Doc V1.0

Steps to use the App Insights Event Logger.

It can be used in production.

This walk-through assumes that you are using Custom Policies with the Starter Pack and your signup and signin journeys are working.

An XML file with all the claims and technical profiles is available.  Copy and paste the elements as identified below in your own version of TrustFrameworkExtensions. If not available, all the needed XML content is described here:

<<yourtenant.onmicrosoft.com-B2C\_1A\_TrustFrameworkExtensionsAppIns SAMPLE no valid keys TESTED.xml>>

1. Create an Account in Application Insights and obtain the Instrumentation Key

Machine generated alternative text:
WingTipB2C 
Application Ins$hts - Last 24 (30 mirwte - ASPNET 
Search Metrics Explorer 
Analytics O Time range Refresh 
Please try some of the new experiences we have in store for you, such as the new 
Composite Application Map, before they become the default experience. 
Overflew 
Activity log 
Access control (IAM) 
Tags 
Diagnose and solve problems 
I WESTIGATE 
group 
W,ngTipB2C 
East US 
Subscriptio 
CPIM Demo Environment 
Subscription ID 
b7f58ee2-Of20-4cb5-g44b-4921a3e 
ASP.NET 
Inst um elation Key 
gac4abef-85g8-4211-bg57-bOdOd1d 

**STEP 2 Open up the extensions file. Usually named yourtenant-B2C\_1A\_TrustFrameworkExtensions.xml.**

Inside the element <BuildingBlocks> add the following ClaimType definitions:

<BuildingBlocks>

<ClaimsSchema>

<ClaimType Id="EventType">

<DisplayName>EventType</DisplayName>

<DataType>string</DataType>

<AdminHelpText />

<UserHelpText />

</ClaimType>

<ClaimType Id="PolicyId">

<DisplayName>PolicyId</DisplayName>

<DataType>string</DataType>

<AdminHelpText />

<UserHelpText />

</ClaimType>

<ClaimType Id="Culture">

<DisplayName>Culture</DisplayName>

<DataType>string</DataType>

<AdminHelpText />

<UserHelpText />

</ClaimType>

<ClaimType Id="CorrelationId">

<DisplayName>CorrelationId</DisplayName>

<DataType>string</DataType>

<AdminHelpText />

<UserHelpText />

</ClaimType>

<!--Additional claims used for passing claims to Application Insights Provider -->

<ClaimType Id="federatedUser">

<DisplayName>federatedUser</DisplayName>

<DataType>boolean</DataType>

<UserHelpText />

</ClaimType>

<ClaimType Id="parsedDomain">

<DisplayName>Parsed Domain</DisplayName>

<DataType>string</DataType>

<UserHelpText>The domain portion of the email address.</UserHelpText>

</ClaimType>

<ClaimType Id="userInLocalDirectory">

<DisplayName>userInLocalDirectory</DisplayName>

<DataType>boolean</DataType>

<UserHelpText />

</ClaimType>

</ClaimsSchema>

**STEP 3 Add the Technical profiles you will need for the Application Insights Provider** to  the extensions file. Usually named **yourtenant-B2C\_1A\_TrustFrameworkExtensions.xml.**

**The technical profiles are:**

**"JourneyContextForInsights" - This one opens up the event in App Insights and sends a correlation Id**

**"AzureInsights-SignInRequest" - To create and Event and send a set of claims when a signin request has been received**

**"AzureInsights-Common" - A common set of parameters to be included in  all Azure-Insights Technical Profiles**

**"AzureInsights-UserSignup" - To create an Event called "UserSignup" when the signup option has been triggered by the user in a signup/signin journey**

**"AzureInsights-SignInComplete" - To record the successful completion of an authentication - a token has been sent to the relying party application.**

