|  |
| --- |
| **[TOC](" \l "toc)** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Draft | N. Sakimura | |  | NRI | |  | J. Bradley, Ed. | |  | Independent | |  | M. Jones | |  | Microsoft | |  | E. Jay | |  | MGI1 | |  | September 30, 2011 | |

# OpenID Connect Discovery 1.0 - draft 06

### Abstract

OpenID Connect is an identity protocol that provides authentication, authorization, and attribute transmission capability. It allows third party attested claims from distributed sources. The specification suite builds on OAuth 2.0 and consists of Building Blocks (Messages, Discovery, Dynamic Client Registration, Session Management, JSON Web Token, JSON Web Signature, JSON WEB Encryption, JSON Web Keys, Simple Web Discovery), Protocol Bindings (e.g., Standard and Basic Client) and Extensions. This specification is the "Discovery" part of the suite that defines how user and server endpoints are discovered.

### Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.)](#RFC2119) [RFC2119].

### Table of Contents

[**1.**](#anchor1)  Introduction  
[**2.**](#terminology)  Terminology  
[**3.**](#ProviderDisc)  Provider Discovery  
    [**3.1.**](#anchor2)  Identifier Normalization  
        [**3.1.1.**](#anchor3)  Identifier Type  
        [**3.1.2.**](#anchor4)  E-mail Address  
        [**3.1.3.**](#anchor5)  URL  
    [**3.2.**](#anchor6)  Non-Normative Examples  
        [**3.2.1.**](#anchor7)  E-Mail Address  
        [**3.2.2.**](#anchor8)  URL  
        [**3.2.3.**](#anchor9)  Account URI  
        [**3.2.4.**](#anchor10)  Hostname & Port  
    [**3.3.**](#anchor11)  Redirection  
    [**3.4.**](#anchor12)  Error  
[**4.**](#ProviderConfig)  Provider Configuration Information  
    [**4.1.**](#anchor13)  Provider Configuration Request  
    [**4.2.**](#anchor14)  Provider Configuration Response  
    [**4.3.**](#anchor15)  Provider Configuration Verification  
[**5.**](#IANA)  IANA Considerations  
[**6.**](#Security)  Security Considerations  
[**7.**](#anchor16)  Open Issues and Things To Be Done (TBD)  
[**8.**](#rfc.references1)  References  
    [**8.1.**](#rfc.references1)  Normative References  
    [**8.2.**](#rfc.references2)  Informative References  
[**Appendix A.**](#Acknowledgements)  Acknowledgements  
[**Appendix B.**](#anchor19)  Document History  
[**§**](#rfc.authors)  Authors' Addresses

|  |
| --- |
| [**TOC**](#toc) |

### 1.  Introduction

In order for an OpenID client to utilize OpenID Connect services for a user, the client needs to know where the OpenID Provider is. OpenID Connect uses [Simple Web Discovery (Jones, M., Ed. and Y. Goland, “Simple Web Discovery,” July 2011.)](#SWD) [SWD] to locate the OpenID Provider for an end-user.

Once an OpenID Provider is identified, the endpoint and other configuration information for that OP is retrieved from a well-known location as a JSON document.

|  |
| --- |
| [**TOC**](#toc) |

### 2.  Terminology

In addition to the terms "Access Token", "Refresh Token", "Authorization Code", "Authorization Grant", "Authorization Server", "Authorization Endpoint", "Client", "Client Identifier", "Client Secret", "Protected Resource", "Resource Owner", "Resource Server", and "Token Endpoint" that are defined by [OAuth 2.0 (Hammer-Lahav, E., Ed., Recordon, D., and D. Hardt, “OAuth 2.0 Authorization Protocol,” July 2011.)](#OAuth2.0) [OAuth2.0], and the terminology defined in the [OpenID Connect Messages 1.0 (Sakimura, N., Recordon, D., Bradley, J., de Medeiros, B., Jones, M., and E. Jay, “OpenID Connect Messages 1.0,” September 2011.)](#OpenID.Messages) [OpenID.Messages] specification, the following terms are defined:

Principal

A resource that is the target of a request in Simple Web Discovery.

Issuer

A verifiable identifier for the OpenID Provider. An issuer is a HTTPS URL with no path component.

Identifier

An Identifier is either an http or https URI (commonly referred to as a URL within this document), or an account URI. This document defines various kinds of Identifiers, designed for use in different contexts.

|  |
| --- |
| [**TOC**](#toc) |

### 3.  Provider Discovery

OpenID Provider discovery is optional; if a Relying Party knows the OP information through an out-of-band mechanism, they can skip this step and proceed to [Section 4 (Provider Configuration Information)](#ProviderConfig).

