



Usage and benefits of Microsoft PowerShell in modern system administration

Increasingly, the modern answer to the question, „How do I efficiently implement this IT-administration task?“ is „With PowerShell!“ Many organizations are already on their way to leveraging this powerful framework to its full potential. But the road is rocky. PowerShell is not trivial to tap into and comes with many hurdles, partly for security reasons and partly because it’s not meant to be used by just anyone. A transfer of knowledge is laborious. So the application often remains reserved for highly skilled administrators, which in turn limits its widespread use.

holistic-net uses Microsoft PowerShell in their daily project work. Especially in migration projects, when setting up Exchange infrastructures, on-premises, or in the cloud, we experience and use PowerShell’s full effectiveness. However, continuous discipline is required when passing knowledge in the form of PowerShell scripts. Over and over, the question is asked:

„Who knows how to solve a specific problem via PowerShell? There was once a script. Where is it? Does it fit the current requirement? Who understands it? Who can customize it? Who can run it?“

In many organizations, this causes a defensive attitude that prevents the beneficial use of PowerShell. People dodge it by reaching for the mouse and lament:

„I repeatedly have to do the same clicks over and over again; isn’t there anything more convenient?“

There is!



„The powershell launch center makes it possible to deploy staff efficiently, manage systems quickly and flexibly, and minimize risk in day-to-day systems management.“

What is the basic approach of Microsoft PowerShell?

Primarily, PowerShell is to be understood as a cross-platform framework freely offered by Microsoft for handling basic administrative tasks in Windows systems. From version 6 on, PowerShell also supports the Linux and macOS platforms and is becoming a powerful tool in this regard.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\Users\Günter Mohren.HOLISTIC-NET>
```

(PowerShell Host)

The framework’s functions enable the user to implement pending tasks for the platforms’ configuration and administration by issuing commands in PowerShell’s user interface, the PowerShell host console. The PowerShell Engine is the command-line interpreter that processes the input and thus forms the heart of PowerShell. The commands are called cmdlets (pronounced commandlets).



Similar functionality is usually more complicated, sometimes even impossible, to implement by clicking through the system settings of Windows. Administration tasks can be executed much more efficiently using PowerShell given that the operator knows the right commands and parameters. The number of pre-installed cmdlets increases steadily with each new version of PowerShell being released.

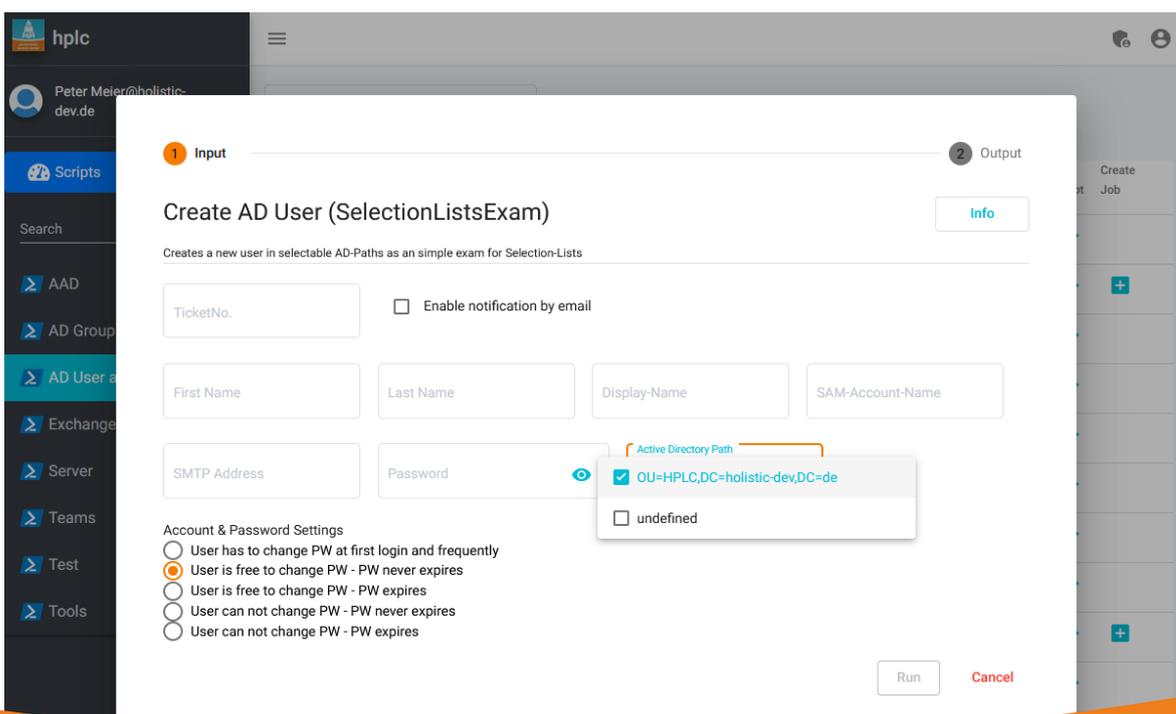
How you work out of the box:

```

111
112
113 $Users | ForEach-Object {
114
115     $firstName = $_.givenName
116     $lastName = $_.surName
117     $displayName = $_.displayName
118     $samAN = $_.sAMAccountName
119     $SMTPAddress = $_.primarySMTP
120     $userPassword = $_.initialPassword
121
122
123     $scn = $displayName
124     $relativDistinguishedName = "CN=$scn"
125     $distinguishedName = "$relativDistinguishedName,$$searchBase"
126
127     $SUPN = $SMTPAddress
128
129     $$SMTPDomain = $SMTPAddress.Split('@')[1]
130     $password = ConvertTo-SecureString $userPassword -AsPlainText -Force
131
132
133     # Check whether SAM-Account-Name is already in use in specified AD-Domain
134     # -----
135
136     $userCheck = Get-ADObject -Filter "samAccountName -eq '$samAN'" -Server $DCServer -Properties * -I
137
138     if ($userCheck -ne $null)
139     {
140         Write-Host "Check of user's AD object status shows conspicuity..."
141         Write-Host "Info: -> The specified user with SAM-Account-Name '$samAN' is already existing in
142
143         # AD
144         Remove-PSession -session $SessionAD
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

So you can work with the holistic powershell launch center:



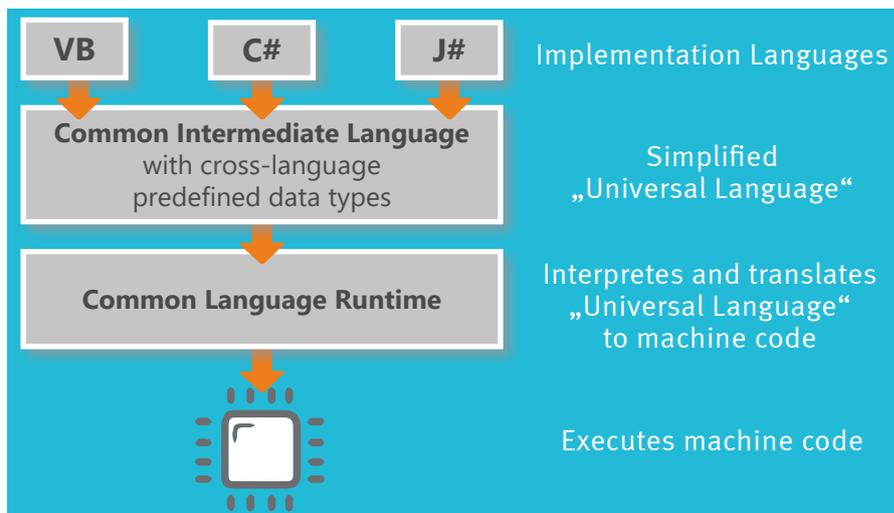


However, PowerShell shows its full strength in automating use cases. For this purpose, complex scripts/programs can be developed with the integrated PowerShell Scripting Language. Command input is done via another PowerShell host, the so-called PowerShell ISE (Integrated Scripting Environment, ISE for short). The ISE is a modern windows-based graphical user interface with an integrated script editor, debugger and IntelliSense. The PowerShell standard cmdlets relieve the user of implementing routine tasks and significantly simplify and accelerate the development process.

