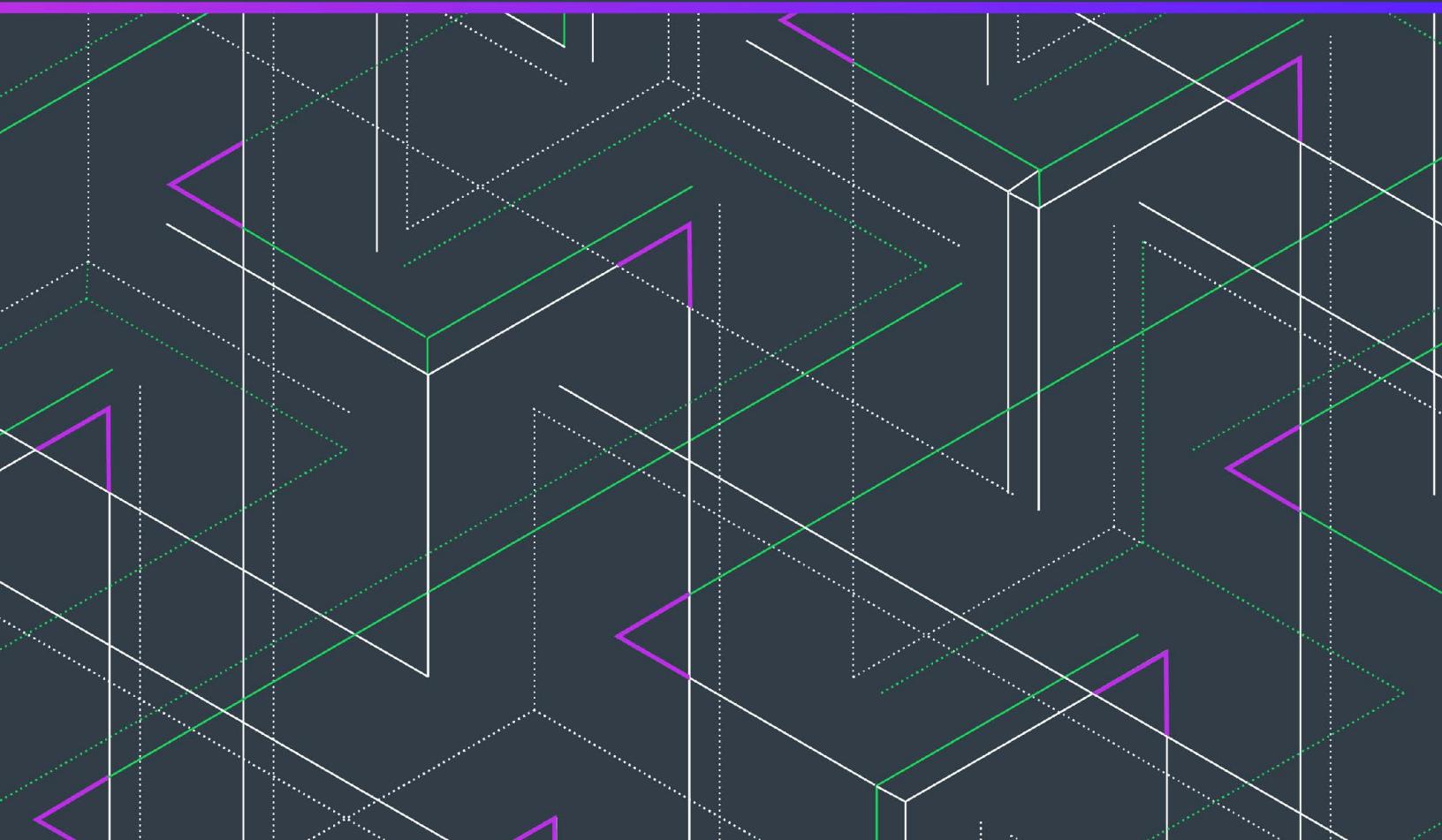


Usage Intelligence

Integration Best Practices Overview v1.6



Legal Information

Book Name: Usage Intelligence Integration Best Practices Overview v1.6

Part Number: RUI-0160-BP02

Product Release Date: September 2024

Copyright Notice

Copyright © 2025 Flexera Software

This publication contains proprietary and confidential information and creative works owned by Flexera Software and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software is strictly prohibited. Except where expressly provided by Flexera Software in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, must display this notice of copyright and ownership in full.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see <https://www.revenera.com/legal/intellectual-property.html>. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

