

電子印鑑システム

パソコン決裁6

D i g i t a l S t a m p S e r i e s

E x t e n s i o n K i t

サービスリリース 1 試用版

リファレンス マニュアル

電子印鑑 XML Extension 編



内容

1	はじめに	1
2	電子印鑑 XML Extension の概要	1
3	電子印鑑 XML Extension のファイル名	1
4	64 ビット版に関する注意事項	1
5	セットアップ	1
6	電子印鑑 XML Extension の操作	2
6.1	VBScript (Visual Basic Script) の場合	2
6.2	Visual Basic (Microsoft Visual Basic 2008) の場合	3
6.3	Visual C# (Microsoft Visual C# 2008) の場合	4
7	メソッドとプロパティ	6
	XF_DsmOpen	6
	XF_DsmClose	7
	XF_GetXml	8
	GetVersion	9
	ResetError	10
	GetLastErrorNumber	11
	GetLastErrorMessage	12
	GetEdition	13
8	XML 定義	14
8.1	XML ヘッダー	14
8.2	XML ルート <dstmpfile>	14
8.3	設定タグ <setting>	14
8.4	捺印用印鑑データファイルのルートタグ <root>	14
8.5	グループタグ <group>	14
8.6	ユーザタグ <user>	15
9	戻り値定数一覧	16

1 はじめに

このドキュメントには、[パソコン決裁6 Extension Kit]内の電子印鑑XML Extension モジュールについての説明がされています。また、本ドキュメントには Windows オペレーティングシステムの操作方法やパソコン決裁6についての操作方法などを理解している方を対象に記載されています。

試用版のご利用について

ダウンロードいただいた、モジュールおよび関連するドキュメントは、ご評価を目的としたモジュールになります。そのため、本モジュールを利用して発生したコンピュータについての障害や現象については、当社は一切の保障を行いません。また、本モジュールにつきましてのご質問やお問い合わせについてはご回答を差し上げることができませんので、あらかじめご了承ください。

2 電子印鑑 XML Extension の概要

電子印鑑 XML Extension は参照された捺印用印鑑データファイル(拡張子.DSM)から、グループやユーザ、捺印ログなどの情報を取得し、XML フォーマットに成型して出力を行います。

3 電子印鑑 XML Extension のファイル名

電子印鑑 XML Extension は次のファイルで構成されています。

名称	ファイル名	機能
電子印鑑 XML Extension (x86)	DstmpXml.dll	電子印鑑の管理状態を XML で出力するモジュール 32 ビット
電子印鑑 XML Extension (x64)	DstmpXml.dll	電子印鑑の管理状態を XML で出力するモジュール 64 ビット

4 64 ビット版に関する注意事項

64 ビット版をご利用される場合には、以下の事項に注意してご利用ください。

- ・32 ビットオペレーティングシステム上では、動作しません。
- ・登録を行うコマンド("Regsvr32"など)は、64 ビット用をご利用ください。
- ・Windows サービスからの呼び出しや Web サービスからの呼び出しなど、呼び出し元による制約を受ける場合があります。

5 セットアップ

電子印鑑 XML Extension のインストールは次のように行います。

1. メディア内の[bin]フォルダから[DstmpXml.dll]を適当なフォルダにコピーします。
登録の際に、英数字以外の文字が利用されている場合(例:"C:¥サンプル" フォルダなど)では正常に動作しない場合がございますので必ず半角英数字(例:"C:¥ComModule" フォルダなど)が利用されたフォルダを指定してください。
2. コマンドプロンプトを開きます。
Windows Vista の場合には管理権限で操作を行う必要があります。
3. 表示されたコマンドプロンプトの画面に次のように入力します。
regsvr32 <コピーしたモジュールのパス>¥DstmpXml.dll
4. 登録が成功した旨のメッセージが表示されます。