Provider discovery requires the following information to make a discovery request:

Principal

Identifier of the target end-user who is the subject of the discovery request

Host

Server where a Simple Web Discovery service is hosted

Service

URI identifying the type of service whose location is requested

OpenID Connect uses the following discoverable service in Simple Web Discovery (SWD):

|  |  |
| --- | --- |
| **Service Type** | **URI** |
| OpenID Connect Issuer | http://openid.net/specs/connect/1.0/issuer |

To start discovery of OpenID end points, the end-user supplies an identifier to the client or relying party. The client applies the normalization rules to the identifier to extract the principal and host. Then it makes a HTTPS request to the host's Simple Web Discovery endpoint with the principal and service parameters to obtain the location of the requested service.

What MUST be returned in the response is the issuer. This includes URI scheme, HOST, and OPTIONALLY, port.

|  |
| --- |
| [**TOC**](#toc) |

### 3.1.  Identifier Normalization

The purpose of normalization is to extract a normalized principal and host from the user input. This is then used as input to SWD to discover the issuer.

The user identifier can be one of the following:

* xri
* E-mail address
* URL

The identifier normalization rules are not extensible.

URLs without a host segment are not supported by this discovery specification.

|  |
| --- |
| [**TOC**](#toc) |

### 3.1.1.  Identifier Type

1. Identifiers starting with the [XRI (Reed, D. and D. McAlpin, “Extensible Resource Identifier (XRI) Syntax V2.0,” November 2005.)](#XRI_Syntax_2.0) [XRI\_Syntax\_2.0] characters ('=','@', and '!') are reserved. Processing of these is out of scope for this document.
2. Any identifier that contains the character '@' in any other position other than the first position must be treated as an e-mail address.
3. All other identifiers are treated as a URL.

|  |
| --- |
| [**TOC**](#toc) |

### 3.1.2.  E-mail Address

If the identifier is an e-mail address, the principal is the e-mail address and the host is the portion to the right of the '@' character.

|  |
| --- |
| [**TOC**](#toc) |

### 3.1.3.  URL

A URL identifier is normalized according to the following rules:

* If the URL does not have a Scheme part, the string "https://" is prefixed to the URL as the [Scheme (Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” January 2005.)](#RFC3986) [RFC3986].
* If the URL contains a fragment part, it MUST be stripped off together with the fragment delimiter character "#".
* The resulting URL is used as the principal and the host is extracted from it according to [URI (Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” January 2005.)](#RFC3986) [RFC3986] syntax rules.

|  |
| --- |
| [**TOC**](#toc) |

### 3.2.  Non-Normative Examples

|  |
| --- |
| [**TOC**](#toc) |

### 3.2.1.  E-Mail Address

To find the issuer for the given e-mail address, joe@example.com, the SWD parameters are as follows:

|  |  |
| --- | --- |
| **SWD Parameter** | **Value** |
| principal | joe@example.com |
| host | example.com |
| service | http://openid.net/specs/connect/1.0/issuer |

Following the SWD specification, the client would make the following request to get the discovery information:

GET /.well-known/simple-web-discovery?principal=joe%40example%2Ecom&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.com

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"locations":["https://example.com/auth"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 3.2.2.  URL

To find the issuer for the given URL, https://example.com/joe, the SWD parameters are as follows:

|  |  |
| --- | --- |
| **SWD Parameter** | **Value** |
| principal | https://example.com/joe |
| host | example.com |
| service | http://openid.net/specs/connect/1.0/issuer |

Following the SWD specification, the client would make the following request to get the discovery information:

GET /.well-known/simple-web-discovery?principal=https%3A%2F%2Fexample%2Ecom%2Fjoe&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.com

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"locations":["https://example.com"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 3.2.3.  Account URI

To find the issuer for the given URI, acct://joe@example.com, the SWD parameters are as follows:

|  |  |
| --- | --- |
| **SWD Parameter** | **Value** |
| principal | acct://joe@example.com/ |
| host | example.com |
| service | http://openid.net/specs/connect/1.0/issuer |

Following the SWD specification, the client would make the following request to get the discovery information:

GET /.well-known/simple-web-discovery?principal=acct%3A%2F%2Fjoe%40example%2Ecom%2F&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.com

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"locations":["https://example.com"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 3.2.4.  Hostname & Port

To find the issuer for the given hostname, example.com:8080, the SWD parameters are as follows:

|  |  |
| --- | --- |
| **SWD Parameter** | **Value** |
| principal | https://example.com:8080/ |
| host | example.com:8080 |
| service | http://openid.net/specs/connect/1.0/issuer |

Following the SWD specification, the client would make the following request to get the discovery information:

GET /.well-known/simple-web-discovery?principal=https%3A%2F%2Fexample%2Ecom%3A8080%2F&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.com:8080

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"locations":["https://example.com"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 3.3.  Redirection

In cases where the SWD request is handled at a host or location other than the one derived from the end-user's identifier, the host will return a JSON object containing the new location.

