



Instructions for the Installation and Configuration of requis_MiX 1.9

Table of Contents

Table of Contents.....	1
1 Introduction	2
2 Requirements	2
3 Installation	2
3.1 Installation of requisis_MiX	2
3.1.1 Local Installation.....	2
3.1.2 Update of a local installation	2
3.1.3 Network installation.....	3
3.1.4 Update of a network installation	3
3.2 Importing and setting up a Mapping Rule Module.....	3
3.3 Setting up the Logging Module (optional)	4
3.4 Configuration of DOORS Next Generation.....	4
3.5 Configure Server Details and OAuth Consumer Key	5
3.5.1 Settings in the configuration.json file	6
3.5.2 SSL-client certificates for DNG Server Connections (optional)	6
3.5.3 Configure Auto Login	7
3.6 Start requisis_MiX	7
3.6.1 PHP settings and usage in terminal server environments (Citrix etc.)	8
4 Advanced Configuration.....	9
4.1 Allow Objects without Type	9
4.2 Customization for merged attributes	9
4.3 Deactivate the Check for Unsupported Fonts	10
4.4 Set the Default Choice for the Object Type Attribute	10
4.5 Save OLE-Objects as .ole instead of .rtf.....	10
4.6 Show Empty Attributes in Collations (Merged Attributes)	10
4.7 Adjust the LOG-Level	10
4.8 Set the Default Link Type for external links.....	10
4.9 Use operating systems default browser for DNG Login.....	11
4.10 Encryption of oauth keys in configuration.json.....	12
4.11 Support for changing the DNG server domain	12
5 API	13



requisis_MiX

5.1	API Functions	13
5.2	DXL Example	14

1 Introduction

requisis_MiX is an extension for IBM Rational DOORS 9 to migrate and synchronize modules between **IBM DOORS** and **IBM DOORS Next Generation (DNG)**. This document provides detailed information for administrators and experienced users about the installation and configuration of the tool.

2 Requirements

To run *requisis_MiX* the following requirements need to be met before installing and running the tool:

- > IBM DOORS 9.6 or later
- > IBM DOORS Next Generation 6.0.5 or later
- > We recommend using *requisis_MiX* with at least 1920x1080 resolution (full HD). Especially if you want to use the Migration Manager.
- > Microsoft Visual C++ Runtime 2019 Redistributable must be installed.

You can download the latest Visual C++ Runtime packages directly from Microsoft:

<https://learn.microsoft.com/en-us/cpp/windows/latest-suppoPHPrted-vc-redis?view=msvc-170>

3 Installation

The default installation of *requisis_MiX* requires 5 steps:

- 1) Run the installer.
- 2) Create a project named "**requisis_MiX**" in your DOORS database.
- 3) Restore the DMA-File containing a template module.
- 4) Create OAuth consumer key from the JTS application of the JAZZ platform.
- 5) Enter server details and OAuth consumer key in the *requisis_MiX* server configuration file.

In the chapters below each step is described in detail, together with some optional configuration steps.

3.1 Installation of *requisis_MiX*

3.1.1 Local Installation

requisis_MiX comes with an installer that installs the tool as a DOORS-Extension by setting the Current-User Registry-Key for DOORS-Addins.

3.1.2 Update of a local installation

To update a local installation, proceed as follows.

- > Perform an installation with the new *requisis_MiX.exe* and adjust the installation path to your current installation path if necessary. The path should correspond to the current installation path.
- > Navigate to the *requisis_MiX* main directory the default path is `c:\REQUISIS\DXL\addins\REQUISIS\`.



requisis_MiX

- > Change to the `php\conf` subfolder and copy the contents of the `configuration.json.bak` file to the `configuration.json` file. Removing the `configuration.json` file and renaming the `configuration.json.bak` files to `configuration.json` is also possible here.
- > Change to the `config` subfolder and copy the contents of the `settings.inc.bak` file to the `settings.inc` file. Removing the `settings.inc` file and renaming the `settings.inc.bak` files to `settings.inc` is also possible here.

Now the newly installed *requisis_MiX* version should start from the menu items in DOORS. If there are any problems or if you have any questions, please contact our product support via the corresponding menu in *requisis_MiX*.

3.1.3 Network installation

If you want to make *requisis_MiX* available to multiple users, you can make an installation on a network share. Unzip the exe (e.g. using 7zip) to the desired location and create the registry keys to add *requisis_MiX* as DOORS-Extensions manually.

