Access Mask

1 Introduction

This document provides a concordance for the ACCESS_MASK data type.

The Windows Software Development Toolkit (SDK) (v7.0) and the Windows Driver Kit (WDK) (v7600.16385.0), which contain at least 46 mask sets and 500+ value declarations.

1.1 References

Unless otherwise noted, all references in this section are published by Microsoft Corporation.

[Windows_Internals]"Windows Internals®, Fifth Edition", Microsoft Press, 2009, Library of Congress Control Number 2009927697

1.1.1 Windows Driver Kit (WDK)

[WDK] "How to Get the Windows Driver Kit", http://go.microsoft.com/fwlink/?LinkID=89050

[MSDN_WDK] "Windows Driver Kit", http://msdn.microsoft.com/en-us/library/aa972908.aspx

1.1.2 Windows Software Development Kit (SDK)

[SDK] "Windows SDK Download", http://go.microsoft.com/fwlink/?LinkID=84091

[MSDN_WIN32_COM_DEV] "Win32 and COM Development", http://msdn.microsoft.com/en-us/library/ee663300.aspx

1.2 Overview (Synopsis)

This document provides an overview of the ACCESS_MASK data type.

Unless otherwise specified, all code declarations in this document are from various header files included in the Windows Software Development Toolkit (SDK v7.0) and the Windows Driver Kit (WDK) v7600.16385.0, and are shown in a `monospace` font.

Example: `typedef DWORD ACCESS_MASK;`

The ACCESS_MASK data type is a flag set that is the primary means of specifying (encoding) the requested or granted access to a (securable) object by a user. Specifically, it is a double word value that defines standard, specific, and generic rights used in access control entries (ACEs).

A securable object is an object that can have a security descriptor. All named Windows objects are securable. Some unnamed objects, such as process and thread objects, can have security descriptors too. For most securable objects, you can specify an object's security descriptor in the function call that creates the object. For example, you can specify a security descriptor in the CreateFile and CreateProcess functions.

Each type of securable object defines its own set of specific access rights and its own mapping of generic access rights. For information about the specific and generic access rights for each type of securable object, see the overview for that type of object [Securable Objects].

The following is a list of some common securable objects [Windows_Internals] p 4548:

- Files, directories and volumes (NTFS file system)
- Devices
- Mailslots
- Named and anonymous pipes
- Jobs
- Processes
- Threads
- Events, keyed events and event pairs
- Mutexes, semaphores
- Shared memory sections
- I/O completion ports
- LPC ports
- Waitable timers
- Access tokens
- Windows stations
- Desktops
- Network shares
- Services

- Registry keys
- Printers
- Active Directory objects

The general rule when setting ACCESS_MASK flags is to always use as few flags as possible

2 ACCESS_MASK

ACCESS_MASK: http://msdn.microsoft.com/en-us/library/aa374892.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ Authorization Reference \ Authorization Data Types

An ACCESS_MASK is a 32-bit (double word) set of flags that is the primary means of specifying (encoding) the requested or granted access to an object by a user. It is used in:

- Parameters to Win32 API functions (such as CreateFile(…), etc.), to indicate the desired access to an object.

  Example: `DWORD dwDesiredAccess;`

- Access control entries (ACEs), both to encode the rights to an object assigned to a principal and to encode the requested access when opening an object.

Windows technologies using the ACCESS_MASK data type include DLLs, Processes, Threads, Fax, File System, Kernel Transaction Manager (KTM), Printing, Security, miscellaneous (but important, nonetheless) system facilities, Window Stations and Desktops, Windows Filtering Platform, Windows Management Instrumentation (WMI) and Wireless Networking.

From WinNT.h ([SDK], [WDK]):

```
//
//  Define the access mask as a longword sized structure divided up as
//  follows:
//
//       3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
//       1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
//      +---------------+---------------+-------------------------------+
//      |G|G|G|G|Res'd|A| StandardRights|         SpecificRights        |
//      |R|W|E|A|     |S|               |                               |
//      +-+-------------+---------------+-------------------------------+
//
//      typedef struct _ACCESS_MASK {
//          WORD  SpecificRights;
//          BYTE  StandardRights;
//          BYTE  AccessSystemAcl : 1;
//          BYTE  Reserved : 3;
//          BYTE  GenericAll : 1;
//          BYTE  GenericExecute : 1;
//          BYTE  GenericWrite : 1;
//          BYTE  GenericRead : 1;
```

```
//      } ACCESS_MASK;
//      typedef ACCESS_MASK *PACCESS_MASK;
//
//  but to make life simple for programmer's we'll allow them to specify
//  a desired access mask by simply OR'ing together mulitple single rights
//  and treat an access mask as a DWORD.  For example
//
//      DesiredAccess = DELETE | READ_CONTROL
//
//  So we'll declare ACCESS_MASK as DWORD
//
// begin_wdm
typedef DWORD ACCESS_MASK;
typedef ACCESS_MASK *PACCESS_MASK;
```

The bits in positions 28 through 31 are generic rights that can be mapped to object-specific user rights by the resource manager for the requested object. The mapping of these rights is implementation-specific.

The bits in positions 24 and 25 are for maximum allowed and access system security rights.

The bits in positions 0 through 15 are standard rights that are common to all objects.

References:

MSDN Page Title	URL
Access Rights and Access Masks	http://msdn.microsoft.com/en-us/library/aa374902.aspx
Standard Access Rights	http://msdn.microsoft.com/en-us/library/aa379607.aspx
SACL Access Right	http://msdn.microsoft.com/en-us/library/aa379321.aspx
AccessCheck Function	http://msdn.microsoft.com/en-us/library/aa374815.aspx
Generic Access Rights	http://msdn.microsoft.com/en-us/library/aa446632.aspx

2.1 Standard Access Rights

Standard Access Rights: http://msdn.microsoft.com/en-us/library/aa379607.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ About Authorization \ Access Control \ Access Control Model \ Access Control Components \ Access Rights and Access Masks

Standard access rights are those rights corresponding to operations common to (most types of) securable objects.

Access Right	Value	Description
DELETE	0x00010000	The right to delete the object.
READ_CONTROL	0x00020000	The right to read the information in the file or directory object's security descriptor. This does not include the information in the SACL.
WRITE_DAC	0x00040000	The right to modify the DACL in the object's security descriptor.

Access Right	Value	Description
WRITE_OWNER	0x00080000	The right to change the owner in the object's security descriptor.
SYNCHRONIZE	0x00100000	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
STANDARD_RIGHTS_REQUIRED	0x000F0000	Combines DELETE, READ_CONTROL, WRITE_DAC, and WRITE_OWNER access. [1]
STANDARD_RIGHTS_READ	READ_CONTROL	Currently defined to equal READ_CONTROL.
STANDARD_RIGHTS_WRITE	READ_CONTROL	Currently defined to equal READ_CONTROL.
STANDARD_RIGHTS_EXECUTE	READ_CONTROL	Currently defined to equal READ_CONTROL.
STANDARD_RIGHTS_ALL	0x001F0000	Combines DELETE, READ_CONTROL, WRITE_DAC, WRITE_OWNER, and SYNCHRONIZE access.

[1] STANDARD_RIGHTS_REQUIRED is a mask meant to be used when defining access masks for object types - it's the set of access masks that all securable objects must support.

From WinNT.h ([SDK], [WDK]):

```
//
//  The following are masks for the predefined standard access types
//
#define DELETE                           (0x00010000L)
#define READ_CONTROL                     (0x00020000L)
#define WRITE_DAC                        (0x00040000L)
#define WRITE_OWNER                      (0x00080000L)
#define SYNCHRONIZE                      (0x00100000L)

#define STANDARD_RIGHTS_REQUIRED         (0x000F0000L)

#define STANDARD_RIGHTS_READ             (READ_CONTROL)
#define STANDARD_RIGHTS_WRITE            (READ_CONTROL)
#define STANDARD_RIGHTS_EXECUTE          (READ_CONTROL)

#define STANDARD_RIGHTS_ALL              (0x001F0000L)
```

2.2 Special Access Rights (ACCESS_SYSTEM_SECURITY and MAXIMUM_ALLOWED)

This ‘special’ grouping is for convenience; there is nothing in the Windows SDK or WDK that classify the ACCESS_SYSTEM_SECURITY and MAXIMUM_ALLOWED flags as ‘special’. Simply put, these are all the defined flags that are not specific, standard or generic.

2.2.1 ACCESS_SYSTEM_SECURITY (SACL Access Right)

SACL Access Right: http://msdn.microsoft.com/en-us/library/aa379321.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ About Authorization \ Access Control \ Access Control Model \ Access Control Components \ Access Rights and Access Masks

Access system security (ACCESS_SYSTEM_SECURITY) is used to request or indicate access to a system access control list (SACL). This is also known as the [SACL Access Right].

This type of access requires the calling process to have the SE_SECURITY_NAME (Manage auditing and security log) privilege. If this flag is set in the access mask of an audit access ACE (successful or unsuccessful access), the SACL access will be audited ([ACCESS_MASK]).

The system grants this access right only if the SE_SECURITY_NAME privilege is enabled in the access token of the requesting thread ([SACL_Access_Right]).

If a backup application must have access to the system-level access control settings, the ACCESS_SYSTEM_SECURITY flag must be specified in the dwDesiredAccess parameter value passed to CreateFile ([File_Access_Rights]).

Privileges determine the type of system operations that a user account can perform. An administrator assigns privileges to user and group accounts. Each user's privileges include those granted to the user and to the groups to which the user belongs. The SE_BACKUP_NAME and SE_RESTORE_NAME privileges are required to grant ACCESS_SYSTEM_SECURITY for backup operations ([Privilege_Constants]).

To read the SACL from a security descriptor, the calling process must have been granted ACCESS_SYSTEM_SECURITY access when the handle was opened. The proper way to get this access is to enable the SE_SECURITY_NAME privilege ([Privilege_Constants]) in the caller's current token, open the handle for ACCESS_SYSTEM_SECURITY access, and then disable the privilege ([GetSecurityInfo]).

From WinNT.h ([SDK], [WDK]):

```
//
// AccessSystemAcl access type
//
#define ACCESS_SYSTEM_SECURITY           (0x01000000L)
```

2.2.2 MAXIMUM_ALLOWED (AccessCheck Function)

AccessCheck Function: http://msdn.microsoft.com/en-us/library/aa374815.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security] Authorization \ Authorization Reference \ Authorization Functions

The MAXIMUM_ALLOWED access type is generally used for the AccessCheck (…) function to determine whether a security descriptor grants a specified set of access rights to the client identified

by an access token. Typically, server applications use this function to check access to a private object. Note that MAXIMUM_ALLOWED cannot be used in an ACE (see access control entries).

When using AccessCheck(…) for this purpose, perform the following steps:

1. Obtain a security descriptor that has owner, group, and DACL information.

   If you are not impersonating a client, obtain an impersonation token by calling ImpersonateSelf(…). This token is passed as the client token in the AccessCheck(…) call.

2. Create a generic mapping structure ([GENERIC_MAPPING]). The contents of this structure will vary depending on the object being used.
3. Call AccessCheck(…) and request MAXIMUM_ALLOWED as the desired access.

   If the AccessCheck(…) call succeeds after the above steps have been completed, the GrantedAccess parameter contains a mask of the object-specific rights that are granted by the security descriptor.

When used in an Access Request operation, the Maximum Allowed bit grants the requestor the maximum permissions allowed to the object through the Access Check Algorithm. This bit can only be requested, it cannot be set in an ACE ([MS-DTYP 2.4.3]).

When used to set the Security Descriptor on an object, the Maximum Allowed bit in the SECURITY_DESCRIPTOR has no meaning. The MA bit SHOULD NOT be set and SHOULD be ignored when part of a SECURITY_DESCRIPTOR structure ([MS-DTYP 2.4.3]).

From WinNT.h ([SDK], [WDK]):

```
//
// MaximumAllowed access type
//
#define MAXIMUM_ALLOWED                  (0x02000000L)
```

2.3 Generic Access Rights

Generic Access Rights: http://msdn.microsoft.com/en-us/library/aa446632.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ About Authorization \ Access Control \ Access Control Model \ Access Control Components \ Access Rights and Access Masks

The following text was adapted from [Generic_Access_Rights]:

The four high-order bits of an ACCESS_MASK specify generic access rights. Each type of securable object maps these bits to a set of its standard and object-specific access rights.

For example, a Windows file object maps the GENERIC_READ bit to the READ_CONTROL and SYNCHRONIZE standard access rights and to the FILE_READ_DATA, FILE_READ_EA, and

FILE_READ_ATTRIBUTES object-specific access rights. Other types of objects map the GENERIC_READ bit to whatever set of access rights is appropriate for that type of object.

You can use generic access rights to specify the type of access you need when you are opening a handle to an object. This is typically simpler than specifying all the corresponding standard and specific rights.

Applications that define private securable objects can also use the generic access rights.

Constant	Value	Description
GENERIC_ALL	0x10000000	The right to read, write, and execute the object [1].
GENERIC_EXECUTE	0x20000000	The right to execute or alternatively look into the object [1].
GENERIC_WRITE	0x40000000	The right to write the information maintained by the object [1].
GENERIC_READ	0x80000000	The right to read the information maintained by the object [1].

