



IBM WORKLOAD SCHEDULER / HCL WORKLOAD AUTOMATION

(Agent Based Credential Provider CP)

CYBERARK AAM INTEGRATION - TECHNICAL DOCUMENTATION

Name of Company: HCL Software

Website: <https://www.hcltechsw.com/>

Name of Product: IBM Workload Scheduler / HCL Workload Automation

Version: 9.5.0.4 +

Date: July 9th 2021

PARTNER SOLUTION OVERVIEW

Workload Automation is a **solution for batch and real-time hybrid workload management**, available for distributed, mainframe or hosted in the cloud. Streamline your workload management with an analytics-fueled solution.

Supported platforms for the integration with CyberArk are as follows:

Windows x86-64

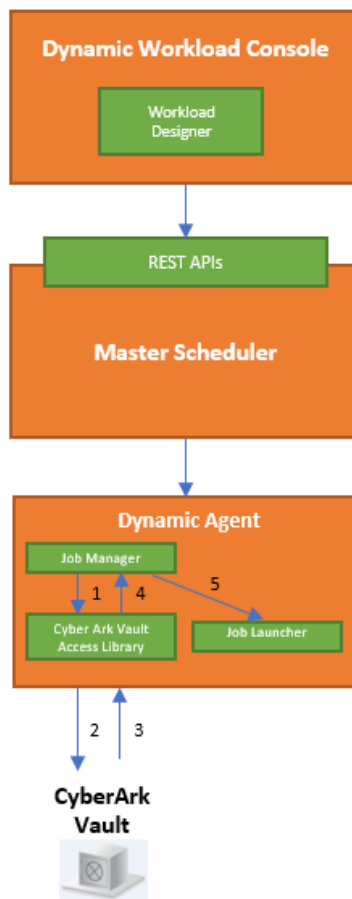
Linux x86-64

KEY BENEFITS

You can use the integration with CyberArk's AAM to submit Workload Automation jobs without specifying any passwords, which are retrieved from the CyberArk vault.

This integration allows you to retrieve the password for a specific user from CyberArk. The password is not stored locally on the Workload Automation workstation, but retrieved from CyberArk while the Workload Automation job is running.

PRODUCT DIAGRAM & DESCRIPTION OF PRODUCT INTEGRATION



You define jobs to run specific actions on target workstations and schedule the execution of the jobs based on their requirements. You can use the Dynamic Workload Console to create job definitions and save them to the database on the master domain manager. Jobs are submitted to the dynamic agents and ran using the users and passwords defined in the CyberArk vault. To retrieve the credentials from CyberArk, the Job Manager component uses the CyberArkVault library, provided with Workload Automation, to connect to the CyberArk vault and sends the retrieved credentials to the Job Launcher component, which is in charge of running the job.

The CyberArkVault library must be installed on all dynamic agents where you plan to use the integration.

The location of the integration lies in the dynamic agent, which is the endpoint.

Using the integration, you can define jobs without having to specify the password for the user. For Windows native jobs only, the password is required when defining Windows jobs. To use the CyberArk mechanism, specify a non-existing user.

On the other hand, for job application plug-ins (such as File transfer, Cognos, Datastage jobs and so on) for all operating systems supported by the dynamic agent, leave the Password field empty, or specify a password using the syntax of the agent user: `${agent:password.USERNAME}`

For more information about supported job types, see [Designing your workload](#) if you are an HCL user. If you are an IBM user, see [Designing your workload](#).

AAM INSTALLATION

Refer to the “Credential Provider and ASCP Implementation Guide” for CyberArk Agent based or Agentless installation and configuration.