Usage Intelligence Integration Best Practices Overview v1.6.....	1
Development Resources.....	2
SDK Configuration.....	3
Basic Configuration & Ordering.....	3
Important Note Regarding Session Duration.....	4
Configuration File	4
Creating the Config File.....	4
Tracking Multiple User Profiles Per Installation.....	5
Managing Config Files During Software Version Updates.....	5
Server-Side Tracking & Multi-Session.....	6
Sync	6
Offline Tracking	6
Opt-In/Opt-Out.....	7
Event Tracking & Custom Properties	7
Events vs. Properties.....	8
Event Tracking Overview	8
Deciding Which Events to Track	9
Naming Convention	9
Event Reports.....	10
Event Tracking API Calls	10
Custom Properties.....	11
ReachOut	11
License Key Tracking & Management.....	11
Integration Checklist	12
Legal Information	14

Usage Intelligence Integration Best Practices Overview v1.6

This is an introductory guide for integrating the Usage Intelligence SDK into applications. The target audience is software developers charged with performing the SDK integration or product managers overseeing the integration.

- [Development Resources](#)
- [SDK Configuration](#)
- [Configuration File](#)
- [Event Tracking & Custom Properties](#)
- [Integration Checklist](#)



Note • This guide is not comprehensive API documentation. For information on the Usage Intelligence Reporting API, see [Usage Intelligence Reporting API \(3.0.0\)](#).

Development Resources

Basic implementation of the SDK requires approximately 30 minutes of integration with approximately 10 lines of code. This basic implementation includes only a list of out-of-the-box metrics. For more comprehensive integration and use of advanced SDK features, Revenera recommends reserving 2–3 developer days for integration.

This target will vary depending on the level of functionality desired (including number of tracked features) and the organization's build, testing, and QA process.

Additional Resources

For additional information, the following resources are available:

- **SDK download links**—To download Usage Intelligence SDKs, see the [Usage Intelligence SDK Download Links](#) page on the Revenera Community (Revenera Community login required).
- **Usage Intelligence product documentation** —To view Usage Intelligence documentation, including the SDK Developer Guides, visit the Revenera Documentation site:
<https://docs.revenera.com>
- **Usage Intelligence Reporting API (3.0.0)**—To view Usage Intelligence Reporting API, see:
<https://rui-api.redoc.ly>
- **Revenera Community**—For answers to commonly asked questions, see the [Usage Intelligence](#) section of the Revenera Community, which includes:
 - Usage Intelligence Knowledge Base
 - Usage Intelligence News
 - Usage Intelligence Forum

Product Registration

Before using the Usage Intelligence software analytics service or integrating the Usage Intelligence SDK, you must first register an account by visiting the [Leverage Real Product Usage Data with a 30-Day Free Trial](#) page of the Revenera website.

After registering a username and creating a new product account for tracking your application, the Product ID, CallHome URL, and AES Key information is accessible from the [Administration > Developer Zone](#) page in the Usage Intelligence application. This page also provides links to the [API Documentation / SDK Download](#) page, [Best Practices Guide](#), and [Support Center](#).

SDK Configuration

Information about SDK configuration is described in the following sections:

- [Basic Configuration & Ordering](#)
- [Important Note Regarding Session Duration](#)

Basic Configuration & Ordering

When integrating the Usage Intelligence SDK, some API calls are necessary to initialize and configure it. Specific ordering of these API calls is required for the SDK to successfully initialize. The following is the order of API operations:

Table 1 • Basic Configuration & Ordering

#	Step	Description
1	Initialize SDK object	Create an instance of the SDK.
2	CreateConfig	Creates a configuration for the SDK instance. A configuration is passed on a file, specified by configFilePath, productID, and appName.
3	SetProductData	Sets or clears the product data.  Note • The product data must be set every time the SDK instance is run.
4	Advanced Integration	(Optional)
5	Custom Properties	Used to collect custom installation metrics.  Note • For more information, see Custom Properties .
6	License properties	Used to collect and/or manage licensing data.  Note • For more information, see License Key Tracking & Management .
7	OptOut	This is an API call that is necessary for any machine opting-out of data collection.  Note • For more information, see Opt-In/Opt-Out .

Table 1 • Basic Configuration & Ordering

#	Step	Description
8	StartSDK / StopSDK	<p>Starts the SDK. StartSDK must be paired with a call to StopSDK.</p> <p>Additionally, when the application is exiting, a StopSDK API call must be made. StopSDK stops the SDK that was started with StartSDK. This function should always be called at the exit point of your application</p>

Important Note Regarding Session Duration

When an application runs for a very short time, the SDK will not have enough time to sync with the server and create a configuration file.