**Inside the element <ClaimsProviders> add:**

<ClaimsProvider>

<DisplayName>Application Insights</DisplayName>

<TechnicalProfiles>

<TechnicalProfile Id="JourneyContextForInsights">

<DisplayName>Application Insights</DisplayName>

<Protocol Name="Proprietary" Handler="Web.TPEngine.Providers.UserJourneyContextProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />

<OutputClaims>

<OutputClaim ClaimTypeReferenceId="CorrelationId" />

</OutputClaims>

</TechnicalProfile>

<TechnicalProfile Id="AzureInsights-SignInRequest">

<InputClaims>

<!--

              An input claim with a PartnerClaimType="eventName" is required. This is used by the AzureApplicationInsightsProvider

              to create an event with the specified value.

            -->

<InputClaim ClaimTypeReferenceId="EventType" PartnerClaimType="eventName" DefaultValue="SignInRequest" />

</InputClaims>

<IncludeTechnicalProfile ReferenceId="AzureInsights-Common" />

</TechnicalProfile>

<TechnicalProfile Id="AzureInsights-SignInComplete">

<InputClaims>

<InputClaim ClaimTypeReferenceId="EventType" PartnerClaimType="eventName" DefaultValue="SignInComplete" />

<InputClaim ClaimTypeReferenceId="federatedUser" PartnerClaimType="{property:FederatedUser}" DefaultValue="false" />

<InputClaim ClaimTypeReferenceId="parsedDomain" PartnerClaimType="{property:FederationPartner}" DefaultValue="Not Applicable" />

</InputClaims>

<IncludeTechnicalProfile ReferenceId="AzureInsights-Common" />

</TechnicalProfile>

<TechnicalProfile Id="AzureInsights-UserSignup">

<InputClaims>

<InputClaim ClaimTypeReferenceId="EventType" PartnerClaimType="eventName" DefaultValue="UserSignup" />

</InputClaims>

<IncludeTechnicalProfile ReferenceId="AzureInsights-Common" />

</TechnicalProfile>

<TechnicalProfile Id="AzureInsights-Common">

<DisplayName>Alternate Email</DisplayName>

<Protocol Name="Proprietary" Handler="Web.TPEngine.Providers.Insights.AzureApplicationInsightsProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />

<Metadata>

<!-- The ApplicationInsights instrumentation key which will be used for logging the events -->

<Item Key="InstrumentationKey">94c4abef-8598-4211-b957-b0d0d1d202d1</Item>

<!--

              A boolean indicating whether delevoper mode is enabled. This controls how events are buffered. In a development environment

              with minimal event volume, enabling developer mode results in events being sent immediately to ApplicationInsights.

            -->

<Item Key="DeveloperMode">false</Item>

<!--

              A boolean indicating whether telemtry should be enabled or not.

            -->

<Item Key="DisableTelemetry ">false</Item>

</Metadata>

<InputClaims>

<!--

              Properties of an event are added using the syntax {property:NAME} where NAME is the name of the property being added

              to the event. DefaultValue can be either a static value or one resolved by one of the supported DefaultClaimResolvers.

            -->

<InputClaim ClaimTypeReferenceId="PolicyId" PartnerClaimType="{property:Policy}" DefaultValue="{Policy:PolicyId}" />

<InputClaim ClaimTypeReferenceId="CorrelationId" PartnerClaimType="{property:JourneyId}" />

<InputClaim ClaimTypeReferenceId="Culture" PartnerClaimType="{property:Culture}" DefaultValue="{Culture:RFC5646}" />

</InputClaims>

</TechnicalProfile>

</TechnicalProfiles>

</ClaimsProvider>

**STEP 4. Lastly, Add the Technical Profiles for Application-Insights as increment orchestration steps in an existing UserJourney.**

1. Invoke the Technical Profile "JourneyContextForInsights" as orchestration step 1

<OrchestrationStep Order="1" Type="ClaimsExchange">

<ClaimsExchanges>

<ClaimsExchange Id="JourneyContextForInsights" TechnicalProfileReferenceId="JourneyContextForInsights" />