6 電子印鑑 XML Extension の操作

電子印鑑 XML Extension で処理を行うために次のような実装を行います。

6.1 VBScript (Visual Basic Script) の場合

電子印鑑 XML Extension を VBScript で実装するためには次のような手順で行います。以下に手順またはコードは、実装を理解しやすくするための例示になります。開発される環境などによって操作方法や名称などが異なる場合がございますので、あらかじめご了承ください。

1. メモ帳を起動します。
2. 起動したメモ帳に以下のコードを入力します。

```
Dim objDstmpXml
Dim strXmlResult
Dim XD_USERANDGROUPS:XD_USERANDGROUPS = 4
Set objDstmpXml = CreateObject("DstmpXml.StmpXml")
objDstmpXml.XF_DsmOpen("<捺印用印鑑データファイルの場所>", "<開くパスワード>")
strXmlResult = objDstmpXml.XF_GetXml(XD_USERANDGROUPS)
objDstmpXml.XF_DsmClose()
Set objDstmpXml = Nothing
msgbox "プログラムの実行が終了しました"
```

3. [ファイル]メニュー内の[名前を付けて保存]メニューを選択します。
4. 表示された[名前を付けて保存]画面で保存する場所を選択して[ファイル名]に"sample.vbs"と入力します。
5. [保存]ボタンを選択します。
6. 保存されたファイルをダブルクリックして実行を行います。
7. "プログラムの実行が終了しました"というメッセージが表示されます。

6.2 Visual Basic(Microsoft Visual Basic 2008)の場合

電子印鑑 XML ExtensionをVisual Basicで実装するためには次のような手順で行います。以下に手順またはコードは、実装を理解しやすくするための例示になります。開発される環境などによって操作方法や名称などが異なる場合がございますので、あらかじめご了承ください。

1. Visual Basic を起動します。
2. [ファイル | 新規作成 | プロジェクト]の順にメニューを選択します。
3. [新しいプロジェクト]画面で[プロジェクトの種類]に[Visual Basic]が選択されていることを確認します。
(選択されていない場合には、[他の言語]などから探して選択します)
4. [テンプレート]から[Windows フォーム アプリケーション]を選択します。
5. [プロジェクト名]に"DstmpWindowsApplication"と入力して[OK]ボタンを選択します。
6. 新しいプロジェクトファイルが作成され、[Form1.vb]のデザイン画面が表示されます。
7. [プロジェクト | 参照の追加]メニューを選択します。
8. 表示された[参照の追加]画面で[COM]タブを選択し表示されている[コンポーネント名]一覧から"DstmpXml"を選択して[OK]ボタンを選択します。
9. [表示 | ツールボックス]メニューを選択します。
10. 表示された[ツールボックス]から[コモン コントロール]内の[Button]をマウスの左ボタンで選択し、マウスの左ボタンを押したままの状態ですべてのフォーム内に移動します。
11. [Form1]フォーム内の適当な場所でマウスの左ボタンを離します。
(フォーム内にボタンコントロールが追加されます)
12. 追加されたボタンコントロールをマウスの左ボタンでダブルクリックします。
13. 次のようなコードが追加されます。

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    End Sub
End Class
```

14. 追加されたコード内の"Private Sub" と、"End Sub" の間に次のコードを挿入します。