[1] [WDK_IFS_Access_Mask]

From WinNT.h ([SDK], [WDK]):

```
//
//  These are the generic Rights
//
#define GENERIC_READ                     (0x80000000L)
#define GENERIC_WRITE                    (0x40000000L)
#define GENERIC_EXECUTE                  (0x20000000L)
#define GENERIC_ALL                      (0x10000000L)
```

2.4 ACCESS_MASK (Redacted)

Just for fun, I defined a new structure, ACCESS_MASKEX, as shown below, with some additional declarations for helpful bit masks. I have perhaps overused anonymous unions and struct definitions; this was done with keeping the structure member reference semantics as flat as possible. Use this at your own risk, of course.

```
//
//    ACCESS_MASKEX
//
//    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
//    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
//   +-------+-------+---------------+---------------------------------+
//   |Generic|Special| StandardRights|         SpecificRights          |
//   |Rights |Rights |   Rights      |                                 |
//   +-------+-------+---------------+---------------------------------+
//   |G|G|G|G|Res|M|A|Res'd|S|W|W|R|D|                                 |
//   |R|W|E|A|'d |A|S|Res'd|Y|O|D|C|E|                                 |
//   +-------+-------+---------------+---------------------------------+
//
//   DE - DELETE
//   RC - READ_CONTROL
//   WD - WRITE_DAC
//   WO - WRITE_OWNER
//   SY - SYNCHRONIZE
```

```
//   AS - ACCESS_SYSTEM_SECURITY
//   MA - MAXIMUM_ALLOWED
//   GA - GENERIC_ALL
//   GE - GENERIC_EXECUTE
//   GW - GENERIC_WRITE
//   GR - GENERIC_READ
//
// * Indicates an existing declaration
//
typedef struct _ACCESS_MASKEX {
  WORD SpecificRights;               // SPECIFIC_RIGHTS_ALL      0x0000FFFF *
  union {
    BYTE StandardRights;             // STANDARD_RIGHTS_MASK     0x00FF0000
    struct {
      BYTE StandardDelete      : 1;  // DELETE                   0x00010000 *
      BYTE StandardReadControl : 1;  // READ_CONTROL             0x00020000 *
      BYTE StandardWriteDAC    : 1;  // WRITE_DAC                0x00040000 *
      BYTE StandardWriteOwner  : 1;  // WRITE_OWNER              0x00080000 *
      BYTE StandardSynchronize : 1;  // SYNCHRONIZE              0x00100000 *
      BYTE StandardReserved    : 3;  // STANDARD_RIGHTS_RESERVED 0x00E00000
    };
  };
  union {
    BYTE SpecialRights : 4;          // SPECIAL_RIGHTS_MASK      0x0F000000
    union {
      BYTE AccessSystemAcl : 1;      // ACCESS_SYSTEM_SECURITY   0x01000000 *
      union {
        BYTE Reserved      : 3;      // ACCESS_MASK_RESERVED     0x0E000000
        struct {
          BYTE MaximumAllowed  : 1;  // MAXIMUM_ALLOWED          0x02000000 *
          BYTE SpecialReserved : 2;  // SPECIAL_RIGHTS_RESERVED  0x0C000000
        };
      };
    };
  };
  union {
    BYTE GenericRights : 4;          // GENERIC_RIGHTS_ALL       0xF0000000
    struct {
      BYTE GenericAll     : 1;       // GENERIC_ALL              0x10000000 *
      BYTE GenericExecute : 1;       // GENERIC_EXECUTE          0x20000000 *
      BYTE GenericWrite   : 1;       // GENERIC_WRITE            0x40000000 *
      BYTE GenericRead    : 1;       // GENERIC_READ             0x80000000 *
    };
  };
} ACCESS_MASKEX;
typedef ACCESS_MASKEX *PACCESS_MASKEX;

// * existing declarations
//#define SPECIFIC_RIGHTS_ALL             (0x0000FFFFL)
//#define DELETE                          (0x00010000L)
//#define READ_CONTROL                    (0x00020000L)
//#define STANDARD_RIGHTS_READ            (READ_CONTROL)
//#define STANDARD_RIGHTS_WRITE           (READ_CONTROL)
//#define STANDARD_RIGHTS_EXECUTE         (READ_CONTROL)
//#define WRITE_DAC                       (0x00040000L)
//#define WRITE_OWNER                     (0x00080000L)
//#define STANDARD_RIGHTS_REQUIRED        (0x000F0000L)
//#define SYNCHRONIZE                     (0x00100000L)
//#define STANDARD_RIGHTS_ALL             (0x001F0000L)
//#define ACCESS_SYSTEM_SECURITY          (0x01000000L)
//#define MAXIMUM_ALLOWED                 (0x02000000L)
//#define GENERIC_ALL                     (0x10000000L)
//#define GENERIC_EXECUTE                 (0x20000000L)
```

```
//#define GENERIC_WRITE                   (0x40000000L)
//#define GENERIC_READ                    (0x80000000L)
//
// Reserved bits
//
#define STANDARD_RIGHTS_RESERVED    (0x00E00000L)
#define ACCESS_MASK_RESERVED        (0x0E000000L)
#define SPECIAL_RIGHTS_RESERVED     (0x0C000000L)
#define RESERVED_RIGHTS_ALL         (STANDARD_RIGHTS_RESERVED |\
                                     ACCESS_MASK_RESERVED     |\
                                     SPECIAL_RIGHTS_RESERVED)
//
// Non-reserved rights
//
#define SPECIAL_RIGHTS_ALL          (ACCESS_SYSTEM_SECURITY | MAXIMUM_ALLOWED)
#define GENERIC_RIGHTS_ALL          (GENERIC_READ           | GENERIC_WRITE       |\
                                     GENERIC_EXECUTE        | GENERIC_ALL)
#define ACCESS_RIGHTS_ALL           (SPECIFIC_RIGHTS_ALL    | STANDARD_RIGHTS_ALL |\
                                     GENERIC_RIGHTS_ALL     | SPECIAL_RIGHTS_ALL)
//
// Masks
//
#define SPECIFIC_RIGHTS_MASK        (SPECIFIC_RIGHTS_ALL)
#define STANDARD_RIGHTS_MASK        (STANDARD_RIGHTS_ALL | STANDARD_RIGHTS_RESERVED)
#define SPECIAL_RIGHTS_MASK         (SPECIAL_RIGHTS_ALL  | SPECIAL_RIGHTS_RESERVED)
#define GENERIC_RIGHTS_MASK         (GENERIC_RIGHTS_ALL)
#define RESERVED_RIGHTS_MASK        (RESERVED_RIGHTS_ALL)
```

3 Specific Access Rights

Each type of securable object has a set of access rights that correspond to operations specific to that type of object. These rights occupy the low order 16 bits of the ACCESS_MASK data type.

From WinNT.h ([SDK], [WDK]):

```
#define SPECIFIC_RIGHTS_ALL             (0x0000FFFFL)
```

3.1 Administration and Management

3.1.1 Active Directory Service Interfaces (ADSI)

See this blog: Active Directory Technical Specification Control Access Rights Concordance

- ADS_RIGHTS_ENUM Enumeration: http://msdn.microsoft.com/en-us/library/aa772285.aspx
- Directory Services Access Rights: http://msdn.microsoft.com/en-us/library/aa446614.aspx
- [MS-ADTS] 5.1.3.2 Access Rights: http://msdn.microsoft.com/en-us/library/cc223511.aspx
- [SDK] NtDsAPI.h

[MS-ADTS] 5.1.3.2 Access Rights	Directory Services Access Rights	ADS_RIGHTS_ENUM	Value
	ACTRL_DS_OPEN		0x00000000

[MS-ADTS] 5.1.3.2 Access Rights	Directory Services Access Rights	ADS_RIGHTS_ENUM	Value
RIGHT_DS_CREATE_CHILD	ACTRL_DS_CREATE_CHILD	ADS_RIGHT_DS_CREATE_CHILD	0x00000001
RIGHT_DS_DELETE_CHILD	ACTRL_DS_DELETE_CHILD	ADS_RIGHT_DS_DELETE_CHILD	0x00000002
RIGHT_DS_LIST_CONTENTS	ACTRL_DS_LIST	ADS_RIGHT_ACTRL_DS_LIST	0x00000004
RIGHT_DS_WRITE_PROPERTY_EXTENDED	ACTRL_DS_SELF	ADS_RIGHT_DS_SELF	0x00000008
RIGHT_DS_READ_PROPERTY	ACTRL_DS_READ_PROP	ADS_RIGHT_DS_READ_PROP	0x00000010
RIGHT_DS_WRITE_PROPERTY	ACTRL_DS_WRITE_PROP	ADS_RIGHT_DS_WRITE_PROP	0x00000020
RIGHT_DS_DELETE_TREE	ACTRL_DS_DELETE_TREE	ADS_RIGHT_DS_DELETE_TREE	0x00000040
RIGHT_DS_LIST_OBJECT	ACTRL_DS_LIST_OBJECT	ADS_RIGHT_DS_LIST_OBJECT	0x00000080
RIGHT_DS_CONTROL_ACCESS	ACTRL_DS_CONTROL_ACCESS	ADS_RIGHT_DS_CONTROL_ACCESS	0x00000100
RIGHT_DELETE	*DELETE*	ADS_RIGHT_DELETE	0x00010000
RIGHT_READ_CONTROL	*READ_CONTROL*	ADS_RIGHT_READ_CONTROL	0x00020000
RIGHT_WRITE_DAC	*WRITE_DAC*	ADS_RIGHT_WRITE_DAC	0x00040000
RIGHT_WRITE_OWNER	*WRITE_OWNER*	ADS_RIGHT_WRITE_OWNER	0x00080000
	SYNCHRONIZE	ADS_RIGHT_SYNCHRONIZE	0x00100000
	ACCESS_SYSTEM_SECURITY	ADS_RIGHT_ACCESS_SYSTEM_SECURITY	0x01000000
	MAXIMUM_ALLOWED		0x02000000
RIGHT_GENERIC_ALL	*GENERIC_ALL*	ADS_RIGHT_GENERIC_ALL	0x10000000
RIGHT_GENERIC_EXECUTE	*GENERIC_EXECUTE*	ADS_RIGHT_GENERIC_EXECUTE	0x20000000
RIGHT_GENERIC_WRITE	*GENERIC_WRITE*	ADS_RIGHT_GENERIC_WRITE	0x40000000
RIGHT_GENERIC_READ	*GENERIC_READ*	ADS_RIGHT_GENERIC_READ	0x80000000

3.1.1.1 Directory Services Access Rights

Directory Services Access Rights: http://msdn.microsoft.com/en-us/library/aa446614.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ About Authorization \ Access Control \ Access Control Model \ Access Control Components

Constant	Value	Description
ACTRL_DS_OPEN	0x00000000	Open a DS object.
ACTRL_DS_CREATE_CHILD	0x00000001	Create a child DS object.
ACTRL_DS_DELETE_CHILD	0x00000002	Delete a child DS object.
ACTRL_DS_LIST	0x00000004	Enumerate a DS object.
ACTRL_DS_SELF	0x00000008	Access allowed only after validated rights checks supported by the object are performed. This flag can be used alone to perform all validated rights checks of the object or it can be combined with an identifier of a specific validated right to perform only that check.
ACTRL_DS_READ_PROP	0x00000010	Read the properties of a DS object.
ACTRL_DS_WRITE_PROP	0x00000020	Write properties for a DS object.
ACTRL_DS_DELETE_TREE	0x00000040	Delete a tree of DS objects.
ACTRL_DS_LIST_OBJECT	0x00000080	List a tree of DS objects.
ACTRL_DS_CONTROL_ACCESS	0x00000100	Access allowed only after extended rights checks supported by the object are performed. This flag can be used alone to perform all extended rights checks on the object or it can be combined with an identifier of a specific extended right to perform only that check.

[MS-ADTS]: Active Directory Technical Specification: http://msdn.microsoft.com/en-us/library/cc223122.aspx

[MS-ADTS] 5.1.3.2 Access Rights: http://msdn.microsoft.com/en-us/library/cc223511.aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Server Protocols (WSPP) \ [MS-ADTS]: Active Directory Technical Specification

Sym	Constant	Value	Description
CC	RIGHT_DS_CREATE_CHILD	0x00000001	The right to create child objects of the object. The ObjectType member of an ACE can contain a GUID that identifies the objectClass of child object whose creation is controlled. If ObjectType does not contain a GUID, the ACE controls the creation of all child object classes allowed by the schema.
DC	RIGHT_DS_DELETE_CHILD	0x00000002	The right to delete child objects of the object. The ObjectType member of an ACE can contain a GUID that identifies the objectClass of the child object whose deletion is controlled. If ObjectType does not contain a GUID, the ACE controls the deletion of all child object classes.
LC	RIGHT_DS_LIST_CONTENTS	0x00000004	The right to list child objects of this object. For more information about this right, see section 3.1.1.4.
VW	RIGHT_DS_WRITE_PROPERTY_EXTENDED	0x00000008	The right to perform an operation controlled by a validated write access right. The ObjectType member of an ACE can contain a GUID that identifies the validated write. If ObjectType does not contain a GUID, the ACE controls the rights to perform all validated write operations associated with the object. For a list of validated write rights, see section 5.1.3.2.2. For specifics of validated write processing, see the Modify operation in section 3.1.1.5.3.
RP	RIGHT_DS_READ_PROPERTY	0x00000010	The right to read properties of the object. The ObjectType member of an ACE can contain a GUID that identifies a property set or an attribute. If ObjectType does not contain a GUID, the ACE controls the right to read all attributes of the object.

Sym	Constant	Value	Description
WP	RIGHT_DS_WRITE_PROPERTY	0x00000020	The right to write properties of the object. The ObjectType member of an ACE can contain a GUID that identifies a property set or an attribute. If ObjectType does not contain a GUID, the ACE controls the right to write all attributes of the object.
DT	RIGHT_DS_DELETE_TREE	0x00000040	The right to perform a Delete-Tree operation on this object. See the Delete operation in section 3.1.1.5.5 for more details.
LO	RIGHT_DS_LIST_OBJECT	0x00000080	The right to list a particular object. If the user is not granted this right, and the user is not granted the RIGHT_DS_LIST_CONTENTS right on the object's parent, the object is hidden from the user. Note that LIST_OBJECT rights are not enforced by Active Directory by default. In order to enable LIST_OBJECT enforcement, a dSHeuristics bit fDoListObject must be set. For more information about this right, see section 7.1.1.2.4.1.2.
CR	RIGHT_DS_CONTROL_ACCESS	0x00000100	The right to perform an operation controlled by a control access right. The ObjectType member of an ACE can contain a GUID that identifies the control access right. If ObjectType does not contain a GUID, the ACE controls the right to perform all control access right controlled operations associated with the object. For a list of control access rights, see section 5.1.3.2.1.
DE	RIGHT_DELETE	0x00010000	The right to delete the object.
RC	RIGHT_READ_CONTROL	0x00020000	The right to read data from the security descriptor of the object, not including the data in the SACL.
WD	RIGHT_WRITE_DAC	0x00040000	The right to modify the DACL in the object security descriptor.
WO	RIGHT_WRITE_OWNER	0x00080000	The right to modify the owner of an object in the object's security descriptor. A user can only take ownership of an object, but cannot transfer ownership of an object to other users.

Sym	Constant	Value	Description
GA	RIGHT_GENERIC_ALL	0x10000000	The right to create or delete child objects, delete a subtree, read and write properties, examine child objects and the object itself, add and remove the object from the directory, and read or write with an extended right.
GX	RIGHT_GENERIC_EXECUTE	0x20000000	The right to read permissions on, and list the contents of, a container object.
GW	RIGHT_GENERIC_WRITE	0x40000000	The right to read permissions on this object, write all the properties on this object, and perform all validated writes to this object.
GR	RIGHT_GENERIC_READ	0x80000000	The right to read permissions on this object, read all the properties on this object, list this object name when the parent container is listed, and list the contents of this object if it is a container.

