11.02.2022

RealMigrator

Product Overview

glueckkanja▪gab

Overview

glueckkanja gab

RealMigrator

Architecture overview

RealMigrator Backend

(SaaS)

RealMigrator Agent

(Windows Client or Server)

Microsoft 365

Control & Inventory
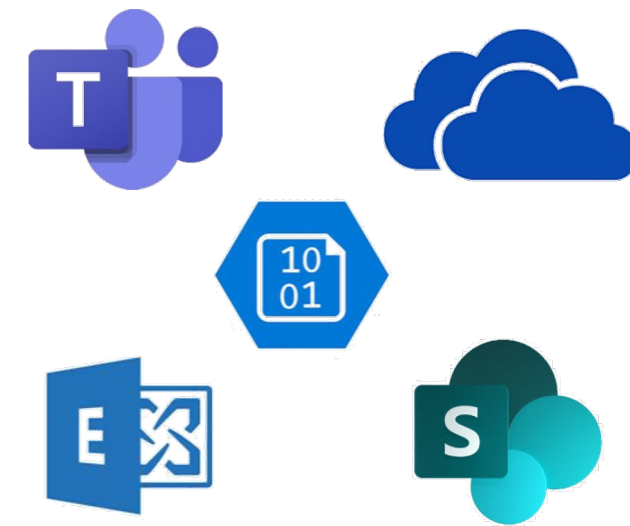
Data Migration

RealMigrator Web Console

PST supported for Windows clients only
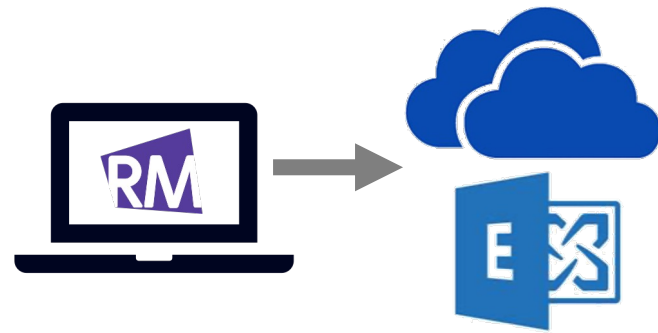
Inventory use cases

Hardware

Software

Configuration (e.g. printer & network drives)
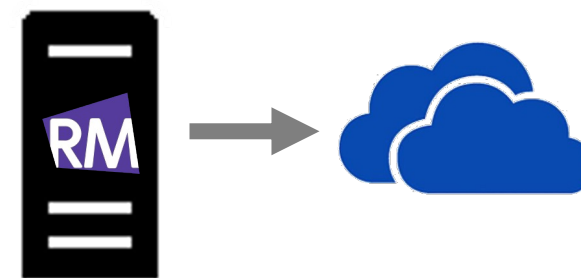
Data (Files + PST)
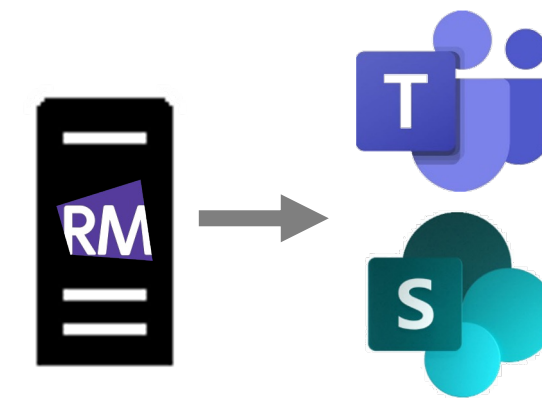
Migration use cases