AAM CONFIGURATION

DEFINING THE APPLICATION ID (APPID) AND AUTHENTICATION DETAILS

To define the Application, here are the instructions to define it manually via CyberArk’s PVWA (Password Vault Web Access) Interface:

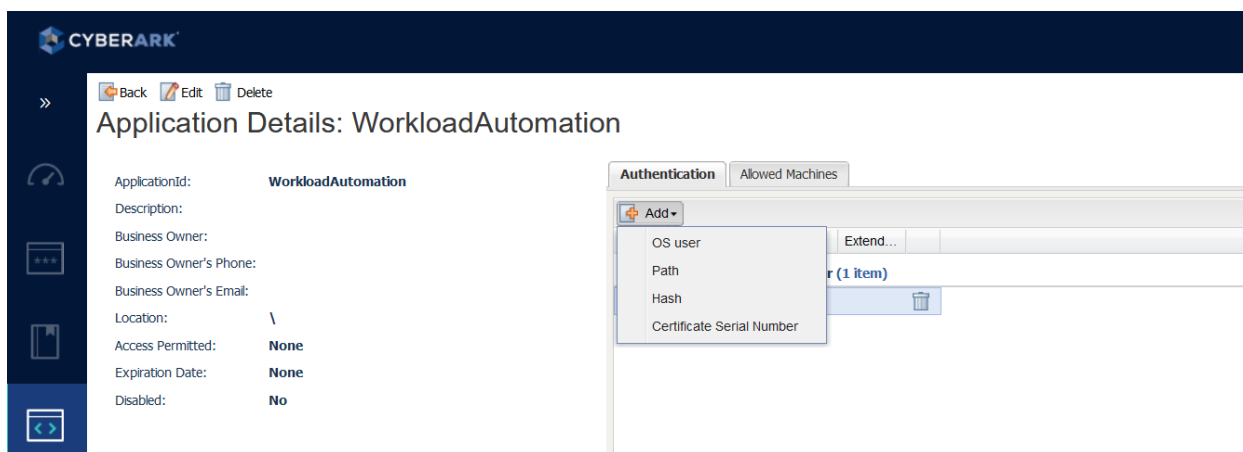
1. Logged in as user allowed to managed applications (it requires Manage Users authorization), in the Applications tab, click **Add Application**; the Add Application page appears.

The screenshot shows the 'Edit Application' dialog box. It contains the following fields and options:

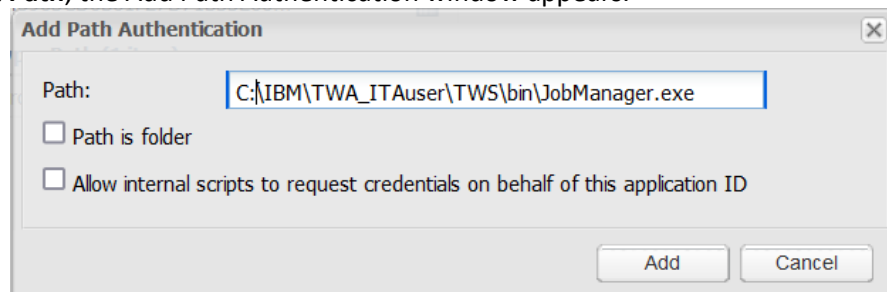
- Name:** WorkloadAutomation
- Description:** Application for Workload Automation
- Business owner:**
 - First Name:** (empty text box)
 - Last Name:** (empty text box)
 - Email:** (empty text box)
 - Phone:** (empty text box)
- Location:** (dropdown menu showing a backslash symbol)
- ☐ **Access Permitted:** From: (dropdown) To: (dropdown)
- ☐ **Expiration Date:** (calendar icon)
- ☐ **Disabled**

At the bottom right, there are two buttons: **Update** and **Cancel**.