For automated QA processes, we recommend adding a 10 second “sleep” function between the StartSDK and StopSDK calls. This delay is necessary because the SDK needs time to sync with the server. However this delay is not required for normal applications with a runtime duration of more than 10 seconds in everyday use.

Configuration File

This section explains how to create and use the Usage Intelligence configuration file.

- [Creating the Config File](#)
- [Tracking Multiple User Profiles Per Installation](#)
- [Managing Config Files During Software Version Updates](#)
- [Server-Side Tracking & Multi-Session](#)
- [Sync](#)
- [Offline Tracking](#)
- [Opt-In/Opt-Out](#)

Creating the Config File

When your software is first run, the Usage Intelligence SDK creates a `ruiconfig*.cfg` file on disk.

- The location of this file is set through the `CreateConfig` function.
- The application must have write permissions to the folder specified. Normally this path is set to the same location where your application stores data/config settings.



Note • On some machine configurations (especially machines running Microsoft Windows Vista or higher with UAC enabled), if the user does not have administrator privileges, the application will be denied access to the `Program Files` directory. Therefore, Revenera recommends avoiding the default path if your application runs from the `Program Files` directory.



Important - The config file should only be used by the Usage Intelligence SDK. Your application should not modify or rely on the contents of this file.

Tracking Multiple User Profiles Per Installation

If the application supports multiple user profiles and each user needs to be tracked separately, then a unique folder must be created for every user when calling the CreateConfig method.

Using independent folders allows the SDK to generate a unique fingerprint for every user and track them as independent installations.

- This can be a folder within the user's OS profile or a folder created manually within the application data directory.
- The folder name could include a unique username/ID/hash identifying that user.

Managing Config Files During Software Version Updates

When a user upgrades product version, we recommend using the same configuration file across multiple product versions. This allows the SDK to track an installation across product versions. This can be handled in one of two ways:

- Using the application data path that will not change during the upgrade process (such as ProgramData, AppData, etc.)
- Having the application upgrade process handle moving the configuration file.



Important - In the case of an application allowing for parallel installs (such as multiple versions installed simultaneously), never share the same configuration file between two installations. Usage Intelligence does not support tracking multiple installations using a single configuration file because each configuration should correspond to its own installation.



Important - The configuration file must not be copied between installations and users, either on the same or different computers. This causes the client fingerprint verification to fail and the server will not track usage statistics for that client. Always allow the SDK API to create the config and log files for new installations.

Server-Side Tracking & Multi-Session

Most applications do not require multi-session tracking. Multi-session tracking is used when a server-side application runtime may host multiple user sessions.

- For more information, see [What Files Are Created by the Usage Intelligence SDK and What Is the Folder Location?](#)
- For the use case of tracking multiple users for a server-side application, contact [Revenera Support](#).

Sync

The Usage Intelligence SDK syncs with the server when the calls to StartSDK and StopSDK are made. Sync also happens automatically every 20 minutes while the application is running.

The SDK also supports manual sync. Manual sync is a complete sync of user data with the server.

- Manual sync should be used when ReachOutOnAutoSync is set to false.
- For this use case, the sync call is used to retrieve ReachOut messages that can be displayed at specific points in the user's workflow.



Note • Manual sync has a 5-minute threshold. If a manual sync is initiated too quickly after another sync has occurred the sync will not be done.

Offline Tracking

Usage Intelligence caches tracking data inside a log file named ruilog.log. This file is created after an initial successful sync with the server. After this file has been created, when an installation cannot connect to the server for a number of days or runtime sessions, all cached data is retained in this log file until the next successful sync.



Note • Offline caching requires an initial successful sync. For machines that are permanently offline, no log file will be created and no offline caching will occur.

The SDK creates a new log file for each application runtime session.

- Each runtime log file has a default maximum size of 1MB.
- The total combined size of all log files is also limited to 1MB by default.
- The maximum log file size limitations can be changed by contacting [Revenera Support](#).

For more information on the list of configuration and log files maintained by the Usage Intelligence SDK, see [What Files Are Created by the Usage Intelligence SDK and What Is the Folder Location?](#).

Opt-In/Opt-Out

Use the Opt-In/Opt-Out mechanism if a user does not want to send tracking information to Revenera.

- OptOut must be called after calling CreateConfig and before RUISDK.StartSDK.
- When OptOut is called, the SDK sends a message informing the server that this client has opted-out.
- The opt-out flag on the server will be used for reporting opt-out statistics only.

If OptOut is called before a new registration, the server will never have any data about that installation. If OptOut is called for a previously tracked installation, the server retains the data that had been collected in the past.