For DOORS 9.6 the key “Addins” located in “HKEY_CURRENT_USER\Software\ Telelogic\DOORS\9.6\Config” or in “HKEY_LOCAL_MACHINE\Software\ Telelogic\DOORS\9.6\Config” must be set the installation path.

Also change the value of the variable `MIX_INSTALL_PATH` in the configuration file (“`config\settings.inc`”) to the correct network path. Please note that Backslashes need to be escaped as `\\`.

Example: `string MIX_INSTALL_PATH=`
`"\\\\myserver\\myAddinsShare\\addins\\REQUISIS\\requisis_MiX";`

3.1.4 Update of a network installation

To update a network installation, proceed as follows:

- > Create a backup copy of the file `mixroot\php\configuration.json` file outside the *requisis_MiX* root directory.
- > Create a backup copy of the file `mixroot\config\settings.inc` file outside the *requisis_MiX* root directory.
- > Unzip the new `mix.exe` (e.g. using 7zip) and replace the content with your previous *requisis_MiX* main directory.
- > Copy the backup configuration.json file back to the `php\conf` subdirectory.
- > Copy the backup settings.inc file back to the `config` subdirectory.

Now the newly installed *requisis_MiX* version should start from the menu items in DOORS. If there are any problems or if you have any questions, please contact our product support via the corresponding menu in *requisis_MiX*.

3.2 Importing and setting up a Mapping Rule Module

To use a mapping rule module, that contains default mapping rules, a template must be imported to the DOORS Database. Therefore, the DMA-File in “`addins\REQUISIS\REQUISIS_MiX\moduleTemplates`” must be restored into a DOORS-Project named “*requisis_MiX*”.

Selecting another location for the mapping rule module (optional):

If the mapping rule module should be stored at another location, the variable `MIX_MAPPING_RULE_MODULE` in the settings file (“`config\settings.inc`”) must be adjusted accordingly to contain the path to the DOORS module.

Disabling the auto mapping functionality (optional):

To deactivate the automapping functionality, simply set the variable `MIX_MAPPING_RULE_MODULE` in the settings file (“`config\settings.inc`”) to an empty string. In this case no template module is required.



3.3 Setting up the Logging Module (optional)

By default, all migrations are logged into a logging module named “Logging Module” located in the project “requisis_MiX”. The module will be created the first time a migration is started. To use another location for the logging module simply adjust the variable `MIX_LOGGING_MODULE` in the settings file (“`config\settings.inc`”), the corresponding module will be created automatically.

3.4 Configuration of DOORS Next Generation

IMPORTANT: To make *requisis_MiX* work with DNG you need to setup an OAuth Consumer Key and Secret in the JTS Application of the Jazz Platform.

Do not setup the oAuth Consumer Key and Secret in the RM Application!

Instructions:

Open “[https://\(your-server:your-port\)/jts/admin#action=com.ibm.team.repository.admin.configureOAuth](https://(your-server:your-port)/jts/admin#action=com.ibm.team.repository.admin.configureOAuth)”.

The following page should open:

OAuth Consumers ?

Use this page to manage your OAuth consumers or to generate keys for new consumers. An OAuth c application, a Jazz Team Server, or a website that is allowed to send requests to this Jazz Team Serve user. Consumers sign their requests with an authorized consumer key and secret. For applications r Server setup wizard, consumer keys were generated automatically. Friend requests from other Jazz T generate a consumer key and secret that then require authorization. These requests that require auth Provisional Key section below. The Authorized Key section lists all authorized consumer keys. You are remove consumer keys. Trusted consumers will be able to share their authorization with other trustee prompted for approval to access data. Websites and applications external to your organization should

Register Consumer

Use the form below to generate an authorized OAuth consumer key.

Property	Value
Consumer Key	1 The key will be generated by the server Click here to pick the consumer key instead.
Consumer Name	2 requisis_dng_migration
Consumer Secret	3
Re-type Consumer Secret	4
Trusted	5 ☒

Register

- 1) Set **Consumer Key**: Either choose one or note down the key generated by the server.
- 2) Set **Consumer Name**: Set to “requisis_MiX” (or whatever you like)
- 3) Set **Secret Key**: Choose a Secret from upper- and lower-case letters and numbers. Special characters are not supported!
- 4) Set **Trusted**: Activate checkbox
- 5) Click on **Register**



3.5 Configure Server Details and OAuth Consumer Key

Open the file “addins/REQUISIS/REQUISIS_MiX/php/conf/configuration.json” in the *requisis_MiX* installation folder.

