

Operational Enhancement Solutions

Xperience™

Release 2018



Hosting a Web App

04/11/2018

© 1999 – 2018 Jack Henry & Associates, Inc.

All rights reserved. Information in this document is subject to change without notice. Dates contained in this document are provided as estimates only and can be changed at any time at the sole discretion of Jack Henry & Associates, Inc.

Printed in the United States of America.

No part of this document may be copied, reproduced, stored in a retrieval system, displayed, distributed or transmitted in any form or any means whatsoever (electronic, mechanical or otherwise), including by photocopying or recording for any purpose, without the prior written permission of Jack Henry & Associates, Inc. Making unauthorized copies of this document for any purpose other than your own personal use is a violation of United States copyright laws.

Any unauthorized use of Jack Henry & Associates, Inc.'s trademarks and service marks is strictly prohibited. The following marks are registered and unregistered trademarks and service marks of Jack Henry & Associates, Inc.:

3rd Party Sweep™; 4|sight™; 4|sight Cash Letter™; 4|sight Exception Processing™; 4|sight PDF Statements™; AccountAssist™; Account Analysis™; Account Cross Sell Jumpstart™; ACH Client™; Advanced Reporting for Credit Unions™; AlertCenter™; AlertManager™; AllAccess™; ARCU Peer Data™; ARCU Third-Party Loan Integration™; ArgoKeys™; ArgoKeys Branch Sales Automation™; ArgoKeys DepositKeys™; ArgoKeys LendingKeys™; ArgoKeys RelationshipKeys™; AudioTel™; Banno Apps™; Banno CMS™; Banno Marketing™; Banno Marketing PS™; Banno Mobile™; Banno Mobile PS™; Banno Monitor™; Banno Online™; Banno People™; Banno People PS™; Banno Platform™; Banno Reports™; Banno Reports PS™; Banno Settings™; Banno Settings PS™; Banno Support™; Banno Support PS™; BladeCenter™; Branch Anywhere™; BusinessManager™; Card Holder Controls and Alerts™; Centurion Business Continuity Planning™; Centurion Business Recovery Consulting Group™; Centurion Co-Location™; Centurion Disaster Recovery™; Centurion Emergency Notification™; Centurion Enterprise-Level Recovery™; Centurion Episys Hosted Failover™; Centurion Hosted High Availability™; Centurion LiveVault™; Check Writer for Core Director™; CIF 20/20™; CIF 20/20 Teller™; CIF 20/20 Teller Capture™; CIF 20/20 Teller Offline™; Core Director™; Core Director Teller™; Core Director Teller Capture™; Cruise™; CruiseNet™; CruiseNet Account Alerts™; CruiseNet Mobile CU™; CruiseNet Mortgage Statement Extract™; CruiseNet Multi-Currency™; CruiseNet Relationship Pricing™; CUPRO™; CUPRO ALM™; CUPRO ALM Express™; DirectLine International™; DirectLine Wires™; ECS OneLook™; Episys™; Episys Anywhere™; Episys Check Ordering Interface™; Episys Collateral and Document Tracking™; Episys Collection Toolkit™; Episys Contact Event Manager™; Episys Continuity Plan™ (ECP); Episys Continuity Services™; Episys Continuity Services Plus™; Episys Data Store™; Episys Dealer Reserve Accounting™; Episys Escrow Module™; Episys External Loan Processing Interface™; Episys Failover Certification™; Episys Failover Self-Certification™; Episys HMDA Solution™; Episys HMDA Reporting™; Episys ID Scanner Interface™; Episys ID Scoring Integration™; Episys Identity and Sanction Screening Integration™; Episys Identity Q&A Integration™; Episys Instant Issue Card Interface™; Episys IRA Interface™; Episys Management Server™; Episys Monitoring™; Episys Multihost™; Episys OFAC and Identity Score Integration™; Episys OFAC Integration™; Episys Overdraw Tolerance™; Episys PowerCheckUp™; Episys PowerSuite™; Episys Quest™; Episys Real Time External Loan Interface™; Episys Replication Failover™; Episys Skip Payment™; Episys SymDocs™; Episys University™; Episys Vaulting™; Episys Virtualization™; Extra Awards™; FactorSoft™; FlexPass™; FormSmart™; Genesys Check Imaging Suite™; Gladiator™; Gladiator Advanced Malware Protection™; Gladiator Consulting Services™; Gladiator CoreDEFENSE Managed Security Services™; Gladiator eBanking Compliance Services™; Gladiator eCommercial SAT™; Gladiator Enterprise Network Design, Implementation & Support Services™; Gladiator Enterprise Security Monitoring™; Gladiator Enterprise Virtualization Services™; Gladiator Enterprise Vulnerability Scanning™; Gladiator eSAT™; Gladiator eShield™; Gladiator Help Desk Service™; Gladiator Hosted Enterprise Mobility Management™; Gladiator Hosted Network Solutions™; Gladiator Incident Alert™; Gladiator IT Regulatory Compliance/Policy Products™; Gladiator Managed IT Services™; Gladiator Managed Unified Communications Services™; Gladiator NetTeller Enterprise Security Monitoring™; Gladiator Network Services™; Gladiator Phishing Defense and Response Service™; Gladiator Social Media Compliance Services™; Gladiator Technology™; Gladiator Total Protect™; Gladiator Unified Communications Services™; Gladiator Virtual Information Security Officer™; Gladiator Website Compliance Review™; goDough™; GoldPass™; iBizManager™; ImageCenter™; ImageCenter ATM Deposit Management™; ImageCenter Cash Letter™; ImageCenter Exception Processing™; ImageCenter Express™; ImageCenter Image Capture™; ImageCenter Interactive Teller Capture™; InTouch Voice Response™; iPay Business Bill Pay™; iPay Check Printing™; iPay Consumer Bill Pay™; iPay QuickPay™; iPay Solutions™; iTalk™; Jack Henry & Associates, Inc.