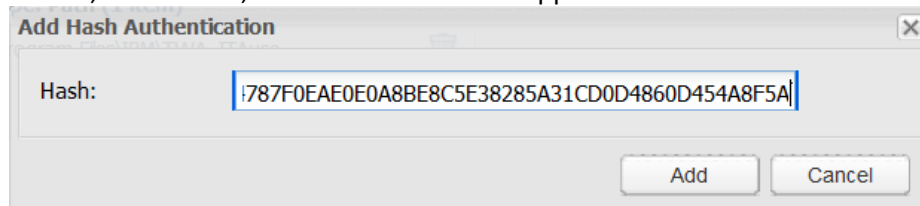
2. Specify the following information:
 - In the Name edit box, specify the unique name (ID) of the application. The recommended Application ID for this integration is: **WorkloadAutomation**
 - In the Description, specify a short description of the application that will help you identify it.
 - In the Business owner section, specify contact information about the application's Business owner.
 - In the lowest section, specify the Location of the application in the Vault hierarchy. If a Location is not selected, the application will be added in the same Location as the user who is creating this application.
3. Click **Add**; the application is added and is displayed in the Application Details page.



- **Allowing extended authentication restrictions.** This enables you to specify an unlimited number of machines and Windows domain OS users for a single application. Please check this box.
4. Specify the application's **Authentication** details. This information enables the Credential Provider to check certain application characteristics before retrieving the application password.
 - In the Authentication tab, click **Add**; a drop-down list of authentication characteristics is displayed.
 - Select the authentication characteristic to specify.
 5. Specify the application path:
 - a. Select **Path**; the Add Path Authentication window appears.



- b. Specify the path where the application will run. The executable file to specify is `<TWA_home>/TWS/bin/JobManager`
 - c. Click **Add**; the Path is added as an authentication characteristic with the information that you specified.
6. Specify a hash:
 - a. Run the AIMGetAppInfo utility to calculate the application's unique hash. The hash value is calculated based on the executable `<TWA_home>/TWS/bin/JobManager`.
 - b. Copy the hash value that is returned by the utility.
 - c. In the PVWA, select **Hash**; the Add Hash window appears.

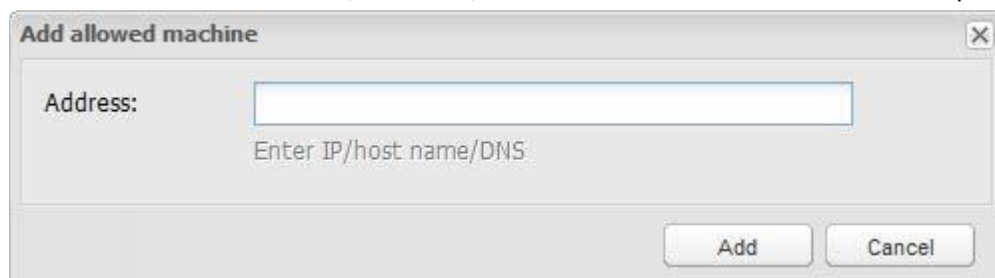


- d. In the Hash edit box, paste the application's unique hash value, or specify multiple hash values with a semi-colon. You can add additional information in a comment after each hash value specified for an application by specifying '#' after the hash value, followed by the comment.

For example, OE883B7OD5B6E3EE37D37198049C9507C8383DB6 #app2

Note: The comment must not include a colon or a semicolon.

- e. Click **Add**; the Hash is added as an authentication characteristic with the information that you specified.
7. Specify the application's **Allowed Machines**. This information enables the Credential Provider to make sure that only applications that run from specified machines can access their passwords.
 - In the Allowed Machines tab, click **Add**; the Add allowed machine window is displayed.



- Specify the IP/hostname/DNS of the machine where the application will run and will request passwords, then click **Add**; the IP address is listed in the Allowed machines tab. Make sure the servers allowed include all mid-tier servers or all endpoints where the AAM Credential Providers were installed.

PROVISIONING ACCOUNTS AND SETTING PERMISSIONS FOR APPLICATION ACCESS

For the application to perform its functionality or tasks, the application must have access to particular existing accounts, or new accounts to be provisioned in CyberArk Vault (Step 1). Once the accounts are managed by CyberArk, make sure to setup the access to both the application and CyberArk Application Password Providers serving the Application (Step 2).