</ClaimsExchanges>

</OrchestrationStep>

1. Invoke the Technical Profile "AzureInsights-SignInRequest" as orchestration step 2

<!-- Track that we have received a sign in request -->

<OrchestrationStep Order="2" Type="ClaimsExchange">

<ClaimsExchanges>

<ClaimsExchange Id="TrackSignInRequest" TechnicalProfileReferenceId="AzureInsights-SignInRequest" />

</ClaimsExchanges>

</OrchestrationStep>

1. Immediately ***before*** the "SendClaims" orchestration step add a step to track if the sign up button has been clicked (in a signup /signin journey). This step uses Technical Profile named "AzureInsights-UserSignup"

<!-- This step is here to handle the user clicking the sign up link in the local account sign in page -->

<OrchestrationStep Order="9" Type="ClaimsExchange">

<Preconditions>

<Precondition Type="ClaimsExist" ExecuteActionsIf="false">

<Value>newUser</Value>

<Action>SkipThisOrchestrationStep</Action>

</Precondition>

<Precondition Type="ClaimEquals" ExecuteActionsIf="true">

<Value>newUser</Value>

<Value>false</Value>

<Action>SkipThisOrchestrationStep</Action>

</Precondition>

</Preconditions>

<ClaimsExchanges>

<ClaimsExchange Id="TrackUserSignUp" TechnicalProfileReferenceId="AzureInsights-UserSignup" />

</ClaimsExchanges>

</OrchestrationStep>

1. And add a new last step, immediately after the "SendClaims" orchestration step. This will be your new last step in the userjourney

<!-- Track that we have successfully sent a token -->

<OrchestrationStep Order="11" Type="ClaimsExchange">

<ClaimsExchanges>

<ClaimsExchange Id="TrackSignInComplete" TechnicalProfileReferenceId="AzureInsights-SignInComplete" />

</ClaimsExchanges>

</OrchestrationStep>

**NOTE That after the insertions above, all orchestration steps must be renumbered sequentially without skipping any integers from 1 to N**

**STEP 5. Upload the new version of TrustFrameworkExtensions.**  You may want to save the previously tested working version or up load this one with a new policyid name to preserve them both.

Now Invoke the SignUp Signin Journey with the new orchestration steps.  You may use the "RUN NOW" option

**Events post to Application Insights within seconds, usually less than 1 minute.**

**STEP 6. Open up Application Insights in your Azure AAD tenant, Azure Portal**

**Look for the Events under the Main menu for Application Insights, Under USAGE and select "Events"**

**You may want to change the resolution from last 24 hours to  last hour for greater resolution.**

**Additional Claims Resolvers**

**Culture Specific Claims**

            Referenced using {Culture:*One of the property names below}*

/// <summary>

        /// An enumeration of the claims supported by the <see cref="JourneyCultureDefaultClaimProcessor"/>

        /// </summary>

        public enum SupportedClaim

        {

            /// <summary>

            /// The two letter ISO code for the language i.e. en

            /// </summary>

            LanguageName,

            /// <summary>

            /// The two letter ISO code for the region i.e. US

            /// </summary>

            RegionName,

            /// <summary>

            /// The RFC5646 language code i.e. en-US

            /// </summary>

            RFC5646,

            /// <summary>

            /// The LCID of language code i.e. 1033

            /// </summary>

            LCID

        }

**Policy Specific Claims**

Referenced using {Policy:*One of the property names below}*

/// <summary>

        /// An enumeration of the claims supported by the <see cref="PolicyDefaultClaimProcessor"/>