Name in Jazz Platform	Parameter in configuration.json
<i>Consumer Key</i>	oauth_clientId
<i>Secret Key</i>	oauth_clientSecret

Place your DNG server URL and OAuth consumer key here. You can configure multiple servers. The file should then look like the image below. (In this case 2 servers were configured)

```
{
  "serverTargets": [
    {
      "targetUrl": "https:\\\\your-server1:9443",
      "oauth_autoLogin": true,
      "oauth_clientId": "tbd",
      "oauth_clientSecret": "tbd",
      "maxSynchronousCurlRequests": 5
    },
    {
      "targetUrl": [
        "https:\\\\your-DNGserver2.net",
        "https:\\\\your-JTSserver.net"
      ],
      "oauth_autoLogin": true,
      "oauth_clientId": "tbd",
      "oauth_clientSecret": "tbd",
      "maxSynchronousCurlRequests": 5
    }
  ],
  "maxSynchronousCurlRequestsDefault": 5,
  "proxy": "",
  "indexerCheckInterval": 100,
  "preferPublicApi": true,
  "oleExtractionEnable": true,
  "zipBinaryData": true,
  "curlTimeout": 120,
  "curlConnectionTimeout": 60
}
```

Please note that / needs to be escaped with an \ in the json format.

You can also specify two different URLs for DNG/RM and JTS. To do this, create an array for the different URLs. The URLs in the array *do not* require a specific order. Note the configuration of the second server in the example above this text.



3.5.1 Settings in the configuration.json file

Under the ServerTargets section are the following options that apply to any connection to the above targets.

- > `maxSynchronousCurlRequestsDefault:`
Number of synchronously executed requests to the server. Defaultvalue: 5
- > `proxy:`
If necessary, an IP address and port for a proxy server can be entered here specify as (hostname/ip address) : (port) for the connections to the JAZZ server. No default value
- > `indexerCheckInterval:`
Interval in seconds while waiting for tasks on the JAZZ server Defaultvalue: 100
- > `preferPublicApi:`
Prefers to use the DNG public API before the private API Defaultvalue: true
- > `oleExtractionEnable:`
When disabled, all OLE containers are migrated as binary data. Defaultvalue: true
- > `zipBinaryData:`
Zips non-extractable OLE container. Defaultvalue: true
- > `curlTimeout:`
Timeout while waiting for a response from the JAZZ server in seconds. Defaultvalue: 120
- > `curlConnectionTimeout:`
Timeout while waiting for the connection to the JAZZ server in seconds. Defaultvalue: 60

3.5.2 SSL-client certificates for DNG Server Connections (optional)

This configuration step is only necessary if *SSL client certificates* are required for communication with the DNG server. To set up SSL client certificates for a specific server you need to edit the *configuration.json* file.

In the relevant section of the server concerned, insert the lines `ssl_client_cert_file`, `ssl_client_cert_keyfile` and `ssl_client_cert_keyfile_password`.

As shown here:

```
{
    "targetUrl": "https://your-server1:9443",
    "oauth_autoLogin": true,
    "oauth_clientId": "123ab12a",
    "oauth_clientSecret": "123ab12a",
    "ssl_client_cert_file": "conf/certs/ssl-client-cert.pem",
    "ssl_client_cert_keyfile": "conf/certs/ssl-client-cert.key.pem",
    "ssl_client_cert_keyfile_password": "yourSecretPassword",
    "maxSynchronousCurlRequests": 5
},
```

Put your SSL public and private certificates for the connection to your DNG server in this subfolder:

`php\conf\certs`

 requisis_MiX

3.5.3 Configure Auto Login

It is possible to add the two parameters *autologinUsername* and *autologinPassword* to a server target to enable an Auto Login for a DNG server. This Auto Login does not work together with the OIDC login.

The user used here must first be created in DNG.

```
{
  "targetUrl": "https:\\\\ your-server1:9443",
  "oauth_autoLogin": true,
  "oauth_clientId": "123ab12a ",
  "oauth_clientSecret": "specialSecret",
  "maxSynchronousCurlRequests": 20,
  "databaseCreatedPriorToVersion7": false,
  "autologinUsername": "dng-auto-user",
  "autologinPassword": "autosecret"
},
```

3.6 Start requisis_MiX

To use *requisis_MiX* use menu “REQUISIS/REQUISIS_MiX” from within a module. During the first use you will be asked for license information or to request an evaluation license.

