This page last changed on Jul 25, 2011 by stevet.

3.0 Console Command Files

Overview

The patch assistant console is a utility enabling creation or modification of patches via the command line. This document explains the available arguments, command file configuration and output for each command.

Command Line Arguments

Option	Parameters	Description
GetPublicKey	file path	Returns the public key used to sign a file
Encrypt	value	Returns an encrypted string for use in password fields
Run	file path	Runs a console command file

*** *all options have a preceding forward slash*

Console Command File

XML file with commands contained in a ConsoleCommands tag.

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<ConsoleCommands xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <!-- Commands Go Here -->
</ConsoleCommands>
```

Console Commands

All commands have an optional IgnoreError boolean parameter to force the console to ignore the error if the command fails.

*** ***Bold*** *values indicate required parameters*

CreateBundle

Key	Type	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
InputPath	path	Local path to bundle
OutputFileName	path	Bundle file location

Sample CreateProduct command:

```
<ConsoleCommand xsi:type="CreateBundle">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <InputPath>d:\patchtestConsole\bundle</InputPath>
  <OutputFileName>d:\patchtestConsole\test.bundle</OutputFileName>
</ConsoleCommand>
```

CreateHostExecutable

Key	Type	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
BaseName	string	Base host to use (typically iehost). iehost_stripped removes some logging from the executable
InputPath	path	Local path to application bundle
OutputFileName	path	Executable file location
ProductName	string	Product name in executable
CompanyName	string	Company name in executable
Copyright	string	Copyright in executable
UPX	bool	UPX compress the executable

Sample CreateProduct command:

```
<ConsoleCommand xsi:type="CreateHostExecutable">
  <UserName>johndoe@email.com</UserName>
```

```
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <BaseName>iehost</BaseName>
  <InputPath>d:\patchtestConsole\appbundle</InputPath>
  <OutputFileName>d:\patchtestConsole\test.exe</OutputFileName>
  <UPX>true</UPX>
</ConsoleCommand>
```

CreateProduct

Key	Type	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
Path	path	Local directory in which the product will be created
Title	string	Title of the product

Sample CreateProduct command:

```
<ConsoleCommand xsi:type="CreateProduct">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <Path>d:\patchtestConsole\</Path>
  <Title>PAConsoleTest</Title>
</ConsoleCommand>
```

Output

- solid.patchproduct file is created at local product path

Sample solid.patchproduct file:

```
<?xml version="1.0" encoding="utf-8"?>
<Product xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" FileFormatVersion="4">
  <Title>Console-docs</Title>
  <Payloads />
</Product>
```

CreatePayload

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN

		password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
Title	string	Title of the payload
TargetDirectory	module (see below) + path	Destination directory for payload to be extracted to
RequiresElevation	boolean	Requires elevation for download and application
Dependencies	dependency list	dependent payloads
Dependency	URL	Full URL to dependent payloads

Target Directory Modules

All modules are encapsulated by curly brackets {}

- ModuleFilename
- ModulePath (active launcher directory)
- TempPath
- ModuleArguments
- ProgramFiles
- StartMenu
- UserDocuments
- UserDesktop
- UserAppData
- UserAppDataRoaming

Sample CreatePayload command:

```
<ConsoleCommand xsi:type="CreatePayload">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <Title>Payload</Title>
  <TargetDirectory>{ModulePath}GameDirectory</TargetDirectory>
  <RequiresElevation>false</RequiresElevation>
  <Dependencies>
    <Dependency>http://localhost/sample1.patchmanifest</Dependency>
    <Dependency>http://localhost/sample2.patchmanifest</Dependency>
  </Dependencies>
</ConsoleCommand>
```

Output

- Creates sub-folder with payload name within product directory
- Creates solid.payload file in project folder
- Adds the <Payload> element to the solid.patchproduct file

Sample solid.payload file:

```
<?xml version="1.0" encoding="utf-8"?>
<Payload xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" FileFormatVersion="4">
  <Title>docs-pay1</Title>
  <RequiresElevation>false</RequiresElevation>
  <TargetDirectory>{ModulePath}GameDirectory</TargetDirectory>
  <RequiredReleaseId>-1</RequiredReleaseId>
  <UpcomingReleaseId>-1</UpcomingReleaseId>
  <LastManifestName />
  <Dependencies>
    <string>http://localhost/sample1.patchmanifest</string>
    <string>http://localhost/sample2.patchmanifest</string>
  </Dependencies>
  <Releases />
</Payload>
```

EditPayload

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
Title	string	Title of the payload
TargetDirectory	module (see below) + path	Destination directory for payload to be extracted to
RequiresElevation	boolean	Requires elevation for download and application
Dependencies	dependency list	dependent payloads
Dependency	URL	Full URL to dependent payloads

Sample EditPayload command:

```
  <ConsoleCommand xsi:type="EditPayload">
    <UserName>johndoe@email.com</UserName>
    <Password>5404419716016175192142035211068183136042206220 70</Password>
    <ProductPath>c:\patchtestConsole\</ProductPath>
    <Title>Payload</Title>
    <TargetDirectory>{ModulePath}Game\Environment</TargetDirectory>
    <RequiresElevation>true</RequiresElevation>
    <Dependencies>
      <Dependency>http://localhost/test.xml</Dependency>
    </Dependencies>
  </ConsoleCommand>
```

Output

- Updates solid.payload file with new element values

CreateRelease

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
SourcePath	path	Path to directory containing release files
DiffType	XDelta/None	Type of differential to use on update paths to and from this release
DisableDifferential	Files listed here are excluded from differential updating (no spaces) For example: file1.txt;file2.txt;*.ini;	
UpdateFrom	List of releases to create an update path from	
Release	string	release name (use **None** when creating the -1toX update path)
RollbackTo	List of releases to create a rollback path to	
Release	string	release name

Sample CreateRelease command:

```
<ConsoleCommand xsi:type="CreateRelease">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>release_1</Name>
  <SourcePath>c:\work\source\gamedirectory\</SourcePath>
  <DiffType>XDelta</DiffType>
  <DisableDifferential></DisableDifferential>
  <UpdateFrom>
    <Release>None</Release>
    <Release>release_0</Release>
  </UpdateFrom>
  <RollbackTo>
    <Release>release_0</Release>
  </RollbackTo>
</ConsoleCommand>
```

Output

- creates a **source** folder within the payload directory
    - copies the source files into sequentially numbered folders (starting at 0) within the source folder
    - each release source folder contains an encrypted .version file with SHA1 values for all files in the release
- updates the solid.payload file with the release information

Sample .version file (decrypted):

```
<?xml version="1.0" encoding="utf-8"?>
<PatchVersion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Files>
    <File>
      <Name>file1.txt</Name>
      <LastModified>Fri, 27 Feb 2009 20:45:21 GMT</LastModified>
      <Size>134457</Size>
      <SHA1>4f991fcbd4ad57e9a024e58606ed7026af530250</SHA1>
    </File>
    <File>
      <Name>file2.txt</Name>
      <LastModified>Fri, 27 Feb 2009 20:39:35 GMT</LastModified>
      <Size>483016</Size>
      <SHA1>cae711a93ba67ca001dde4e821dd0eca28550043</SHA1>
    </File>
    <File>
      <Name>file3.txt</Name>
      <LastModified>Fri, 27 Feb 2009 20:46:25 GMT</LastModified>
      <Size>152405</Size>
      <SHA1>ba024dc112dab72af8afa979e650486688e0fe8c</SHA1>
    </File>
    <File>
      <Name>file4.txt</Name>
      <LastModified>Fri, 27 Feb 2009 20:46:59 GMT</LastModified>
      <Size>486849</Size>
      <SHA1>16797d93ddfcc4b065bd1778bb3d2d45b5b39ee2</SHA1>
    </File>
  </Files>
</PatchVersion>
```

Sample solid.payload file after CreateRelease:

```
<?xml version="1.0" encoding="utf-8"?>
<Payload xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" FileFormatVersion="4">
  <Title>docs-pay1</Title>
  <RequiresElevation>true</RequiresElevation>
  <TargetDirectory>{ModulePath}Game\Environment</TargetDirectory>
  <RequiredReleaseId>-1</RequiredReleaseId>
  <UpcomingReleaseId>-1</UpcomingReleaseId>
  <LastManifestName />
  <Dependencies>
    <string>http://localhost/test.xml</string>
  </Dependencies>
  <Releases>
    <Release>
      <Id>0</Id>
      <Name>release_0</Name>
      <SHA1>1c34b05cf41b87adbcf1ad4bb55f8243eb94c225</SHA1>
      <DiffType>None</DiffType>
      <DisableDifferential />
      <UpdateFromList>
        <int>-1</int>
      </UpdateFromList>
      <RollbackToList />
      <NetworkGroupId>0</NetworkGroupId>
      <ExtraData />
    </Release>
    <Release>
      <Id>1</Id>
```

```
      <Name>release_1</Name>
      <SHA1>6190b168f48ffcb9b8aa42f2d8ddd79f7ab90fb9</SHA1>
      <DiffType>XDelta</DiffType>
      <DisableDifferential>file1.txt</DisableDifferential>
      <UpdateFromList>
        <int>-1</int>
        <int>0</int>
      </UpdateFromList>
      <RollbackToList>
        <int>0</int>
      </RollbackToList>
      <NetworkGroupId>0</NetworkGroupId>
      <ExtraData />
    </Release>
  </Releases>
</Payload>
```

EditRelease

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
DiffType	XDelta/None	Type of differential to use on update paths to and from this release
DisableDifferential	Files listed here are excluded from differential updating (no spaces) For example: file1.txt;file2.txt;*.ini;	
UpdateFrom	List of releases to create an update path from	
Release	string	release name
RollbackTo	List of releases to create a rollback path to	
Release	string	release name

Sample EditRelease command:

```
  <ConsoleCommand xsi:type="EditRelease">
    <UserName>johndoe@email.com</UserName>
    <Password>5404419716016175192142035211068183136042206220 70</Password>
    <ProductPath>c:\patchtestConsole\</ProductPath>
    <PayloadTitle>Payload</PayloadTitle>
    <Name>release_2</Name>
```

```
    <DiffType>XDelta</DiffType>
    <DisableDifferential>file1.txt</DisableDifferential>
    <UpdateFrom>
      <Release>release_1</Release>
    </UpdateFrom>
    <RollbackTo>
      <Release>release_1</Release>
    </RollbackTo>
  </ConsoleCommand>
```

Output

- updates solid.payload file with new values

RemoveRelease

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release

Sample RemoveRelease command:

```
<ConsoleCommand xsi:type="RemoveRelease">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>release_3</Name>
</ConsoleCommand>
```

Output

- removes the specified <Release> element from the solid.payload file
- deletes the numbered release folder from within the source directory for the payload

BuildRelease

Key	Value	Description
UserName	string	Solid State Networks ADMIN email

Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
Clean	boolean	Delete existing release paths from /out/ directory

Sample BuildRelease command:

```
<ConsoleCommand xsi:type="BuildRelease">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>1</Name>
  <Clean>false</Clean>
</ConsoleCommand>
```

Output

- Creates an **out** folder within the payload directory
  - creates sub-directories for each update path (-1to2, 0to2, 1to2, etc...)
  - each update path folder contains the patch data in .zip and .z0X files (1GB chunks)

InjectRelease

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
NetworkGroup	string	Name of Solid State Networks Admin network group
ReliableSourceUrl	string	Root URL where the content should be downloaded from the server
DownloadConfigurationUrl	string	path to skin configuration file (.solidconfig). Can be within skin or external URL

Sample InjectRelease command:

```
  <ConsoleCommand xsi:type="InjectRelease">
    <UserName>johndoe@email.com</UserName>
    <Password>5404419716016175192142035211068183136042206220 70</Password>
    <ProductPath>c:\patchtestConsole\</ProductPath>
    <PayloadTitle>Payload</PayloadTitle>
    <Name>release_1</Name>
    <NetworkGroup>_QA Internal</NetworkGroup>
    <ReliableSourceUrl>http://localhost/</ReliableSourceUrl>
    <DownloadConfigurationUrl>{AppContentUrl}download.solidconfig</DownloadConfigurationUrl>
  </ConsoleCommand>
```

Output

- Creates .solid file within **out** directory for every update path in release
    - .solid file naming convention: payload title + update path
- Updates solid.payload file with injection data for each update path

Sample solid.payload file after InjectRelease:

```
<?xml version="1.0" encoding="utf-8"?>
<Payload xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" FileFormatVersion="4">
  <Title>docs-pay1</Title>
  <RequiresElevation>true</RequiresElevation>
  <TargetDirectory>{ModulePath}Game\Environment</TargetDirectory>
  <RequiredReleaseId>-1</RequiredReleaseId>
  <UpcomingReleaseId>-1</UpcomingReleaseId>
  <LastManifestName />
  <Dependencies>
    <string>http://localhost/test.xml</string>
  </Dependencies>
  <Releases>
    <Release>
      <Id>0</Id>
      <Name>release_0</Name>
      <SHA1>1c34b05cf41b87adbcf1ad4bb55f8243eb94c225</SHA1>
      <DiffType>XDelta</DiffType>
      <DisableDifferential>file1.txt</DisableDifferential>
      <UpdateFromList>
        <int>-1</int>
      </UpdateFromList>
      <RollbackToList />
      <NetworkGroupId>16022</NetworkGroupId>
      <ReliableSource>http://qa.solidstatenetworks.com/pa/2.1/</ReliableSource>
      <DownloadConfig>{AppContentUrl}download.solidconfig</DownloadConfig>
      <ExtraData>
        <PatchExtraDataItem>
          <FromId>-1</FromId>
          <ToId>0</ToId>
          <Key>MetafileUrl</Key>
          <Value>http://qa.solidstatenetworks.com/pa/2.1/docs-pay1_-1to0.solid</Value>
        </PatchExtraDataItem>
        <PatchExtraDataItem>
          <FromId>-1</FromId>
          <ToId>0</ToId>
          <Key>ConfigurationUrl</Key>
          <Value>{AppContentUrl}download.solidconfig</Value>
        </PatchExtraDataItem>
      </ExtraData>
    </Release>
  </Releases>
</Payload>
```

CreatePatchManifest

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the patch manifest
IsMaintenance	boolean	Indicates if the launcher should display a maintenance message when running this payload
Required	string	Name of release required to use the application
Upcoming	string	Name of release that can be optionally downloaded

Sample CreatePatchManifest command:

```
<ConsoleCommand xsi:type="CreatePatchManifest">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>Payload</Name>
  <IsMaintenance>false</IsMaintenance>
  <Required>1</Required>
  <Upcoming>None</Upcoming>
</ConsoleCommand>
```

Output

- Creates an encrypted .patchmanifest file within the **out** directory for the payload
  - the .patchmanifest file contains versioning and patch configuration information used by the patcher

Sample .patchmanifest file (decrypted):

```
<?xml version="1.0" encoding="utf-8"?>
<PatchManifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Dependencies>
    <Dependency>http://localhost/test.xml</Dependency>
  </Dependencies>
  <Name>docs-pay1</Name>
  <RequiredRelease>0</RequiredRelease>
  <UpcomingRelease>-1</UpcomingRelease>
  <TargetDirectory>{ModulePath}Game\Environment</TargetDirectory>
  <RequiresElevation>true</RequiresElevation>
  <Maintenance>false</Maintenance>
```