Client

— Personal Data

— Files & PST

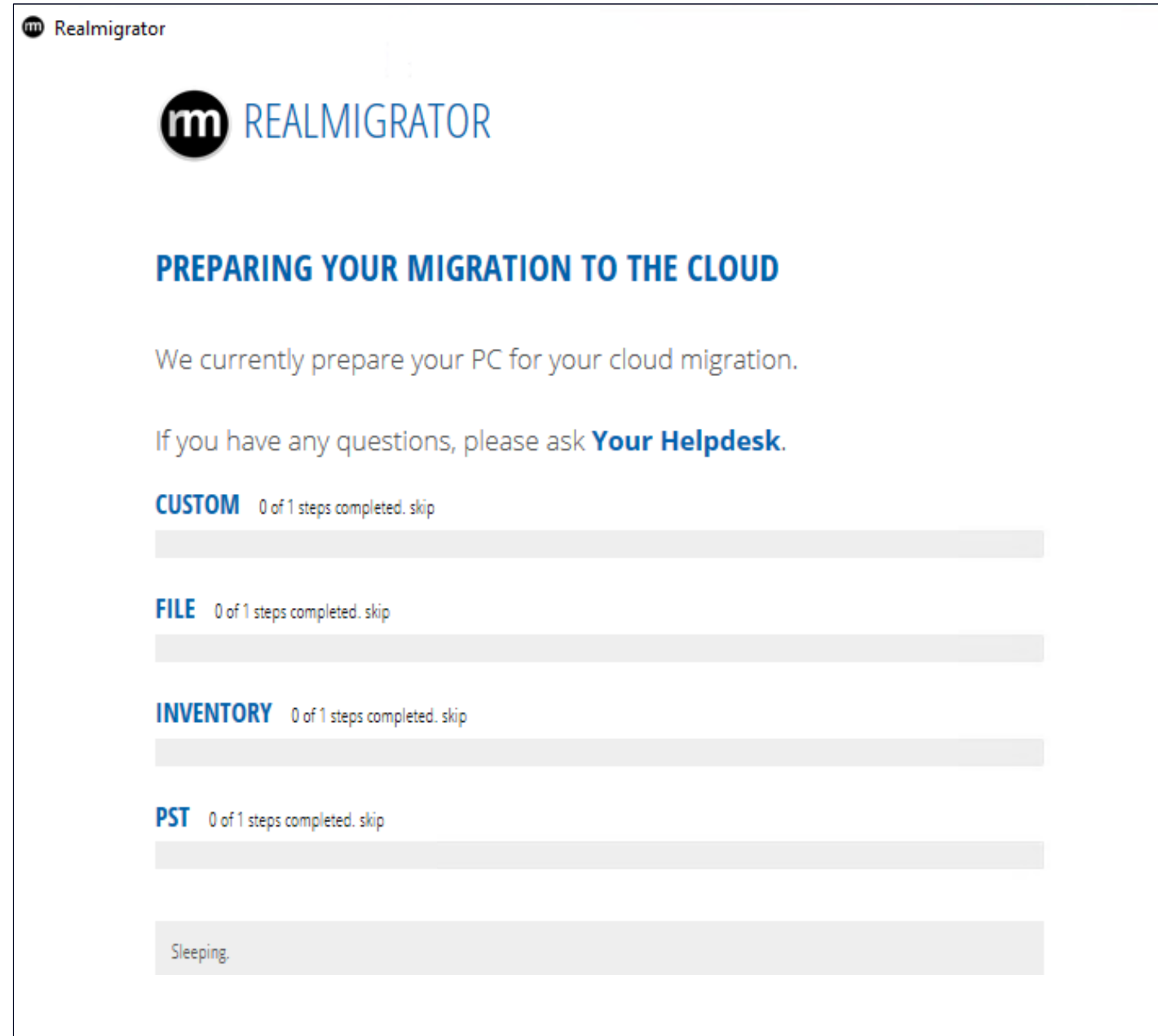— Win7/10/11 => Win10/11
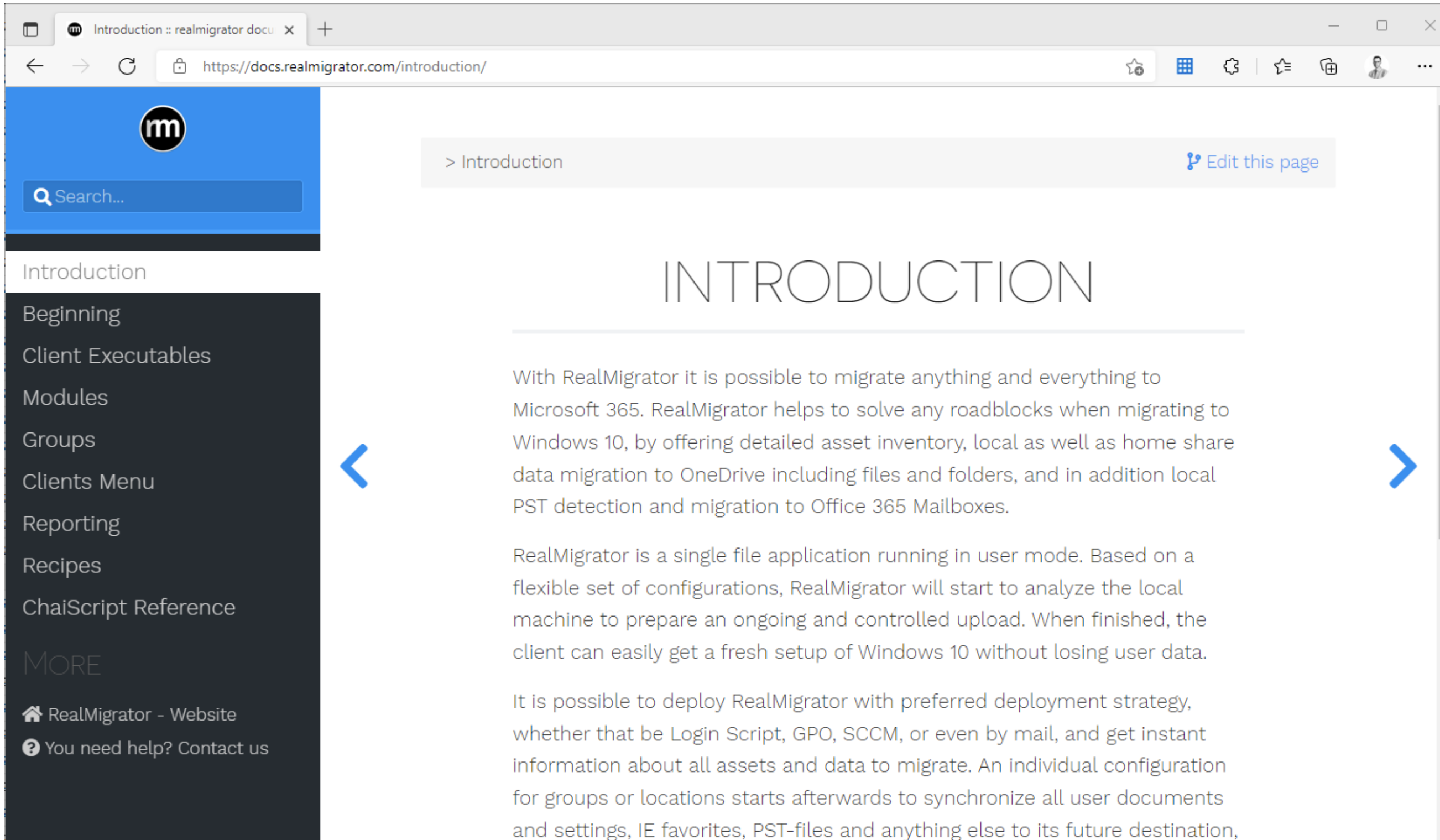
Home Share