1. In the Password Safe, provision the privileged accounts that will be required by the application. You can do this in either of the following ways:
 - **Manually** – Add accounts manually one at a time, and specify all the account details.
 - **Automatically** – Add multiple accounts automatically using the Password Upload feature.

For this step, you require the **Add accounts** authorization in the Password Safe.

For more information about adding and managing privileged accounts, refer to **the Privileged Access Security Implementation Guide**.

2. Add the Credential Provider and application users as members of the Password Safes where the application passwords are stored. This can either be done manually in the Safes tab, or by specifying the Safe names in the CSV file for adding multiple applications.
 - i. Add the Provider user as a Safe Member with the following authorizations:
 - List accounts
 - Retrieve accounts
 - View Safe Members

Note: When installing multiple Providers for this integration, it is recommended to create a group for them, and add the group to the Safe once with the above authorization.

Add Safe Member

Search:

Search In:

Vault

Search

Selected Search: Vault

Name	Business Email	Full Name
------	----------------	-----------

☐ Access

☐ Use accounts

☒ Retrieve accounts

☒ List accounts

☐ Account Management

☐ Safe Management

☐ Monitor

☐ View Audit log

☒ View Safe Members

Add

Close

- ii. Add the application(the APPID) as a Safe Member with the following authorizations:

- Retrieve accounts

Add Safe Member

Search:

Search In:

Vault

Search

Selected Search: Vault

Name	Business Email	Full Name
------	----------------	-----------

☐ Access

☐ Use accounts

☒ Retrieve accounts

☐ List accounts

☐ Account Management

☐ Safe Management

☐ Monitor

☐ View Audit log

☐ View Safe Members

▲

▼

Add

Close

- iii. If the Safe is configured for object level access, make sure that both the Provider user and the application have access to the password(s) to retrieve.

For more information about configuring Safe Members, refer to the **Privileged Access Security Implementation Guide**.

PARTNER PRODUCT INSTALLATION & INTEGRATION CONFIGURATION

Refer to CyberArk.ini configuration file.

CyberArk.ini configuration file:

```
[CyberArk.Config]
CPAccessLibrary =
HandlePasswordChangeInProgress = true
RetryIntervalForPasswordChangeInProgress =
RetryAttemptsForPasswordChangeInProgress =

[CyberArk.CP.Connection]
Port =
ConnectionTimeout = 180

[CyberArk.AppDescs]
AppID = WorkloadAutomation
|
[CyberArk.Query]
Safe =
Folder =
Object =
Username =
Address =
PolicyID =
Database =
```

SYSTEM REQUIREMENTS

Requires the dynamic agent, Version 9.5, Fix Pack 4, or later.

INSTALLING

Integrate CyberArk with Workload Automation by deploying the `CyberArkVault_version_number` library file and customizing the CyberArk.ini properties file, available for download at [Automation Hub](#). The files are available in the compressed archive. The `CyberArkVault_version_number` library file allows you to retrieve the password for a specific user from CyberArk. The password is not stored locally on the Workload Automation workstation, but retrieved from CyberArk while the job is running.

The `CyberArkVault_version_number` library file supports two different integration methods to CyberArk:

CCP

This method starts an HTTP call to CyberArk Central Credential Provider (CCP).

CP

This method starts a call to CyberArk Credential Provider (CP) using a proprietary CyberArk library file.

For dynamic agents installed on all supported operating systems, with the exception of IBM i, use the GSKit mode. For dynamic agents installed on IBM i, use the OpenSSL mode.