™; Jack Henry Banking™; JHA Card Processing Solutions™; JHA Cash Automation™; JHA Commercial Cash Management™; JHA Compliance Access™; JHA Consumer Pieces™; JHA Electronic Statements – Interactive™; JHA EMV™; JHA Enterprise Risk Mitigation Solutions™; JHA Failover™; JHA Get Smart™; JHA Image ATM™; JHA Image ATM Capture and Reconciliation™; JHA Marketplace™; JHA Money Center™; JHA OutLink Processing Services™; JHA PayCenter™; JHA Payment Solutions™; JHA Real Time™; JHA Safeguard™; JHA SmartMonitor™; JHA SmartPay ACH™; JHA SmartPay Biller Direct™; JHA SmartPay Business™; JHA SmartPay Card™; JHA SmartPay Commercial Mobile™; JHA SmartPay Consumer/Micro Business™; JHA SmartPay Express™; JHA SmartPay Remote Deposit Anywhere™; JHA SmartPay Remote Deposit Complete™; JHA SmartPay Remote Deposit Now™; JHA SmartPay Remote Deposit Scan™; JHA Sweep Account Processing™; jhaAddress Verify™; jhaAnalytics™; jhaCall Center™; jhaCall Center In-House™; jhaCall Center Outsourced Services™; jhaCall Center Outsourced Services After Hours™; jhaCall Center Outsourced Full Business Services™; jhaCall Center Outsourced Select Services™; jhaData Insights™; jhaDirect™; jhaEnterprise Notifications System™; jhaEnterprise Workflow™; jhaID Scan™; jhaKnow™; jhaKnow Express™; jhaPassPort Debit Optimizer™; jhaPassPort™; jhaPassPort Direct™; jhaPassPort pro™; jhaPassPort Fraud Center™; jhaPassPort Hot Card Center™; jhaPassPort Promotions and Consulting Services™; jhaPassPort Switch™; jhaPredictive Models™; jhaArchive™; jVault™; jXchange™; Know-It-All Credit Programs™; Know-It-All Education™; Know-It-All Learning Management Portal™; Know-It-All Now™; Known Payee Disbursements™; LendingNetwork™; Margin Maximizer Interactive™; Margin Maximizer MaxConnect™; Margin Maximizer Pronto™; Margin Maximizer Suite™; Masterlink™; MaxConnect Interactive™; MedCashManager™; MyCardRules™; Net.Check™; NetTeller™; NetTeller Bill Pay™; NetTeller Cash Management™; NetTeller MemberConnect™; NetTeller Online Banking™; NetTeller Security Manager™; NetTeller Text Alerts™; NetTeller Vanity URL™; NetTeller Vanity URL.bank™; OFX Gateway™; OnBoard Deposits™; OnBoard Loans™; OnNet™; OnTarget™; Opening Act™; Optimizer™; Optimus Credit Union™; Pay Anyone Disbursements™; Point™; PointMobility™; PowerOn™; PowerOn2™; PowerOn2 Batch™; PowerOn2 Docs™; PowerOn2 Editor™; PowerOn2 Screens™; PowerOn Marketplace™; PowerOn Studio™; PROFITability™; Organizational PROFITability Analysis System™; Product PROFITability Analysis System™; PROFITability Budget™; PROFITability Reporting Service™; PROFITstar™; PROFITstar ALM Budgeting™; PROFITstar Budget™; PROFITstar Classic™; PROFITstar Reporting Service™; ProfitStars™; ProfitStars CECL Analysis and Reporting™; ProfitStars CECL Data Store and Validation™; ProfitStars Customer Profitability and Pricing™; ProfitStars Direct™; ProfitStars Financial Performance Suite™; ProfitStars Synergy™; Relationship 360™; RemitCentral™; RemitPlus™; RemitPlus Express™; RemitPlus HRCM™; RemitPlus Remittance/Lockbox™; RemitWeb™; Remote Deposit Express™; ReportHub™; Silhouette Document Imaging™; SilverLake Real Time™; SilverLake System™; SilverLake Teller™; SilverLake Teller Capture™; SilverLake Teller Offline™; Smart EIP™; Smart GL™; SmartSight™; SmartSight for Business™; smsGuardian™; Store & Forward™; StreamLine Platform Automation™; StreamLine Platform Automation – Deposits™; StreamLine Platform Automation – Loans™; Summit Support™; SymAdvisor™; SymApp™; SymChoice Loan™; SymConnect™; SymForm™; SymForm PDF™; Symitar™; Symitar ATM Services™; Symitar Database Cleansing Package™; Symitar eNotifications™; Symitar Fraud Management™; Symitar Member Business Services™; Symitar Member Privilege™; Symitar Wire Management™; Symitar EASE™; SymX™; SymXchange™; Synapsys™; Synapsys Express™; Synapsys Lobby Tracking™; Synapsys Member Relationship Management™; Synergy AdvancedPDF™; Synergy API Integration Toolkit™; Synergy AutoImport™; Synergy Automated Document Recognition™ (ADR); Synergy Batch Document Recognition™ (BDR); Synergy Check Archive™; Synergy DataMart™; Synergy Document Management™; Synergy Document Recognition™; Synergy Document Tracking™; Synergy eDistribution™; Synergy eMailAssist™; Synergy Enterprise Content Management™ (ECM); Synergy eSign™; Synergy eSignWeb™; Synergy eSign Integration Module™; Synergy eStorage™; Synergy Express™; Synergy Express Custom Data Storage Report™; Synergy Express Report Parameter Service™; Synergy ID Scan™; Synergy iSign™; Synergy Kofax Capture™; Synergy One™; Synergy PowerSearch™; Synergy Reports™; Synergy Workflow Management™; TellerMaster™; Teller BondMaster™; Teller CheckMaster™; Teller CheckMaster Plus™; Teller Co-Mingle™; Teller CTRMaster™; Teller Isosceles™; Teller PassBook™; Teller SigMaster™; Teller Orbograph™; TimeTrack Human Resources™; TimeTrack Payroll System™; TimeTrack Time and Attendance™; ValuePass™; Vertex Teller Automation System™; WebEpisys™; Yellow Hammer™; Yellow Hammer Express™; Yellow Hammer ACH Origination™; Yellow Hammer BSA™; Yellow Hammer BSA Express™; Yellow Hammer BSA Regulatory Consulting Service™; Yellow Hammer EFT Fraud Detective™; Yellow Hammer Fraud Detective™; Yellow Hammer SAR Center™; Yellow Hammer Wire Origination™; Xperience™