```
Dim objDstmpXml As DstmpXml.StmpXml
objDstmpXml = CreateObject("DstmpXml.StmpXml")
objDstmpXml.XF_DsmOpen("<捺印用印鑑データファイルの場所>", "<開くパスワード>")
objDstmpXml.XF_GetXml(4)
objDstmpXml.XF_DsmClose()
objDstmpXml = Nothing
MsgBox("プログラムの実行が終了しました")
```

15. [デバッグ | デバッグ開始]メニューを選択します。
16. ビルドが開始され、フォームが表示されます。
17. 表示された[Button1]ボタンを選択します。
18. "プログラムの実行が終了しました"というメッセージが表示されます。

6.3 Visual C#(Microsoft Visual C# 2008)の場合

電子印鑑 XML Extension を C# で実装するためには次のような手順で行います。以下に手順またはコードは、実装を理解しやすくするための例示になります。開発される環境などによって操作方法や名称などが異なる場合がございますので、あらかじめご了承ください。

1. Visual C#を起動します。
2. [ファイル | 新規作成 | プロジェクト]の順にメニューを選択します。
3. [新しいプロジェクト]画面で[プロジェクトの種類]に[Visual C#]が選択されていることを確認します。
(選択されていない場合には、[他の言語]などから探して選択します)
4. [テンプレート]から[Windows フォーム アプリケーション]を選択します。
5. [プロジェクト名]に"DstmpWindowsApplication"と入力して[OK]ボタンを選択します。
6. 新しいプロジェクトファイルが作成され、[Form1.cs]のデザイン画面が表示されます。
7. [プロジェクト | 参照の追加]メニューを選択します。
8. 表示された[参照の追加]画面で[COM]タブを選択し表示されている[コンポーネント名]一覧から"DstmpXml"を選択して[OK]ボタンを選択します。
9. [表示 | ツールボックス]メニューを選択します。
10. 表示された[ツールボックス]から[コモン コントロール]内の[Button]をマウスの左ボタンで選択し、マウスの左ボタンを押したままの状態ですべてのフォーム内に移動します。
11. [Form1]フォーム内の適当な場所でマウスの左ボタンを離します。
(フォーム内にボタンコントロールが追加されます)
12. 追加されたボタンコントロールをマウスの左ボタンでダブルクリックします。
13. 次のようなコードが追加されます。

```
private partial class Form1 : Form
{
    ...省略...
    private void button1_Click(object sender, EventArgs e)
    {