— Personal Data

— Files

Group Share

— Group Data

— Multiple destinations per source

User UI

- MultiLanguage

- State

- Error / Retry

docs.realmigrator.com
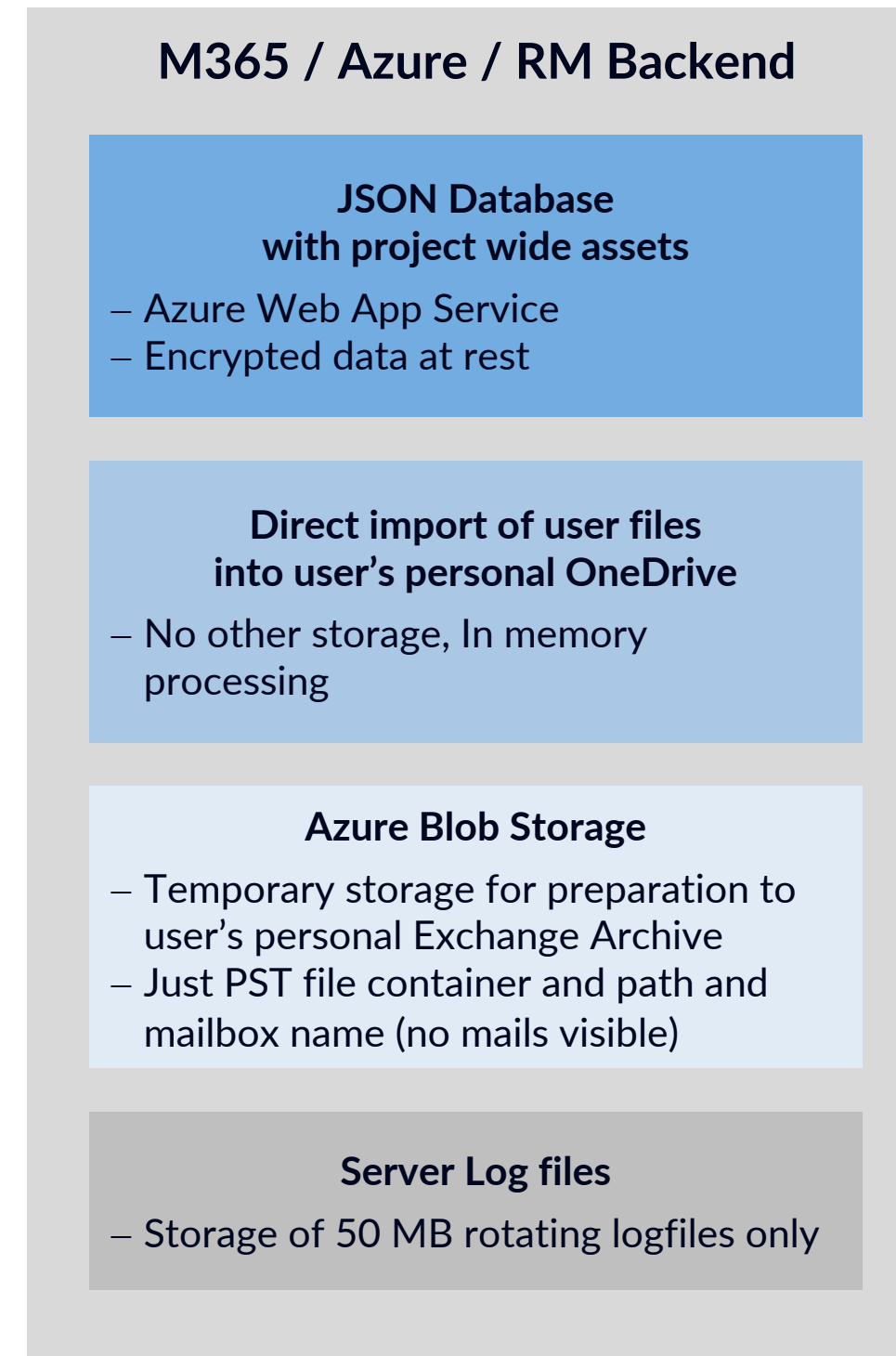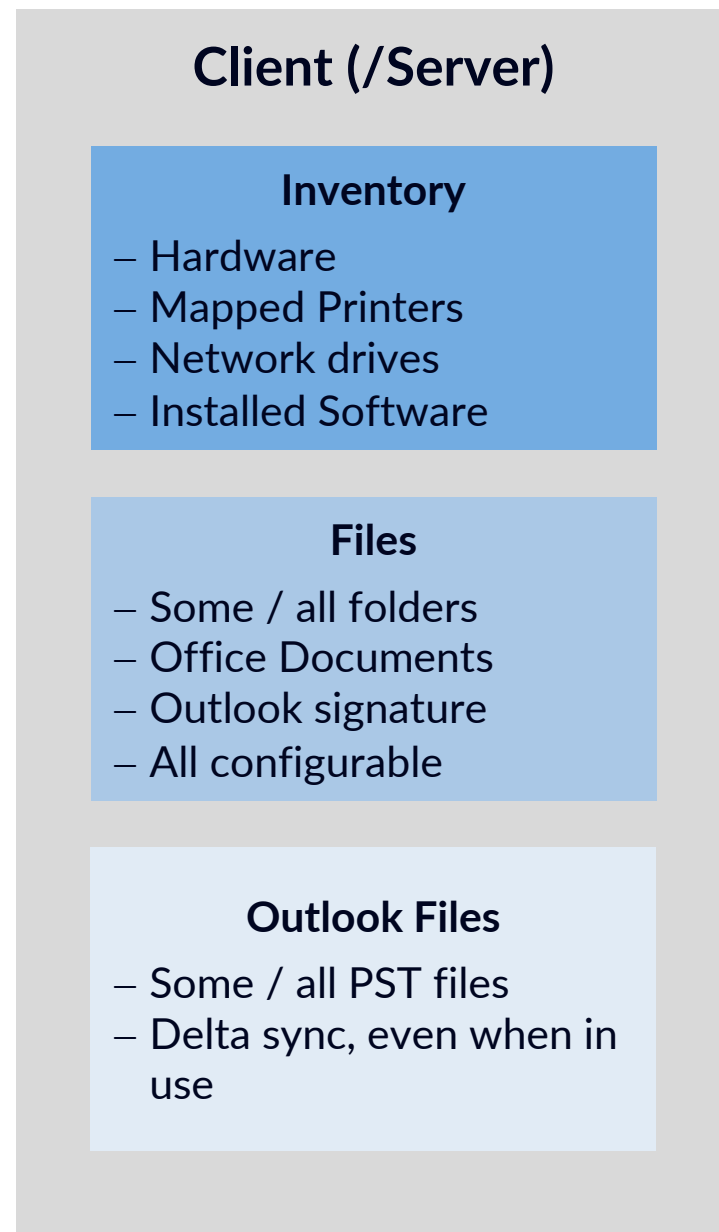