NOTE: Evaluation licenses are limited to migrate 500 objects per module.

A user manual that describes the usage of *requisis_MiX* can be found in “addins/REQUISIS/REQUISIS_MiX/manual” within the installation folder. The user manual can also be opened by clicking on any *Help* button in the UI.



requisis_MiX

3.6.1 PHP settings and usage in terminal server environments (Citrix etc.)

If you run *requisis_MiX* in a terminal server environment such as Citrix. You must release the following files for execution in your Terminal Server environment:

- > php.exe
- > launcher.exe
- > php-win.exe

You will find these files in the *php/bin* subfolder of your *requisis_MiX* installation folder. You can test whether your terminal server environment allows execution by double-clicking the corresponding exe file. If you want to use your own PHP installation, you can specify this in the *settings.inc* file in the *config* subfolder under the parameters *MIX_PHP_CALLER* and *MIX_PATH_TO_PHP*. If the exe files are not released for execution, you will receive a DXL error message that looks something like this when executing a migration with *requisis_MiX*:

```
-R-E- DXL: <REQUISIS/requisis_MiX/lib/Shell.inc:75> SHELL: executing command
failed with following error - OLE problem: Access is denied.
, OLE error code: 0x%08x
Backtrace:
<REQUISIS/requisis_MiX/lib/Shell.inc:290>
<REQUISIS/requisis_MiX/lib/helpFunctions.inc:1793>
<REQUISIS/requisis_MiX/lib/helpFunctions.inc:1824>
<REQUISIS/requisis_MiX/controller/phpCaller.inc:26>
<REQUISIS/requisis_MiX/controller/callbackFunctions.inc:30>
<REQUISIS/requisis_MiX/controller/callbackFunctions.inc:98>
<REQUISIS/requisis_MiX/controller/callbackFunctions.inc:104>
<REQUISIS/requisis_MiX/gui/gui.inc:872>
<\\RBGS064A
ADAS_RM_Toolbox
NonPublic/addins/REQUISIS/requisis_MiX/doorsToDng.dxl:25>
```



4 Advanced Configuration

For most use cases *requisis_MiX* should work well with its default behavior and the settings provided by the UI. However, in some special use cases the user may want to adjust a certain behavior of the tool. Therefore, the file **"settings.inc"** offers additional parameters to adjust. You find this file in the subfolder **"config"** of your *requisis_MiX* installation. It is recommended creating a backup of this file before making any changes. This section gives a short overview over the available parameter.

4.1 Allow Objects without Type

Variable Name: `MIX_ALLOW_OBJECTS_WITHOUT_TYPE`

Description: Determines how to handle objects whose type is not defined. The object type is defined by setting a value for the object type definition attribute (Can be chosen in the UI).

If set to **false** (default): To start a migration the user has to set a value for the object type definition attribute for each object.

If set to **true**: In the "Type Mapping" dialog a new entry will become available in the list of DOORS types ("[[Object Type not set]]"). The type mapping of all objects with no value in their object type definition attribute will be determined by this entry.

BE CAREFUL: In this case all objects with undefined object type are mapped to the same DNG type!

4.2 Customization for merged attributes

If you merge multiple Doors attributes into one DNG attribute, these attributes are separated by a line break. Each new line starts with the attribute name of the Doors attribute. Attribute value and attribute name are separated by a colon and a space. It is possible to customize this behavior. For example, that each line after the attribute value ends with a semicolon. In the **"config"** folder, navigate to the **"userFunctions.inc"** file. There you will find the following `addAttributeLabel` function, which controls the output.

```
void addAttributeLabel(Buffer resultBuffer, Buffer textBuffer, string
attributeName, bool bShowAttributeLabelsInCollection, bool isHtml)
{
    if (bShowAttributeLabelsInCollection) {
        if (isHtml) {
            //add label only if not Object Text and Object Heading
            if (attributeName != "Object Text" && attributeName != "Object
Heading") {
                resultBuffer += "<b>" attributeName ":</b>";
            } else { //stop here, no \n or space
                return;
            }
        } else {
            resultBuffer += attributeName":";
        }
        //MIX_ATTRIBUTE_FLOATING_AMOUNT defined in settings.inc default:20;
        if (length(textBuffer) > MIX_ATTRIBUTE_FLOATING_AMOUNT) {
            resultBuffer += "\n";
        } else {
            resultBuffer += " ";
        }
    }
}
```



4.3 Deactivate the Check for Unsupported Fonts

Variable Name: `MIX_CHECK_FOR_UNSUPPORTED_FONTS`

Description: By default, *requisis_MiX* will check for unsupported fonts when preparing the migration. Setting this variable to "false" will deactivate this check.