    }
}
```

14. 追加されたコード内の"private void button1_Click(...{" と、"}" の間に次のコードを挿入します。

```
Type type;
int XD_USERANDGROUPS = 4;
type = Type.GetTypeFromProgID("DstmpXml.StmpXml");
DstmpXml.IStmpXml objDstmpXml = (DstmpXml.IStmpXml)Activator.CreateInstance(type);
objDstmpXml.XF_DsmOpen("<捺印用印鑑データファイルの場所>", "<開くパスワード>");
MessageBox.Show(objDstmpXml.XF_GetXml(XD_USERANDGROUPS));
objDstmpXml.XF_DsmClose();
MessageBox.Show(objDstmpXml.XF_GetXml(XD_USERANDGROUPS));
MessageBox.Show("プログラムの実行が終了しました");
```

15. [デバッグ | デバッグ開始]メニューを選択します。
16. ビルドが開始され、フォームが表示されます。
17. 表示された[Button1]ボタンを選択します。
18. "プログラムの実行が終了しました"というメッセージが表示されます。

7 メソッドとプロパティ

電子印鑑 XML Extension > IStmpXml	メソッド
XF_DsmOpen	
捺印用印鑑データファイルを開きます。	
long XF_DsmOpen(string strFilePath, string strPassword)	
<p>パラメータ</p> <p>strFilePath: 対象となる捺印用印鑑データファイルの場所 strPassword: 設定されている開くパスワード</p> <p>戻り値</p> <p>long</p> <p>成功した場合には DSM_SUCCESS(1)が戻されます。 失敗した場合には次の値が戻されます。 E_DSM_CANNOT_OPEN(-3) : 捺印用印鑑データファイルを開くことができません E_STFV3LIB(-4) : 内部ライブラリエラー E_UNMATCH_PASSWORD(-6) : 開くパスワードが不正です E_DSM_PATH(-8) : 捺印用印鑑データファイルの場所が不正です 詳細なエラーを取得するには GetLastErrorNumber メソッドまたは GetLastErrorMessage メソッドを利用します。</p> <p>備考</p>	
<p>使用例</p> <p>long nResult = object.XF_DsmOpen("捺印用印鑑データファイルの場所", "開くパスワード")</p>	
<p>サンプルコード</p> <pre>Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルを開く res = stamp.XF_DsmOpen("C:\¥STMP¥STMPDAT.DSM", "1111") '結果表示 If res = 1 Then WScript.Echo "捺印用印鑑データファイルを開きました。" Else WScript.Echo "捺印用印鑑データファイルを開けませんでした。" & vbCrLf & stamp.GetLastErrorMessage WScript.Quit End If '捺印用印鑑データファイルを閉じる stamp.XF_DsmClose() WScript.Echo "捺印用印鑑データファイルを閉じました。" Set stamp =Nothing</pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
XF_DsmClose	
捺印用印鑑データファイルを閉じます。	
void XF_DsmClose(void)	
<p>パラメータ ありません</p> <p>戻り値 ありません</p> <p>備考</p>	
<p>使用例 object.XF_DsmClose()</p>	
<p>サンプルコード</p> <pre> Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルを開く res = stamp.XF_DsmOpen("C:\YSTMP\YSTMPDAT.DSM", "1111") '結果表示 If res = 1 Then WScript.Echo "捺印用印鑑データファイルを開きました。" Else WScript.Echo "捺印用印鑑データファイルを開けませんでした。" & vbCrLf & stamp.GetLastErrorMessage WScript.Quit End If '捺印用印鑑データファイルを閉じる stamp.XF_DsmClose() WScript.Echo "捺印用印鑑データファイルを閉じました。" Set stamp =Nothing </pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
XF_GetXml	
ユーザまたはグループなどの情報を XML 形式で取得します。	
string XF_GetXml(long nItemKind)	