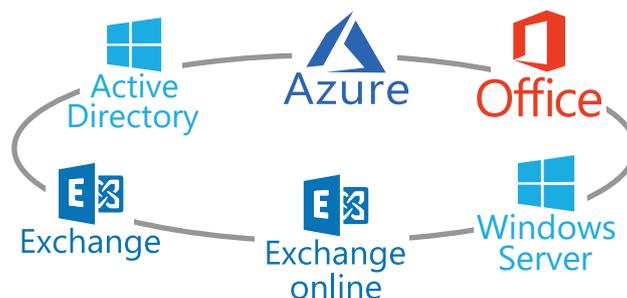
Users can leverage the full power of the .NET framework with PowerShell scripts.

PowerShell works with objects. Cmdlets are designed to perform exactly one particular task and return a specific type of object as a result. The PowerShell engine’s cmdlet library consists of a collection of C# classes based on the Common Language Runtime (CLR) of the .NET framework, e.g. WMI classes, COM objects.

PowerShell manages the whole world of Microsoft services



The commands provided by installing PowerShell are not limited to functions for Windows Client & Server administration. The command library also provides access to the directory services Active Directory and Azure AD, Exchange on-premise/online, and other parts of the Microsoft online service - Office 365.





Control and safety are paramount

However, PowerShell’s strength lies in its ability to automate everyday activities and complex procedures by scripting. Scripts ensure processing commands under full control in every individual step including necessary checks of operators input. Incorrect system settings are eliminated and the security of system operation is significantly increased. Additionally scripting is increasing efficiency of resources. Even less trained users can use PowerShell scripts to operate the infrastructure entrusted to them in less time, with maximum security and control. The resources for creating scripts and those that are using them can be separated as well.



holistic powershell launch center (hplc)

is the answer to the challenge faced by many companies that want to operate their own system landscape efficiently, in a controlled manner and, above all, with sufficient security.

From valuable experiences holistic-net developed an idea

For many years holistic-net have successfully supported a large number of its customers in the migration from HCL Notes to Microsoft Outlook. The basis of our success lies in our vast knowledge and experience in the installation and operation of infrastructures based on Notes/Domino and Outlook/Exchange. If you want to be efficient, you need excellent tools and knowledge of available migration paths. Migrations of any size require security, control, as well as efficiency. Without the skilled use of PowerShell, holistic-net would not be able to offer its customers the support they need.

Important use cases with high security requirements

- User provisioning, e.g. onboarding & offboarding
- Resource management, e.g. meeting room management
- Set up mailboxes and distribution groups
- Set up out-of-office messages for mailboxes
- Manage Windows Services

Furthermore, the potential of PowerShell does not end with the migration. For our customers, a new world opens up, and that world needs to be managed. Their Active Directory is extended in both the range and variety of entries, the Exchange environment is added. System administrators are confronted with a wide range of tasks that have to be processed securely in recurring workflows.



holistic powershell launch center (hplc)

is the solution for everyday modern system administration. hplc allows anyone to execute PowerShell scripts in a controlled manner without the need for PowerShell expertise.

More security, control & efficiency is not possible!



For most companies, regardless of size, it isn't easy to keep information about all their systems' administration and distribute that knowledge among several shoulders. A challenge not only real for companies that have completed a mailing platform migration. Systems evolve, and their complexity increases in both functionality as well as their areas of application, but they also become more powerful. Care must always be taken to ensure that all operations are carried out securely. Understandably, many IT departments cannot provide the necessary resources and, above all, the mandatory technical knowledge. More often than not, the goal of having redundant access to these competencies seems far fetched.

holistic-net were able to experience PowerShell's many advantages in developing and using complex scripts in their daily operation and support of crucial systems. From this valuable experience, an idea was born.