3.1.1.2 ADS_RIGHTS_ENUM

ADS_RIGHTS_ENUM Enumeration: http://msdn.microsoft.com/en-us/library/aa772285.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Administration and Management \ Directory, Identity and Access Services \ Directory Services \ Directory Services Technologies \ Active Directory Service Interfaces \ ADSI Enumerations

Constant	Value
ADS_RIGHT_DS_CREATE_CHILD	0x1
ADS_RIGHT_DS_DELETE_CHILD	0x2
ADS_RIGHT_ACTRL_DS_LIST	0x4
ADS_RIGHT_DS_SELF	0x8
ADS_RIGHT_DS_READ_PROP	0x10
ADS_RIGHT_DS_WRITE_PROP	0x20
ADS_RIGHT_DS_DELETE_TREE	0x40
ADS_RIGHT_DS_LIST_OBJECT	0x80
ADS_RIGHT_DS_CONTROL_ACCESS	0x100
ADS_RIGHT_DELETE	0x10000
ADS_RIGHT_READ_CONTROL	0x20000
ADS_RIGHT_WRITE_DAC	0x40000
ADS_RIGHT_WRITE_OWNER	0x80000
ADS_RIGHT_SYNCHRONIZE	0x100000
ADS_RIGHT_ACCESS_SYSTEM_SECURITY	0x1000000
ADS_RIGHT_GENERIC_READ	0x80000000
ADS_RIGHT_GENERIC_WRITE	0x40000000

Constant	Value
ADS_RIGHT_GENERIC_EXECUTE	0x20000000
ADS_RIGHT_GENERIC_ALL	0x10000000

From Iads.h ([SDK]):

```
typedef /* [public] */
enum __MIDL___MIDL_itf_ads_0001_0043_0001
    {    ADS_RIGHT_DELETE                    = 0x10000,
         ADS_RIGHT_READ_CONTROL              = 0x20000,
         ADS_RIGHT_WRITE_DAC                 = 0x40000,
         ADS_RIGHT_WRITE_OWNER               = 0x80000,
         ADS_RIGHT_SYNCHRONIZE               = 0x100000,
         ADS_RIGHT_ACCESS_SYSTEM_SECURITY    = 0x1000000,
         ADS_RIGHT_GENERIC_READ              = 0x80000000,
         ADS_RIGHT_GENERIC_WRITE             = 0x40000000,
         ADS_RIGHT_GENERIC_EXECUTE           = 0x20000000,
         ADS_RIGHT_GENERIC_ALL               = 0x10000000,
         ADS_RIGHT_DS_CREATE_CHILD           = 0x1,
         ADS_RIGHT_DS_DELETE_CHILD           = 0x2,
         ADS_RIGHT_ACTRL_DS_LIST             = 0x4,
         ADS_RIGHT_DS_SELF                   = 0x8,
         ADS_RIGHT_DS_READ_PROP              = 0x10,
         ADS_RIGHT_DS_WRITE_PROP             = 0x20,
         ADS_RIGHT_DS_DELETE_TREE            = 0x40,
         ADS_RIGHT_DS_LIST_OBJECT            = 0x80,
         ADS_RIGHT_DS_CONTROL_ACCESS         = 0x100
    }    ADS_RIGHTS_ENUM;
```

3.1.1.3 ACTRL_DS_* (NtDsAPI.h, SDK)

See Also COM.

From NtDsAPI.h ([SDK]):

```
// Permissions bits used in security descriptors in the directory.
#ifndef _DS_CONTROL_BITS_DEFINED_
#define _DS_CONTROL_BITS_DEFINED_
#define ACTRL_DS_OPEN                           0x00000000
#define ACTRL_DS_CREATE_CHILD                   0x00000001
#define ACTRL_DS_DELETE_CHILD                   0x00000002
#define ACTRL_DS_LIST                           0x00000004
#define ACTRL_DS_SELF                           0x00000008
#define ACTRL_DS_READ_PROP                      0x00000010
#define ACTRL_DS_WRITE_PROP                     0x00000020
#define ACTRL_DS_DELETE_TREE                    0x00000040
#define ACTRL_DS_LIST_OBJECT                    0x00000080
#define ACTRL_DS_CONTROL_ACCESS                 0x00000100

// generic read
#define DS_GENERIC_READ          ((STANDARD_RIGHTS_READ)     | \
                                  (ACTRL_DS_LIST)            | \
                                  (ACTRL_DS_READ_PROP)       | \
                                  (ACTRL_DS_LIST_OBJECT))

// generic execute
#define DS_GENERIC_EXECUTE       ((STANDARD_RIGHTS_EXECUTE)  | \
                                  (ACTRL_DS_LIST))
```

```
// generic right
#define DS_GENERIC_WRITE         ((STANDARD_RIGHTS_WRITE)     | \
                                  (ACTRL_DS_SELF)             | \
                                  (ACTRL_DS_WRITE_PROP))
// generic all

#define DS_GENERIC_ALL           ((STANDARD_RIGHTS_REQUIRED) | \
                                  (ACTRL_DS_CREATE_CHILD)     | \
                                  (ACTRL_DS_DELETE_CHILD)     | \
                                  (ACTRL_DS_DELETE_TREE)      | \
                                  (ACTRL_DS_READ_PROP)        | \
                                  (ACTRL_DS_WRITE_PROP)       | \
                                  (ACTRL_DS_LIST)             | \
                                  (ACTRL_DS_LIST_OBJECT)      | \
                                  (ACTRL_DS_CONTROL_ACCESS)   | \
                                  (ACTRL_DS_SELF))
#endif
```

3.2 Component Development (COM)

ACTRL_ACCESS_ENTRY Structure: http://msdn.microsoft.com/en-us/library/ms692524.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Component Development \ COM \ Component Object Model (COM) \ Com Fundamentals \ Reference \ Structures

From AccCtrl.h ([SDK], [WDK]):

```
//
// Generic permission values
//
#define ACTRL_RESERVED          0x00000000
#define ACTRL_PERM_1            0x00000001
#define ACTRL_PERM_2            0x00000002
#define ACTRL_PERM_3            0x00000004
#define ACTRL_PERM_4            0x00000008
#define ACTRL_PERM_5            0x00000010
#define ACTRL_PERM_6            0x00000020
#define ACTRL_PERM_7            0x00000040
#define ACTRL_PERM_8            0x00000080
#define ACTRL_PERM_9            0x00000100
#define ACTRL_PERM_10           0x00000200
#define ACTRL_PERM_11           0x00000400
#define ACTRL_PERM_12           0x00000800
#define ACTRL_PERM_13           0x00001000
#define ACTRL_PERM_14           0x00002000
#define ACTRL_PERM_15           0x00004000
#define ACTRL_PERM_16           0x00008000
#define ACTRL_PERM_17           0x00010000
#define ACTRL_PERM_18           0x00020000
#define ACTRL_PERM_19           0x00040000
#define ACTRL_PERM_20           0x00080000

//
// Standard and object rights
//
#define ACTRL_SYSTEM_ACCESS         0x04000000
#define ACTRL_DELETE                0x08000000
#define ACTRL_READ_CONTROL          0x10000000
#define ACTRL_CHANGE_ACCESS         0x20000000
#define ACTRL_CHANGE_OWNER          0x40000000
```

```
#define ACTRL_SYNCHRONIZE           0x80000000
#define ACTRL_STD_RIGHTS_ALL        0xf8000000
#define ACTRL_STD_RIGHT_REQUIRED    ( ACTRL_STD_RIGHTS_ALL & ~ACTRL_SYNCHRONIZE )

#ifndef _DS_CONTROL_BITS_DEFINED_
#define _DS_CONTROL_BITS_DEFINED_
#define ACTRL_DS_OPEN                           ACTRL_RESERVED
#define ACTRL_DS_CREATE_CHILD                   ACTRL_PERM_1
#define ACTRL_DS_DELETE_CHILD                   ACTRL_PERM_2
#define ACTRL_DS_LIST                           ACTRL_PERM_3
#define ACTRL_DS_SELF                           ACTRL_PERM_4
#define ACTRL_DS_READ_PROP                      ACTRL_PERM_5
#define ACTRL_DS_WRITE_PROP                     ACTRL_PERM_6
#define ACTRL_DS_DELETE_TREE                    ACTRL_PERM_7
#define ACTRL_DS_LIST_OBJECT                    ACTRL_PERM_8
#define ACTRL_DS_CONTROL_ACCESS                 ACTRL_PERM_9
#endif

#define ACTRL_FILE_READ                         ACTRL_PERM_1
#define ACTRL_FILE_WRITE                        ACTRL_PERM_2
#define ACTRL_FILE_APPEND                       ACTRL_PERM_3
#define ACTRL_FILE_READ_PROP                    ACTRL_PERM_4
#define ACTRL_FILE_WRITE_PROP                   ACTRL_PERM_5
#define ACTRL_FILE_EXECUTE                      ACTRL_PERM_6
#define ACTRL_FILE_READ_ATTRIB                  ACTRL_PERM_8
#define ACTRL_FILE_WRITE_ATTRIB                 ACTRL_PERM_9
#define ACTRL_FILE_CREATE_PIPE                  ACTRL_PERM_10
#define ACTRL_DIR_LIST                          ACTRL_PERM_1
#define ACTRL_DIR_CREATE_OBJECT                 ACTRL_PERM_2
#define ACTRL_DIR_CREATE_CHILD                  ACTRL_PERM_3
#define ACTRL_DIR_DELETE_CHILD                  ACTRL_PERM_7
#define ACTRL_DIR_TRAVERSE                      ACTRL_PERM_6
#define ACTRL_KERNEL_TERMINATE                  ACTRL_PERM_1
#define ACTRL_KERNEL_THREAD                     ACTRL_PERM_2
#define ACTRL_KERNEL_VM                         ACTRL_PERM_3
#define ACTRL_KERNEL_VM_READ                    ACTRL_PERM_4
#define ACTRL_KERNEL_VM_WRITE                   ACTRL_PERM_5
#define ACTRL_KERNEL_DUP_HANDLE                 ACTRL_PERM_6
#define ACTRL_KERNEL_PROCESS                    ACTRL_PERM_7
#define ACTRL_KERNEL_SET_INFO                   ACTRL_PERM_8
#define ACTRL_KERNEL_GET_INFO                   ACTRL_PERM_9
#define ACTRL_KERNEL_CONTROL                    ACTRL_PERM_10
#define ACTRL_KERNEL_ALERT                      ACTRL_PERM_11
#define ACTRL_KERNEL_GET_CONTEXT                ACTRL_PERM_12
#define ACTRL_KERNEL_SET_CONTEXT                ACTRL_PERM_13
#define ACTRL_KERNEL_TOKEN                      ACTRL_PERM_14
#define ACTRL_KERNEL_IMPERSONATE                ACTRL_PERM_15
#define ACTRL_KERNEL_DIMPERSONATE               ACTRL_PERM_16
#define ACTRL_PRINT_SADMIN                      ACTRL_PERM_1
#define ACTRL_PRINT_SLIST                       ACTRL_PERM_2
#define ACTRL_PRINT_PADMIN                      ACTRL_PERM_3
#define ACTRL_PRINT_PUSE                        ACTRL_PERM_4
#define ACTRL_PRINT_JADMIN                      ACTRL_PERM_5
#define ACTRL_SVC_GET_INFO                      ACTRL_PERM_1
#define ACTRL_SVC_SET_INFO                      ACTRL_PERM_2
#define ACTRL_SVC_STATUS                        ACTRL_PERM_3
#define ACTRL_SVC_LIST                          ACTRL_PERM_4
#define ACTRL_SVC_START                         ACTRL_PERM_5
#define ACTRL_SVC_STOP                          ACTRL_PERM_6
#define ACTRL_SVC_PAUSE                         ACTRL_PERM_7
#define ACTRL_SVC_INTERROGATE                   ACTRL_PERM_8
#define ACTRL_SVC_UCONTROL                      ACTRL_PERM_9
#define ACTRL_REG_QUERY                         ACTRL_PERM_1
```

```
#define ACTRL_REG_SET                       ACTRL_PERM_2
#define ACTRL_REG_CREATE_CHILD              ACTRL_PERM_3
#define ACTRL_REG_LIST                      ACTRL_PERM_4
#define ACTRL_REG_NOTIFY                    ACTRL_PERM_5
#define ACTRL_REG_LINK                      ACTRL_PERM_6
#define ACTRL_WIN_CLIPBRD                   ACTRL_PERM_1
#define ACTRL_WIN_GLOBAL_ATOMS              ACTRL_PERM_2
#define ACTRL_WIN_CREATE                    ACTRL_PERM_3
#define ACTRL_WIN_LIST_DESK                 ACTRL_PERM_4
#define ACTRL_WIN_LIST                      ACTRL_PERM_5
#define ACTRL_WIN_READ_ATTRIBS              ACTRL_PERM_6
#define ACTRL_WIN_WRITE_ATTRIBS             ACTRL_PERM_7
#define ACTRL_WIN_SCREEN                    ACTRL_PERM_8
#define ACTRL_WIN_EXIT                      ACTRL_PERM_9
```

3.3 Diagnostics

3.3.1 WMI Access Rights

EventAccessControl Function: http://msdn.microsoft.com/en-us/library/aa363717.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Diagnostics \ Windows Events \ Event Tracing \ Event Tracing Reference \ Event Tracing Functions

From wmistr.h ([WDK]):

```
//
// Specific rights for WMI guid objects. These are available from 0x0001 to
// 0xffff (ie up to 16 rights)
//
#define WMIGUID_QUERY                0x0001
#define WMIGUID_SET                  0x0002
#define WMIGUID_NOTIFICATION         0x0004
#define WMIGUID_READ_DESCRIPTION     0x0008
#define WMIGUID_EXECUTE              0x0010
#define TRACELOG_CREATE_REALTIME     0x0020
#define TRACELOG_CREATE_ONDISK       0x0040
#define TRACELOG_GUID_ENABLE         0x0080
#define TRACELOG_ACCESS_KERNEL_LOGGER 0x0100
#define TRACELOG_LOG_EVENT           0x0200 // used on Vista and greater
#define TRACELOG_CREATE_INPROC       0x0200 // used pre-Vista
#define TRACELOG_ACCESS_REALTIME     0x0400
#define TRACELOG_REGISTER_GUIDS      0x0800

#define WMIGUID_ALL_ACCESS_WIN2K (STANDARD_RIGHTS_READ          | \
                                  WMIGUID_QUERY                 | \
                                  WMIGUID_SET                   | \
                                  WMIGUID_NOTIFICATION          | \
                                  WMIGUID_READ_DESCRIPTION      | \
                                  WMIGUID_EXECUTE               | \
                                  TRACELOG_CREATE_REALTIME      | \
                                  TRACELOG_CREATE_ONDISK        | \
                                  TRACELOG_GUID_ENABLE          | \
                                  TRACELOG_ACCESS_KERNEL_LOGGER | \
                                  TRACELOG_CREATE_INPROC        | \
                                  TRACELOG_ACCESS_REALTIME)
```

```
#define WMIGUID_ALL_ACCESS_WINXP (WMIGUID_ALL_ACCESS_WIN2K | \
                                  SYNCHRONIZE               | \
                                  TRACELOG_REGISTER_GUIDS)

#if (NTDDI_VERSION >= NTDDI_WINXP)

#define WMIGUID_ALL_ACCESS WMIGUID_ALL_ACCESS_WINXP

#else

#define WMIGUID_ALL_ACCESS WMIGUID_ALL_ACCESS_WIN2K

#endif
```