Slogans

Cutting-Edge IT Solutions for the Future of Credit Unions™; Know-It-All – Empowering Users Through Knowledge™; Leading through technology ... guiding through support™; Powering Actionable Insight™; The Depth of Financial Intelligence™; We Are Looking Out For You™; Where Tradition Meets Technology™

Various other trademarks and service marks used or referenced in this document are the property of their respective companies/owners.

Xperience Programming Reference Guide for a Web App.....	4
Web Application Setup.....	4
TypeScript Definition Files.....	4
JavaScript Files.....	4
Integrating with Xperience.....	4
Startup Arguments.....	5
Security Concerns.....	7
Sending and Receiving Messages.....	7
Connecting to the Message Bus.....	7
Encryption.....	7
Connect.....	8
Disconnect.....	8
Subscribing to the Message Bus.....	8
Publishing.....	9
Sending.....	10
Specifying an App.....	10
Messages.....	10
Identity Management.....	12
Debugging Web/JavaScript.....	15
Debugging JavaScript While Running in Xperience.....	17

Xperience Programming Reference Guide for a Web App

This document covers the basics of getting a product up and running using Xperience. A simple Angular application is available to accompany this document, as well as further information for developing with Xperience once completing the basics.

Web Application Setup

Three TypeScript Definition files must be included before getting a web application running in Xperience.

TypeScript Definition Files

- `jackhenry.enterprise.businessObjects.d.ts`
- `jackhenry.xperience.d.ts`
- `xml2json.d.ts`

JavaScript Files

- `jackhenry.enterprise.businessObjects.js`
- `jackhenry.enterprise.js`
- `jackhenry.xperience.js`
- `xml2json.min.js`

Integrating with Xperience

A web application can determine if it is running in Xperience by:

- API - `jackHenry.xperience.isFrameworkHosted`

The API returns `true` if the application is being hosted in Xperience.

Application setup and message bus connections that happen when hosted in Xperience should be done when Xperience is ready for communication. Xperience calls the function registered by the `onReady` function.

NOTE

Certain functions can be hidden if running outside of Xperience.

Startup Arguments

When the `onReady` callback function is raised, there are several startup arguments that are available for use by the web application.

These startup arguments can be accessed through `jackHenry.xperience.startupArguments`.

The following image shows the `startupArguments`:

Startup Arguments	
MessageBusSessionId	13452
ClientId	WebBrowser WebBrowser[0]
Theme	Pacific
ThemeType	Qrtz
ColorType	Pear
ThemeModeType	Reg
StartUpUri	null
TraceEnabled	true
HelpPopupState	Disabled
AnimationState	Enabled
ZoomState	100
AuthenMode	Online
NotHostedInXpTab	false
UseXmlSerializerEndpoint	true
InstitutionColors	

Security Concerns

The security features/restrictions that are defined in the *Internet Properties* on the client restrict the web browser in Xperience.

If a web-product injects script directly into the browser, the default security features prevent the code from executing as in the following example:

```
<html>
<head>
</head>
<body onLoad="document.myForm.submit()">
  <form action="https://someothersite.com/workers/dosomething.do"
    name="myForm"
    method="post">
    <input type="hidden"
      name="myToken"
      value="Example">
  </form>
</body>
</html>
```

Sending and Receiving Messages

The Message Bus is the mechanism that provides communication between applications executing within the Xperience Framework. The applications hosted within the framework communicate with each other using the Message Bus to coordinate visual user interface activity, and to deliver minor data and application activity among themselves.