> Introduction

Edit this page

INTRODUCTION

With RealMigrator it is possible to migrate anything and everything to Microsoft 365. RealMigrator helps to solve any roadblocks when migrating to Windows 10, by offering detailed asset inventory, local as well as home share data migration to OneDrive including files and folders, and in addition local PST detection and migration to Office 365 Mailboxes.

RealMigrator is a single file application running in user mode. Based on a flexible set of configurations, RealMigrator will start to analyze the local machine to prepare an ongoing and controlled upload. When finished, the client can easily get a fresh setup of Windows 10 without losing user data.

It is possible to deploy RealMigrator with preferred deployment strategy, whether that be Login Script, GPO, SCCM, or even by mail, and get instant information about all assets and data to migrate. An individual configuration for groups or locations starts afterwards to synchronize all user documents and settings, IE favorites, PST-files and anything else to its future destination,
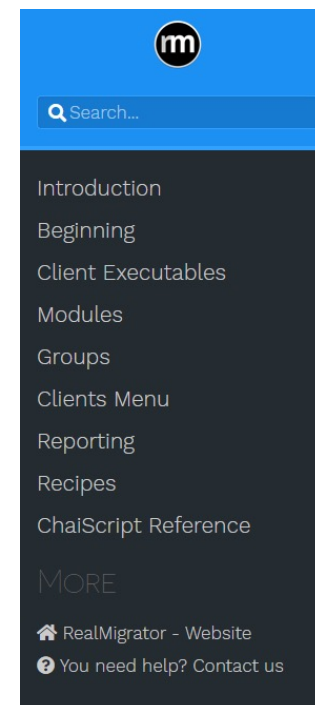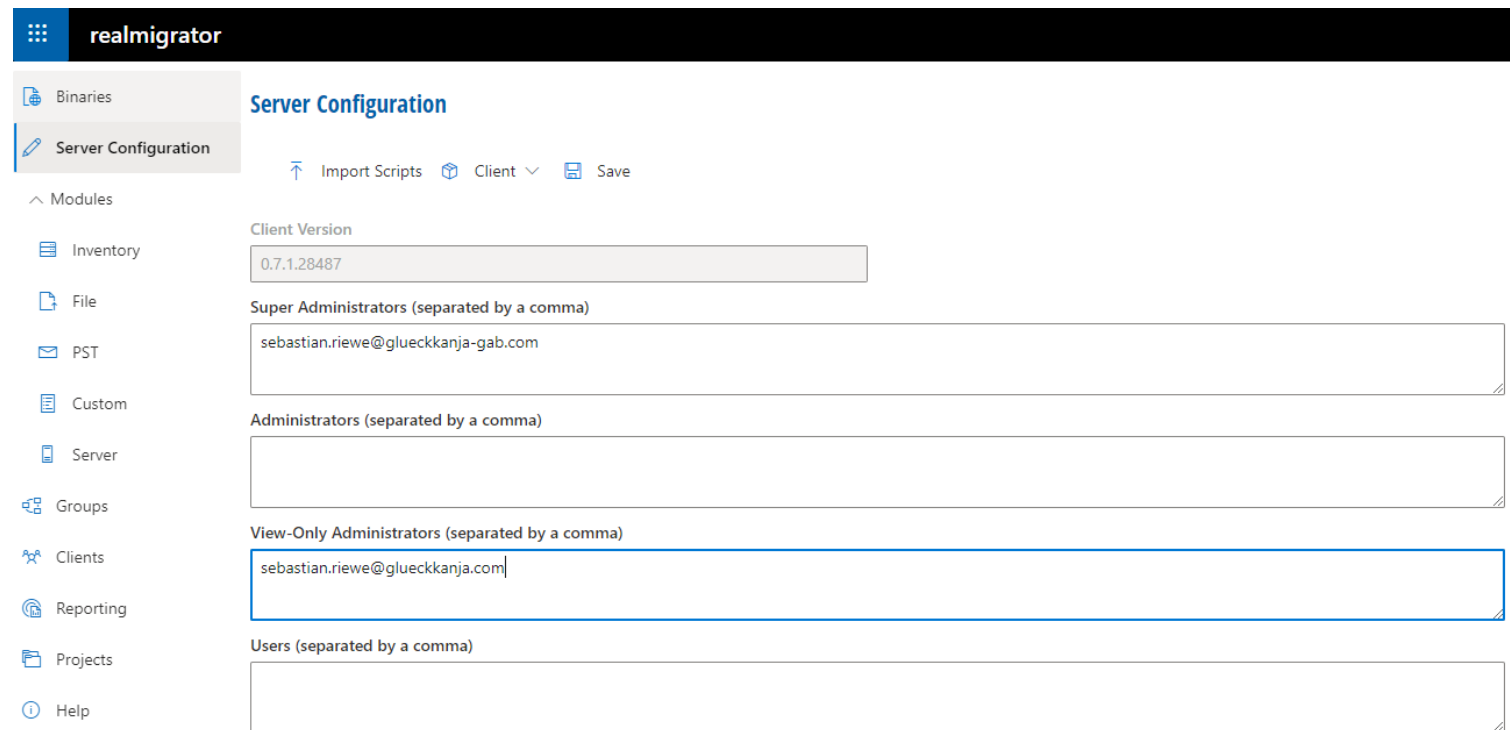
Introduction