3.3.2 Namespace Access Rights

Namespace Access Rights Constants (WMI): http://msdn.microsoft.com/en-us/library/aa392710.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Administration and Management \ Windows Management Instrumentation \ WMI Reference \ WMI Infrastructure Objects and Values \ WMI Security \ WMI Security Constants

From WbemCli.h ([SDK], [WDK]):

```
enum tag_WBEM_SECURITY_FLAGS
    {   WBEM_ENABLE             = 1,
        WBEM_METHOD_EXECUTE     = 2,
        WBEM_FULL_WRITE_REP     = 4,
        WBEM_PARTIAL_WRITE_REP  = 8,
        WBEM_WRITE_PROVIDER     = 0x10,
        WBEM_REMOTE_ACCESS      = 0x20,
        WBEM_RIGHT_SUBSCRIBE    = 0x40,
        WBEM_RIGHT_PUBLISH      = 0x80
    }   WBEM_SECURITY_FLAGS;
```

3.4 Networking

3.4.1 Fax Service

Fax Client User Access Rights: http://msdn.microsoft.com/en-us/library/ms692351.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Networking \ Network Communications \ Fax Service \ Fax Service Client API for Windows 2000 \ About the Fax Service Client API

Specific Fax Access Rights: http://msdn.microsoft.com/en-us/library/ms692302.aspx

Generic Fax Access Rights: http://msdn.microsoft.com/en-us/library/ms691834.aspx

Required Fax Access Rights by Function: http://msdn.microsoft.com/en-us/library/ms692330.aspx

From WinFax.h ([SDK]):

```
//
// Fax Specific Access Rights
//

#define FAX_JOB_SUBMIT          (0x0001)
#define FAX_JOB_QUERY           (0x0002)
#define FAX_CONFIG_QUERY        (0x0004)
#define FAX_CONFIG_SET          (0x0008)
#define FAX_PORT_QUERY          (0x0010)
#define FAX_PORT_SET            (0x0020)
#define FAX_JOB_MANAGE          (0x0040)

#define FAX_READ                (STANDARD_RIGHTS_READ        |\
                                 FAX_JOB_QUERY               |\
                                 FAX_CONFIG_QUERY            |\
                                 FAX_PORT_QUERY)

#define FAX_WRITE               (STANDARD_RIGHTS_WRITE       |\
                                 FAX_JOB_SUBMIT )

#define FAX_ALL_ACCESS          (STANDARD_RIGHTS_ALL         |\
                                 FAX_JOB_SUBMIT              |\
                                 FAX_JOB_QUERY               |\
                                 FAX_CONFIG_QUERY            |\
                                 FAX_CONFIG_SET              |\
                                 FAX_PORT_QUERY              |\
                                 FAX_PORT_SET                |\
                                 FAX_JOB_MANAGE)
```

3.4.2 Windows Filtering Platform (WFP)

WFP Access Control: http://msdn.microsoft.com/en-us/library/bb442405.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Networking \ Network Firewall and Routing \ Windows Filtering Platform \ About Windows Filtering Platform \ WFP Configuration

From fwpmk.h ([SDK], [WDK]):

From fwpmu.h ([WDK]):

```
// Specific access rights.
#define FWPM_ACTRL_ADD                (0x00000001)
#define FWPM_ACTRL_ADD_LINK           (0x00000002)
#define FWPM_ACTRL_BEGIN_READ_TXN     (0x00000004)
#define FWPM_ACTRL_BEGIN_WRITE_TXN    (0x00000008)
#define FWPM_ACTRL_CLASSIFY           (0x00000010)
#define FWPM_ACTRL_ENUM               (0x00000020)
#define FWPM_ACTRL_OPEN               (0x00000040)
#define FWPM_ACTRL_READ               (0x00000080)
#define FWPM_ACTRL_READ_STATS         (0x00000100)
#define FWPM_ACTRL_SUBSCRIBE          (0x00000200)
#define FWPM_ACTRL_WRITE              (0x00000400)

// Generic access Rights
#define FWPM_GENERIC_READ \
      ( STANDARD_RIGHTS_READ       | \
        FWPM_ACTRL_BEGIN_READ_TXN  | \
```

```
        FWPM_ACTRL_CLASSIFY        | \
        FWPM_ACTRL_OPEN            | \
        FWPM_ACTRL_READ            | \
        FWPM_ACTRL_READ_STATS      )

#define FWPM_GENERIC_EXECUTE \
      ( STANDARD_RIGHTS_EXECUTE    | \
        FWPM_ACTRL_ENUM            | \
        FWPM_ACTRL_SUBSCRIBE       )

#define FWPM_GENERIC_WRITE \
      ( STANDARD_RIGHTS_WRITE      | \
        DELETE                     | \
        FWPM_ACTRL_ADD             | \
        FWPM_ACTRL_ADD_LINK        | \
        FWPM_ACTRL_BEGIN_WRITE_TXN | \
        FWPM_ACTRL_WRITE           )

#define FWPM_GENERIC_ALL \
      ( STANDARD_RIGHTS_REQUIRED   | \
        FWPM_ACTRL_ADD             | \
        FWPM_ACTRL_ADD_LINK        | \
        FWPM_ACTRL_BEGIN_READ_TXN  | \
        FWPM_ACTRL_BEGIN_WRITE_TXN | \
        FWPM_ACTRL_CLASSIFY        | \
        FWPM_ACTRL_ENUM            | \
        FWPM_ACTRL_OPEN            | \
        FWPM_ACTRL_READ            | \
        FWPM_ACTRL_READ_STATS      | \
        FWPM_ACTRL_SUBSCRIBE       | \
        FWPM_ACTRL_WRITE           )
```

3.4.3 Wireless Networking

WlanGetSecuritySettings Function: http://msdn.microsoft.com/en-us/library/ms706746.aspx

WlanSetSecuritySettings Function: http://msdn.microsoft.com/en-us/library/ms706819.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Networking \ Wireless Networking \ Native WiFi Reference \ Native WiFi Functions

From wlanapi.h ([SDK]):

```
// Definition of access masks for setting non-default security
// settings on WLAN configuration objects and connection profiles.

#define WLAN_READ_ACCESS     ( STANDARD_RIGHTS_READ | FILE_READ_DATA )
#define WLAN_EXECUTE_ACCESS  ( WLAN_READ_ACCESS     | STANDARD_RIGHTS_EXECUTE |\
                               FILE_EXECUTE )
#define WLAN_WRITE_ACCESS    ( WLAN_READ_ACCESS     | WLAN_EXECUTE_ACCESS |\
                               STANDARD_RIGHTS_WRITE | FILE_WRITE_DATA     |\
                               DELETE                | WRITE_DAC )
```

3.5 Authorization

3.5.1 Token

Access Rights for Access-Token Objects: http://msdn.microsoft.com/en-us/library/aa374905.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ About Authorization \ Access Control \ Access Control Model \ Access Control Components \ Access Tokens

From WinNT.h ([SDK], [WDK]):

From ntifs.h ([WDK]):

```
//
// Token Specific Access Rights.
//

#define TOKEN_ASSIGN_PRIMARY    (0x0001)
#define TOKEN_DUPLICATE         (0x0002)
#define TOKEN_IMPERSONATE       (0x0004)
#define TOKEN_QUERY             (0x0008)
#define TOKEN_QUERY_SOURCE      (0x0010)
#define TOKEN_ADJUST_PRIVILEGES (0x0020)
#define TOKEN_ADJUST_GROUPS     (0x0040)
#define TOKEN_ADJUST_DEFAULT    (0x0080)
#define TOKEN_ADJUST_SESSIONID  (0x0100)

#define TOKEN_ALL_ACCESS_P (STANDARD_RIGHTS_REQUIRED | \
                            TOKEN_ASSIGN_PRIMARY     | \
                            TOKEN_DUPLICATE          | \
                            TOKEN_IMPERSONATE        | \
                            TOKEN_QUERY              | \
                            TOKEN_QUERY_SOURCE       | \
                            TOKEN_ADJUST_PRIVILEGES  | \
                            TOKEN_ADJUST_GROUPS      | \
                            TOKEN_ADJUST_DEFAULT )

#if ((defined(_WIN32_WINNT) && (_WIN32_WINNT > 0x0400)) ||
(!defined(_WIN32_WINNT)))
#define TOKEN_ALL_ACCESS  (TOKEN_ALL_ACCESS_P |\
                           TOKEN_ADJUST_SESSIONID )
#else
#define TOKEN_ALL_ACCESS  (TOKEN_ALL_ACCESS_P)
#endif

#define TOKEN_READ       (STANDARD_RIGHTS_READ     | \
                          TOKEN_QUERY)


#define TOKEN_WRITE      (STANDARD_RIGHTS_WRITE    | \
                          TOKEN_ADJUST_PRIVILEGES  | \
                          TOKEN_ADJUST_GROUPS      |\
                          TOKEN_ADJUST_DEFAULT)

#define TOKEN_EXECUTE    (STANDARD_RIGHTS_EXECUTE)
```

3.5.2 Access Control Entry (ACE)

ACE: http://msdn.microsoft.com/en-us/library/aa374912.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ Authorization Reference \ Authorization Structures

See also: ACE Strings: http://msdn.microsoft.com/en-us/library/aa374928.aspx

'Mask' field…

```
typedef struct _X_ACE {
  ACE_HEADER  Header;
  ACCESS_MASK Mask;
  .
  .
  .
} X_ACE, *PX_ACE;
```

[MS-DTYP] 2.4.4 ACE	Authorization Structures	Use
2.4.4.2 ACCESS_ALLOWED_ACE	ACCESS_ALLOWED_ACE Structure	DACL
2.4.4.5 ACCESS_ALLOWED_CALLBACK_ACE	ACCESS_ALLOWED_CALLBACK_ACE Structure	DACL
2.4.4.7 ACCESS_ALLOWED_CALLBACK_OBJECT_ACE	ACCESS_ALLOWED_CALLBACK_OBJECT_ACE Structure	DACL
2.4.4.3 ACCESS_ALLOWED_OBJECT_ACE	ACCESS_ALLOWED_OBJECT_ACE Structure	DACL
2.4.4.4 ACCESS_DENIED_ACE	ACCESS_DENIED_ACE Structure	DACL
2.4.4.6 ACCESS_DENIED_CALLBACK_ACE	ACCESS_DENIED_CALLBACK_ACE Structure	DACL
2.4.4.8 ACCESS_DENIED_CALLBACK_OBJECT_ACE	ACCESS_DENIED_CALLBACK_OBJECT_ACE Structure	DACL
	ACCESS_DENIED_OBJECT_ACE Structure	DACL
2.4.4.9 SYSTEM_AUDIT_ACE	SYSTEM_AUDIT_ACE Structure	SACL
2.4.4.10 SYSTEM_AUDIT_CALLBACK_ACE	SYSTEM_AUDIT_CALLBACK_ACE Structure	SACL
2.4.4.12 SYSTEM_AUDIT_CALLBACK_OBJECT_ACE	SYSTEM_AUDIT_CALLBACK_OBJECT_ACE Structure	SACL
	SYSTEM_AUDIT_OBJECT_ACE Structure	SACL
2.4.4.11 SYSTEM_MANDATORY_LABEL_ACE	SYSTEM_MANDATORY_LABEL_ACE Structure	SACL

From Iads.h ([SDK]):

```
typedef /* [public] */
enum __MIDL___MIDL_itf_ads_0001_0043_0001
    {    ADS_RIGHT_DELETE                    = 0x10000,
         ADS_RIGHT_READ_CONTROL              = 0x20000,
         ADS_RIGHT_WRITE_DAC                 = 0x40000,
         ADS_RIGHT_WRITE_OWNER               = 0x80000,
         ADS_RIGHT_SYNCHRONIZE               = 0x100000,
         ADS_RIGHT_ACCESS_SYSTEM_SECURITY  = 0x1000000,
         ADS_RIGHT_GENERIC_READ              = 0x80000000,
         ADS_RIGHT_GENERIC_WRITE             = 0x40000000,
         ADS_RIGHT_GENERIC_EXECUTE           = 0x20000000,
         ADS_RIGHT_GENERIC_ALL               = 0x10000000,
         ADS_RIGHT_DS_CREATE_CHILD           = 0x1,
         ADS_RIGHT_DS_DELETE_CHILD           = 0x2,
         ADS_RIGHT_ACTRL_DS_LIST             = 0x4,
         ADS_RIGHT_DS_SELF                   = 0x8,
         ADS_RIGHT_DS_READ_PROP              = 0x10,
         ADS_RIGHT_DS_WRITE_PROP             = 0x20,
         ADS_RIGHT_DS_DELETE_TREE            = 0x40,
         ADS_RIGHT_DS_LIST_OBJECT            = 0x80,
         ADS_RIGHT_DS_CONTROL_ACCESS         = 0x100
    }    ADS_RIGHTS_ENUM;
```

Constant	Value	Description
ADS_RIGHT_DS_CREATE_CHILD	0X00000001	The ObjectType GUID identifies a type of child object. The ACE controls the trustee's right to create this type of child object.
ADS_RIGHT_DS_SELF	0x00000008	The ObjectType GUID identifies a validated write.
ADS_RIGHT_DS_READ_PROP	0x00000010	The ObjectType GUID identifies a property set or property of the object. The ACE controls the trustee's right to read the property or property set.
ADS_RIGHT_DS_WRITE_PROP	0x00000020	The ObjectType GUID identifies a property set or property of the object. The ACE controls the trustee's right to write the property or property set.
ADS_RIGHT_DS_CONTROL_ACCESS	0X00000100	The ObjectType GUID identifies an extended access right.

