

2.3 Directory Service Schema Elements

The Terminal Services Terminal Server Runtime Interface accesses the following **directory service** attributes in the **user** class listed in the following table.

For the syntactic specifications of the following attributes, refer to Active Directory Domain Services (AD/DS) ([\[MS-ADA1\]](#), [\[MS-ADA2\]](#), [\[MS-ADA3\]](#), and [\[MS-ADSC\]](#)).

Directory service attributes	Description
msTSAllowLogon	Not used by Microsoft Terminal Services.
msTSBrokenConnectionAction	Not used by Microsoft Terminal Services.
msTSConnectClientDrives	Not used by Microsoft Terminal Services.
msTSConnectPrinterDrives	Not used by Microsoft Terminal Services.
msTSDefaultToMainPrinter	Not used by Microsoft Terminal Services.
msTSHomeDirectory	Not used by Microsoft Terminal Services.
msTSHomeDrive	Not used by Microsoft Terminal Services.
msTSInitialProgram	Not used by Microsoft Terminal Services.
msTSMaxConnectionTime	Not used by Microsoft Terminal Services.
msTSMaxDisconnectionTime	Not used by Microsoft Terminal Services.
msTSMaxIdleTime	Not used by Microsoft Terminal Services.
msTSProfilePath	Not used by Microsoft Terminal Services.
msTSReconnectionAction	Not used by Microsoft Terminal Services.
msTSRemoteControl	Not used by Microsoft Terminal Services.
msTSWorkDirectory	Not used by Microsoft Terminal Services.
userParameters	This attribute contains a binary blob composed of various fields of the USERCONFIG structure returned by RpcGetConfigData . For the binary blob structure, please see UserParameters (section 2.3.1) .

Note Any of the previously defined properties can be set for either a user or a machine. Machine properties MAY override user properties and hence the value returned by the RPC calls as previously defined will depend on whether machine properties are set in addition to the user properties.

2.3.1 UserParameters

The userParameters attribute is used by Microsoft Terminal Services to store the configuration data associated with the user connected to a particular session running on a terminal server. This configuration data is returned in a [USERCONFIG](#) structure by the [RpcGetConfigData](#) method. Microsoft Terminal Services stores the user configuration data in the userParameter attribute in the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MSProductData																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(MSProductData cont'd for 16 rows)																															
Signature																TSPropertyCount															
TSPropertyArray (variable)																															
...																															

MSProductData (96 bytes): A 96 byte Unicode character array containing 48 Unicode characters. This field is not used by Microsoft Terminal Services.

Signature (2 bytes): A 2 byte Unicode character. This field is used by Microsoft Terminal Services to assert the validity of the TSPropertyArray field. It can only contain Unicode character 'P'.

TSPropertyCount (2 bytes): A 2 byte unsigned integer indicating the number of elements in TSPropertyArray.

TSPropertyArray (variable): A variable length array of [TSProperty](#) structures. The number of elements in this array is specified by the field TSPropertyCount.

2.3.2 TSProperty

Following is the format of each TSProperty structure:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NameLength																ValueLength															

Type	PropName (variable)
...	
PropValue (variable)	
...	

NameLength (2 bytes): A 2 byte unsigned integer indicating length of the PropName field in bytes.

ValueLength (2 bytes): A 2 byte unsigned integer indicating length of the PropValue field in bytes.

Type (2 bytes): A 2 byte unsigned integer indicating Type of the PropValue field. It can have the following values:

Value	Meaning
PROP_TYPE_ITEM 0x01	Indicates that the property contains one item.

PropName (variable): A Unicode string whose length is indicated by the NameLength field. The string indicates the name of the property.

PropValue (variable): An encoded binary blob of length indicated by the ValueLength field. This field either contains a 32-bit unsigned integer or an array of ASCII characters. See [Encoding and decoding PropValue field](#) for encoding and decoding this blob.

The following table describes various PropName fields and associated PropValue fields used by Microsoft Terminal Services.