Connecting to the Message Bus

Connecting to the Message Bus enables the WCF framework to be established between Xperience and the application.

Encryption

Xperience 2016.0 or later uses encryption to secure the message bus. To be backwards compatible, Xperience allows unencrypted connections for applications not using the newest code release. If you do not want Xperience to allow any unencrypted connections, the `allowunencrypted` property of `IMessageBus` can be set to `false` before a connection is established. Otherwise, the `allowunencrypted` property defaults to `true`.

```
jackHenry.xperience.messageBus.allowUnencrypted = false;
jackHenry.xperience.messageBus.connect();
```

Connect

`connect()` throws a `connectionException` if a failure to connect to the message bus occurs. The following image is an example of a successful connection to the message bus:

```
jackHenry.xperience.messageBus.connect();

// The following call will return true if the application is connected to the message
bus.
jackHenry.xperience.messageBus.isConnected
```

Disconnect

The method parameter clears the message bus subscriptions. Setting to *true* clears any subscriptions. Setting to *false* leaves the subscriptions active, and any reconnection to the message bus handles those subscriptions.

The following is an example of disconnecting from the message bus. The `bool` being passed clears all the subscriptions if *true*.

```
jackHenry.xperience.messageBus.disconnect(true);
```

Subscribing to the Message Bus

Subscriptions can be made for messages at any time, even before `connect()` is called. In the examples below, you can see it is possible to reply to subscribed messages by using the `context.Reply` method.

NOTE

Message Action (string) subscription with in-line delegate. You are required to deserialize the object.

```

jackHenry.xperience.messageBus.subscribe('AppCls', onAppCls);

function onAppCls(context: any) {
    const message = context.messageToJson();

    const response = new jackHenry.enterprise.businessObjects.AppClsResponse();
    const closeStatus = 'ClsAck';

    response.ClsStat = closeStatus;
    response.RsStat = 'Success';

    context.reply(response);
    jackHenry.xperience.messageBus.disconnect(true);
    window.close();
}

```

Subscription methods return a `MessageSubscriptionToken` that can be used to unregister any subscription:

```

const token = jackHenry.xperience.messageBus.subscribe("ThemeChng", onThemeChng);
jackHenry.xperience.messageBus.unsubscribe(token);

```

Publishing

Publishing a message is asynchronous (fire and forget).

```

const message = new XpAngular.CustomMessage();
    message.Text = "Publish CustomMessage";

    const receipt = jackHenry.xperience.messageBus.publish(message);

```

A receipt can be requested using `messageOptions`.

NOTE

If a receipt is not requested, the publish call does not wait for it.

```

const messageOptions = new jackHenry.xperience.MessageOptions();
messageOptions.destinationIds.push(jackHenry.xperience.startupArguments.ClientId);

const responseCallback = jackHenry.xperience.messageBus.send(message,
messageOptions).registerResponseHandler(onSendCallback);

if (!responseCallback.receipt.isValid()) {
    const error = Helpers.ErrorsToString(responseCallback.receipt.errors);
    displayOutput(error);
} else {
    displayOutput("Send Message Receipt ID: " + responseCallback.receipt.messageId);
}

```

Sending

Send is used to receive a response from a destination that is subscribing to the message being sent. This example sends asynchronously and calls the delegate passed into the `registerResponseHandler`.

```

const message = new CustomMessage();
message.Text = "Sending CustomMessage";
const responseCallback =
jackHenry.xperience.messageBus.send(message).registerResponseHandler(onSendCallback);

```

Specifying an App

To send a message to a specific app, use `messageOptions` and the `prodId`.

```

const message = new CustomMessage();
message.Text = "Publish CustomMessage";
const messageOptions = new jackHenry.xperience.MessageOptions();
messageOptions.destinationIds.push("Synergy");
jackHenry.xperience.messageBus.publish(message, messageOptions);

```

Messages

To make sure serialization works correctly for objects sent over the message bus, the class must be able to be serialized. The class name must match the expected message name or `xmlRoot` attribute must be used as found in the EBO.

Using TypeScript, there are specific methods to be implemented to get the correct serialization for the message bus as it is XML and not JSON based.

```

export class CustomMessage {
    public Text: string;
    private _xmlns: string;

    constructor() {
        this._xmlns = 'http://jackhenry.com/XpAngular';
    }

    public ToJsonString(): string {
        const result = "{ 'CustomMessage': { '@xmlns':
'http://jackhenry.com/XpAngular', 'Text': " + this.PropertyToJsonString(this.Text) +
"} }";
        return result;
    }

    private ArrayToJsonString(prop: Array<any>): string {
        let result = '';

        if (prop == null || prop.length <= 0) {
            return '';
        }

        result = "[";

        for (let i = 0; i < prop.length; i++) {
            if (prop[i].Name.ToJsonString != null) {
                result += prop[i].ToJsonString();
                if (i != prop.length - 1) {
                    result += ", ";
                }
            }
        }

        result += "];";

        return result;
    }

    private PropertyToJsonString(prop: any): string {
        if (prop == null) {
            return null;
        }

        if (prop.ToJsonString != null) {
            return prop.ToJsonString();
        }

        return "" + prop + "";
    }
}

