u_maxboxsize 25000

This single command will change the mailbox size of every user under domain.com to 25MB, and inform all account owners. Attempting this in the GUI would require manually clicking on each individual user and changing the respective option for each. This is naturally an onerous and tedious task for domains with multiple accounts.

Working with the Tool Commands

The following is an overview of the syntax required for building commands and how they can be formatted. Refer to the `{install_folder}/api/Delphi/apiconst.pas` file for all of the IceWarp API calls.

- To display all system settings using a wildcard:

tool display system *

- To export all system settings and values to a file:

tool display system * > path to file

- To change the port SMTP uses for SSL:

**tool modify system C_System_Services_SMTP_SSLPort
(Port Number)**

- To request the number of messages caught by the content filters:

Tool display system system st_failedcf

- To export all users from the server with their passwords, full names, account type, and mailbox limits:

**tool export account *@* u_password u_name u_type
u_maxboxsize >File path**

When *exporting* users from a server into a file, the administrator must first note the Commands used and the order in which they were used.

The Command for *importing* the users back into the server is:

**Tool import account filename u_password u_name
u_type u_maxboxsize**

Command variables (u_password, u_name, etc.) are in the same order they were originally exported. Administrators should follow this rule each time they export and import users. If the variable pattern is different than when originally exported, data will appear in the wrong fields. For instance, *name* as *password*, *mailbox limit* as *name*, etc.

- Creating a new domain:

Tool create domain domain.com

- Editing the maximum message size for all domains:

Tool modify domain domain.com d_usermsg (Number)

The tool can connect to remote servers, run commands on them, and gather information from them.

- To connect to another server:

**Tool -r =user,pass, server:port,["SSL " if used]
[options] [Command] [Object] [Parameters]**

**Tool -r =admin,password,mail.domain.com:32000
create domain domain2.com**

Only the Administrator Account login credentials are required to access a remote server and perform commands using the remote connection string outlined here.

Uses for tool.exe and API

Administrators who understand how to manipulate the command structure and know where to find all of the API calls can start putting them to good use.

For instance, commands can be used in web applications for reporting purposes. We see many instances where these commands are used in web applications for reporting purposes.

Using the API, it is easy for administrators to make custom scripts that interface with IceWarp Server. This gives administrators the means to enhance their infrastructure, web applications, and entire processing when building on top of the product.