PropName	PropValue Type	Description	MS-TSTS RPC call that can be used to query this value
CtxCfgPresent	32 bit unsigned integer	It indicates presence of valid TSPROPERTY structures in TSPROPERTYArray. If the TSPROPERTYArray does not contain a TSPROPERTY structure containing this PropName and PropValue, Terminal Services ignore the remaining contents of TSPROPERTYArray. The PropValue field can only contain value 0xB00B1E55.	None.
CtxCfgFlags1	32 bit unsigned integer	Each bit in the PropValue maps to a boolean field of USERCONFIG structure returned by RpcGetConfigData method. For details about each bit, see the table of CtxCfgFlags1 values in this section.	RpcGetConfigData.

PropName	PropValue Type	Description	MS-TSTS RPC call that can be used to query this value
CtxCallBack	32 bit unsigned integer	The callback class for callback operations.	Returned by RpcGetConfigData method in Callback field of USERCONFIG structure.
CtxKeyboardLayout	32 bit unsigned integer	The keyboard layout (HKL) of the user session.	Returned by RpcGetConfigData method in KeyboardLayout field of USERCONFIG structure.
CtxMinEncryptionLevel	32 bit unsigned integer	The minimum allowed encryption level of the user session.	Returned by RpcGetConfigData method in MinEncryptionLevel field of USERCONFIG structure.
CtxNWLogonServer	32 bit unsigned integer	The NetWare logon server name.	Returned by RpcGetConfigData method in NWLogonServer field of USERCONFIG structure.
CtxWFHomeDir	Variable Length ASCII character array	This attribute specifies the home directory for the user. Each user on a terminal server has a unique home directory. This ensures that application information is stored separately for each user in a multi-user environment. To set a home directory on the local computer, the implementer specifies a local path; for example, C:\Path. To set a home directory in a network environment, the implementer MUST first set the CtxWFHomeDrive property, and then set this property to a Universal Naming Convention (UNC) path.	Returned by RpcGetConfigData method in WFHomeDir field of USERCONFIG structure.
CtxWFHomeDrive	Variable Length ASCII character array	This attribute specifies a home drive for the user. In a network environment, this property is a string containing a drive specification (a drive letter followed by a colon) to which the UNC path specified in the TerminalServicesCtxWFHomeDir property is mapped. To set a home directory in a network environment, the implementer MUST first set this property, and then set the	Returned by RpcGetConfigData method in WFHomeDirDrive field of USERCONFIG structure.

PropName	PropValue Type	Description	MS-TSTS RPC call that can be used to query this value
		CtxWFHomeDir property.	
CtxInitialProgram	Variable Length ASCII character array	This attribute specifies the path and file name of the application that the user requires to start automatically when the user logs on to the terminal server. To set an initial application to start when the user logs on, the implementer MUST first set this property, and then set the CtxWorkDirectory property. If the implementer sets only the CtxInitialProgram property, the application starts in the user's session in the default user directory.	Returned by RpcGetConfigData method in InitialProgram field of USERCONFIG structure.
CtxMaxConnectionTime	32 bit unsigned integer	This attribute specifies the maximum duration (in minutes) of the Terminal Services session. After the specified number of minutes have elapsed, the session can be disconnected or terminated.	Returned by RpcGetConfigData method in MaxConnectionTime field of USERCONFIG structure.
CtxMaxDisconnectionTime	32 bit unsigned integer	This attribute specifies the maximum amount of time (in minutes) that a disconnected Terminal Services session remains active on the terminal server. After the specified number of minutes have elapsed, the session is terminated.	Returned by RpcGetConfigData method in MaxDisconnectionTime field of USERCONFIG structure.
CtxMaxIdleTime	32 bit unsigned integer	This attribute specifies the maximum amount of time (in minutes) that the Terminal Services session can remain idle. After the specified number of minutes have elapsed, the session can be disconnected or terminated.	Returned by RpcGetConfigData method in MaxIdleTime field of USERCONFIG structure.
CtxWFProfilePath	Variable Length ASCII character array	This attribute specifies a roaming or mandatory profile path to use when the user logs on to the terminal server. The profile path is in the following network path format: \\servername\profiles folder name\username.	Returned by RpcGetConfigData method in WFProfilePath field of USERCONFIG structure.
CtxShadow	32 bit unsigned integer	This attribute specifies whether to allow remote observation or remote control of the user's Terminal Services session. The values are as follows: 0. Disable	Returned by RpcGetConfigData method in Shadow field of USERCONFIG structure.