To install and configure the integration with CyberArk, perform the following steps:

1. Download and extract the `IBM_Workload_Automation_CyberArk_Vault_version_number.zip` archive from [Automation Hub](#) to all your agents.
2. Save the `CyberArkVault_version_number` library file and the `CyberArk.ini` properties file to a path of your choice. Ensure the library and properties files are stored in the same path and the `twuser` defined on the agent is authorized to read them.
3. Browse to the `JobManager.ini` file and add the following property, if not already present, in the Launchers section:

```
PasswordResolver
```

The `JobManager.ini` file is located in

On UNIX™ operating systems

```
TWA_DATA_DIR/ITA/cpa/config/JobManager.ini
```

On Windows™ operating systems

```
TWA_home\TWS\ITA\cpa\config\JobManager.ini
```

4. Set the `PasswordResolver` property to the full path to the `CyberArkVault_version_number` library file.
5. Browse to the path where you stored the `CyberArk.ini` properties file.
6. Edit the `CyberArk.ini` properties file. For a full description of all properties, see [CyberArk.ini file details](#).
7. Add the necessary accounts in CyberArk, using the dedicated interface. For more information, see CyberArk documentation.

DEFINING AND SUBMITTING A JOB

You can now define a job without having to specify the password for the user. Note the following points:

Windows native jobs only

The password is required when defining Windows jobs. To use the CyberArk mechanism, specify a non-existing user.

Job application plug-ins (such as File transfer, Cognos, Datastage jobs and so on)

Leave the Password field empty, or specify a password using the syntax of the agent user:

```
${agent:password.USERNAME}
```

For more information about supported job types, see [Designing your workload](#) if you are an HCL user. If you are an IBM user, see [Designing your workload](#).

You can also define the job user in a query, by combining the parameters in the AppDescs and Query sections of the CyberArk.ini file. Specify one or more parameters to build a query using the following syntax:

```
section_name::parameter_name=parameter_value;section_name::parameter_name=parameter_value...
```

For example:

```
AppDescs::AppID=TestApplicationCert;Query::Safe=TestSafe;Query::Username=ITAuth;...:PolicyID=UnixSSH
```

CYBERARK.INI FILE DETAILS

This section describes the properties of the CyberArk.ini file.

CyberArk.Config

CPAccessLibrary

The full path to the CyberArk proprietary library file contained in ApplicationPasswordSdk. For example, on Windows systems, point to the CPPasswordSDK64.dll library file. Leave this property empty to use CCP.

HandlePasswordChangeInProcess

The operation to be performed if another user changes the password while you are requesting it from CyberArk. Supported values are true and false. The default value is true.

If you set this property to true, the job remains in waiting status and password retrieval is attempted again, based on the values set for the following parameters:

RetryIntervalForPasswordChangeInProcess and RetryAttemptsForPasswordChangeInProcess

If you set this property to false, the operation fails if another user changes the password while you are requesting it from CyberArk.

RetryIntervalForPasswordChangeInProgress

The time interval in seconds IBM Workload Scheduler waits before sending a new password request to CyberArk.

RetryAttemptsForPasswordChangeInProgress

The number of times IBM Workload Scheduler retries to obtain the password from CyberArk. If the specified number of retries is exceeded, the operation fails.

CyberArk.CP.Connection

This section applies only when you use the CP.

Port

The port that is used to connect to the Credential Provider.

ConnectionTimeout

The timeout period in seconds before a timeout message is issued when a password is requested. The default value is 180 seconds.

CyberArk.CCP.Connection

This section applies only when you use the CCP and is used when you leave the CPAccessLibrary property empty. These properties are mandatory.

Host

The host name of the workstation where CyberArk Central Credential Provider is installed.

Protocol

The protocol used to connect to the host.

Port

The port used to connect to the host.

Path

The path where the REST API is located.

ConnectionTimeout

The time interval in seconds IBM Workload Scheduler waits for the host to answer the connection request. The default value is 180 seconds.

Timeout

The time interval in seconds IBM Workload Scheduler waits for CyberArk to return the password. The default value is 3600 seconds.

FollowLocation

Set this property to `true` to enable the HTTP protocol. You cannot enable the HTTP protocol from the command line. This property instructs the composer command to follow any 'Location: header' that the server sends as part of the HTTP header in a 3xx response. The 'Location: header' can specify a relative or an absolute URL to follow.