```

Identity Management

Obtaining the Current User

```
const currentUserRequest = new jackHenry.enterprise.businessObjects.UsrSecTokenRetrv();  
// Note: setting this to true will get a fresh token from IMS server for  
// the current user  
currentUserRequest.RefreshFromServer = 'false';  
  
Helpers.FillHeader(currentUserRequest);  
  
const responseCallback =  
jackHenry.xperience.messageBus.send(currentUserRequest).registerResponseHandler(onGetCurrentUserCallback);
```

NOTE

If the refreshed token is no longer valid for any reason, the following error information is returned in the response:

- Message - Token Renewal Failed - User is being logged off and a token is not available.
- Error Code - 4200.
- Category - Error.
- Element - "".
- Location - Xperience - userSecurityTokenRetrievalStrategy.

All products receive the `appCls` message when this condition happens. This message is received because the current user is no longer authenticated and Xperience is attempting to log out. In this condition, any products that attempt to cancel the `appCls` in the response are ignored.

Alias Claim Record

```

// Alias - The IMS named alias.
// AliasContent - The context value for the Alias.
// AliasDesc - General description
// AliasDft - Is the alias the default alias in IMS? Canonicals: True/False
// AliasInstRtId
// The identification of the institution related to the Alias. A financial
// institution entity will utilize the routing transit or ABA nine (9) digit
// number assigned to financial institutions for the purpose of routing as
// assigned by the American Bankers Association. Any leading zeros must be
// provided for a complete routing and transit number. A non-financial
// institution entity will use a mutually agreed upon identification that must
// contain at least one non-integer character.
// AppliesToApp - A change/modification being applied to an application
AliasClaimRec_CType newAlias = new AliasClaimRec_CType()
{
    Alias = Guid.NewGuid().ToString(),
    AliasContext = "test",
    AliasDesc = "generic",
    AliasDft = bool.FalseString,
    AliasInstRtId = "12345678",
    AppliesToApp = "http://jackhenry.com/application/XpAngular",
    UniqueId = Guid.NewGuid().ToString()
};

```

Add an Alias

```

const request = new jackHenry.enterprise.businessObjects.AliasAdd();
    request.AliasClaimRec = new
jackHenry.enterprise.businessObjects.AliasClaimRec_CType();
    request.AliasClaimRec.Alias = jackHenry.xperience.Guid.NewGuid();
    request.AliasClaimRec.AliasContext = 'XpAngular';
    request.AliasClaimRec.AliasDesc = 'This is a test alias from - XpAngular';
    request.AliasClaimRec.AliasDft = 'false';
    request.AliasClaimRec.AliasInstRtId = '12345678';
    request.AliasClaimRec.AppliesToApp =
"http://jackhenry.com/application/XpAngular";
    request.AliasClaimRec.UniqueId = jackHenry.xperience.Guid.NewGuid();
    request.IMSSubj = currentUser.IMSSubj;
    Helpers.FillHeader(request);

    const responseCallback =
jackHenry.xperience.messageBus.send(request).registerResponseHandler(onAddAliasCallback
);

```

Modify an Alias

```

lastAddedAlias.AliasDesc = 'This is a test alias from - XpAngular - Updated Desc';

    const request = new jackHenry.enterprise.businessObjects.AliasMod();
    request.AliasClaimRec = lastAddedAlias;
    request.Dlt = 'false';
    request.IMSSubj = currentUser.IMSSubj;
    Helpers.FillHeader(request);

    const responseCallback =
jackHenry.xperience.messageBus.send(request).registerResponseHandler(onModifyAliasCallb
ack);

```

Delete an Alias

Uses the same `AliasModRq_MType`, but with the `Dlt` element set to "True".

Obtain a Kerberos Token for the Current User

```

const request = new jackHenry.enterprise.businessObjects.GetKerberosToken();

    // Service Principal Name (SvcPrincName) is *required* in the request.
    // TokenTimeoutMin is optional.
    // Defaults to 60 minutes if omitted.
    // Must be > zero if supplied.
    request.SvcPrincName = "testsvc/testresource.dev.jha";
    request.TokenTimeoutMin = 30;

    Helpers.fillHeader(request);

    const responseCallback =
jackHenry.xperience.messageBus.send(request).registerResponseHandler(onGetKerberosToken
Callback);

    if (!responseCallback.receipt.isValid()) {
        const error = Helpers.errorsToString(responseCallback.receipt.errors);
        displayOutput(error);
    }

```

If the `svcPrincName` is not supplied, the following error information is returned in the response:

- Message - `Missing Information.`
- Error Code - `2010.`
- Category - `Error.`
- Element - `svcPrincName.`

- Element Value - "".
- Location - Xperience - xpMessageBusService.

If an invalid `tokenTimeoutMin` value (less than or equal to zero) is supplied, the following error information is returned in the response:

- Message - `tokenTimeoutMin` must be greater than zero.
- Error Code - 2012.
- Category - Error.
- Element - `tokenTimeoutMin`.
- Element Value - (The supplied value).
- Location - Xperience - xpMessageBusService.