        /// </summary>

        public enum SupportedClaim

        {

            /// <summary>

            /// The trustframework tenant id

            /// </summary>

            TrustFrameworkTenantId,

            /// <summary>

            /// The tenant id of the relying party

            /// </summary>

            RelyingPartyTenantId,

            /// <summary>

            /// The policy id of the policy

            /// </summary>

            PolicyId,

            /// <summary>

            /// The tenant object id of the policy

            /// </summary>

            TenantObjectId

}

**OpenIDConnect Specific Claims**

Referenced using {OIDC:*One of the property names below}*

/// <summary>

        /// An enumeration of the claims supported by the <see cref="OpenIdConnectDefaultClaimProcessor"/>

        /// </summary>

        public enum SupportedClaim

        {

            /// <summary>

            /// The OpenIdConnect prompt parameter

            /// </summary>

            Prompt,

            /// <summary>

            /// The OpenIdConnect login\_hint parameter

            /// </summary>

            LoginHint,

            /// <summary>

            /// The OpenIdConnect domain\_hint parameter

            /// </summary>

            DomainHint,

            /// <summary>

            /// The OpenIdConnect max\_age parameter

            /// </summary>

            MaxAge,

            /// <summary>

            /// The OpenIdConnect client\_id parameter

            /// </summary>

            ClientId,

            /// <summary>

            /// The OpenIdConnect username parameter

            /// </summary>

            Username,

            /// <summary>

            /// The OpenIdConnect password parameter

            /// </summary>

            Password,

            /// <summary>

            /// The OpenIdConnect resource type parameter

            /// </summary>

            Resource,

            /// <summary>

            /// The OpendIdConext acr\_values parameter

            /// </summary>

            AuthenticationContextReferences

        }

**Non-protocol parameters included with OIDC & OAuth2 requests**

Referenced using { OAUTH-KV:*Querystring parameter name }*

We also have a provider that can inject claims into the claim bag when executed as part of a claims exchange. This provider is referenced like this. When I have a free moment, I turning the UJC provider into a IDefaultClaimResolver which would provide a consistent way of referencing these types of claims.

*<ClaimsProvider>*

*<DisplayName>User Journey Context Provider</DisplayName>*

*<TechnicalProfiles>*

*<TechnicalProfile Id="SimpleUJContext">*

*<DisplayName>User Journey Context Provide</DisplayName>*

*<Protocol Name="Proprietary" Handler="Web.TPEngine.Providers.UserJourneyContextProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />*

*<OutputClaims>*

*<OutputClaim ClaimTypeReferenceId="****IP-Address****" />*

*<OutputClaim ClaimTypeReferenceId="****CorrelationId****" />*

*<OutputClaim ClaimTypeReferenceId="****DateTimeInUtc****" />*

*<OutputClaim ClaimTypeReferenceId="****Build****" />*

*</OutputClaims>*

*</TechnicalProfile>*

*</TechnicalProfiles>*

*</ClaimsProvider>*

Machine generated alternative text:
All 
Events 
> WngTlp82C - 
+ New 
Open Save Save As CD Refresh Share pin 
Show occurrences for All users 
Add filters 
20 events 
EVENT STATISTICS 
NAME 
Overall 
SignInRequest 
userSignup 
SignInComplete 
O Who used Any Event 
Feedback 
O During 
05:18 
Last hour 
3 minutes 
UNIQUE USERS 
O Split by 
UNIQUE SESSIONS 
1 (Icosa 
1 (Icosa 
1 (Icosa 
Bar chart v 
000 
Flter by name„_ 
COUNT 
20 
11 
Last hour 
USERS 
SESSIONS Machine generated alternative text:
Show occurrences for All users 
Split by FederatedlJser 
Add filters 
13 events 
Who used 
Any Event 
UNIQUE USERS 
O During 
Last hour 
3 minutes 
Bar chart v 
Cd:18 
Flter by name„ 
COUNT 
13 
Last hour 
USERS 
SESSIONS 
8 
EVENT STATISTICS 
NAME 
Overall 
UserSignup 
SignlnRequest 
SignlnComplete 
5 
UNIQUE SESSIONS 
1 (100%) 
1 (100%) 
1 (100%) 