<p>パラメータ</p> <p>nItemKind: 次のいずれかの値を指定します。</p> <p>XD_USERS(1) : ユーザに設定されている値を取得します</p> <p>XD_USERS_FULL(2) : ユーザに設定されているすべての値を取得します</p> <p>XD_GROUPS(3) : グループに設定されている値を取得します</p> <p>XD_USERANDGROUPS(4) : ユーザとグループに設定されている値を取得します</p> <p>XD_USERANDGROUPS_FULL(5) : ユーザとグループに設定されているすべての値を取得します</p> <p>戻り値</p> <p>string</p> <p>成功した場合には取得された情報が XML 形式で戻されます。</p> <p>失敗した場合には空文字が戻されます。</p> <p>詳細なエラーを取得するには GetLastErrorNumber メソッドまたは GetLastErrorMessage メソッドを利用します。</p> <p>備考</p> <p>取得される値については「XML 出力される属性情報」を参照してください</p>	
<p>使用例</p> <pre>string strResult = object.XF_GetXml(1)</pre>	
<p>サンプルコード</p> <pre>Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルを開く res = stamp.XF_DsmOpen("C:\YSTMP\YSTMPDAT.DSM", "1111") 'ユーザに設定されている値を取得 users = stamp.XF_GetXml(1) 'ユーザに設定されているすべての値を取得 usersFull = stamp.XF_GetXml(2) 'グループに設定されている値を取得 groups = stamp.XF_GetXml(3) 'ユーザとグループに設定されている値を取得 userandgroups = stamp.XF_GetXml(4) 'ユーザとグループに設定されているすべての値を取得 userandgroupsFull = stamp.XF_GetXml(5) '捺印用印鑑データファイルを閉じる stamp.XF_DsmClose() Set stamp = Nothing '結果をファイルに保存 Set FS = CreateObject("Scripting.FileSystemObject") Set file = FS.CreateTextFile("C:\test\XD_USERS.XML", True, True) '第 2 引数 : 上書きモード、第 3 引数 : Unicode file.WriteLine(users) file.Close Set file = FS.CreateTextFile("C:\test\XD_USERSFULL.XML", True, True) '第 2 引数 : 上書きモード、第 3 引数 : Unicode file.WriteLine(usersFull) file.Close Set file = FS.CreateTextFile("C:\test\XD_GROUPS.XML", True, True) '第 2 引数 : 上書きモード、第 3 引数 : Unicode file.WriteLine(groups) file.Close Set file = FS.CreateTextFile("C:\test\XD_USERANDGROUPS.XML", True, True) '第 2 引数 : 上書きモード、第 3 引数 : Unicode file.WriteLine(userandgroups) file.Close Set file = FS.CreateTextFile("C:\test\XD_USERANDGROUPFULL.XML", True, True) '第 2 引数 : 上書きモード、第 3 引数 : Unicode file.WriteLine(userandgroupsFull) file.Close Set FS = Nothing</pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
GetVersion	
バージョン情報を取得します。	
string GetVersion(void)	
<p>パラメータ ありません</p> <p>戻り値 string ライブラリのバージョン情報</p> <p>備考 取得されるバージョン情報は 0.00.000.0000 形式で取得されます</p>	
<p>使用例 string strResult = object.GetVersion()</p>	
<p>サンプルコード</p> <pre>Set stamp = CreateObject("DSTMPXml.StmpXml") 'ライブラリモジュールのバージョンを取得 res = stamp.GetVersion() '結果表示 WScript.Echo res set stamp = Nothing</pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
ResetError	
内部エラー情報を初期化します。	
void ResetError(void)	
<p>パラメータ ありません</p> <p>戻り値 ありません</p> <p>備考 初期化後の GetLastErrorNumber メソッドは DSM_SUCCESS(1)を GetLastErrorMessage メソッドは"エラーはありません"を戻します。</p>	
<p>使用例 object.ResetError()</p>	
<p>サンプルコード</p> <pre> Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルパスを指定しないことで、エラーを発生させる res = stamp.XF_DsmOpen("", "") 'エラー情報取得 If res <> 1 Then 'エラー番号を取得 msg = "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf 'エラーメッセージを取得 msg = msg & "ErrorMessage: " & stamp.GetLastErrorMessage() End If '結果表示 WScript.Echo msg 'エラー情報を初期化 stamp.ResetError() '結果表示 WScript.Echo "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf & _ "ErrorMessage: " & stamp.GetLastErrorMessage() Set stamp =Nothing </pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