Our product "holistic powershell launch center" uses PowerShell's essential possibilities and features and provides them to our customers packed in a user-friendly web application.

hplc
holistic powershell launch center

```
PS COMPANY > hplc --version  
hplc Version 1.0.8062.28026
```

```
PS COMPANY> hplc --help  
This tool provides a collection of helpful PowerShell scripts
```

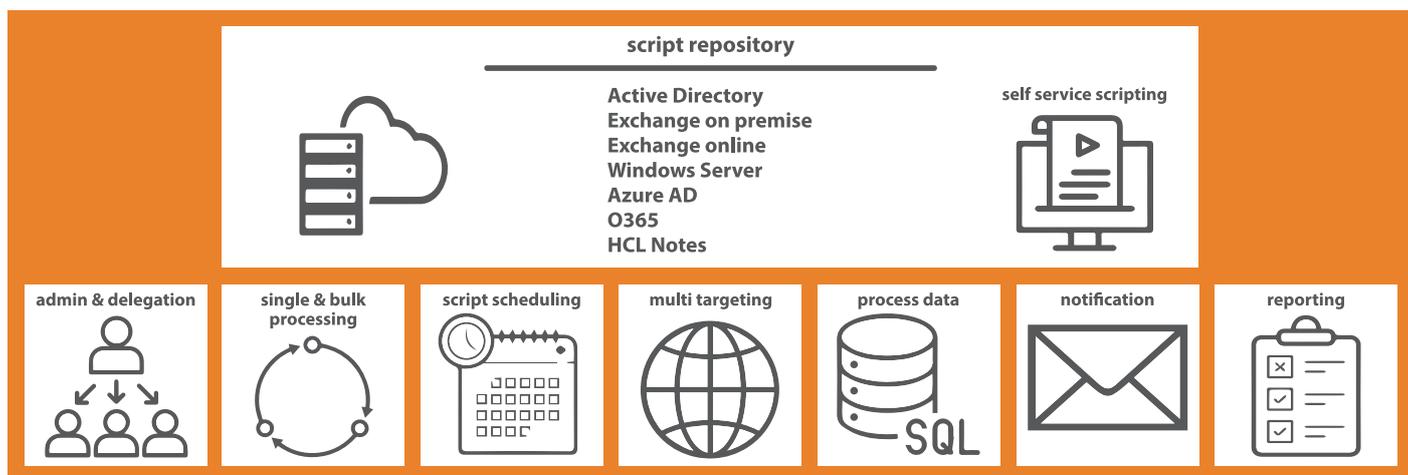
Usage:
Click on a category on the left side to get an overview of available scripts
Then, click on a specific script to fill a script with parameters
Finally, click on the 'Run' button to execute the script

Good luck and be careful :)

```
PS COMPANY>
```

41 Available scripts	7 Current jobs	2 Current templates	Jan 27, 2022 Last repository update
0 Open user deletion requests			

Copyright © 2022 holistic-net. All rights reserved. Version 1.0.8062.28026



The holistic powershell launch center leverages the Microsoft PowerShell framework and its capabilities in combination with functional extensions such as

- Delegation
- Bulk processing
- Script repository
- Script scheduling
- Email notifications
- Process data reporting
- Credential management

hplc - Advantages & Benefits

- Applicability to major current and future target system platforms
- PowerShell & target system skills are not a prerequisite
- Avoidance of incorrect settings with all adverse consequences
- Creation of resource redundancy through delegation
- Efficiency increase
- Reduction of total cost of ownership

Have we piqued your interest?

Then contact: Günter Mohren, +49 511 760728-46, guenter.mohren@holistic-net.de