To enable service level redirection a SWD server MAY return a 200 O.k. to an HTTPS request with a content type of application/json (or whatever other content type has been negotiated) that contains a JSON object that contains a member pair whose name is the string "SWD\_service\_redirect" whose value is a JSON object with a member pair whose name is "location" and whose value is a string that encodes a URI. Optionally the JSON object value of "SWD\_service\_redirect" MAY also contain a member whose name is "expires" and whose value is a JSON number that encodes an integer.

The "location" member identifies the URI that the caller MUST redirect all SWD requests for that domain to until the "expires" time is met. SWD requests for the redirected domain MUST be constructed by taking the URI returned in the "location" and using it as the base URI to which the SWD form arguments are then added as query parameters. The location URI MUST NOT include a query component.

GET /.well-known/simple-web-discovery?principal=joe%40example%2Ecom&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.com

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"SWD\_service\_redirect":

{

"location":"https://example.net/swd\_server"

}

}

GET /swd\_server?principal=joe%40example%2Ecom&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer HTTP/1.1

Host: example.net

HTTP/1.1 200 O.K.

Content-Type: application/json

{

"locations":["https://example.net/auth"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 3.4.  Error

If the Simple Web Discovery endpoint is unreachable or returns an error, then the RP may prepend https: to the host from [Sec 3.1 (Provider Discovery)](#ProviderDisc) and use that as the issuer.

|  |
| --- |
| [**TOC**](#toc) |

### 4.  Provider Configuration Information

This step is optional. The OpenID Provider endpoints and configuration information may be obtained out-of-band.

Using the issuer discovered in [Section 3 (Provider Discovery)](#ProviderDisc) or through direct configuration the OpenID Provider's configuration can be retrieved.

OpenID Providers MUST make available a JSON document at the path .well-known/openid-configuration. The syntax and semantics of .well-known are defined in [RFC 5785 (Nottingham, M. and E. Hammer-Lahav, “Defining Well-Known Uniform Resource Identifiers (URIs),” April 2010.)](#RFC5785) [RFC5785]. openid-configuration MUST point to a JSON document compliant with this specification.

OpenID Providers MUST support receiving SWD requests via TLS 1.2 as defined in [RFC 5246 (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.)](#RFC5246) [RFC5246] and MAY support other transport layer security mechanisms of equivalent security.

|  |
| --- |
| [**TOC**](#toc) |

### 4.1.  Provider Configuration Request

An OpenID Provider Configuration Document is queried using a HTTPS GET request with the previously specified path.

The client would make the following request to the issuer using TLS to get the Configuration information

GET /.well-known/openid-configuration HTTP/1.1

Host: example.com

|  |
| --- |
| [**TOC**](#toc) |

### 4.2.  Provider Configuration Response

The response is a set of claims about the OpenID Provider's configuration, including all necessary endpoint, supported scope, and public key location information.

The response MUST return a plain text JSON object that contains a set of claims that are a subset of those defined below.

Claims that return multiple values are JSON arrays. Claims with 0 elements must be omitted from the response.

Other claims MAY also be returned.