The SDK will send no further information to the server while the user is opted-out. The application must call OptOut before every startup as long as the user wants to stay opted-out. If this function is not called, then the SDK assumes that the user is opting-in again and will start tracking normally.

Event Tracking & Custom Properties

You can choose to track custom events. You can also track extra custom properties that are related to your software or the environment that it runs on.

- [Events vs. Properties](#)
- [Event Tracking Overview](#)
- [Deciding Which Events to Track](#)
- [Naming Convention](#)
- [Event Reports](#)
- [Event Tracking API Calls](#)
- [Custom Properties](#)
- [ReachOut](#)
- [License Key Tracking & Management](#)

Events vs. Properties

For a developer integrating the Usage Intelligence SDK, it is beneficial to conceptually understand the difference between an Event and a Property. Some aspects of a product should be tracked as events and other aspects as properties.

- **Events**—Events are actions that occur at specific points in time, such as *user clicking on a button*.
- **Properties**—Properties are attributes of a product or application, such as *product language*.

In the object-oriented programming paradigm, an *event* is analogous to a *method* and a *property* is analogous to a *field*.

Events

Usage Intelligence gives users the ability to define and track events as they are triggered. Examples of events are:

- User interacting with UI elements such as:
 - Button clicks
 - Menu launch
- Feature and function usage
- Application state-change

Properties

While developers can define their own custom properties, the SDK automatically collects a list of properties by default. Examples of properties collected by default include:

- Screen resolution
- RAM
- Continent/country/U.S. state
- Lifetime number of sessions
- Operating system

Event Tracking Overview

The terms **Feature** and **Event** are used interchangeably in the scope of Usage Intelligence. In the Dashboard, event-related reports are displayed in the section titled **Feature & Event Tracking**.

Deciding Which Events to Track

When integrating the SDK, some users may attempt to track every function and action that exists in their application.

Revenera strongly recommends researching and identifying your *most important* software features. While Usage Intelligence is capable of tagging thousands of unique features, tagging too many complicates the process of extracting information from dashboard reports.

Upon initial integration, we recommend tracking a conservative number of features. This is typically less than 300 unique features. This will help to lessen the development load while still providing valuable insights.

In the Dashboard, users have the capability to enable/disable tracking for particular features. This allows the dashboard user flexibility when selecting which features are tracked for a specific period of time.

Naming Convention

In Usage Intelligence, events are defined by a **Category** and **Name**. Events can be segmented by category in the dashboard for reporting.

Revenera recommends implementing an appropriate naming convention for your tracked Events. Examples of naming conventions are:

Table 2 • Naming Convention

Category	Name
File Operation	Open
File Operation	Save
File Operation	Print
Install Wizard	Step 1
Install Wizard	Step 2

Two common naming schemes use the following conventions:

category=<function>; name=<interface component>;

OR

category=<interface component>; name=<function>;

If automated or unexposed processes require tracking, we recommend categorizing these processes together. An example is a button-click UI element which triggers three additional automated processes.

- The button-click operation could be tracked on its own.
- If the three automated processes require tracking, these processes should be categorized together.

Event Reports

In the Usage Intelligence Dashboard, many reports are generated using event counts. The following reports can be configured to display event count data for any tracked event. No matter what API call is used for event tracking, the data will be displayed in the following reports:

- **Event Usage Timeline**—This report shows aggregate usage statistics for all tracked events, highlighting usage trends over time.
- **Lifetime Event Usage**—This report shows lifetime usage statistics for all tracked events. Lifetime usage statistics are calculated for each client by counting the amount of times an event was fired during their tracked lifetime (such as starting from the first install until the last seen date). This enables you to determine how many times users fired an event throughout their lifetime.

Any tracked events can be used as filter criteria in most dashboard reports and also for targeting in-app messaging campaigns via ReachOut.

Event Tracking API Calls

TrackEvent is the standard event tracking API call and is used to record the fact that something occurred (such as an action). This is for any event occurring without additional custom data collection.

When an event occurs, you can additionally collect some form of custom data including numeric/text/name-value pairs. For this, use the following API calls (in place of TrackEvent):

Table 3 • Event Tracking API Calls

API Call	Description
TrackEventNumeric	<p>Logs an event and adds a custom numeric value.</p> <p>This numeric value is used to compute the aggregate and average Event Custom Value table in the Event Usage Timeline report.</p>
TrackEventText	<p>Logs an event and adds a custom string (with a maximum size limit of 4096 characters).</p>
TrackEventCustom	<p>Logs an event and adds a list of name-value pairs (with a maximum size limit of 128 characters per value).</p>

The data collected with TrackEventText and TrackEventCustom is available in the Custom Events dashboard report. This report contains a table view of the most recent custom events and allows a dashboard user to download a daily/monthly archive of custom event data.