3.5.3 Audit

AuditSetSystemPolicy Function: http://msdn.microsoft.com/en-us/library/aa375712.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Authorization \ Authorization Reference \ Authorization Functions

From NTSecAPI.h ([SDK], [WDK]):

```
#define AUDIT_SET_SYSTEM_POLICY                (0x0001)
#define AUDIT_QUERY_SYSTEM_POLICY              (0x0002)
#define AUDIT_SET_USER_POLICY                  (0x0004)
#define AUDIT_QUERY_USER_POLICY                (0x0008)
#define AUDIT_ENUMERATE_USERS                  (0x0010)
#define AUDIT_SET_MISC_POLICY                  (0x0020)
#define AUDIT_QUERY_MISC_POLICY                (0x0040)

#define AUDIT_GENERIC_ALL            (STANDARD_RIGHTS_REQUIRED  |\
                                      AUDIT_SET_SYSTEM_POLICY   |\
                                      AUDIT_QUERY_SYSTEM_POLICY |\
                                      AUDIT_SET_USER_POLICY     |\
                                      AUDIT_QUERY_USER_POLICY   |\
                                      AUDIT_ENUMERATE_USERS     |\
                                      AUDIT_SET_MISC_POLICY     |\
                                      AUDIT_QUERY_MISC_POLICY)

#define AUDIT_GENERIC_READ           (STANDARD_RIGHTS_READ      |\
                                      AUDIT_QUERY_SYSTEM_POLICY |\
                                      AUDIT_QUERY_USER_POLICY   |\
                                      AUDIT_ENUMERATE_USERS     |\
                                      AUDIT_QUERY_MISC_POLICY)

#define AUDIT_GENERIC_WRITE          (STANDARD_RIGHTS_WRITE     |\
                                      AUDIT_SET_USER_POLICY     |\
                                      AUDIT_SET_MISC_POLICY     |\
```

```
                                        AUDIT_SET_SYSTEM_POLICY)

#define AUDIT_GENERIC_EXECUTE         (STANDARD_RIGHTS_EXECUTE)
```

3.5.4 Local Security Authority (LSA)

Security Management Objects: http://msdn.microsoft.com/en-us/library/ms721854.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Security Management \ Security Management Reference \ Security Management Objects

3.5.4.1 Local Security Authority Account Specific Access Rights

Account Object Access Rights: http://msdn.microsoft.com/en-us/library/ms721750.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Security Management \ Security Management Reference \ Security Management Objects

From ntlsa.h ([WDK]):

```
//
// Account object type-specific Access Types
//

#define ACCOUNT_VIEW                        0x00000001L
#define ACCOUNT_ADJUST_PRIVILEGES           0x00000002L
#define ACCOUNT_ADJUST_QUOTAS               0x00000004L
#define ACCOUNT_ADJUST_SYSTEM_ACCESS        0x00000008L

#define ACCOUNT_ALL_ACCESS    (STANDARD_RIGHTS_REQUIRED         |\
                               ACCOUNT_VIEW                     |\
                               ACCOUNT_ADJUST_PRIVILEGES        |\
                               ACCOUNT_ADJUST_QUOTAS            |\
                               ACCOUNT_ADJUST_SYSTEM_ACCESS)

#define ACCOUNT_READ          (STANDARD_RIGHTS_READ             |\
                               ACCOUNT_VIEW)

#define ACCOUNT_WRITE         (STANDARD_RIGHTS_WRITE            |\
                               ACCOUNT_ADJUST_PRIVILEGES        |\
                               ACCOUNT_ADJUST_QUOTAS            |\
                               ACCOUNT_ADJUST_SYSTEM_ACCESS)

#define ACCOUNT_EXECUTE       (STANDARD_RIGHTS_EXECUTE)
```

3.5.4.2 Local Security Authority Policy Specific Access Rights

Policy Object Access Rights: http://msdn.microsoft.com/en-us/library/ms721916.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Security Management \ Security Management Reference \ Security Management Objects

From ntlsa.h ([WDK]):

```
//
// Access types for the Policy object
//

#define POLICY_VIEW_LOCAL_INFORMATION             0x00000001L
#define POLICY_VIEW_AUDIT_INFORMATION             0x00000002L
#define POLICY_GET_PRIVATE_INFORMATION            0x00000004L
#define POLICY_TRUST_ADMIN                        0x00000008L
#define POLICY_CREATE_ACCOUNT                     0x00000010L
#define POLICY_CREATE_SECRET                      0x00000020L
#define POLICY_CREATE_PRIVILEGE                   0x00000040L
#define POLICY_SET_DEFAULT_QUOTA_LIMITS           0x00000080L
#define POLICY_SET_AUDIT_REQUIREMENTS             0x00000100L
#define POLICY_AUDIT_LOG_ADMIN                    0x00000200L
#define POLICY_SERVER_ADMIN                       0x00000400L
#define POLICY_LOOKUP_NAMES                       0x00000800L
#define POLICY_NOTIFICATION                       0x00001000L

#define POLICY_ALL_ACCESS     (STANDARD_RIGHTS_REQUIRED         |\
                               POLICY_VIEW_LOCAL_INFORMATION    |\
                               POLICY_VIEW_AUDIT_INFORMATION    |\
                               POLICY_GET_PRIVATE_INFORMATION   |\
                               POLICY_TRUST_ADMIN               |\
                               POLICY_CREATE_ACCOUNT            |\
                               POLICY_CREATE_SECRET             |\
                               POLICY_CREATE_PRIVILEGE          |\
                               POLICY_SET_DEFAULT_QUOTA_LIMITS  |\
                               POLICY_SET_AUDIT_REQUIREMENTS    |\
                               POLICY_AUDIT_LOG_ADMIN           |\
                               POLICY_SERVER_ADMIN              |\
                               POLICY_LOOKUP_NAMES)


#define POLICY_READ           (STANDARD_RIGHTS_READ             |\
                               POLICY_VIEW_AUDIT_INFORMATION    |\
                               POLICY_GET_PRIVATE_INFORMATION)

#define POLICY_WRITE          (STANDARD_RIGHTS_WRITE            |\
                               POLICY_TRUST_ADMIN               |\
                               POLICY_CREATE_ACCOUNT            |\
                               POLICY_CREATE_SECRET             |\
                               POLICY_CREATE_PRIVILEGE          |\
                               POLICY_SET_DEFAULT_QUOTA_LIMITS  |\
                               POLICY_SET_AUDIT_REQUIREMENTS    |\
                               POLICY_AUDIT_LOG_ADMIN           |\
                               POLICY_SERVER_ADMIN)

#define POLICY_EXECUTE        (STANDARD_RIGHTS_EXECUTE          |\
                               POLICY_VIEW_LOCAL_INFORMATION    |\
                               POLICY_LOOKUP_NAMES)
```

3.5.4.3 Local Security Authority Trusted Domain Specific Access Rights

TrustedDomain Object Access Rights: http://msdn.microsoft.com/en-us/library/ms722466.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Security Management \ Security Management Reference \ Security Management Objects

From ntlsa.h ([WDK]):

```
//
// Trusted Domain object specific access types
//

#define TRUSTED_QUERY_DOMAIN_NAME                0x00000001L
#define TRUSTED_QUERY_CONTROLLERS                0x00000002L
#define TRUSTED_SET_CONTROLLERS                  0x00000004L
#define TRUSTED_QUERY_POSIX                      0x00000008L
#define TRUSTED_SET_POSIX                        0x00000010L
#define TRUSTED_SET_AUTH                         0x00000020L
#define TRUSTED_QUERY_AUTH                       0x00000040L


#define TRUSTED_ALL_ACCESS     (STANDARD_RIGHTS_REQUIRED     |\
                                TRUSTED_QUERY_DOMAIN_NAME    |\
                                TRUSTED_QUERY_CONTROLLERS    |\
                                TRUSTED_SET_CONTROLLERS      |\
                                TRUSTED_QUERY_POSIX          |\
                                TRUSTED_SET_POSIX            |\
                                TRUSTED_SET_AUTH             |\
                                TRUSTED_QUERY_AUTH)

#define TRUSTED_READ           (STANDARD_RIGHTS_READ         |\
                                TRUSTED_QUERY_DOMAIN_NAME)

#define TRUSTED_WRITE          (STANDARD_RIGHTS_WRITE        |\
                                TRUSTED_SET_CONTROLLERS      |\
                                TRUSTED_SET_POSIX            |\
                                TRUSTED_SET_AUTH )

#define TRUSTED_EXECUTE        (STANDARD_RIGHTS_EXECUTE      |\
                                TRUSTED_QUERY_CONTROLLERS    |\
                                TRUSTED_QUERY_POSIX)
```

3.5.4.4 Local Security Authority Secret Specific Access Rights

Private Data Object Access Rights: http://msdn.microsoft.com/en-us/library/ms722417.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Security \ Security Management \ Security Management Reference \ Security Management Objects

From ntlsa.h ([WDK]):

```
//
// Secret object specific access types
//

#define SECRET_SET_VALUE                         0x00000001L
#define SECRET_QUERY_VALUE                       0x00000002L

#define SECRET_ALL_ACCESS     (STANDARD_RIGHTS_REQUIRED         |\
                               SECRET_SET_VALUE                 |\
                               SECRET_QUERY_VALUE)

#define SECRET_READ           (STANDARD_RIGHTS_READ             |\
                               SECRET_QUERY_VALUE)
```

```
#define SECRET_WRITE           (STANDARD_RIGHTS_WRITE            |\
                                SECRET_SET_VALUE)

#define SECRET_EXECUTE         (STANDARD_RIGHTS_EXECUTE)
```

3.5.5 Security Accounts Manager (SAM)

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

[MS-SAMR] 2.2.1.1 Common ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245511(PROT.10).aspx

[MS-SAMR] 2.2.1.2 Generic ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245520(PROT.10).aspx

3.5.5.1 Security Accounts Manager Alias Specific Access Rights

[MS-SAMR] 2.2.1.6 Alias ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245524(PROT.10).aspx

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)

From ntsam.h ([WDK]):

```
//
// Access rights for alias object
//

#define ALIAS_ADD_MEMBER                0x0001
#define ALIAS_REMOVE_MEMBER             0x0002
#define ALIAS_LIST_MEMBERS              0x0004
#define ALIAS_READ_INFORMATION          0x0008
#define ALIAS_WRITE_ACCOUNT             0x0010

#define ALIAS_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED  |\
                          ALIAS_READ_INFORMATION    |\
                          ALIAS_WRITE_ACCOUNT       |\
                          ALIAS_LIST_MEMBERS        |\
                          ALIAS_ADD_MEMBER          |\
                          ALIAS_REMOVE_MEMBER)


#define ALIAS_READ       (STANDARD_RIGHTS_READ      |\
                          ALIAS_LIST_MEMBERS)


#define ALIAS_WRITE      (STANDARD_RIGHTS_WRITE     |\
                          ALIAS_WRITE_ACCOUNT       |\
```

```
                            ALIAS_ADD_MEMBER           |\
                            ALIAS_REMOVE_MEMBER)

#define ALIAS_EXECUTE     (STANDARD_RIGHTS_EXECUTE    |\
                            ALIAS_READ_INFORMATION)
```

3.5.5.2 Security Accounts Manager Domain Specific Access Rights

[MS-SAMR] 2.2.1.4 Domain ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245522(PROT.10).aspx

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)

From ntsam.h ([WDK]):

```
//
// Access rights for domain object
//

#define DOMAIN_READ_PASSWORD_PARAMETERS  0x0001
#define DOMAIN_WRITE_PASSWORD_PARAMS     0x0002
#define DOMAIN_READ_OTHER_PARAMETERS     0x0004
#define DOMAIN_WRITE_OTHER_PARAMETERS    0x0008
#define DOMAIN_CREATE_USER               0x0010
#define DOMAIN_CREATE_GROUP              0x0020
#define DOMAIN_CREATE_ALIAS              0x0040
#define DOMAIN_GET_ALIAS_MEMBERSHIP      0x0080
#define DOMAIN_LIST_ACCOUNTS             0x0100
#define DOMAIN_LOOKUP                    0x0200
#define DOMAIN_ADMINISTER_SERVER         0x0400

#define DOMAIN_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED         |\
                           DOMAIN_READ_OTHER_PARAMETERS     |\
                           DOMAIN_WRITE_OTHER_PARAMETERS    |\
                           DOMAIN_WRITE_PASSWORD_PARAMS     |\
                           DOMAIN_CREATE_USER               |\
                           DOMAIN_CREATE_GROUP              |\
                           DOMAIN_CREATE_ALIAS              |\
                           DOMAIN_GET_ALIAS_MEMBERSHIP      |\
                           DOMAIN_LIST_ACCOUNTS             |\
                           DOMAIN_READ_PASSWORD_PARAMETERS  |\
                           DOMAIN_LOOKUP                    |\
                           DOMAIN_ADMINISTER_SERVER)

#define DOMAIN_READ        (STANDARD_RIGHTS_READ            |\
                           DOMAIN_GET_ALIAS_MEMBERSHIP      |\
                           DOMAIN_READ_OTHER_PARAMETERS)


#define DOMAIN_WRITE       (STANDARD_RIGHTS_WRITE           |\
                           DOMAIN_WRITE_OTHER_PARAMETERS    |\
                           DOMAIN_WRITE_PASSWORD_PARAMS     |\
                           DOMAIN_CREATE_USER               |\
```

```
                            DOMAIN_CREATE_GROUP             |\
                            DOMAIN_CREATE_ALIAS             |\
                            DOMAIN_ADMINISTER_SERVER)

#define DOMAIN_EXECUTE      (STANDARD_RIGHTS_EXECUTE        |\
                            DOMAIN_READ_PASSWORD_PARAMETERS |\
                            DOMAIN_LIST_ACCOUNTS            |\
                            DOMAIN_LOOKUP)
```

3.5.5.3 Security Accounts Manager Group Specific Access Rights

[MS-SAMR] 2.2.1.5 Group ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245523(PROT.10).aspx

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)

From ntsam.h ([WDK]):

```
//
// Access rights for group object
//

#define GROUP_READ_INFORMATION          0x0001
#define GROUP_WRITE_ACCOUNT             0x0002
#define GROUP_ADD_MEMBER                0x0004
#define GROUP_REMOVE_MEMBER             0x0008
#define GROUP_LIST_MEMBERS              0x0010

#define GROUP_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED  |\
                          GROUP_LIST_MEMBERS        |\
                          GROUP_WRITE_ACCOUNT       |\
                          GROUP_ADD_MEMBER          |\
                          GROUP_REMOVE_MEMBER       |\
                          GROUP_READ_INFORMATION)


#define GROUP_READ       (STANDARD_RIGHTS_READ      |\
                          GROUP_LIST_MEMBERS)


#define GROUP_WRITE      (STANDARD_RIGHTS_WRITE     |\
                          GROUP_WRITE_ACCOUNT       |\
                          GROUP_ADD_MEMBER          |\
                          GROUP_REMOVE_MEMBER)

#define GROUP_EXECUTE    (STANDARD_RIGHTS_EXECUTE   |\
                          GROUP_READ_INFORMATION)
```