If there is no authenticated user, the following error information is returned in the response:

- Message - A user has not been authenticated.
- Error Code - 4017.
- Category - Error.
- Element - "".
- Element Value - "".
- Location - Xperience - xpMessageBusService.

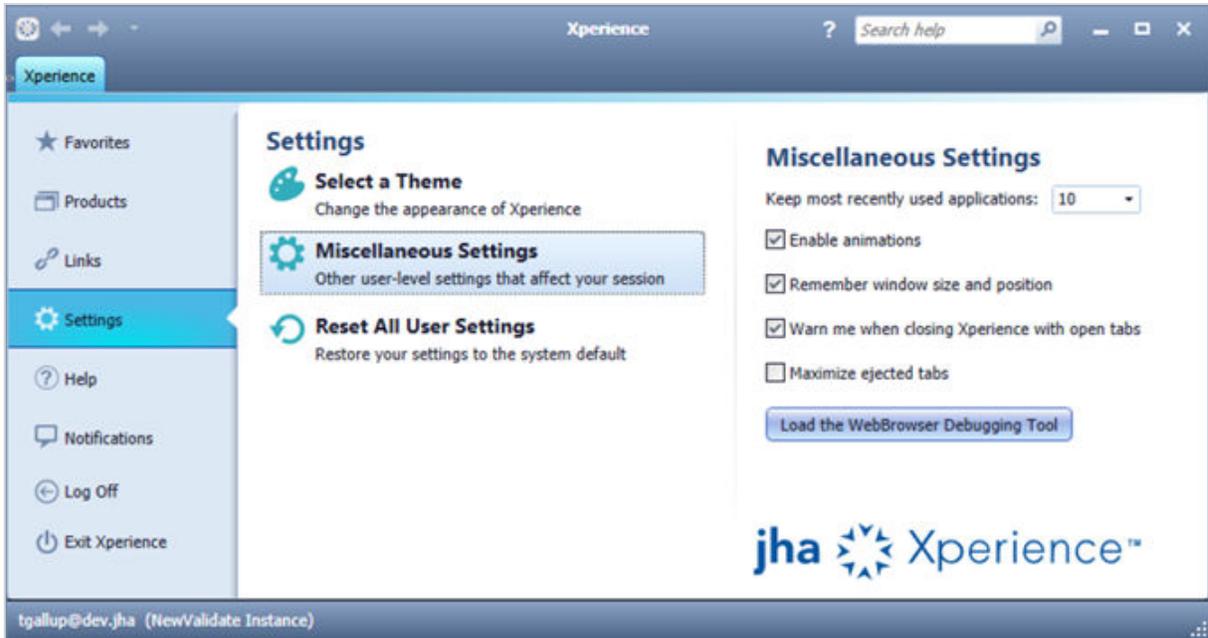
Debugging Web/JavaScript

When debugging web applications running inside of Xperience, use the **WebBrowser Debugging Tool**.

NOTE

This is a tool built into Xperience that is intended only for development. By default, this tool is unavailable so that the financial institutions do not see it. Contact Product Adoption to enable the tool.