GetLastErrorNumber	
インタフェース内のメソッドで最後に発生したエラー番号を取得します。	
long GetLastErrorNumber(void)	
<p>パラメータ ありません</p> <p>戻り値 long 最終エラー番号</p> <p>備考</p>	
<p>使用例 long nResult = object.GetLastErrorNumber()</p>	
<p>サンプルコード</p> <pre> Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルパスを指定しないことで、エラーを発生させる res = stamp.XF_DsmOpen("", "") 'エラー情報取得 If res <> 1 Then 'エラー番号を取得 msg = "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf 'エラーメッセージを取得 msg = msg & "ErrorMessage: " & stamp.GetLastErrorMassage() End If '結果表示 WScript.Echo msg 'エラー情報を初期化 stamp.ResetError() '結果表示 WScript.Echo "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf & _ "ErrorMessage: " & stamp.GetLastErrorMassage() Set stamp =Nothing </pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
GetLastErrorMessage	
インタフェイス内のメソッドで最後に発生した定義済みのエラーメッセージを取得します。	
string GetLastErrorMessage(void)	
<p>パラメータ ありません</p> <p>戻り値 string 最終エラーメッセージ</p> <p>備考</p>	
<p>使用例</p> <pre>string strResult = object.GetLastErrorMessages();</pre>	
<p>サンプルコード</p> <pre>Set stamp = CreateObject("DSTMPXml.StmpXml") '捺印用印鑑データファイルパスを指定しないことで、エラーを発生させる res = stamp.XF_DsmOpen("", "") 'エラー情報取得 If res <> 1 Then 'エラー番号を取得 msg = "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf 'エラーメッセージを取得 msg = msg & "ErrorMessage: " & stamp.GetLastErrorMessages() End If '結果表示 WScript.Echo msg 'エラー情報を初期化 stamp.ResetError() '結果表示 WScript.Echo "ErrorNumber: " & stamp.GetLastErrorNumber() & vbCrLf & _ "ErrorMessage: " & stamp.GetLastErrorMessages() Set stamp = Nothing</pre>	

電子印鑑 XML Extension > IStmpXml	メソッド
GetEdition	
ライブラリのエディションを取得します。	
long GetEdition(void)	
<p>パラメータ ありません</p> <p>戻り値 long ライブラリのエディションが戻ります。 LIB_EDITION_TRIAL(0) : 試用版 LIB_EDITION_STANDARD(1) : 製品版 LIB_EDITION_TRIAL_X64(2) : 64 ビット製品版 LIB_EDITION_STANDARD_X64(3) : 64 ビット試用版</p> <p>備考 製品のエディションは、ファイルのプロパティのバージョン情報の製品名から確認することができます。</p>	
<p>使用例 long nResult = object.GetEdition()</p>	
<p>サンプルコード</p> <pre>Set stamp = CreateObject("DSTMPXml.StmpXml") 'ライブラリモジュールのバージョンを取得 version = stamp.GetVersion() 'ライブラリモジュールのエディションを取得 edition = stamp.GetEdition() '結果表示 WScript.Echo "バージョン" & vbTab & version & vbCrLf & _ "エディション" & vbTab & edition Set stamp = Nothing</pre>	

8 XML 定義

IStmpXml::GetXml メソッドで取得される XML 形式のタグと属性は次のように定義される

8.1 XML ヘッダー

すべての XML 形式の情報については次のヘッダー情報が先頭部に出力されます。

出力される XML ヘッダー