TrackEventCustom can be used to view custom numeric event reports. For more information, see [Custom Numeric Event Tracking Reporting](#) in the Revenera Community.

For additional information on the event tracking API calls, see [Feature / Event Tracking](#) in the *Usage Intelligence 5.6.1 C++ for Windows SDK Developer Guide*.

Custom Properties

Usage Intelligence offers the capability to track additional custom properties or attributes that are related to your software or its environment.

- For information on supported types of custom properties, see [What Type of Data Can I Store In Custom Properties?](#)
- For information on adding Custom Properties to your account, contact [Revenera Support](#).

ReachOut

ReachOut in-application messaging is a powerful marketing tool that allows you to interact with your users while they are engaged with your application. By using ReachOut you can take advantage of the analytics data that you collect from your clients to create personalized campaigns and target your audience based on product, licensing, platform, and usage properties.

License Key Tracking & Management

For software having its own licensing mechanism, this data can be tracked through client-managed license tracking. `SetLicenseData` is the API call used to collect external license information.

Usage Intelligence gives you the ability to manage your license key status via the product dashboard. The application queries the server to determine the status of a particular license key. This is server-managed license tracking.

For more information, see [Usage Intelligence Reporting API \(3.0.0\)](#).

Integration Checklist

Before beginning integration steps, complete the items on this checklist to verify the application configuration and configuration directory settings

Table 4 • Integration Checklist

Step	Description
<input type="checkbox"/>	Verify correct order of API calls
	<p>Verify that you have configured the correct order of API calls for basic integration:</p> <ol style="list-style-type: none"> 1. Initialize SDK object 2. CreateConfig 3. SetProductData 4. Custom properties (<i>Optional</i>) 5. License properties (<i>Optional</i>) 6. OptOut (<i>Optional</i>) 7. StartSDK 8. StopSDK
<input type="checkbox"/>	Config path name
	<p>Config path name:</p> <hr/>
<input type="checkbox"/>	“Write” permissions to config directory
	<p>Verify that the application has write permissions to the config file directory.</p>
<input type="checkbox"/>	Application supports tracking multiple installs on the same physical machine
	<p>Verify that application supports tracking multiple installs on the same physical machine. (<i>Optional</i>)</p> <ul style="list-style-type: none"> • Each installation has its own config directory
<input type="checkbox"/>	For version upgrades, application uses the same config file
	<p>Verify that for version upgrades, application uses the same config file.</p> <ul style="list-style-type: none"> • Application handles moving the configuration file across product versions, <p><i>or</i></p> <ul style="list-style-type: none"> • Does your application use same config directory across versions?
<input type="checkbox"/>	Application allows for parallel installs (multiple versions simultaneously)
	<p>Verify that application allows for parallel installs (multiple versions simultaneously). (<i>Optional</i>)</p> <ul style="list-style-type: none"> • Parallel installs do not share the same configuration file.

Table 4 • Integration Checklist

Step	Description
<input type="checkbox"/> Application is server-side and supports multiple user sessions simultaneously <i>(Optional)</i>	Verify that application is server-side and supports multiple user sessions simultaneously. <ul style="list-style-type: none"> Multi-session mode is implemented
<input type="checkbox"/> Application has opt-out mechanism <i>(Optional)</i>	Verify that application has opt-out mechanism. <ul style="list-style-type: none"> Application calls OptOut before every startup as long as the user wants to stay opted-out.
<input type="checkbox"/> Approximate number of unique features tracked	Determine the approximate number of unique features tracked: <ul style="list-style-type: none"> Determine a standard naming convention which groups event names by product feature or UI location. Application logs a single event each time a tracked feature is used. Ensure the same event is not duplicated by different components.
<input type="checkbox"/> Application uses custom properties <i>(Optional)</i>	Verify that application uses custom properties.
<input type="checkbox"/> Application uses ReachOut <i>(Optional)</i>	Verify that application uses ReachOut.
<input type="checkbox"/> Application uses license tracking <i>(Optional)</i>	Verify that application uses license tracking. <ul style="list-style-type: none"> Application uses client-managed license tracking. <i>or</i> Application uses server-managed license tracking.

Legal Information

Copyright Notice

Copyright © 2025 Flexera Software

This publication contains proprietary and confidential information and creative works owned by Flexera Software and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software is strictly prohibited. Except where expressly provided by Flexera Software in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, must display this notice of copyright and ownership in full.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see <https://www.revenera.com/legal/intellectual-property.html>. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.