3.5.5.4 Security Accounts Manager Server Specific Access Rights

[MS-SAMR] 2.2.1.3 Server ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245521(PROT.10).aspx

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)

From ntsam.h ([WDK]):

```
//
// Access rights for server object
//

#define SAM_SERVER_CONNECT               0x0001
#define SAM_SERVER_SHUTDOWN              0x0002
#define SAM_SERVER_INITIALIZE            0x0004
#define SAM_SERVER_CREATE_DOMAIN         0x0008
#define SAM_SERVER_ENUMERATE_DOMAINS     0x0010
#define SAM_SERVER_LOOKUP_DOMAIN         0x0020

#define SAM_SERVER_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED     |\
                               SAM_SERVER_CONNECT           |\
                               SAM_SERVER_INITIALIZE        |\
                               SAM_SERVER_CREATE_DOMAIN     |\
                               SAM_SERVER_SHUTDOWN          |\
                               SAM_SERVER_ENUMERATE_DOMAINS |\
                               SAM_SERVER_LOOKUP_DOMAIN)

#define SAM_SERVER_READ       (STANDARD_RIGHTS_READ         |\
                               SAM_SERVER_ENUMERATE_DOMAINS)

#define SAM_SERVER_WRITE      (STANDARD_RIGHTS_WRITE        |\
                               SAM_SERVER_INITIALIZE        |\
                               SAM_SERVER_CREATE_DOMAIN     |\
                               SAM_SERVER_SHUTDOWN)

#define SAM_SERVER_EXECUTE    (STANDARD_RIGHTS_EXECUTE      |\
                               SAM_SERVER_CONNECT           |\
                               SAM_SERVER_LOOKUP_DOMAIN)
```

3.5.5.5 Security Accounts Manager User Specific Access Rights

[MS-SAMR] 2.2.1.7 User ACCESS_MASK Values: http://msdn.microsoft.com/en-us/library/cc245525(PROT.10).aspx

[MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server), http://msdn.microsoft.com/en-us/library/cc245476(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-SAMR]: Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)

From ntsam.h ([WDK]):

```
//
// Access rights for user object
//

#define USER_READ_GENERAL               0x0001
#define USER_READ_PREFERENCES           0x0002
#define USER_WRITE_PREFERENCES          0x0004
#define USER_READ_LOGON                 0x0008
#define USER_READ_ACCOUNT               0x0010
#define USER_WRITE_ACCOUNT              0x0020
#define USER_CHANGE_PASSWORD            0x0040
#define USER_FORCE_PASSWORD_CHANGE      0x0080
#define USER_LIST_GROUPS                0x0100
#define USER_READ_GROUP_INFORMATION     0x0200
#define USER_WRITE_GROUP_INFORMATION    0x0400

#define USER_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED       |\
                         USER_READ_PREFERENCES          |\
                         USER_READ_LOGON                |\
                         USER_LIST_GROUPS               |\
                         USER_READ_GROUP_INFORMATION    |\
                         USER_WRITE_PREFERENCES         |\
                         USER_CHANGE_PASSWORD           |\
                         USER_FORCE_PASSWORD_CHANGE     |\
                         USER_READ_GENERAL              |\
                         USER_READ_ACCOUNT              |\
                         USER_WRITE_ACCOUNT             |\
                         USER_WRITE_GROUP_INFORMATION)


#define USER_READ       (STANDARD_RIGHTS_READ           |\
                         USER_READ_PREFERENCES          |\
                         USER_READ_LOGON                |\
                         USER_READ_ACCOUNT              |\
                         USER_LIST_GROUPS               |\
                         USER_READ_GROUP_INFORMATION)


#define USER_WRITE      (STANDARD_RIGHTS_WRITE          |\
                         USER_WRITE_PREFERENCES         |\
                         USER_CHANGE_PASSWORD)

#define USER_EXECUTE    (STANDARD_RIGHTS_EXECUTE        |\
                         USER_READ_GENERAL              |\
                         USER_CHANGE_PASSWORD)
```

3.6 System Services

3.6.1 DLLs, Processes and Threads

3.6.1.1 Console

Console Buffer Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms682062.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ DLLs, Processes and Threads \ Character Mode Applications

The Windows security model enables you to control access to console input buffers and console screen buffers. For more information about security, see Access-Control Model.

You can specify a security descriptor for the console input and console screen buffers when you call the **CreateFile** or **CreateConsoleScreenBuffer** function. If you specify NULL, the object gets a default security descriptor. The ACLs in the default security descriptor for a console buffer come from the primary or impersonation token of the creator.

The handles returned by **CreateFile** or **CreateConsoleScreenBuffer**, and **GetStdHandle** have the GENERIC_READ and GENERIC_WRITE access rights.

The valid access rights include the GENERIC_READ and GENERIC_WRITE generic access rights.

Constant	Value	Description
GENERIC_WRITE	0x40000000	Requests write access to the console screen buffer, enabling the process to write data to the buffer.
GENERIC_READ	0x80000000	Requests read access to the console screen buffer, enabling the process to read data from the buffer.

3.6.1.2 Processes and Threads

3.6.1.2.1 Process

Process Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms684880.aspx

From WinNT.h ([SDK], [WDK]):

```
#define PROCESS_TERMINATE                  (0x0001)
#define PROCESS_CREATE_THREAD              (0x0002)
#define PROCESS_SET_SESSIONID              (0x0004)
#define PROCESS_VM_OPERATION               (0x0008)
#define PROCESS_VM_READ                    (0x0010)
#define PROCESS_VM_WRITE                   (0x0020)
#define PROCESS_DUP_HANDLE                 (0x0040)
#define PROCESS_CREATE_PROCESS             (0x0080)
#define PROCESS_SET_QUOTA                  (0x0100)
```

```
#define PROCESS_SET_INFORMATION            (0x0200)
#define PROCESS_QUERY_INFORMATION          (0x0400)
#define PROCESS_SUSPEND_RESUME             (0x0800)
#define PROCESS_QUERY_LIMITED_INFORMATION  (0x1000)
#if (NTDDI_VERSION >= NTDDI_VISTA)
#define PROCESS_ALL_ACCESS        (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | \
                                   0xFFFF)
#else
#define PROCESS_ALL_ACCESS        (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | \
                                   0xFFF)
#endif
```

3.6.1.2.2 Job

Job Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms684164.aspx

From WinNT.h ([SDK], [WDK]):

```
#define JOB_OBJECT_ASSIGN_PROCESS           (0x0001)
#define JOB_OBJECT_SET_ATTRIBUTES           (0x0002)
#define JOB_OBJECT_QUERY                    (0x0004)
#define JOB_OBJECT_TERMINATE                (0x0008)
#define JOB_OBJECT_SET_SECURITY_ATTRIBUTES  (0x0010)
#define JOB_OBJECT_ALL_ACCESS               (STANDARD_RIGHTS_REQUIRED |\
                                             SYNCHRONIZE | 0x1F)
```

3.6.1.2.3 Thread

Thread Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686769.aspx

From WinNT.h ([SDK], [WDK]):

```
#define THREAD_TERMINATE                 (0x0001)
#define THREAD_SUSPEND_RESUME            (0x0002)
#define THREAD_GET_CONTEXT               (0x0008)
#define THREAD_SET_CONTEXT               (0x0010)
#define THREAD_QUERY_INFORMATION         (0x0040)
#define THREAD_SET_INFORMATION           (0x0020)
#define THREAD_SET_THREAD_TOKEN          (0x0080)
#define THREAD_IMPERSONATE               (0x0100)
#define THREAD_DIRECT_IMPERSONATION      (0x0200)
// begin_wdm
#define THREAD_SET_LIMITED_INFORMATION   (0x0400)  // winnt
#define THREAD_QUERY_LIMITED_INFORMATION (0x0800)  // winnt
#if (NTDDI_VERSION >= NTDDI_VISTA)
#define THREAD_ALL_ACCESS                (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE |\
                                          0xFFFF)
#else
#define THREAD_ALL_ACCESS                (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE |\
                                          0x3FF)
#endif
```

3.6.1.3 Window Station

Window Station Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms687391.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ DLLs, Processes and Threads \ Processes and Threads \ Windows Stations and Desktops \ About Window Stations and Desktops

From WinUser.h ([SDK], [WDK]):

```
/*
 * Windowstation-specific access flags
 */
#define WINSTA_ENUMDESKTOPS         0x0001L
#define WINSTA_READATTRIBUTES       0x0002L
#define WINSTA_ACCESSCLIPBOARD      0x0004L
#define WINSTA_CREATEDESKTOP        0x0008L
#define WINSTA_WRITEATTRIBUTES      0x0010L
#define WINSTA_ACCESSGLOBALATOMS    0x0020L
#define WINSTA_EXITWINDOWS          0x0040L
#define WINSTA_ENUMERATE            0x0100L
#define WINSTA_READSCREEN           0x0200L

#define WINSTA_ALL_ACCESS     (WINSTA_ENUMDESKTOPS    | WINSTA_READATTRIBUTES | \
                               WINSTA_ACCESSCLIPBOARD | WINSTA_CREATEDESKTOP  | \
                               WINSTA_WRITEATTRIBUTES                         | \
                               WINSTA_ACCESSGLOBALATOMS | WINSTA_EXITWINDOWS  | \
                               WINSTA_ENUMERATE         | WINSTA_READSCREEN)
```

3.6.1.4 Desktop

Desktop Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms682575.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ DLLs, Processes and Threads \ Processes and Threads \ Windows Stations and Desktops \ About Window Stations and Desktops

From WinUser.h ([SDK], [WDK]):

```
/*
 * Desktop-specific access flags
 */
#define DESKTOP_READOBJECTS         0x0001L
#define DESKTOP_CREATEWINDOW        0x0002L
#define DESKTOP_CREATEMENU          0x0004L
#define DESKTOP_HOOKCONTROL         0x0008L
#define DESKTOP_JOURNALRECORD       0x0010L
#define DESKTOP_JOURNALPLAYBACK     0x0020L
#define DESKTOP_ENUMERATE           0x0040L
#define DESKTOP_WRITEOBJECTS        0x0080L
#define DESKTOP_SWITCHDESKTOP       0x0100L
```

3.6.1.5 Services

3.6.1.5.1 Service Control Manager

Service Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms685981.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ DLLs, Processes and Threads \ Services \ About Services

From WinSvc.h ([SDK], [WDK]):

```
//
// Service Control Manager object specific access types
//
#define SC_MANAGER_CONNECT             0x0001
#define SC_MANAGER_CREATE_SERVICE      0x0002
#define SC_MANAGER_ENUMERATE_SERVICE   0x0004
#define SC_MANAGER_LOCK                0x0008
#define SC_MANAGER_QUERY_LOCK_STATUS   0x0010
#define SC_MANAGER_MODIFY_BOOT_CONFIG  0x0020

#define SC_MANAGER_ALL_ACCESS          (STANDARD_RIGHTS_REQUIRED      | \
                                        SC_MANAGER_CONNECT            | \
                                        SC_MANAGER_CREATE_SERVICE     | \
                                        SC_MANAGER_ENUMERATE_SERVICE  | \
                                        SC_MANAGER_LOCK               | \
                                        SC_MANAGER_QUERY_LOCK_STATUS  | \
                                        SC_MANAGER_MODIFY_BOOT_CONFIG)
```

3.6.1.5.2 Service

Service Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms685981.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ DLLs, Processes and Threads \ Services \ About Services

From WinSvc.h ([SDK], [WDK]):

```
//
// Service object specific access type
//
#define SERVICE_QUERY_CONFIG           0x0001
#define SERVICE_CHANGE_CONFIG          0x0002
#define SERVICE_QUERY_STATUS           0x0004
#define SERVICE_ENUMERATE_DEPENDENTS   0x0008
#define SERVICE_START                  0x0010
#define SERVICE_STOP                   0x0020
#define SERVICE_PAUSE_CONTINUE         0x0040
#define SERVICE_INTERROGATE            0x0080
#define SERVICE_USER_DEFINED_CONTROL   0x0100

#define SERVICE_ALL_ACCESS             (STANDARD_RIGHTS_REQUIRED     | \
                                        SERVICE_QUERY_CONFIG         | \
                                        SERVICE_CHANGE_CONFIG        | \
                                        SERVICE_QUERY_STATUS         | \
                                        SERVICE_ENUMERATE_DEPENDENTS | \
                                        SERVICE_START                | \
                                        SERVICE_STOP                 | \
                                        SERVICE_PAUSE_CONTINUE       | \
```

```
                                    SERVICE_INTERROGATE           | \
                                    SERVICE_USER_DEFINED_CONTROL)
```

3.6.1.6 Synchronization Objects

Synchronization Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686670.aspx

The Windows security model enables you to control access to event, mutex, semaphore, and waitable timer objects. Timer queues, interlocked variables, and critical section objects are not securable. For more information, see Access-Control Model.

You can specify a security descriptor for an interprocess synchronization object when you call the **CreateEvent**, **CreateMutex**, **CreateSemaphore**, or **CreateWaitableTimer** function. If you specify NULL, the object gets a default security descriptor. The Access-Control Lists (ACLs) in the default security descriptor for a synchronization object come from the primary or impersonation token of the creator.

To get or set the security descriptor of an event, mutex, semaphore, or waitable timer object, call the **GetNamedSecurityInfo**, **SetNamedSecurityInfo**, **GetSecurityInfo**, or **SetSecurityInfo** functions.

The handles returned by **CreateEvent**, **CreateMutex**, **CreateSemaphore**, or **CreateWaitableTimer** have full access to the new object. When you call the **OpenEvent**, **OpenMutex**, **OpenSemaphore**, and **OpenWaitableTimer** functions, the system checks the requested access rights against the object's security descriptor.

The valid access rights for the interprocess synchronization objects include the standard access rights and some object-specific access rights. The following table lists the standard access rights used by all objects.

Constant	Value	Description
DELETE	0x00010000	Required to delete the object.
READ_CONTROL	0x00020000	Required to read information in the security descriptor for the object, not including the information in the SACL. To read or write the SACL, you must request the ACCESS_SYSTEM_SECURITY access right. For more information, see SACL Access Right.
WRITE_DAC	0x00040000	Required to modify the DACL in the security descriptor for the object.
WRITE_OWNER	0x00080000	Required to change the owner in the security descriptor for the object.
SYNCHRONIZE	0x00100000	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state.

3.6.1.6.1 Event

Synchronization Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686670.aspx

The following table lists the object-specific access rights for event objects. These rights are supported in addition to the standard access rights.