Beginning

Client Executables

Modules

Groups

Clients Menu

Reporting

Recipes

ChaiScript Reference

MORE

RealMigrator - Website

You need help? Contact us

Features & config

RealMigrator

glueckkanja-gab

Main Features

Client (/Server)

Inventory
– Hardware
– Mapped Printers
– Network drives
– Installed Software

Files
– Some / all folders
– Office Documents
– Outlook signature
– All configurable

Outlook Files
– Some / all PST files
– Delta sync, even when in use

Client initiated
TLS 1.2 encrypted
communication

Throttled upload
over weeks

M365 / Azure / RM Backend

JSON Database
with project wide assets
– Azure Web App Service
– Encrypted data at rest

Direct import of user files
into user's personal OneDrive
– No other storage, In memory processing

Azure Blob Storage
– Temporary storage for preparation to user's personal Exchange Archive
– Just PST file container and path and mailbox name (no mails visible)

Server Log files
– Storage of 50 MB rotating logfiles only

RealMigrator

Where to start?

— general information refer to
https://docs.realmigrator.com

— Client inventory including data such as RAM, hard drives, monitors, mapped drives and printers

— Collect data about installed software from registry and MSI Installer Database

— Virtually extensionable to everything what WMI can report

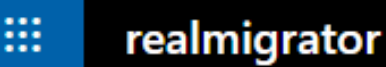— Data is uploaded into a cloud based JSON database for further reporting

WELCOME TO

REALMIGRATOR
MIGRATE THE UNMIGRATABLE

Server Configuration

— Each RealMigrator project has its own purpose and is accessible via a distinct URL.

— The RealMigrator Console is accessible at https://<project>.realmigrator.com

— Only authorized AzureAD users have access to the console

— Role assignments and configuration scripts are maintained within this section

RealMigrator

Server Configuration

glueckkanja-gab.com
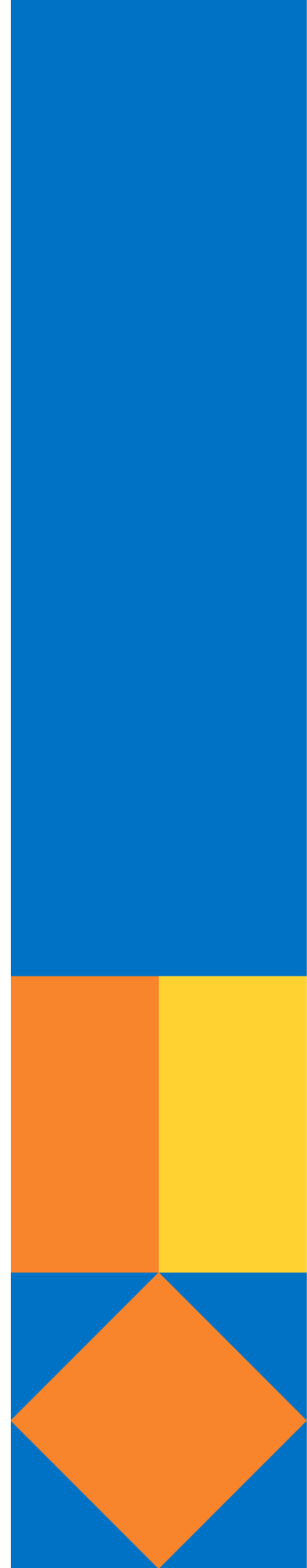
Modules

— Each module has a different scope for migration

— With **Inventory** collect Data (Hardware, Software) from a computer

— **File Module** and **PST Module:** migrate settings of files and PST files.