|  |  |  |
| --- | --- | --- |
| **Claim** | **Type** | **Description** |
| version | string | Version of the provider response. "3.0" is the default. |
| issuer | string | The https: URL with no path component the OP asserts as its issuer identifier |
| authorization\_endpoint | string | URL of the OP's Authentication and Authorization Endpoint [[OpenID.Messages] (Sakimura, N., Recordon, D., Bradley, J., de Medeiros, B., Jones, M., and E. Jay, “OpenID Connect Messages 1.0,” September 2011.)](#OpenID.Messages) |
| token\_endpoint | string | URL of the OP's OAuth 2.0 Token Endpoint [[OpenID.Messages] (Sakimura, N., Recordon, D., Bradley, J., de Medeiros, B., Jones, M., and E. Jay, “OpenID Connect Messages 1.0,” September 2011.)](#OpenID.Messages) |
| user\_info\_endpoint | string | URL of the OP's UserInfo Endpoint [[OpenID.Messages] (Sakimura, N., Recordon, D., Bradley, J., de Medeiros, B., Jones, M., and E. Jay, “OpenID Connect Messages 1.0,” September 2011.)](#OpenID.Messages) |
| check\_id\_endpoint | string | URL of the OP's Check ID Endpoint [[OpenID.Session] (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Session Management 1.0,” September 2011.)](#OpenID.Session) |
| refresh\_session\_endpoint | string | URL of the OP's Refresh Session Endpoint [[OpenID.Session] (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Session Management 1.0,” September 2011.)](#OpenID.Session) |
| end\_session\_endpoint | string | URL of the OP's End Session Endpoint [[OpenID.Session] (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Session Management 1.0,” September 2011.)](#OpenID.Session) |
| jwk\_document | string | URL of the OP's JSON Web Key [[JWK] (Jones, M., “JSON Web Key (JWK),” July 2011.)](#JWK) document |
| x509\_url | string | URL of the OP's X.509 certificates in PEM format. |
| registration\_endpoint | string | URL of the OP's Dynamic Client Registration Endpoint [[OpenID.Registration] (Sakimura, N., Bradley, J., Ed., and M. Jones, “OpenID Connect Dynamic Client Registration 1.0,” September 2011.)](#OpenID.Registration) |
| scopes\_supported | array | A JSON array containing a list of the [OAuth 2.0 (Hammer-Lahav, E., Ed., Recordon, D., and D. Hardt, “OAuth 2.0 Authorization Protocol,” July 2011.)](#OAuth2.0) [OAuth2.0] scopes that this server supports. The server MUST support the openid scope. |
| flows\_supported | array | A JSON array containing a list of the OAuth 2.0 flows that this server supports. The server MUST support the code flow. |
| iso29115\_supported | array | A JSON array containing a list of the ISO 29115 assurance contexts that this server supports. |
| identifiers\_supported | array | A JSON array containing a list of the user identifier types that this server supports |

|  |
| --- |
| **Table 1: Reserved Claim Definitions** |

Example response

{

"authorization\_endpoint": "https://example.com/connect/authorize",

"issuer" : "https://example.com",

"token\_endpoint": "https://example.com/connect/token"

"user\_info\_endpoint": "https://example.com/connect/user",

"check\_id\_endpoint": "https://example.com/connect/check\_id",

"refresh\_session\_endpoint": "https://example.com/connect/refresh\_session",

"end\_session\_endpoint": "https://example.com/connect/end\_session",

"jwk\_document": "https://example.com/jwk.json",

"registration\_endpoint": "https://example.com/connect/register",

"scopes\_supported": ["openid"],

"flows\_supported": ["code", "token"],

"iso29115\_supported": ["http://www.idmanagement.gov/schema/2009/05/icam/openid-trust-level1.pdf"],

"identifiers\_supported": ["public", "ppid"]

}

|  |
| --- |
| [**TOC**](#toc) |

### 4.3.  Provider Configuration Verification

If the configuration response contains the issuer element, the value MUST exactly match the issuer URL that was used to retrieve it.

|  |
| --- |
| [**TOC**](#toc) |

### 5.  IANA Considerations

This document makes no request of IANA.

|  |
| --- |
| [**TOC**](#toc) |

### 6.  Security Considerations

TBD

|  |
| --- |
| [**TOC**](#toc) |

### 7.  Open Issues and Things To Be Done (TBD)

[[ To be removed from the final specification ]]

Following items remain to be done in this draft:

* Should issuer be in the Provider Configuration Response?
* Should issuer be explicitly restricted to the https:// scheme?