Constant	Value	Description
EVENT_MODIFY_STATE	0x0002	Modify state access, which is required for the **SetEvent**, **ResetEvent** and **PulseEvent** functions.
EVENT_ALL_ACCESS	0x1F0003 (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)	All possible access rights for an event object. Use this right only if your application requires access beyond that granted by the standard access rights and EVENT_MODIFY_STATE. Using this access right increases the possibility that your application must be run by an Administrator.

From WinNT.h ([SDK], [WDK]):

```
#define EVENT_MODIFY_STATE      0x0002
#define EVENT_ALL_ACCESS       (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
```

3.6.1.6.2 Mutex

Synchronization Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686670.aspx

From WinNT.h ([SDK], [WDK]):

```
#define MUTANT_QUERY_STATE      0x0001
#define MUTANT_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED|SYNCHRONIZE|\
                           MUTANT_QUERY_STATE)
```

From winbase.h ([SDK], [WDK]):

```
#define MUTEX_MODIFY_STATE MUTANT_QUERY_STATE
#define MUTEX_ALL_ACCESS MUTANT_ALL_ACCESS
```

The following table lists the object-specific access rights for mutex objects. These rights are supported in addition to the standard access rights.

Constant	Value	Description

Constant	Value	Description
MUTEX_QUERY_STATE (MUTANT_QUERY_STATE)	0x0001	Reserved for future use.
MUTEX_ALL_ACCESS (MUTANT_ALL_ACCESS)	0x1F0001 (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | MUTANT_QUERY_STATE)	All possible access rights for a mutex object. Use this right only if your application requires access beyond that granted by the standard access rights. Using this access right increases the possibility that your application must be run by an Administrator.

3.6.1.6.3 Semaphore

Synchronization Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686670.aspx

From WinNT.h ([SDK], [WDK]):

```
#define SEMAPHORE_MODIFY_STATE      0x0002
#define SEMAPHORE_ALL_ACCESS       (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
```

The following table lists the object-specific access rights for semaphore objects. These rights are supported in addition to the standard access rights.

Constant	Value	Description
SEMAPHORE_MODIFY_STATE	0x0002	Modify state access, which is required for the **ReleaseSemaphore** function.
SEMAPHORE_ALL_ACCESS	0x1F0003 (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)	All possible access rights for a semaphore object. Use this right only if your application requires access beyond that granted by the standard access rights and SEMAPHORE_MODIFY_STATE. Using this access right increases the possibility that your application must be run by an Administrator.

From WinNT.h ([SDK], [WDK]):

```
#define SEMAPHORE_MODIFY_STATE      0x0002
#define SEMAPHORE_ALL_ACCESS       (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
```

3.6.1.6.4 Timer

Synchronization Object Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms686670.aspx

From WinNT.h ([SDK], [WDK]):

```
#define TIMER_QUERY_STATE       0x0001
#define TIMER_MODIFY_STATE      0x0002

#define TIMER_ALL_ACCESS       (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE |\
                                TIMER_QUERY_STATE | TIMER_MODIFY_STATE)
```

3.6.2 File Services

3.6.2.1 File Access Rights

File Security and Access Rights: http://msdn.microsoft.com/en-us/library/aa364399.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ File Services \ File Systems \ File Management \ About File Management

From WinNT.h ([SDK], [WDK]):

```
#define FILE_READ_DATA            ( 0x0001 )    // file & pipe
#define FILE_LIST_DIRECTORY       ( 0x0001 )    // directory

#define FILE_WRITE_DATA           ( 0x0002 )    // file & pipe
#define FILE_ADD_FILE             ( 0x0002 )    // directory

#define FILE_APPEND_DATA          ( 0x0004 )    // file
#define FILE_ADD_SUBDIRECTORY     ( 0x0004 )    // directory
#define FILE_CREATE_PIPE_INSTANCE ( 0x0004 )    // named pipe


#define FILE_READ_EA              ( 0x0008 )    // file & directory

#define FILE_WRITE_EA             ( 0x0010 )    // file & directory

#define FILE_EXECUTE              ( 0x0020 )    // file
#define FILE_TRAVERSE             ( 0x0020 )    // directory

#define FILE_DELETE_CHILD         ( 0x0040 )    // directory

#define FILE_READ_ATTRIBUTES      ( 0x0080 )    // all

#define FILE_WRITE_ATTRIBUTES     ( 0x0100 )    // all

#define FILE_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x1FF)

#define FILE_GENERIC_READ         (STANDARD_RIGHTS_READ     |\
                                   FILE_READ_DATA           |\
                                   FILE_READ_ATTRIBUTES     |\
                                   FILE_READ_EA             |\
```

```
                                    SYNCHRONIZE)

#define FILE_GENERIC_WRITE         (STANDARD_RIGHTS_WRITE    |\
                                    FILE_WRITE_DATA          |\
                                    FILE_WRITE_ATTRIBUTES    |\
                                    FILE_WRITE_EA            |\
                                    FILE_APPEND_DATA         |\
                                    SYNCHRONIZE)

#define FILE_GENERIC_EXECUTE       (STANDARD_RIGHTS_EXECUTE  |\
                                    FILE_READ_ATTRIBUTES     |\
                                    FILE_EXECUTE             |\
```

3.6.2.2 File Mapping

File Mapping Security and Access Rights: http://msdn.microsoft.com/en-us/library/aa366559.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Memory Management \ About Memory Management \ File Mapping

ZwCreateSection

From WinNT.h ([SDK], [WDK]):

```
#define SECTION_QUERY                0x0001
#define SECTION_MAP_WRITE            0x0002
#define SECTION_MAP_READ             0x0004
#define SECTION_MAP_EXECUTE          0x0008
#define SECTION_EXTEND_SIZE          0x0010
#define SECTION_MAP_EXECUTE_EXPLICIT 0x0020 // not included in SECTION_ALL_ACCESS

#define SECTION_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED|SECTION_QUERY|\
                            SECTION_MAP_WRITE |      \
                            SECTION_MAP_READ |       \
                            SECTION_MAP_EXECUTE |    \
                            SECTION_EXTEND_SIZE)
```

From WinBase.h ([SDK], [WDK]):

```
#define FILE_MAP_COPY       SECTION_QUERY
#define FILE_MAP_WRITE      SECTION_MAP_WRITE
#define FILE_MAP_READ       SECTION_MAP_READ
#define FILE_MAP_ALL_ACCESS SECTION_ALL_ACCESS
#define FILE_MAP_EXECUTE    SECTION_MAP_EXECUTE_EXPLICIT    // not included in
                                                            // FILE_MAP_ALL_ACCESS
```

3.6.2.3 Pipes

About Pipes: http://msdn.microsoft.com/en-us/library/aa365780.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Interprocess Communications \ Pipes \ About Pipes

3.6.2.3.1 Anonymous Pipes

Anonymous Pipe Security and Access Rights: http://msdn.microsoft.com/en-us/library/aa365142.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Interprocess Communications \ Pipes \ About Pipes \ Anonymous Pipes

Windows security enables you to control access to anonymous pipes. For more information about security, see Access-Control Model.

You can specify a security descriptor for a pipe when you call the **CreatePipe** function. The security descriptor controls access to both the read and write ends of the pipe. If you specify NULL, the pipe gets a default security descriptor. The ACLs in the default security descriptor for a pipe come from the primary or impersonation token of the creator.

To retrieve a pipe's security descriptor, call the **GetSecurityInfo** function. To change a pipe's security descriptor, call the **SetSecurityInfo** function.

The **CreatePipe** function returns two handles to the anonymous pipe: a read handle with GENERIC_READ and SYNCHRONIZE access; and a write handle with GENERIC_WRITE and SYNCHRONIZE access. GENERIC_READ and GENERIC_WRITE access use the same access rights mapping as for named pipes.

GENERIC_READ access for an anonymous pipe combines the rights to read data from the pipe, read pipe attributes, read extended attributes, and read the pipe's DACL.

GENERIC_WRITE access for an anonymous pipe combines the rights to write data to the pipe, append data to it, write pipe attributes, write extended attributes, and read the pipe's DACL.

3.6.2.3.2 Named Pipes

Named Pipe Security and Access Rights: http://msdn.microsoft.com/en-us/library/aa365600.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Interprocess Communications \ Pipes \ About Pipes \ Named Pipes

From WinBase.h ([SDK], [WDK]):

```
//
// Define the dwOpenMode values for CreateNamedPipe
//
#define PIPE_ACCESS_INBOUND         0x00000001
#define PIPE_ACCESS_OUTBOUND        0x00000002
#define PIPE_ACCESS_DUPLEX          0x00000003
```

3.6.2.4 Registry

Registry Key Security and Access Rights: http://msdn.microsoft.com/en-us/library/ms724878.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Windows System Information \ Registry \ About the Registry

From WinNT.h ([SDK], [WDK]):

```
//
// Registry Specific Access Rights.
//

#define KEY_QUERY_VALUE         (0x0001)
#define KEY_SET_VALUE           (0x0002)
#define KEY_CREATE_SUB_KEY      (0x0004)
#define KEY_ENUMERATE_SUB_KEYS  (0x0008)
#define KEY_NOTIFY              (0x0010)
#define KEY_CREATE_LINK         (0x0020)
#define KEY_WOW64_32KEY         (0x0200)
#define KEY_WOW64_64KEY         (0x0100)
#define KEY_WOW64_RES           (0x0300)

#define KEY_READ                ((STANDARD_RIGHTS_READ       |\
                                  KEY_QUERY_VALUE            |\
                                  KEY_ENUMERATE_SUB_KEYS     |\
                                  KEY_NOTIFY)                 \
                                  &                           \
                                 (~SYNCHRONIZE))


#define KEY_WRITE               ((STANDARD_RIGHTS_WRITE      |\
                                  KEY_SET_VALUE              |\
                                  KEY_CREATE_SUB_KEY)         \
                                  &                           \
                                 (~SYNCHRONIZE))

#define KEY_EXECUTE             ((KEY_READ)                   \
                                  &                           \
                                 (~SYNCHRONIZE))

#define KEY_ALL_ACCESS          ((STANDARD_RIGHTS_ALL        |\
                                  KEY_QUERY_VALUE            |\
                                  KEY_SET_VALUE              |\
                                  KEY_CREATE_SUB_KEY         |\
                                  KEY_ENUMERATE_SUB_KEYS     |\
                                  KEY_NOTIFY                 |\
                                  KEY_CREATE_LINK)            \
                                  &                           \
                                 (~SYNCHRONIZE))
```

3.6.3 Kernel Transaction Manager (KTM)

Kernel Transaction Manager Constants: http://msdn.microsoft.com/en-us/library/aa366267.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Kernel Transaction Manager \ KTM Reference

3.6.3.1 Enlistment (KTM)

Enlistment Access Masks: http://msdn.microsoft.com/en-us/library/aa366021.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Kernel Transaction Manager \ KTM Reference \ Kernel Transaction Manager Constants

From WinNT.h ([SDK], [WDK]):

```
//
// KTM enlistment object rights.
//
#define ENLISTMENT_QUERY_INFORMATION     ( 0x0001 )
#define ENLISTMENT_SET_INFORMATION       ( 0x0002 )
#define ENLISTMENT_RECOVER               ( 0x0004 )
#define ENLISTMENT_SUBORDINATE_RIGHTS    ( 0x0008 )
#define ENLISTMENT_SUPERIOR_RIGHTS       ( 0x0010 )

//
// Generic mappings for enlistment rights.
//
#define ENLISTMENT_GENERIC_READ        (STANDARD_RIGHTS_READ           |\
                                        ENLISTMENT_QUERY_INFORMATION)

#define ENLISTMENT_GENERIC_WRITE       (STANDARD_RIGHTS_WRITE          |\
                                        ENLISTMENT_SET_INFORMATION     |\
                                        ENLISTMENT_RECOVER             |\
                                        ENLISTMENT_SUBORDINATE_RIGHTS  |\
                                        ENLISTMENT_SUPERIOR_RIGHTS)

#define ENLISTMENT_GENERIC_EXECUTE     (STANDARD_RIGHTS_EXECUTE        |\
                                        ENLISTMENT_RECOVER             |\
                                        ENLISTMENT_SUBORDINATE_RIGHTS  |\
                                        ENLISTMENT_SUPERIOR_RIGHTS)

#define ENLISTMENT_ALL_ACCESS          (STANDARD_RIGHTS_REQUIRED       |\
                                        ENLISTMENT_GENERIC_READ        |\
                                        ENLISTMENT_GENERIC_WRITE       |\
                                        ENLISTMENT_GENERIC_EXECUTE)
```

3.6.3.2 Resource Manager (KTM)

Resource Manager Access Masks: http://msdn.microsoft.com/en-us/library/aa366359.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Kernel Transaction Manager \ KTM Reference \ Kernel Transaction Manager Constants

From WinNT.h ([SDK], [WDK]):

```
//
// KTM resource manager object rights.
//
#define RESOURCEMANAGER_QUERY_INFORMATION     ( 0x0001 )
#define RESOURCEMANAGER_SET_INFORMATION       ( 0x0002 )
#define RESOURCEMANAGER_RECOVER               ( 0x0004 )
#define RESOURCEMANAGER_ENLIST                ( 0x0008 )
#define RESOURCEMANAGER_GET_NOTIFICATION      ( 0x0010 )
```

```
#define RESOURCEMANAGER_REGISTER_PROTOCOL     ( 0x0020 )
#define RESOURCEMANAGER_COMPLETE_PROPAGATION  ( 0x0040 )

//
// Generic mappings for resource manager rights.
//
#define RESOURCEMANAGER_GENERIC_READ       (STANDARD_RIGHTS_READ                 |\
                                            RESOURCEMANAGER_QUERY_INFORMATION    |\
                                            SYNCHRONIZE)

#define RESOURCEMANAGER_GENERIC_WRITE      (STANDARD_RIGHTS_WRITE                |\
                                            RESOURCEMANAGER_SET_INFORMATION      |\
                                            RESOURCEMANAGER_RECOVER              |\
                                            RESOURCEMANAGER_ENLIST               |\
                                            RESOURCEMANAGER_GET_NOTIFICATION     |\
                                            RESOURCEMANAGER_REGISTER_PROTOCOL    |\
                                            RESOURCEMANAGER_COMPLETE_PROPAGATION |\
                                            SYNCHRONIZE)

#define RESOURCEMANAGER_GENERIC_EXECUTE    (STANDARD_RIGHTS_EXECUTE              |\
                                            RESOURCEMANAGER_RECOVER              |\
                                            RESOURCEMANAGER_ENLIST               |\
                                            RESOURCEMANAGER_GET_NOTIFICATION     |\
                                            RESOURCEMANAGER_COMPLETE_PROPAGATION |\
                                            SYNCHRONIZE)

#define RESOURCEMANAGER_ALL_ACCESS         (STANDARD_RIGHTS_REQUIRED             |\
                                            RESOURCEMANAGER_GENERIC_READ         |\
                                            RESOURCEMANAGER_GENERIC_WRITE        |\
                                            RESOURCEMANAGER_GENERIC_EXECUTE)
```