— **Server Module:** for PST import or sync Azure Blob to MS Teams

glueckkanja-gab.com

Modules

realmigrator
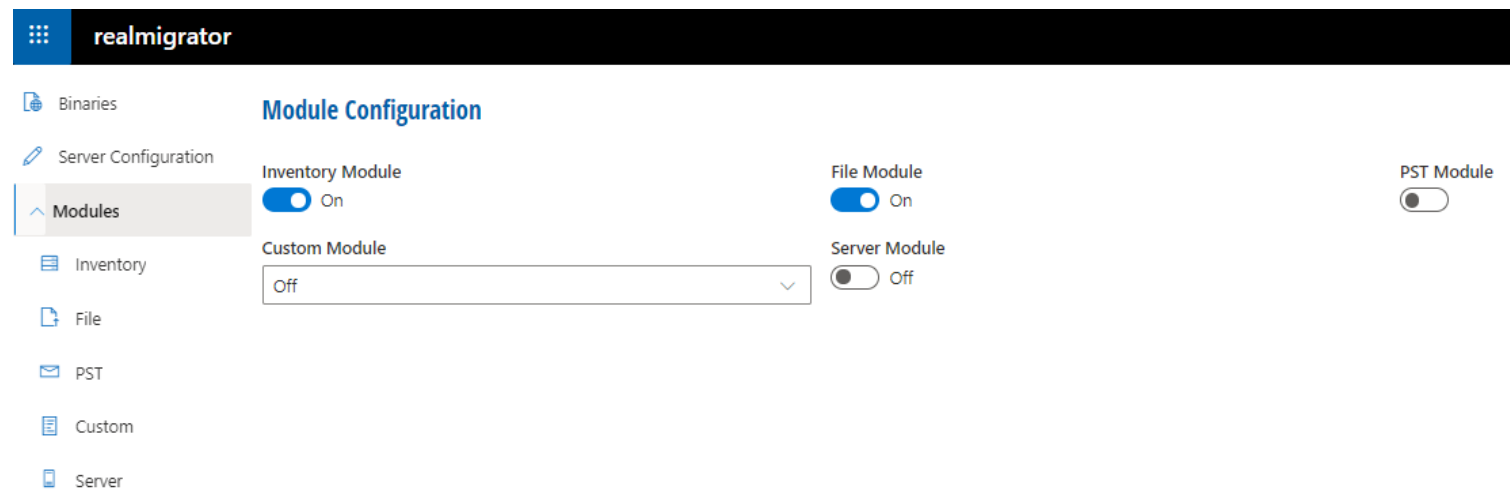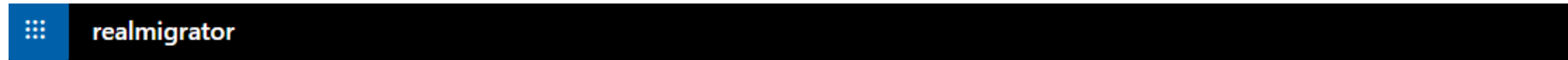
- Binaries
- Server Configuration
- **Modules**
  - Inventory
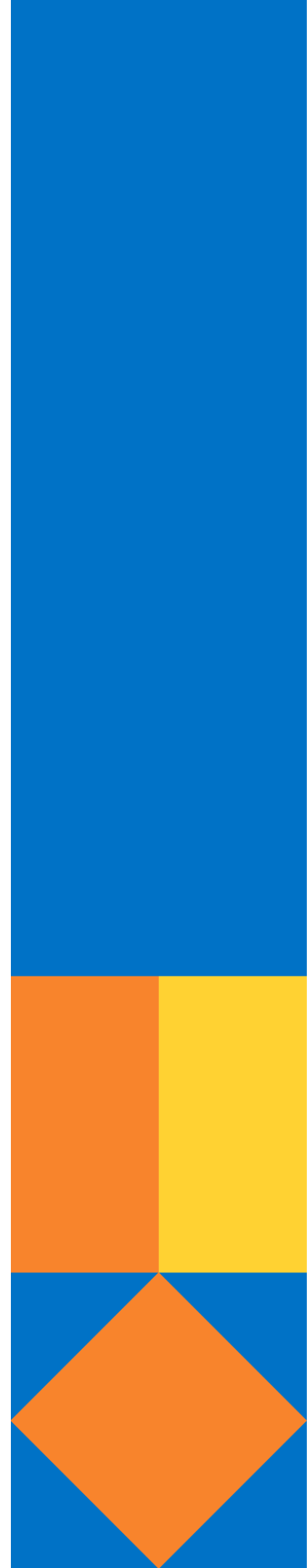 - File
  - PST
  - Custom
  - Server

Module Configuration

Inventory Module
🔵 On

File Module
🔵 On

PST Module
⚫

Custom Module
Off ⌄

Server Module
⚫ Off

glueckkanja-gab.com

Groups & Clients

— For each client an admin can create migration catalogs

— These clients are grouped in groups

— Groups have different purposes:

　— Module settings → skip, discover, migrate

　— Group based settings via script (e.g. bandwidth usage limitations)

　— Management in waves or locations

　— Fill groups with api
　https://<project>.realmigrator.com/help/api

RealMigrator

Groups & Clients

Binaries

Server Configuration

Modules

Inventory

File

PST

Custom

Server

Groups

Groups (Total 5)

+ Add   ▦ Start selection

Filter by Name (5 items displayed):

Name	Finalize Date
1 - Pilot HO - Discover	
2 - Pilot HO - Migration	
3 - Pilot HO - Finalization	
Test - Staged Teams Migration	
Unassigned Clients	
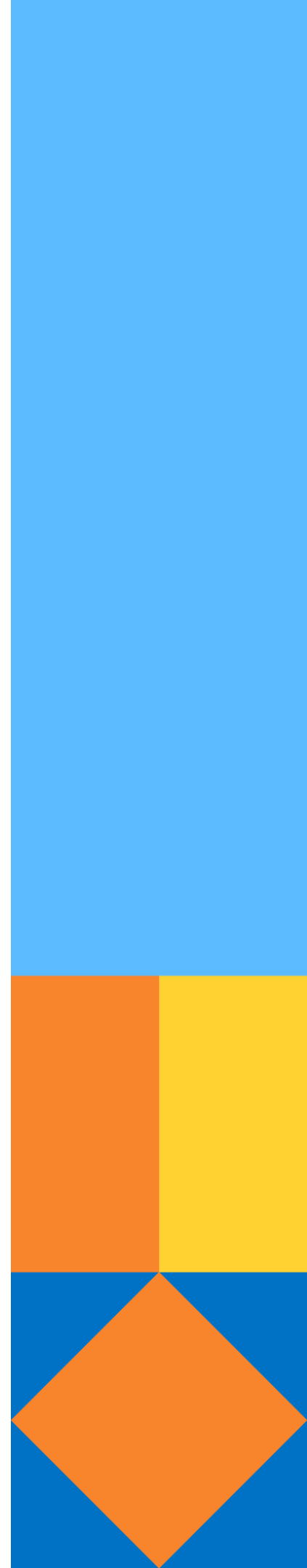
Customization – with ChaiScript