UseGSKit

Specifies whether GSKit should be enabled for secure communication. Supported values are `yes` and `no`. The default value is `yes`.

SSLVerifyServer

Specify `yes` if server authentication is to be used in SSL communications. The default value is `yes`.

Proxy

The name of the proxy server used when connecting to the specified host.

ProxyPort

The TCP/IP port number of the proxy server used when connecting to the specified host.

TLS10

Specifies whether version 10 of the TLS protocol is used. The default value is `DFLT`.

TLS11

Specifies whether version 11 of the TLS protocol is used. The default value is `NONE`.

TLS12

Specifies whether version 12 of the TLS protocol is used. The default value is `NONE`.

SSLV3

Specifies whether SSL version 3.0 is supported. The default value is `NONE`.

CyberArk.CCP.Connection.GSKit

This section applies only when you use the CCP and is used if you leave the `CPAccessLibrary` property empty. These properties are mandatory if the `Protocol` property is set to `https` and the `UseGSKit` property is set `yes`. You can use this mode for dynamic agents installed on all supported operating systems, with the exception of IBM i. If your dynamic agents are installed on IBM i, use the OpenSSL mode.

`SSLKeystoreFile`

The path to the keystore file.

`SSLKeystorePwd`

The password for the keystore file.

`SSLCertificateKeystoreLabel`

The label which identifies the certificate in the keystore when using SSL authentication.

CyberArk.CCP.Connection.OpenSSL

This section applies only when you use the CCP and is used only if you leave the `CPAccessLibrary` property empty. These properties are mandatory if the `Protocol` property is set to `https` and the `UseGSKit` property is set `no`. You can use this mode only for dynamic agents installed on IBM i. If your dynamic agents are installed on other supported operating systems, use the GSKit mode.

`SSLKey`

The name of the private key file.

`SSLCertificate`

The name of the local certificate file used in SSL communication.

`SSLKeyPwd`

The name of the file containing the password for the stashed key.

`SSLCACertificate`

The name of the file containing the trusted certification authority (CA) certificates required for SSL authentication.

`SSLRandomSeed`

The pseudo random number file used by OpenSSL on some operating systems.

CyberArk.AppDescs

AppID

The unique ID of the application issuing the password request. This parameter is required.

CyberArk.Query

Safe

The name of the Safe where the password is stored.

Folder

The name of the folder where the password is stored.

Object

The name of the password object to retrieve.

Username

Defines search criteria according to the UserName account property.

Address

Defines search criteria according to the Address account property.

PolicyID

Defines the format that will be used in the setPolicyID method.

Database

Defines search criteria according to the Database account property.

CyberArk.Logging

This section is reserved for the Software Support team.

CyberArk.Logging.cclog

This section is reserved for the Software Support team.

ENABLING THE CYBERARK INTEGRATION ON ALL YOUR AGENTS

To enable the integration, you need to deploy the `CyberArkVault_<version_number>` library file and the `CyberArk.ini` properties file to all your agents, and modify the `JobManager.ini` to use CyberArk.