3.6.3.3 Transaction (KTM)

Transaction Access Masks: http://msdn.microsoft.com/en-us/library/aa366384.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Kernel Transaction Manager \ KTM Reference \ Kernel Transaction Manager Constants

From WinNT.h ([SDK], [WDK]):

```
//
// KTM transaction object rights.
//
#define TRANSACTION_QUERY_INFORMATION     ( 0x0001 )
#define TRANSACTION_SET_INFORMATION       ( 0x0002 )
#define TRANSACTION_ENLIST                ( 0x0004 )
#define TRANSACTION_COMMIT                ( 0x0008 )
#define TRANSACTION_ROLLBACK              ( 0x0010 )
#define TRANSACTION_PROPAGATE             ( 0x0020 )
#define TRANSACTION_RIGHT_RESERVED1       ( 0x0040 )

//
// Generic mappings for transaction rights.
// Resource managers, when enlisting, should generally use the macro
// TRANSACTION_RESOURCE_MANAGER_RIGHTS when opening a transaction.
// It's the same as generic read and write except that it does not allow
// a commit decision to be made.
//

#define TRANSACTION_GENERIC_READ            (STANDARD_RIGHTS_READ            |\
```

```
                                        TRANSACTION_QUERY_INFORMATION   |\
                                        SYNCHRONIZE)

#define TRANSACTION_GENERIC_WRITE       (STANDARD_RIGHTS_WRITE          |\
                                        TRANSACTION_SET_INFORMATION     |\
                                        TRANSACTION_COMMIT              |\
                                        TRANSACTION_ENLIST              |\
                                        TRANSACTION_ROLLBACK            |\
                                        TRANSACTION_PROPAGATE           |\
                                        SYNCHRONIZE)

#define TRANSACTION_GENERIC_EXECUTE     (STANDARD_RIGHTS_EXECUTE        |\
                                        TRANSACTION_COMMIT              |\
                                        TRANSACTION_ROLLBACK            |\
                                        SYNCHRONIZE)

#define TRANSACTION_ALL_ACCESS          (STANDARD_RIGHTS_REQUIRED       |\
                                        TRANSACTION_GENERIC_READ        |\
                                        TRANSACTION_GENERIC_WRITE       |\
                                        TRANSACTION_GENERIC_EXECUTE)

#define TRANSACTION_RESOURCE_MANAGER_RIGHTS (TRANSACTION_GENERIC_READ   |\
                                        STANDARD_RIGHTS_WRITE           |\
                                        TRANSACTION_SET_INFORMATION     |\
                                        TRANSACTION_ENLIST              |\
                                        TRANSACTION_ROLLBACK            |\
                                        TRANSACTION_PROPAGATE           |\
                                        SYNCHRONIZE)
```

3.6.3.4 Transaction Manager

Transaction Manager Access Masks: http://msdn.microsoft.com/en-us/library/aa366390.aspx

MSDN: MSDN Library \ Win32 and COM Development \ System Services \ Kernel Transaction Manager \ KTM Reference \ Kernel Transaction Manager Constants

From WinNT.h ([SDK], [WDK]):

```
//
// KTM Tm object rights
//
#define TRANSACTIONMANAGER_QUERY_INFORMATION     ( 0x0001 )
#define TRANSACTIONMANAGER_SET_INFORMATION       ( 0x0002 )
#define TRANSACTIONMANAGER_RECOVER               ( 0x0004 )
#define TRANSACTIONMANAGER_RENAME                ( 0x0008 )
#define TRANSACTIONMANAGER_CREATE_RM             ( 0x0010 )

// The following right is intended for DTC's use only; it will be
// deprecated, and no one else should take a dependency on it.
#define TRANSACTIONMANAGER_BIND_TRANSACTION      ( 0x0020 )

//
// Generic mappings for transaction manager rights.
//
#define TRANSACTIONMANAGER_GENERIC_READ     (STANDARD_RIGHTS_READ               |\
                                             TRANSACTIONMANAGER_QUERY_INFORMATION)

#define TRANSACTIONMANAGER_GENERIC_WRITE    (STANDARD_RIGHTS_WRITE              |\
                                             TRANSACTIONMANAGER_SET_INFORMATION |\
                                             TRANSACTIONMANAGER_RECOVER         |\
```

```
                                                TRANSACTIONMANAGER_RENAME          |\
                                                TRANSACTIONMANAGER_CREATE_RM)

#define TRANSACTIONMANAGER_GENERIC_EXECUTE  (STANDARD_RIGHTS_EXECUTE)

#define TRANSACTIONMANAGER_ALL_ACCESS       (STANDARD_RIGHTS_REQUIRED          |\
                                             TRANSACTIONMANAGER_GENERIC_READ   |\
                                             TRANSACTIONMANAGER_GENERIC_WRITE  |\
                                             TRANSACTIONMANAGER_GENERIC_EXECUTE |\
                                             TRANSACTIONMANAGER_BIND_TRANSACTION)
```

3.6.4 Memory Management

ZwCreateSection: http://msdn.microsoft.com/en-us/library/ms804361.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Windows Driver Kit \ Kernel-Mode Driver Architecture \ Reference \ Driver Support Routines \ ZwXxx Routines

Section Objects and Views, http://msdn.microsoft.com/en-us/library/ms796304.aspx

From WinNT.h ([SDK], [WDK]):

```
#define SECTION_QUERY               0x0001
#define SECTION_MAP_WRITE           0x0002
#define SECTION_MAP_READ            0x0004
#define SECTION_MAP_EXECUTE         0x0008
#define SECTION_EXTEND_SIZE         0x0010
#define SECTION_MAP_EXECUTE_EXPLICIT 0x0020 // not included in SECTION_ALL_ACCESS

#define SECTION_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED|SECTION_QUERY|\
                            SECTION_MAP_WRITE |      \
                            SECTION_MAP_READ |       \
                            SECTION_MAP_EXECUTE |    \
                            SECTION_EXTEND_SIZE)
```

3.7 Installable File System Drivers (Windows Driver Kit)

FltBuildDefaultSecurityDescriptor: http://msdn.microsoft.com/en-us/library/aa488551.aspx

MSDN: MSDN Library \ Win32 and COM Development \ Windows Driver Kit \ Installable File System Drivers \ Reference \ FltXxx (Minifilter Driver) Routines

From fltKernel.h ([WDK]):

```
//
//  Access masks for filter communication ports
//

#define FLT_PORT_CONNECT        0x0001
#define FLT_PORT_ALL_ACCESS     (FLT_PORT_CONNECT | STANDARD_RIGHTS_ALL)
```

3.8 Open Specifications

3.8.1 Printing (Windows Communication Protocols (MCPP))

[MS-RPRN]: Print System Remote Protocol Specification: http://msdn.microsoft.com/en-us/library/cc244528(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-RPRN]: Print System Remote Protocol Specification

3.8.1.1.1 Print Jobs

[MS-RPRN] 2.2.3.1 Access Values: http://msdn.microsoft.com/en-us/library/cc244650(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-RPRN]: Print System Remote Protocol Specification

Using the Windows Headers: http://msdn.microsoft.com/en-us/library/aa383745(VS.85).aspx

From WinSpool.h ([SDK], [WDK]):

```
#define JOB_ACCESS_ADMINISTER       0x00000010

#if ((NTDDI_VERSION >= NTDDI_WINXPSP1) || \
     ((OSVER(NTDDI_VERSION) == NTDDI_WIN2K) && (SPVER(NTDDI_VERSION) >= 3)))
    #define JOB_ACCESS_READ         0x00000020
#endif // ((NTDDI_VERSION >= NTDDI_WINXPSP1) ...

#if ((NTDDI_VERSION >= NTDDI_WINXPSP1) || \
     ((OSVER(NTDDI_VERSION) == NTDDI_WIN2K) && (SPVER(NTDDI_VERSION) >= 3)))
    #define JOB_ALL_ACCESS          (STANDARD_RIGHTS_REQUIRED    |\
                                     JOB_ACCESS_ADMINISTER       |\
                                     JOB_ACCESS_READ)
#else
    #define JOB_ALL_ACCESS          (STANDARD_RIGHTS_REQUIRED    |\
                                     JOB_ACCESS_ADMINISTER)
#endif

#if ((NTDDI_VERSION >= NTDDI_WINXPSP1) || \
     ((OSVER(NTDDI_VERSION) == NTDDI_WIN2K) && (SPVER(NTDDI_VERSION) >= 3)))
    #define JOB_READ                (STANDARD_RIGHTS_READ        |\
                                     JOB_ACCESS_READ)
#else
    #define JOB_READ                (STANDARD_RIGHTS_READ        |\
                                     JOB_ACCESS_ADMINISTER)

#endif

#define JOB_WRITE               (STANDARD_RIGHTS_WRITE       |\
                                 JOB_ACCESS_ADMINISTER)

#define JOB_EXECUTE             (STANDARD_RIGHTS_EXECUTE     |\
                                 JOB_ACCESS_ADMINISTER)
```

Minimum system required	Value for NTDDI_VERSION

Minimum system required	Value for NTDDI_VERSION
Windows XP with Service Pack 1 (SP1)	NTDDI_WINXPSP1
Windows 2000 with Service Pack 3 (SP3)	NTDDI_WIN2KSP3
Windows 2000	NTDDI_WIN2K

For Windows 2000 Service Pack 3, or Windows XP Service Pack 1 and beyond:

Constant	Value	Description
JOB_ACCESS_ADMINISTER	0x00000010	Printing-specific authorization to cancel, pause, resume, or restart the job.
JOB_ACCESS_READ	0x00000020	Printing-specific read rights for the spool file.
JOB_READ	(STANDARD_RIGHTS_READ | JOB_ACCESS_READ)	
JOB_WRITE	(STANDARD_RIGHTS_WRITE | JOB_ACCESS_ADMINISTER)	
JOB_EXECUTE	(STANDARD_RIGHTS_EXECUTE | JOB_ACCESS_ADMINISTER)	
JOB_ALL_ACCESS	(STANDARD_RIGHTS_REQUIRED | JOB_ACCESS_ADMINISTER | JOB_ACCESS_READ)	Access rights for printers to perform all administrative tasks and basic printing operations except SYNCHRONIZE.

For < Windows 2000 Service Pack 3, or < Windows XP Service Pack 1:

Constant	Value	Description
JOB_ACCESS_ADMINISTER	0x00000010	Printing-specific authorization to cancel, pause, resume, or restart the job.
JOB_ACCESS_READ	0x00000020	Printing-specific read rights for the spool file.
JOB_READ	(STANDARD_RIGHTS_READ | JOB_ACCESS_ADMINISTER)	
JOB_WRITE	(STANDARD_RIGHTS_WRITE | JOB_ACCESS_ADMINISTER)	
JOB_EXECUTE	(STANDARD_RIGHTS_EXECUTE | JOB_ACCESS_ADMINISTER)	
JOB_ALL_ACCESS	(STANDARD_RIGHTS_REQUIRED | JOB_ACCESS_ADMINISTER)	Access rights for printers to perform all administrative tasks and basic printing operations except SYNCHRONIZE.

3.8.1.1.2 Print Server Printer

[MS-RPRN] 2.2.3.1 Access Values: http://msdn.microsoft.com/en-us/library/cc244650(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-RPRN]: Print System Remote Protocol Specification

From WinSpool.h ([SDK], [WDK]):

```
/*
 * Access rights for printers
 */
#define PRINTER_ACCESS_ADMINISTER   0x00000004
#define PRINTER_ACCESS_USE          0x00000008

#define PRINTER_ALL_ACCESS    (STANDARD_RIGHTS_REQUIRED     |\
                               PRINTER_ACCESS_ADMINISTER    |\
                               PRINTER_ACCESS_USE)

#define PRINTER_READ          (STANDARD_RIGHTS_READ         |\
                               PRINTER_ACCESS_USE)

#define PRINTER_WRITE         (STANDARD_RIGHTS_WRITE        |\
                               PRINTER_ACCESS_USE)

#define PRINTER_EXECUTE       (STANDARD_RIGHTS_EXECUTE      |\
                               PRINTER_ACCESS_USE)



#define PRINTER_ACCESS_ADMINISTER   0x00000004
#define PRINTER_ACCESS_USE          0x00000008
```

3.8.1.1.3 Print Server Remote Protocol

[MS-RPRN] 2.2.3.1 Access Values: http://msdn.microsoft.com/en-us/library/cc244650(PROT.10).aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Windows Communication Protocols (MCPP) \ [MS-RPRN]: Print System Remote Protocol Specification

From WinSpool.h ([SDK], [WDK]):

```
#define SERVER_ACCESS_ADMINISTER    0x00000001
#define SERVER_ACCESS_ENUMERATE     0x00000002

/*
 * Access rights for print servers
 */

#define SERVER_ALL_ACCESS    (STANDARD_RIGHTS_REQUIRED      |\
                              SERVER_ACCESS_ADMINISTER      |\
                              SERVER_ACCESS_ENUMERATE)

#define SERVER_READ          (STANDARD_RIGHTS_READ          |\
```

```
                                SERVER_ACCESS_ENUMERATE)

#define SERVER_WRITE           (STANDARD_RIGHTS_WRITE         |\
                                SERVER_ACCESS_ADMINISTER      |\
                                SERVER_ACCESS_ENUMERATE)

#define SERVER_EXECUTE         (STANDARD_RIGHTS_EXECUTE       |\
                                SERVER_ACCESS_ENUMERATE)
```

3.8.2 Windows Internet Naming Service (WINS)

[MS-RAIW]: Remote Administrative Interface: WINS Specification: http://msdn.microsoft.com/en-us/library/dd240484.aspx

MSDN: MSDN Library \ Open Specifications \ Windows Protocols \ Application Services and .NET Framework Protocols

[MS-RAIW] 2.1.1 Server Security Settings: http://msdn.microsoft.com/en-us/library/dd240499.aspx

Please note that at the time this document was created (March 30, 2010), the below values were referenced, but not defined in [MS-RAIW].

```
WINS_QUERY_ACCESS                ( 0x0002 ) // 0: No Query  1: Query
WINS_CONTROL_ACCESS              ( 0X0001 ) // 0: Read Only 1: Read/Write
```