— ChaiScript - an easy to use embedded scripting language for C++

— https://chaiscript.com

Global Client Configuration [⚡ Reset]
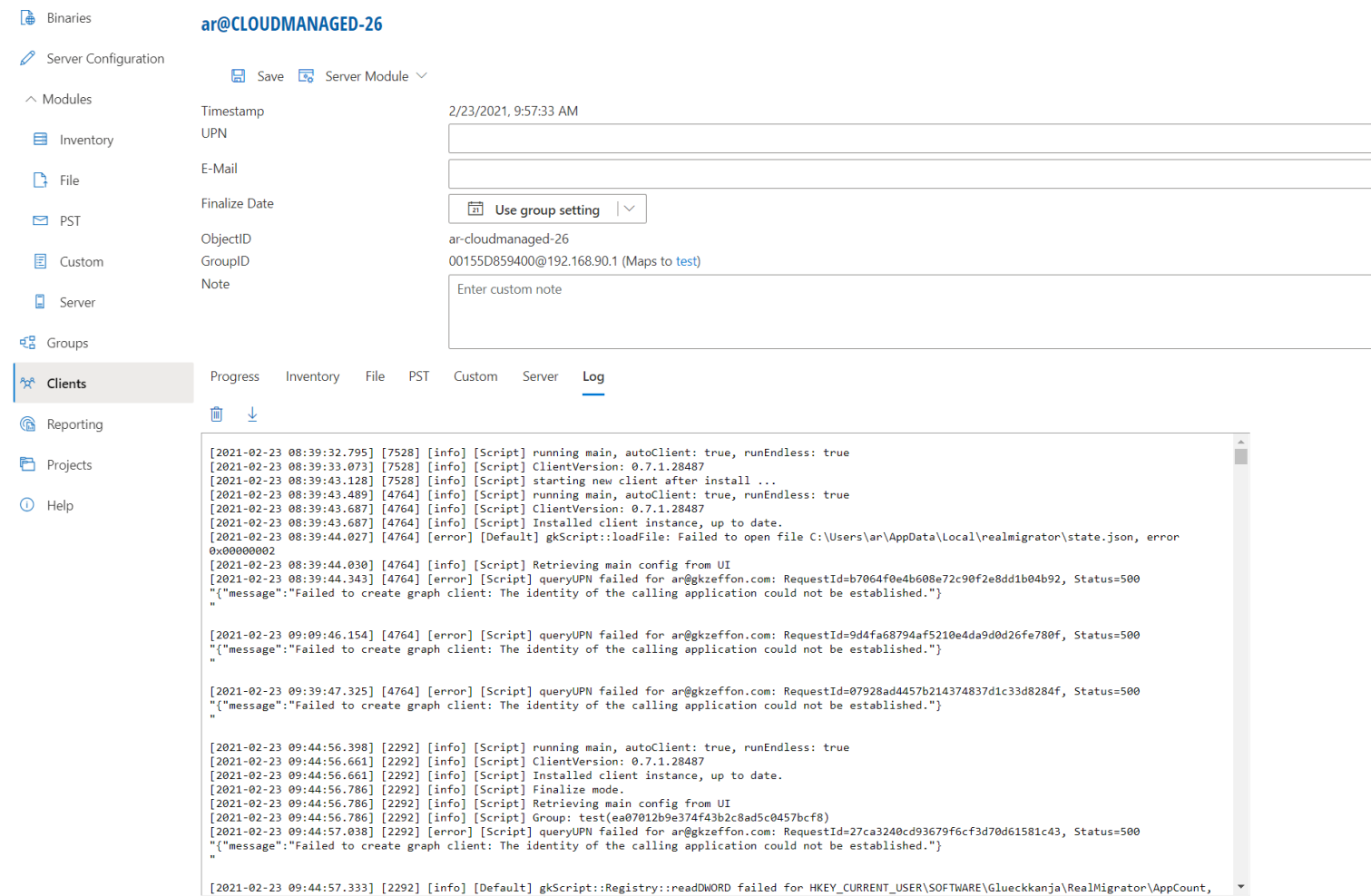
Config | **Script**

```
 1  //
 2  // config_main script
 3  //
 4
 5
 6  def getClientId() {
 7      gkScript.logDebug("getClientId");
 8
 9      return gkScript.toLower(gkScript.expandString("%username%-%computername%"));
10  }
11
12  def getGroupId() {
13      gkScript.logDebug("getGroupId");
14
15      return gkScript.getNetworkId(true);
16  }
17
18  def getDisplayName() {
19      gkScript.logDebug("getDisplayName");
20
21      return gkScript.expandString("%username%@%computername%");
22  }
23
24  def mainConfigCheck(globalSettings, config) {
25      var group = globalSettings["Group"];
26      if (type_name(group)=="Map") {
27          gkScript.logInfo("Group: " + group["DisplayName"] + "(" + group["ObjectID"] + ")");
28      }
29  }
30
31  def beforeFinalize() {
32      gkScript.logInfo("beforeFinalize");
33      gkScript.logInfo("beforeFinalize: Quitting programms");
34      quitProcesses();
35      gkScript.sleep(5000);
36      gkScript.logInfo("beforeFinalize: Killing processes");
37      killProcesses();
38  }
39
40  def afterFinalize() {
41      gkScript.logInfo("afterFinalize");
42  }
43
44  def quitProcesses() {
45      var pList=gkScript.wmiQuery("ROOT\\CIMV2", "Select ProcessId, Handle, Name from Win32_Process");
46      for (p:pList) {
47          if (gkScript.toLower(p["Name"].substr(0,7)) != "client-") {
48              var process = gkScript.openProcess(p["ProcessId"]);
49
50              if (process) {
51                  gkScript.logInfo("Send WM_QUIT to " + p["Name"] + " (" + p["Handle"] + ") ...");
52                  gkScript.wmQuitProcess(p["ProcessId"]);
53                  if (gkScript.wait(process, 1000)) {
```