1. Log on to Xperience.
2. Select **Settings > Miscellaneous Settings**.

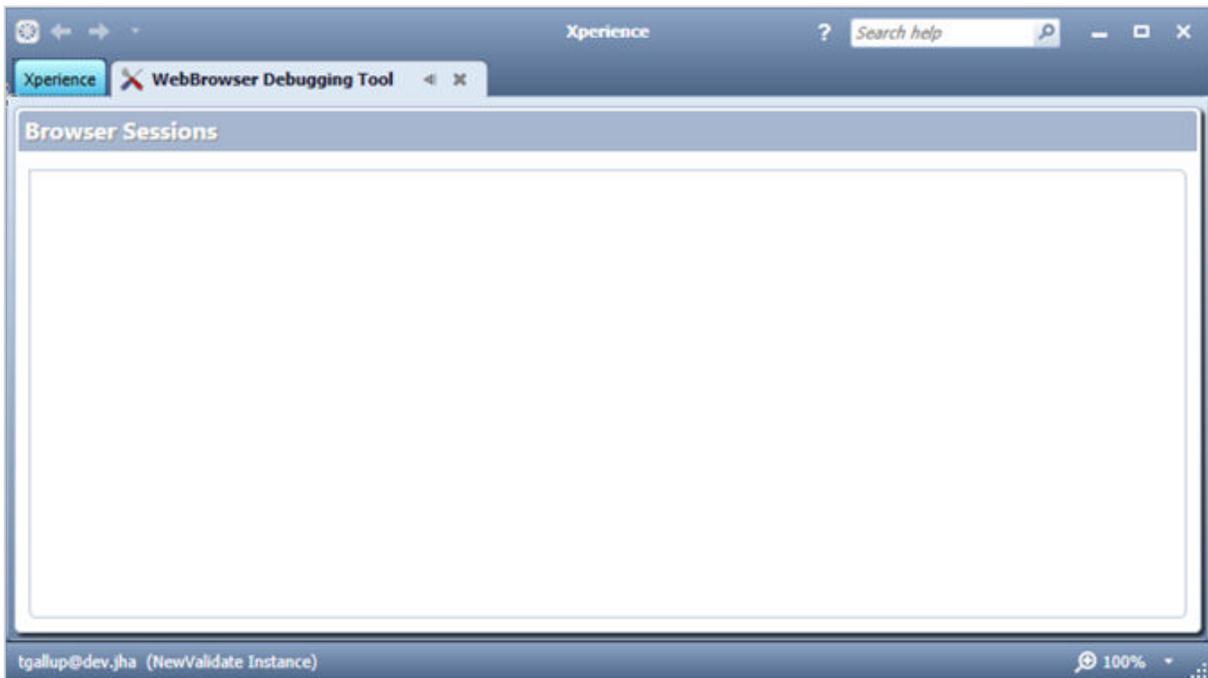


3. Select the **Load the WebBrowser Debugging Tool** button.

NOTE

This option is not shown when the **Show Browser Debugging Tool** is unavailable on the server instance.

A new *Xperience* tab is opened for the tool:



4. Load the product/website to debug as you normally would.

Products can log directly to the Web API with `window.external.log(message)`.

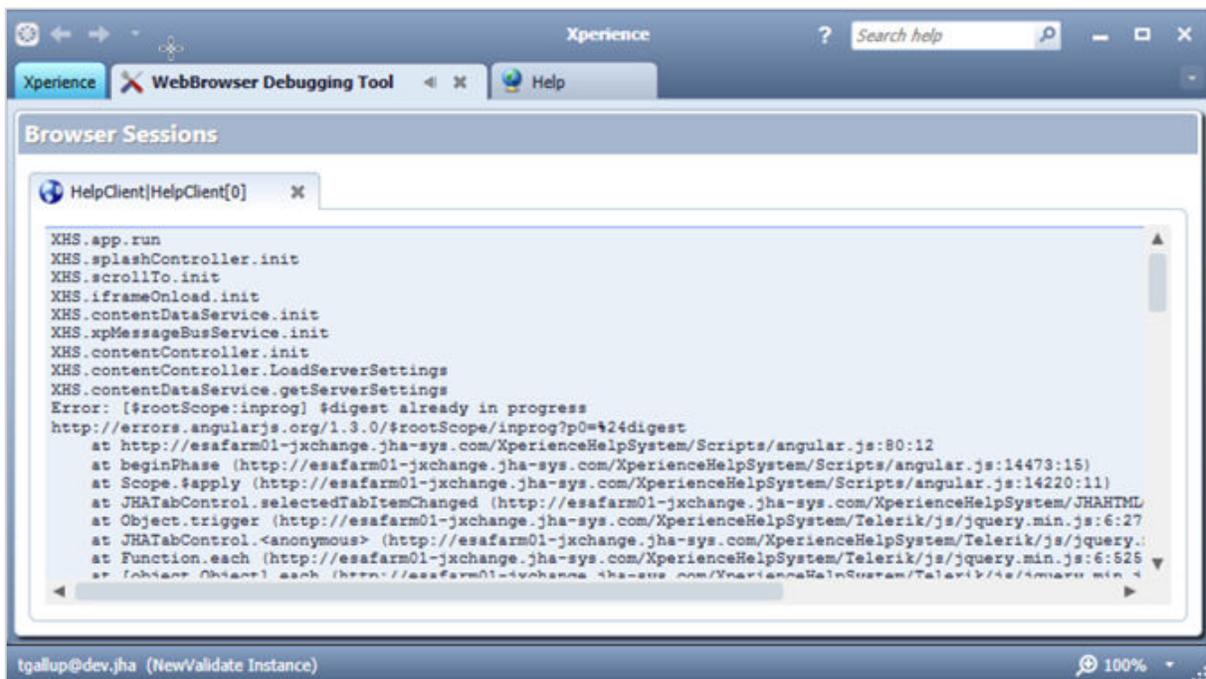
It is up to the product development team to include the logging in the JavaScript source.

Closing a tab in the WebBrowser Debugging Tool clears all information for that setting. However, if the WebBrowser is still active and more logging is done from the application, a new tab is created.

NOTE

All previous logging messages are lost when a tab is closed.

Xperience automatically injects JavaScript into all loaded pages that redirect any console (debug, info, log, error) messages (in the F12 screen) to the Web API logging function. These messages are then shown in the loaded *WebBrowser Debugging Tool*. Each tab indicates the session ID of the WebBrowser:



Debugging JavaScript While Running in Xperience

Follow this process to attach and debug scripts running inside the Xperience web browser.

