



Additional Programming information for Instant Bank Verification (IBV)

Data that is not sent in the request will be required by the end user on the verification personal information form. All fields must be URL encoded if present; this is done automatically in C# if using the WebClient object and if the content type is set as indicated above.

Note: If you use an LMS, you may have special programming requirements from the LMS.

Use of IFrames and Overlay

The IBV verification screens can be presented within an Inline Frame (IFrame) from your web page. IFrames allow you to include content from external sources inside your web pages. One means of presenting the IBV verification screens is with the use of an overlay.

Sample Code for Overlay

The sample code below is one example for creating the pop-up screen overlay that pre-populates the applicant's information.

Using fancybox library (based on jquery javascript library):

```
<!DOCTYPE html> <!-- Valid DOCTYPE is required for FancyBox to look and function correctly. Full
documentation at http://fancybox.net/-->
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<script type="text/javascript" src="jquery.fancybox.js"></script> <!-- full documentation and download at
http://fancybox.net/ -->
<link rel="stylesheet" type="text/css" href="jquery.fancybox.css" media="screen" /> <!-- full documentation
and download at http://fancybox.net/ -->
```

```
<!-- You will need fancybox_loading.png, fancybox_overlay.png and fancybox_sprite.png. These files along
with full documentation can be downloaded at http://fancybox.net/ -->
```

```
<script type="text/javascript">
$(document).ready(function() {
    $(".MBIBV").fancybox({
        'width': '60%',
        'height': '70%',
        'fancybox-skin': 'background: white'
    });
});
</script>
```



```
</head>

<body>

<!--your code here-->

<a class="MBIBV fancybox.iframe" href = "http://www.microbilt.com" data-fancybox-type="iframe">Show
external page within iframe</a>
    <!-- href = "Replace with URL from CreateForm response" -->
<!--your code here-->

</body>
</html>
```

Size and position of the pop-up can be determined by the properties specified in the CSS style.

In the html code, you will need to replace the link to ["http://www.microbilt.com"](http://www.microbilt.com) with ["URL from CreateForm response"](#)

Sample screen follows and more information regarding the pop-up follows in the next section.

Callback URL Sample Code

Sample code in C# .NET for Callback implementation on the listening site (as specified in CallbackUrl on request).

```
using System.Web;

namespace MicroBilt.Callback
{
    public class Callback : IHttpHandler
    {
        public bool IsReusable
        {
            get { return true; }
        }

        public void ProcessRequest(HttpContext context)
        {
            var results = new Results
            {
                Reference = context.Request["Reference"],
                Success = context.Request["Success"],
                AccountOwnerFirstName = context.Request["AccountOwnerFirstName"],
                AccountOwnerLastName = context.Request["AccountOwnerLastName"],
                AccountOwnerMatches = context.Request["AccountOwnerMatches"],
                AccountName = context.Request["AccountName"],
            }
        }
    }
}
```



```
InstitutionName = context.Request["InstitutionName"],
InstitutionNameMatches = context.Request["InstitutionNameMatches"],
AccountNumber = context.Request["AccountNumber"],
AccountNumberMatches = context.Request["AccountNumberMatches"],
RoutingNumber = context.Request["RoutingNumber"],
RoutingNumberMatches = context.Request["RoutingNumberMatches"],
AccountType = context.Request["AccountType"],
AccountTypeMatches = context.Request["AccountTypeMatches"],
DirectDepositAmount = context.Request["DirectDepositAmount"],
DirectDepositAmountDiff = context.Request["DirectDepositAmountDiff"],
DirectDepositPayCycle = context.Request["DirectDepositPayCycle"],
DirectDepositPayCycleMatches = context.Request["DirectDepositPayCycleMatches"],
Balance = context.Request["Balance"],
DaysActive = context.Request["DaysActive"],
NSFCount = context.Request["NSFCount"],
GrossIncome = context.Request["GrossIncome"],
DirectDepositor = context.Request["DirectDepositor"],
};
```

```
// process the results
// note that you don't have to send any response back, MicroBilt is not expecting any
}
```

```
//
```

```
private class Results
```

```
{
    public string Reference { get; set; }
    public string Success { get; set; }
    public string AccountOwnerFirstName { get; set; }
    public string AccountOwnerLastName { get; set; }
    public string AccountOwnerMatches { get; set; }
    public string AccountName { get; set; }
    public string InstitutionName { get; set; }
    public string InstitutionNameMatches { get; set; }
    public string AccountNumber { get; set; }
    public string AccountNumberMatches { get; set; }
    public string RoutingNumber { get; set; }
    public string RoutingNumberMatches { get; set; }
    public string AccountType { get; set; }
    public string AccountTypeMatches { get; set; }
    public string DirectDepositAmount { get; set; }
    public string DirectDepositAmountDiff { get; set; }
    public string DirectDepositPayCycle { get; set; }
    public string DirectDepositPayCycleMatches { get; set; }
    public string Balance { get; set; }
    public string DaysActive { get; set; }
    public string NSFCount { get; set; }
    public string GrossIncome { get; set; }
    public string DirectDepositor { get; set; }
}
}
```



Mobile Devices

The verification screens are built using a responsive design and will adjust according to the device. To present in a mobile version, IBV can be used in an iFrame with a workaround (such as Mobify). As a preferred alternative, we suggest that you *not* use the iFrame for a mobile version and make use of the FinalURL to redirect the end user back to your site.