**Add-in Central for Developers (quick notes)**

Add-in Central is a community powered add-in manager. Add-in Central is a Windows Home Server add-in and service designed to help you discover and track useful add-ins right from within the Windows Home Server Console. Enhance your Windows Home Server experience by browsing for add-ins by category, keyword, or even by community rated popularity. Useful summaries, screenshots, and documentation links are provided to help guide your search. Once you’ve located an add-in that looks interesting, Add-in Central can automatically download it to your home server and help you along with a hassle-free installation process.

The project is organized by and spans the leading home server destinations such as (homeserverland.com, home-server-blog.de etc) with the ultimate goal to help users explore and get the most out of Windows Home Server.

**The Add-in Central API**

Add-in Central uses a publicly accessibly application programming interface (API). The protocol is HTTP which makes it relatively easy to communicate with + additional protocols such as Simple Object Access Protocol (SOAP) and the most direct Representational State Transfer (REST) are supported as well.

The static web service URL is:

<http://api.addincentral.com/v1services.asmx>

These web services easily consumable by third party developers. Note, that all “Get” prefixed methods are FREE for all and do not require authentication. The “Post” methods require a special authentication token.

**Method 1: Web Reference**

Since these APIs are standard ASP.net web services, developers can easily add a reference which would generate a proxy class. This class defines all of the methods exposed and at design time, the added reference enables the developer use the statement completion in the code editor using the IntelliSense. The actual implementation of the methods in the proxy class is comprised of code to package and send SOAP request message and to receive and un package any returned SOAP response message.





[This works great, is super easy to use but may add additional overhead]

**Method 2: Lightweight Update Checking Example**

The following shows how we could query the Add-in Central system for the version information of Grid Junction.

// we start with our destination URL, notice the ? param which jumps directly to the webmethod
private static string url = "http://api.addincentral.com/v1services.asmx/GetAddinByTitle?title=gridjunction";

/// <summary>

/// Uses an very fast HTTP GET to query latest version

/// </summary>

/// <returns></returns>

private bool UpdateAvailableCheckExample()

{

 // Create web request

 HttpWebRequest httpWebRequest = null;

 try

 {

 // Initialize web request

 httpWebRequest = (HttpWebRequest)WebRequest.Create(url);

 httpWebRequest.Method = "GET";

 httpWebRequest.UserAgent = "GridJunction Update Checker";

 httpWebRequest.KeepAlive = false;

 httpWebRequest.Timeout = 15 \* 1000; // Set timeout to 15 seconds

 // Get response

 using (HttpWebResponse response = httpWebRequest.GetResponse() as HttpWebResponse)

 {

 // Get the response stream

 StreamReader streamReader = new StreamReader(response.GetResponseStream());

 // Read the Response Stream using XmlTextReader

 XmlTextReader xmlTextReader = new XmlTextReader(streamReader);

 //read through all the nodes

 while (xmlTextReader.Read())

 {

 if (xmlTextReader.NodeType.Equals(XmlNodeType.Element))

 {

 switch (xmlTextReader.LocalName)

 {

 case "Version":

 liveVersion = xmlTextReader.ReadString();

 WriteDebug("GridJunctionService UpdateAvailableCheck liveVersion = "+ liveVersion, EventLogEntryType.Information);

 break;

 }

 }

 }

 streamReader.Close();

 response.Close();

 }

 if (IsNewerVersion(liveVersion))

 {

 return true;

 }

 }

 catch (Exception exc)

 {

 WriteDebug("GridJunctionService UpdateAvailableCheck: " + exc.Message, EventLogEntryType.Error);

 }

 finally

 {

 if (httpWebRequest != null)

 httpWebRequest = null;

 }

 return false;

}

That’s about it. Where “IsNewerVersion(liveVersion)” would be a method which self-reflects to compare itself against the version in Add-in Central.