PropName	PropValue Type	Description	MS-TSTS RPC call that can be used to query this value
		1. EnableInputNotify 2. EnableInputNoNotify 3. EnableNoInputNotify 4. EnableNoInputNoNotify For a description of these values, see the RemoteControl method of the Win32_TSRemoteControlSetting WMI class.	
CtxWorkDirectory	Variable Length ASCII character array	This attribute specifies the working directory path for the user. To set an initial application to start when the user logs on to the terminal server, the implementer MUST first set the CtxInitialProgram property, and then set this property.	Returned by RpcGetConfigData method in WorkDirectory field of USERCONFIG structure.
CtxCallbackNumber	Variable Length ASCII character array	This attribute specifies the call back number provided to the user on the client side for technical support.	Returned by RpcGetConfigData method in CallbackNumber field of USERCONFIG structure.

Following table provides the details of each bit in the PropValue associated with the PropName 'CtxCfgFlags1'.

Bit mask in CtxCfgFlags1 PropValue	MS-TSTS RPC call that can be used to query this value
0x10000000	fInheritInitialProgram field of USERCONFIG structure.
F1MSK_INHERITCALLBACK 0x08000000	Returned by RpcGetConfigData method in fInheritCallback field of USERCONFIG structure.
F1MSK_INHERITCALLBACKNUMBER 0x04000000	Returned by RpcGetConfigData method in fInheritCallbackNumber field of USERCONFIG structure
F1MSK_INHERITSHADOW 0x02000000	Returned by RpcGetConfigData method in fInheritShadow field of USERCONFIG structure
F1MSK_INHERITMAXSESSIONTIME 0x01000000	Returned by RpcGetConfigData method in fInheritMaxSessionTime field of USERCONFIG structure
F1MSK_INHERITMAXDISCONNECTIONTIME 0x00800000	Returned by RpcGetConfigData method in fInheritMaxDisconnectionTime field USERCONFIG structure
F1MSK_INHERITMAXIDLETIME 0x00400000	Returned by RpcGetConfigData method in fInheritMaxIdleTime field of USERCONFIG structure
F1MSK_INHERITAUTOCLIENT 0x00200000	Returned by RpcGetConfigData method in fInheritAutoClient field of USERCONFIG structure