To automate the `JobManager.ini` customization, you can customize and submit the following sample jobs, replacing paths and other relevant details to meet the specifics of your environment:

```

VARIABLE MAIN_TABLE
  ISDEFAULT
  MEMBERS
    CYBER_ARK_DLL_PATH_UNIX
    "path_to_CyberArk_library/libCyberArkVault_version_number.so"
    CYBER_ARK_DLL_PATH_WIN
    "path_to_CyberArk_library\CyberArkVault_version_number.dll"
  END
SCHEDULE CAJS
DESCRIPTION "Added by composer."
VARIABLE MAIN_TABLE
:
CAJOB AS CAJOB_UNIX
  TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jSDL:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/XMLSchema"
      xmlns:jSDL="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDL"
      xmlns:jSDLE="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDLE"
      XMLSchema:text="resolveVariableTable" name="executable">
      <jSDL:application name="executable">
        <jSDLE:executable interactive="false">
          <jSDLE:script>echo
$UNISONHOME/_uninstall/ACTIONTOOLS/TWSupdate_file -delRow
$AGENT_INI/JobManager.ini PasswordResolver
$UNISONHOME/_uninstall/ACTIONTOOLS/TWSupdate_file -delRow
$AGENT_INI/JobManager.ini PasswordResolver
echo $UNISONHOME/_uninstall/ACTIONTOOLS/TWSupdate_file -addRow
$AGENT_INI/JobManager.ini PasswordResolver ${CYBER_ARK_DLL_PATH_UNIX}
Launchers
$UNISONHOME/_uninstall/ACTIONTOOLS/TWSupdate_file -addRow
$AGENT_INI/JobManager.ini PasswordResolver ${CYBER_ARK_DLL_PATH_UNIX}
Launchers
$UNISONHOME/ShutDownLwa
$UNISONHOME/StartUpLwa</jSDLE:script>
        </jSDLE:executable>
      </jSDL:application>
    </jSDL:jobDefinition>
    DESCRIPTION "Added by composer."
    RECOVERY STOP

CAJOB AS CAJOB_WIN
  TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jSDL:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/XMLSchema"
      xmlns:jSDL="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDL"
      xmlns:jSDLE="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDLE"
      XMLSchema:text="resolveVariableTable" name="executable">
      <jSDL:application name="executable">
        <jSDLE:executable interactive="false">
          <jSDLE:script>
echo &quot;%UNISONHOME%\_uninstall\ACTIONTOOLS\TWSupdate_file&quot;
-delRow &quot;%UNISONHOME%\ITA\cpa\config\JobManager.ini&quot;
PasswordResolver
          </jSDLE:script>
        </jSDLE:executable>
      </jSDL:application>
    </jSDL:jobDefinition>
    DESCRIPTION "Added by composer."
    RECOVERY STOP

```

```

&quot;%UNISONHOME%\_uninstall\ACTIONTOOLS\TWSupdate_file&quot;;
-delRow &quot;%UNISONHOME%\ITA\cpa\config\JobManager.ini&quot;;
PasswordResolver
echo &quot;%UNISONHOME%\_uninstall\ACTIONTOOLS\TWSupdate_file&quot;;
-addRow &quot;%UNISONHOME%\ITA\cpa\config\JobManager.ini&quot;;
PasswordResolver ${CYBER_ARK_DLL_PATH_WIN} Launchers
&quot;%UNISONHOME%\_uninstall\ACTIONTOOLS\TWSupdate_file&quot;;
-addRow &quot;%UNISONHOME%\ITA\cpa\config\JobManager.ini&quot;;
PasswordResolver ${CYBER_ARK_DLL_PATH_WIN} Launchers
&quot;%UNISONHOME%\ShutDownLwa&quot;; &amp;;
&quot;%UNISONHOME%\StartUpLwa&quot;;
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
  DESCRIPTION "Added by composer."
  RECOVERY STOP

END

```

PARTNER CONTACT INFO

	Name	Email
Business Contact	Vincenzo Spinelli	vincenzo.spinelli@hcl.com
	Umberto Caselli	umberto.caselli@hcl.com
	Antonella Godino	antonella.godino@hcl.com
Technical Contact	Franco Mossotto	franco.mossotto@hcl.com
	Silvano Lutri	silvano.lutri@hcl.com
	Gianluca Seghetti	gianluca.seghetti@hcl.com
	Michele Pescuma	michele.pescuma@hcl.com
Support Contact	Maria Sbriccoli	maria.sbriccoli@hcl.com
	Paolo Salerno	paolo.salerno@hcl.com
	Donatella Sabellico	donatella.sabellico@hcl.com