|  |
| --- |
| [**TOC**](#toc) |

### 8.  References

|  |
| --- |
| [**TOC**](#toc) |

### 8.1. Normative References

|  |  |
| --- | --- |
| **[JWK]** | Jones, M., “[JSON Web Key (JWK)](http://self-issued.info/docs/draft-jones-json-web-key.html),” July 2011. |
| **[OpenID.Messages]** | Sakimura, N., Recordon, D., Bradley, J., de Medeiros, B., Jones, M., and E. Jay, “[OpenID Connect Messages 1.0](http://openid.net/specs/openid-connect-messages-1_0.html),” September 2011. |
| **[OpenID.Registration]** | Sakimura, N., Bradley, J., Ed., and M. Jones, “[OpenID Connect Dynamic Client Registration 1.0](http://openid.net/specs/openid-connect-registration-1_0.html),” September 2011. |
| **[OpenID.Session]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “[OpenID Connect Session Management 1.0](http://openid.net/specs/openid-connect-session-1_0.html),” September 2011. |
| **[RFC2119]** | [Bradner, S.](mailto:sob@harvard.edu), “[Key words for use in RFCs to Indicate Requirement Levels](http://tools.ietf.org/html/rfc2119),” BCP 14, RFC 2119, March 1997 ([TXT](http://www.rfc-editor.org/rfc/rfc2119.txt), [HTML](http://xml.resource.org/public/rfc/html/rfc2119.html), [XML](http://xml.resource.org/public/rfc/xml/rfc2119.xml)). |
| **[RFC3986]** | [Berners-Lee, T.](mailto:timbl@w3.org), [Fielding, R.](mailto:fielding@gbiv.com), and [L. Masinter](mailto:LMM@acm.org), “[Uniform Resource Identifier (URI): Generic Syntax](http://tools.ietf.org/html/rfc3986),” STD 66, RFC 3986, January 2005 ([TXT](http://www.rfc-editor.org/rfc/rfc3986.txt), [HTML](http://xml.resource.org/public/rfc/html/rfc3986.html), [XML](http://xml.resource.org/public/rfc/xml/rfc3986.xml)). |
| **[RFC5246]** | Dierks, T. and E. Rescorla, “[The Transport Layer Security (TLS) Protocol Version 1.2](http://tools.ietf.org/html/rfc5246),” RFC 5246, August 2008 ([TXT](http://www.rfc-editor.org/rfc/rfc5246.txt)). |
| **[RFC5785]** | Nottingham, M. and E. Hammer-Lahav, “[Defining Well-Known Uniform Resource Identifiers (URIs)](http://tools.ietf.org/html/rfc5785),” RFC 5785, April 2010 ([TXT](http://www.rfc-editor.org/rfc/rfc5785.txt)). |
| **[SWD]** | Jones, M., Ed. and Y. Goland, “[Simple Web Discovery](http://self-issued.info/docs/draft-jones-simple-web-discovery.html),” July 2011. |

|  |
| --- |
| [**TOC**](#toc) |

### 8.2. Informative References

|  |  |
| --- | --- |
| **[OAuth2.0]** | Hammer-Lahav, E., Ed., Recordon, D., and D. Hardt, “[OAuth 2.0 Authorization Protocol](http://tools.ietf.org/html/draft-ietf-oauth-v2),” July 2011. |
| **[XRI\_Syntax\_2.0]** | Reed, D. and D. McAlpin, “Extensible Resource Identifier (XRI) Syntax V2.0,” November 2005 ([HTML](http://www.oasis-open.org/committees/download.php/15376/xri-syntax-V2.0-cs.html), [PDF](http://www.oasis-open.org/committees/download.php/15377/xri-syntax-V2.0-cs.pdf)). |

|  |
| --- |
| [**TOC**](#toc) |

### Appendix A.  Acknowledgements

|  |
| --- |
| [**TOC**](#toc) |

### Appendix B.  Document History

[[ To be removed from the final specification ]]

-06

* Changed Check Session Endpoint to Check ID Endpoint to match Basic.
* Changed certs\_url to x509\_url to match registration and JWE format.

-05

* Ticket #46 Added text to 3.3
* Deleted duplicate check session endpoint from 4.2
* Ticket #40 Added clarification of issuer url to 4.2
* Ticket #39 Cleaned up issuer examples and added verification text.

-04

* Changes associated with renaming "Lite" to "Basic Client" and replacing "Core" and "Framework" with "Messages" and "Standard".
* Numerous cleanups, including updating references.

-03

* Corrected examples.

-02

* Correct issues raised by Johnny Bufu and discussed on the 7-Jul-11 working group call.

-01

* Incorporate working group decisions from 5-Jul-11 spec call.
* Consistency and cleanup pass, including removing unused references.

-00

* Initial version based upon former openid-connect-swd-1\_0 spec.

|  |
| --- |
| [**TOC**](#toc) |

### Authors' Addresses

|  |  |
| --- | --- |
|  | Nat Sakimura |
|  | Nomura Research Institute, Ltd. |
| **Email:** | [n-sakimura@nri.co.jp](mailto:n-sakimura@nri.co.jp) |
|  |  |
|  | John Bradley (editor) |
|  | Independent |
| **Email:** | [ve7jtb@ve7jtb.com](mailto:ve7jtb@ve7jtb.com) |
|  |  |
|  | Michael B. Jones |
|  | Microsoft Corporation |
| **Email:** | [mbj@microsoft.com](mailto:mbj@microsoft.com) |
|  |  |
|  | Edmund Jay |
|  | MGI1 |
| **Email:** | [ejay@mgi1.com](mailto:ejay@mgi1.com) |