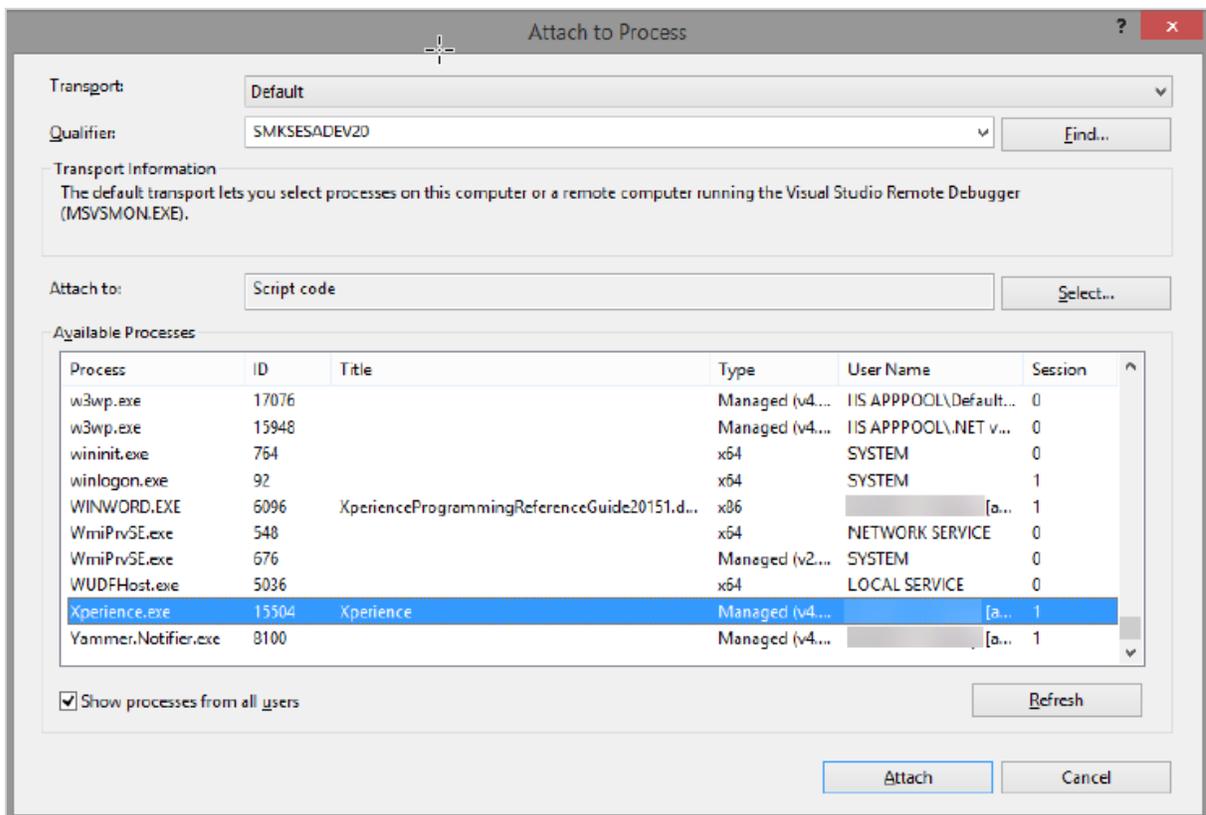
Start and log in to Xperience (as needed).

1. Load your web application project in Visual Studio™ if not already loaded.

2. Choose **Attach to Process** on the **Debug** menu.
3. Select **Xperience.exe** from the list of **Available Processes**.

In the **Attach to:** box, you should see **Script Code** or **Automatic: Script Code**. If anything else, complete these steps:

- a) Click **Select**.
- b) In the *Select Code Type* dialog box, click **Debug These Code Types** and select **Script**.
- c) Click **OK**.



In the **Attach to:** box, you should see **Script Code** or **Automatic: Script Code**. If anything else, complete these steps:

- a) Click **Attach**.
- b) Set any breakpoints in the JavaScript code as needed.
- c) Start the application inside Xperience.

The code stops at the breakpoints that were set, and normal Visual Studio™ debugging tools are available, such as **Watch Windows** and **Output Windows**:

```
jackHenry.xperience.messageBus.subscribe("ThemeCng", function (messageContext) {...});
jackHenry.xperience.messageBus.subscribe("AppClis", function (messageContext) {...});
jackHenry.xperience.messageBus.subscribe("HistClrResponse", function (messageContext) {...});
// These are the product specific messages that this product handles:
jackHenry.xperience.messageBus.subscribe("ContentShow", function (messageContext) {...});
jackHenry.xperience.messageBus.subscribe("ContentSrch", function (messageContext) {...});
jackHenry.xperience.messageBus.subscribe("HistoryNavigation", function (messageContext) {...});
jackHenry.xperience.messageBus.onReady(function () {
    $log.debug("XIS.app:jackHenry.xperience.messageBus.onReady");
    if (jackHenry.xperience.isFrameworkHosted) {
        jackHenry.xperience.messageBus.connect();
        if (jackHenry.xperience.messageBus.isConnected) {
            $rootScope.$broadcast("xpMessageBusConnected", jackHenry.xperience.messageBus.isConnected);
            if (angular.isDefined(jackHenry.xperience) && angular.isDefined(jackHenry.xperience.startupArguments)) {
                LoadTheme(jackHenry.xperience.startupArguments.ThemeType, jackHenry.xperience.startupArguments.ColorType, jackHenry.xperience.startupArguments.ThemeModeType);
                // This will skip the loading of a default theme below:
                return;
            }
        }
    }
    // This is the default theme to use, if we are running outside of XP:
    LoadTheme("Quartz", "Pacific", "Regular");
}, 1500);
function LoadTheme(themeType, themeColor, themeMode) {
    $log.debug("XIS.app:LoadTheme");
    switch (themeColor) {
        case 'Inst':
            themeColor = "Institution";
            break;
        case 'Tang':
            break;
    }
}
```

See this Microsoft® link for additional information: <https://msdn.microsoft.com/en-us/library/bb385613.aspx>