```
  <Releases>
    <Release>
      <Id>0</Id>
      <SHA1>1c34b05cf41b87adbcf1ad4bb55f8243eb94c225</SHA1>
      <Name>release_0</Name>
    </Release>
    <Release>
      <Id>1</Id>
      <SHA1>6190b168f48ffcb9b8aa42f2d8ddd79f7ab90fb9</SHA1>
      <Name>release_1</Name>
    </Release>
  </Releases>
  <ReleaseUpdatePaths>
    <ReleaseUpdatePath>
      <From>-1</From>
      <To>0</To>
      <ExtraData>
        <ExtraDataItem>
          <Key>MetafileUrl</Key>
          <Value>http://qa.solidstatenetworks.com/pa/2.1/docs-pay1_-1to0.solid</Value>
        </ExtraDataItem>
        <ExtraDataItem>
          <Key>ConfigurationUrl</Key>
          <Value>{AppContentUrl}download.solidconfig</Value>
        </ExtraDataItem>
      </ExtraData>
    </ReleaseUpdatePath>
    <ReleaseUpdatePath>
      <From>-1</From>
      <To>1</To>
      <ExtraData />
    </ReleaseUpdatePath>
    <ReleaseUpdatePath>
      <From>0</From>
      <To>1</To>
      <ExtraData />
    </ReleaseUpdatePath>
    <ReleaseUpdatePath>
      <From>1</From>
      <To>0</To>
      <ExtraData />
    </ReleaseUpdatePath>
  </ReleaseUpdatePaths>
</PatchManifest>
```

Patch Extra Data

AddPatchExtraData

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
From	string	Name of release whose update

		path the extra data should apply from
To	string	Name of release whose update path the extra data should apply to
Key	string	Title of extra data key
Value	string	Title of extra data key value

Sample AddPatchExtraData command:

```
<ConsoleCommand xsi:type="AddPatchExtraData">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>release_1</Name>
  <From>None</From>
  <To>release_1</To>
  <Key>Sample Key Name</Key>
  <Value>Sample Key Value 1</Value>
</ConsoleCommand>
```

Output

- Adds extra patch data to specified <Release> element and update path within solid.payload file

Sample solid.payload file after AddPatchExtraData:

```
  <Release>
    <Id>1</Id>
    <Name>release_1</Name>
    <SHA1>6190b168f48ffcb9b8aa42f2d8ddd79f7ab90fb9</SHA1>
    <DiffType>XDelta</DiffType>
    <DisableDifferential>file1.txt</DisableDifferential>
    <UpdateFromList>
      <int>-1</int>
      <int>0</int>
    </UpdateFromList>
    <RollbackToList>
      <int>0</int>
    </RollbackToList>
    <NetworkGroupId>0</NetworkGroupId>
    <ExtraData>
      <PatchExtraDataItem>
        <FromId>1</FromId>
        <ToId>0</ToId>
        <Key>Sample Key Name</Key>
        <Value>Sample Key Value 1</Value>
      </PatchExtraDataItem>
    </ExtraData>
  </Release>
```

RemovePatchExtraData

Key	Value	Description
UserName	string	Solid State Networks ADMIN email
Password	string	Solid State Networks ADMIN

		password in encrypted format (using Console /encrypt argument)
ProductPath	path	Local directory of the product
PayloadTitle	string	Title of the payload
Name	string	Title of the release
From	string	Name of release whose update path the extra data should apply from
To	string	Name of release whose update path the extra data should apply to
Key	string	Title of extra data key
Value	string	Title of extra data key value

Sample RemovePatchExtraData command:

```
<ConsoleCommand xsi:type="RemovePatchExtraData">
  <UserName>johndoe@email.com</UserName>
  <Password>5404419716016175192142035211068183136042206220 70</Password>
  <ProductPath>c:\patchtestConsole\</ProductPath>
  <PayloadTitle>Payload</PayloadTitle>
  <Name>release_1</Name>
  <From>None</From>
  <To>1</To>
  <Key>Test</Key>
</ConsoleCommand>
```

Output

- Removes specified extra patch data from <Release> element within solid.payload file