```
<?xml version="1.0" encoding="utf-8">
```

8.2 XML ルート <dstmpfile>

すべての XML 形式の情報については次のルートタグで囲まれます。

タグ名	属性名	内容
dstmpfile	なし	出力される XML ファイルのルートを意味します

8.3 設定タグ <setting>

捺印用印鑑データファイルの設定情報が含まれます

タグ名	属性名	内容
setting	なし	捺印用印鑑データファイルの設定情報を意味します
link	なし	ActiveDirectory/ドメインへの関連付けの設定情報
	id	関連付けの設定 0=なし 1=ドメインコントローラ 2=Active Directory
	type	関連付けの設定 "none"=なし "DC"=ドメインコントローラ "LDAP"=Active Directory
	domainname	関連付けされるドメイン名
	domaincontrollername	関連付けされているドメインコントローラ名

8.4 捺印用印鑑データファイルのルートタグ <root>

捺印用印鑑データファイルのルート情報が含まれます

タグ名	属性名	内容
root	name	捺印用印鑑データファイルの「ルート名」
	description	捺印用印鑑データファイルの「説明」
	id	捺印用印鑑データファイルの「CID」
	position	ルートを示すファイル内の位置情報

8.5 グループタグ <group>

捺印用印鑑データファイルのグループ情報が含まれます

タグ名	属性名	内容
group	name	グループの「グループ名」または組織単位 (OU) の「名前」
	description	グループの「説明」(ローカルグループのみ)
	id	グループの「CID」(ローカルグループのみ)
	GUID	組織単位 (OU) の GUID 値 (Active Directory のみ)
	type	グループの種別 0=ローカルグループ 1=ドメインコントローラ 2=組織単位 (OU)
	position	グループを示すファイル内の位置情報

8.6 ユーザタグ <user>

捺印用印鑑データファイルのユーザ情報が含まれます

XD_USERS の場合

タグ名	属性名	内容
user	name	ユーザの「ユーザ名」またはドメインユーザの「samAccountName」
	id	ユーザの「CID」
	GUID	ドメインユーザの SID 値(ドメインコントローラのみ) ドメインユーザの GUID 値(Active Directory のみ)
	type	ユーザの種別 0=ローカルユーザ 1=ドメインユーザ
	username	ユーザの「姓」と「氏名」の組み合わせ

XD_USERS_FULL の場合

タグ名	属性名	内容
user	name	ユーザの「ユーザ名」またはドメインユーザの「samAccountName」
	description	ユーザの「説明」(ローカルユーザのみ)
	id	ユーザの「CID」
	GUID	ドメインユーザの SID 値(ドメインコントローラのみ) ドメインユーザの GUID 値(Active Directory のみ)
	type	ユーザの種別 0=ローカルユーザ 1=ドメインユーザ
	passwordminlength	ユーザの「パスワード最小文字数」(ローカルユーザのみ)
	firstname	ユーザの「氏名」
	middlename	ユーザの「ミドルネーム」
	lastname	ユーザの「姓」
	email	ユーザの「電子メールアドレス」
	enable	ユーザの「ユーザを無効にする」
	listup	ユーザの「捺印ツールのユーザ名リストに表示する」
	disablecheckout	ユーザの「チェックアウト機能の利用を禁止する」
	level	ユーザの「役職 ID」
	levelname	ユーザの「役職名」
	disableconfig	ユーザの「ユーザによるオプション設定の変更を禁止する」
	windowsauth	ユーザの「Windows 認証」に関連付けされたドメインアカウント名
	kind	ユーザの区分(通常ユーザ/管理者ユーザ)

9 戻り値定数一覧

	定数	内容
DSM_SUCCESS	1	エラーはありません
DSM_SUCCESS_FALSE	0	エラーが発生しました
E_DSM_NOT_OPEN	-1	捺印用印鑑データファイルが開いていませ
E_GROUP_NOT_FOUND	-2	グループが見つかりません
E_DSM_CANNOT_OPEN	-3	捺印用印鑑データファイルが開けません
E_STFV3LIB	-4	内部ライブラリエラー
E_USER_NOT_FOUND	-5	ユーザが見つかりません
E_UNMATCH_PASSWORD	-6	パスワードが一致しません
E_NOT_CERTIFICATION	-7	認証されていません
E_DSM_PATH	-8	捺印用印鑑データファイルのパスに問題があります
E_UNMATCH_VALUE	-22	引数で設定された値が不正です
E_XML_CREATE_ELEMENT	-3001	XML 要素の作成に失敗しました
E_XML_SET_ELEMENT_ATTRIBUTE	-3002	XML 要素の属性情報の追加に失敗しました
E_XML_APPEND_ELEMENT	-3003	XML 要素の追加に失敗しました
E_XML_APPEND_DOCUMENT	-3004	XML ドキュメントの追加に失敗しました
E_DWA_CANNOT_OPEN	-4001	DWA ファイルのオープンに失敗しました
E_CONNECT_DOMAIN	-4002	ドメインへの接続に失敗しました
E_UNKNOWN_ERROR	-9000	エラーメッセージが定義されていません

著作権情報

© 2010 Shachihata Inc. All rights reserved.

本マニュアルの内容は、著作権法により保護されています。

本マニュアルが使用許諾契約を含むソフトウェアと共に提供される場合、本マニュアルおよびその中に記載されているソフトウェアは、使用許諾契約にもとづいて提供されるものであり、当該使用許諾契約の契約条件に従ってのみ使用または複製することが可能となるものです。本マニュアルのいかなる部分も、発行者（シャチハタ株式会社）への書面による許可なしに、いかなる形式・手段でも、複製、検索システムへの保存、または伝送を行うことはできません。

本マニュアルに記載される内容は、可能な限り正確であることを考慮しますが予告なしに変更されることを条件として提供されるものであります。従って、本マニュアルの情報がシャチハタ株式会社による確約として解釈されるものではありません。シャチハタ株式会社は、本マニュアルにおけるいかなる誤りまたは不正確な記述に対しても、いかなる義務や責任を負うものではありません。

本マニュアルに記載されている会社名、製品名などは各社の商標または商標登録です。

電子印鑑 XML Extension サービスリリース 1
リファレンスマニュアル

2010/06 発行

発行者

シャチハタ株式会社

<http://www.shachihata.co.jp/>

パソコン決裁についての情報

<http://www.shachihata.co.jp/interweb/>