4.4 Set the Default Choice for the Object Type Attribute

Variable Name: `MIX_DEFAULT_MAPPING_ATTRIBUTE`

Description: On the first page of the *requisis_MiX* GUI, the user can specify which DOORS attribute determines the object type of each object. This attribute determines which DOORS attribute is preselected when *requisis_MiX* is started for the first time in a module. If there is no attribute with this name in the module, the selection is empty.

4.5 Save OLE-Objects as .ole instead of .rtf

Variable Name: `MIX_EXPORT_GUI_OLE_RTF_OPTION`

Description: This parameter determines if DOORS OLE-Objects are embedded in RTF-files (Rich Text) before they are transferred to DNG. This has the advantage, that the files can be opened with MS-Word from DNG. The functionality is activated by default. If the parameter is set to false, all OLE-Objects are transferred as .ole files instead.

4.6 Show Empty Attributes in Collations (Merged Attributes)

Variable Name: `MIX_EMPTY_ATTRIBUTES_IN_COLLECTION`

Description: If several DOORS attributes are mapped to the same DNG attribute in the attribute mapping dialog, the values are merged. Thereby entries in the shape of "<attribute name>: <attribute value>" are concatenated. The parameter below decides what happens, if one of the values to merge is empty.

If set to **true**: An entry "<attribute name>:" is added for empty values.

If set to **false** (default): Nothing is added for empty values.

4.7 Adjust the LOG-Level

Variable Name: `MIX_LOGLEVEL`

Description: Defines how detailed the log messages are. The default is **WARN**, increase to **INFO** or **DEBUG** to create more detailed logs that can be provided to the support.

Choices are (In order of decreasing amount of information provided): **DEBUG**, **INFO**, **WARN**, **ERROR**, **FATAL_ERROR**, **PROGRESS**

4.8 Set the Default Link Type for external links

Variable Name: `MIX_EXTERNAL_LINK_TYPE_ALIAS`

Description: Defines the Default DNG link type alias for external link creation. To change this type copy a URI from the *link types* tab of the *components properties* of your target module.

Default setting: "<http://www.ibm.com/xmlns/rdm/types/Link>" (Link to)



4.9 Use operating systems default browser for DNG Login

Variable Name: `MIX_EXTERNAL_LOGIN`

Description: In some network environments or when using terminal server environments, it is possible that the browser configured in Doors cannot be used for logging into DNG.

If set to **true**: *requisis_MiX* uses the standard browser of the operating system on the host.

If set to **false** (default): *requisis_MiX* uses the browser configured in the Doors client.



4.10 Encryption of oauth keys in configuration.json

It is possible to encrypt the keys `oauth_clientId` and `oauth_clientSecret` as well as all values with the index `ssl_client_cert_keyfile_password` in the *configuration.json* file.

To do this,

- > Open the php/config subfolder of your *requisis_MiX* installation and locate the file *configuration.json*.
- > Copy *configuration.json* into the php subfolder of the same installation.
- > Start encryption by dragging *configuration.json* onto *encrypt_configuration.cmd* in the php subfolder.
- > When the console window opens, confirm with a key press. A new encrypted file named *configuration.encrypted.json* is then created, with the keys `oauth_clientId` and `oauth_clientSecret`, as well as all values with the index `ssl_client_cert_keyfile_password`, encrypted.
- > Copy *configuration.encrypted.json* back into the php/config subfolder so that *requisis_MiX* uses it.
- > Keep the original unencrypted *configuration.json* in a secure location outside the installation directory.

If you want to disable encryption, copy the unencrypted file back into the php/config subfolder and remove the file *configuration.encrypted.json*.

4.11 Support for changing the DNG server domain

requisis_MiX supports you when you move your DNG server to another domain. To ensure that modules that have already been migrated can still be updated, you will find the DXL script *changeDomain.dxl* in the *tools* subfolder of your *requisis_MiX* installation directory. The script can rewrite all MIX attributes of the current module from your old DNG domain to the new domain. Before you run the script, you must fill in the two parameters `OLD_DOMAIN` and `NEW_DOMAIN` with the respective domain.