Bit mask in CtxCfgFlags1 PropValue	MS-TSTS RPC call that can be used to query this value
F1MSK_INHERITSECURITY 0x00100000	Returned by RpcGetConfigData method in fInheritSecurity field of USERCONFIG structure
F1MSK_PROMPTFORPASSWORD 0x00080000	Returned by RpcGetConfigData method in fPromptForPassword field of USERCONFIG structure
F1MSK_RESETBROKEN 0x00040000	Returned by RpcGetConfigData method in fResetBroken field of USERCONFIG structure
F1MSK_RECONNECTSAME 0x00020000	Returned by RpcGetConfigData method in fReconnectSame field of USERCONFIG structure
F1MSK_LOGONDISABLED 0x00010000	Returned by RpcGetConfigData method in fLogonDisabled field of USERCONFIG structure.
F1MSK_AUTOCLIENTDRIVES 0x00008000	Returned by RpcGetConfigData method in fAutoClientDrives field of USERCONFIG structure
F1MSK_AUTOCLIENTLPTS 0x00004000	Returned by RpcGetConfigData method in fAutoClientLpts field of USERCONFIG structure
F1MSK_FORCECLIENTLPTDEF 0x00002000	Returned by RpcGetConfigData method in fForceClientLptDef field of USERCONFIG structure
F1MSK_DISABLEENCRIPTION 0x00001000	Returned by RpcGetConfigData method in fDisableEncryption field of USERCONFIG structure
F1MSK_HOMEDIRECTORYMAPROOT 0x00000800	Returned by RpcGetConfigData method in fHomeDirectoryMapRoot field of USERCONFIG structure
F1MSK_USEDEFAULTGINA 0x00000400	Returned by RpcGetConfigData method in fUseDefaultGina field of USERCONFIG structure
F1MSK_DISABLECPM 0x00000200	Returned by RpcGetConfigData method in fDisableCpm field of USERCONFIG structure
F1MSK_DISABLECDM 0x00000100	Returned by RpcGetConfigData method in fDisableCdm field of USERCONFIG structure
F1MSK_DISABLECCM 0x00000080	Returned by RpcGetConfigData method in fDisableCcm field of USERCONFIG structure
F1MSK_DISABLELPT 0x00000040	Returned by RpcGetConfigData method in fDisableLPT field of USERCONFIG structure
F1MSK_DISABLECLIP 0x00000020	Returned by RpcGetConfigData method in fDisableClip field of USERCONFIG structure
F1MSK_DISABLEEXE 0x00000010	Returned by RpcGetConfigData method in fDisableExe field of USERCONFIG structure
F1MSK_WALLPAPERDISABLED 0x00000008	Returned by RpcGetConfigData method in fWallPaperDisabled field of USERCONFIG structure
F1MSK_DISABLECAM 0x00000004	Returned by RpcGetConfigData method in fDisableCam field of USERCONFIG structure

2.3.3 Encoding and decoding PropValue field in TSProperty Structure

To create the encoded binary blob for the PropValue field, each byte in the input is replaced with two hexadecimal characters; equivalent to the binary representation of the input byte. For example, character "A" is replaced with two characters "41". The string containing the hexadecimal characters is then placed in the blob.

For example, the ASCII string "ABCDE\0" is converted to "414243444500" and placed in the binary blob. The encoded binary blob for the ASCII string "ABCDE" will look like:

```
34 31 34 32 34 33 34 34 34 35 30 30
```

See [Encoding/Decoding Example \(section 4.5\)](#) for example code to encode and decode the PropValue field.

4.5 Encoding/Decoding Example

The following is the example of encoding and decoding the PropValue field in the [TSProperty](#) structure.

```
DWORD
EncodePropValue
(
    __in BYTE* pbSource,
    __in DWORD dwSourceLength,
    __deref_out_bcount(*pdwDestLength) BYTE** ppbDest,
    __out DWORD* pdwDestLength
)
{
    *pdwDestLength = (dwSourceLength*2)+1;
    *ppbDest = (BYTE*)LocalAlloc(LPTR, *pdwDestLength);

    for(DWORD i=0; i<dwSourceLength; i++)
    {
        StringCbPrintfA((char*)((*ppbDest)+(i*2)), 3, "%02x", pbSource[i]);
    }

    return 0;
}

#define MAPHEXTODIGIT(x) ( x >= '0' && x <= '9' ? (x-'0') :      \
                          x >= 'A' && x <= 'F' ? (x-'A'+10) :    \
                          x >= 'a' && x <= 'f' ? (x-'a'+10) : 0 )

DWORD
DecodePropValue
(
    __in BYTE* pbSource,
    __in DWORD dwSourceLength,
    __deref_out_bcount(*pdwDestLength) BYTE** ppbDest,
    DWORD *pdwDestLength
)
{
    *pdwDestLength = (dwSourceLength/2) + 1;
    (*ppbDest) = (BYTE*)LocalAlloc(LPTR, (*pdwDestLength));

    for(DWORD i=0; i<(*pdwDestLength); i++)
    {
        (*ppbDest)[i] = MAPHEXTODIGIT( pbSource[2*i]) * 16 +
            MAPHEXTODIGIT( pbSource[2*i+1]);
    }
    return 0;
}
```