group & client configuration

Clients Menu

— This menu reports the current status of the client

- — Module progress

- — Last timestamp

- — Assigned admin (defined in the UPN-field)

- — Group assignment

— Log Files

RealMigrator

Clients Menu

Binaries

Server Configuration

Modules
- Inventory
- File
- PST
- Custom
- Server

Groups

Clients

Reporting

Projects

Help

ar@CLOUDMANAGED-26

💾 Save   🖥 Server Module ⌄

Timestamp	2/23/2021, 9:57:33 AM
UPN	
E-Mail	
Finalize Date	📅 Use group setting ⌄
ObjectID	ar-cloudmanaged-26
GroupID	00155D859400@192.168.90.1 (Maps to test)
Note	Enter custom note

Progress   Inventory   File   PST   Custom   Server   **Log**

🗑   ↓

```
[2021-02-23 08:39:32.795] [7528] [info] [Script] running main, autoClient: true, runEndless: true
[2021-02-23 08:39:33.073] [7528] [info] [Script] ClientVersion: 0.7.1.28487
[2021-02-23 08:39:43.128] [7528] [info] [Script] starting new client after install ...
[2021-02-23 08:39:43.489] [4764] [info] [Script] running main, autoClient: true, runEndless: true
[2021-02-23 08:39:43.687] [4764] [info] [Script] ClientVersion: 0.7.1.28487
[2021-02-23 08:39:43.687] [4764] [info] [Script] Installed client instance, up to date.
[2021-02-23 08:39:44.027] [4764] [error] [Default] gkScript::loadFile: Failed to open file C:\Users\ar\AppData\Local\realmigrator\state.json, error
0x00000002
[2021-02-23 08:39:44.030] [4764] [info] [Script] Retrieving main config from UI
[2021-02-23 08:39:44.343] [4764] [error] [Script] queryUPN failed for ar@gkzeffon.com: RequestId=b7064f0e4b608e72c90f2e8dd1b04b92, Status=500
"{"message":"Failed to create graph client: The identity of the calling application could not be established."}
"
[2021-02-23 08:39:46.154] [4764] [error] [Script] queryUPN failed for ar@gkzeffon.com: RequestId=9d4fe68794ef521994de9d9d26fe780f, Status=500
```

RealMigrator

Reporting

— Default Reports

— inventory

— Software

— System

— Drivemapping

— Printer

— File

— File state

— File anonym

— Individuelle Reports

Reporting

Run query   Download report ∨   Advanced Report

Data Source	Report	Group
Inventory ∨	System details ∨	All Clients ∨

Total 2 items

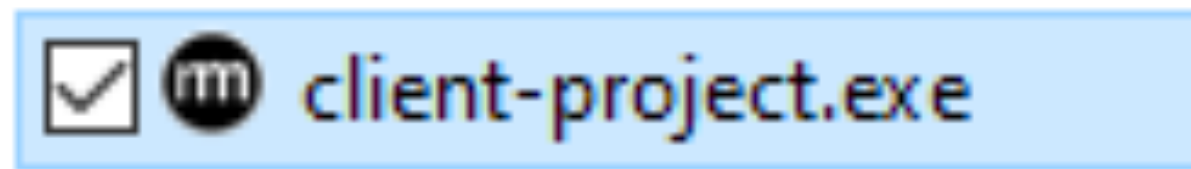| Client ID	Display Name	RAM size	Disk	Disk size	Operating System	Architecture	PC manufacturer
br-ap-566979684374	br@AP-566979684374	2374971392	Microsoft virtueller Datenträger	136358691840	Microsoft Windows 10 Enterprise	64-Bit	Microsoft Corporation
ar-cloudmanaged-26	ar@CLOUDMANAGED-26	1911500800	Microsoft Virtual Disk	136358691840	Microsoft Windows 10 Enterprise	64-bit	Microsoft Corporation

Reporting

Run query   Download report ∨   Advanced Report

Data Source	Report	Group
Inventory ∨	Software summary ∨	All Clients

Total 31 items

Name	Version	Package Code	Install Count
Microsoft OneDrive	21.016.0124.0003		2
AddressBook			4
Connection Manager			4
DirectDrawEx			4
DXM_Runtime			4

Agent

— Small Agent (single executeable) to deploy weeks before effective deployment date

— No admin permissions needed on the client

— All happens in user space context.

— Client Deployment

  — SCCM

  — GPO

  — Login Script

  — Mail

— Copy exe on Client

  — %localappdata%\realmigrator

— Run in User context

☑ 🔴 client-project.exe

glueckkanja▪gab

Thank You. Questions?