Example:

```
OLD_DOMAIN = "my.old-jazz.com"
```

```
NEW_DOMAIN = "the.new-jazz.com"
```

To make the fully configured script available from the *Tools* menu in the *requisis_MiX* entry of the corresponding DOORS module, you also need to edit the *tools.idx* file. Add the following line to the file:

```
changeDomain _ _ Change DNG Server Domain for current module
```

You can customize the text after the two underscores (`_ _`) as needed.

In order for *requisis_MiX* to connect to the new DNG domain, the *configuration.json* file must also be adjusted. For more information, see chapter 3.5 Configure Server Details and OAuth Consumer Key of this manual.

It is also possible to transfer this script to your own DXL script to migrate more than one module to the new domain.



requisis_MiX

5 API

It is possible to control *requisis_MiX* also via API. Before you start the migration via API, the modules to be migrated must be configured accordingly via GUI for the migration. To use the API you have to include the file *mixBatchIncludes.inc* from the subfolder *dxl/config* into your script. In the subfolder *dxl/api* you will find further examples for the use of the API.

5.1 API Functions

```
syncToDng(string sFullModuleName, bool bShowGui)
```

Function to migrate a DOORS formal module into a DNG module artifact.

Parameters:

- > sFullModuleName: DXL fullname of the formal module you want to migrate.
- > bShowGui: true shows progressbar and result window, false shows nothing

```
syncLinksToDng(string sLinkSyncModuleName, bool bShowGui)
```

Function to migration links configured in a *requisis_MiX* link module to DNG.

Parameters:

- > sLinkSyncModuleName: DXL fullname of the *requisis_MiX* link module. The module from which you start the *requisis_MiX* link migrations gui.
- > bShowGui: true shows progressbar and result window, false shows nothing

```
syncMigrationManagerToDng(string sMigrationManagerModuleName, bool bShowGui)
```

Function to migration formal modules configured in a *requisis_MiX* migration manager module to DNG.

Parameters:

- > sMigrationManagerModuleName: DXL fullname of the *requisis_MiX* migration manager module. The module from which you start the *requisis_MiX* migration manager.
- > bShowGui: true shows progress bar and result window, false shows nothing



5.2 DXL Example

Code example for DXL users: Migration/synchronization of a module with preparation steps.

```
#include <../addins/REQUISIS/requisis_MiX/config/mixIncludes.inc>

#pragma encoding, "UTF-8"

#pragma runLim, 0;

Regexp resultIsError          = regexp2 "ERROR:.*"
Regexp resultLogFileName      = regexp2 "syncResult:(.*)"

string modNameDC = "/ImportantProject/Specifications"
string viewName = "Migration"
string attrType = "ObjectType"
string TargetFolderURI = "https://myDNGServer.de/rm/folders/FR_ABC"
string dngModuri = "newModcheck"
string dngModType = "Requirements Document"

//should only be run once per module

string modcheck = initNewModule(modNameDC, TargetFolderURI, dngModuri, dngModType, viewName,
attrType, true, true)

print modcheck "\n"

//init module data and run automapping
Buffer bError = create();
//important! variable name must be mCurrent
mCurrent= edit(modName, true, true, true);
syncConfiguration = getSyncConfiguration(mCurrent);
initObjectMapping(mCurrent, objectMapping);

//load Datamodel
artifactMapping = getDngDatamodel(getDngModuleUrl(), uniqueID(mCurrent), false);
purifyArtifactMapping(artifactMapping);
```



```
//run automapping
bool hasUnmappedElements = runAutomapping(mCurrent, bError)
//save data
saveSyncConfiguration(mCurrent);
saveCurrentModule(mCurrent);
if (hasUnmappedElements ) {
    print stringOf(bError)
} else {
    //try to sync module to dng
    string result = syncToDng(modName , true)
    print result
}
delete(bError)
```




REQUISIS GmbH

Walter-Benjamin-Platz 8, D-10629 Berlin, Germany

Telefon +49 30 / 53 65 06 - 700; Fax +49 30 / 53 65 06 - 300

REQUISIS GmbH

Willy-Brandt-Str. 54, D-70173 Stuttgart, Germany

Telefon +49 711 / 528 529 - 700; Fax +49 711 / 528 529 - 300

REQUISIS GmbH Product Support

DE-Hotline +49 30 536 506 - 666

US-Hotline 1-844-REQUISI, ext. 666

product-support@requisis.com, <https://requisis.com>