PREPAID WEB SERVICES

SERVICE DESCRIPTION

Effective: October 25, 2014

© *Visa Inc.* 2014

NOTICE: This information is Confidential and must be used exclusively by the Visa customer in connection with Visa programs. The information contained in this media may be copied and distributed for internal use by the Visa customer in connection with Visa programs. It may not otherwise be duplicated, published, or disclosed without written permission from Visa.

D-0273-07

Table of Contents

Revision History

Section	Change Description
Changes Made in This Version:	
Available Services	Updated sections: • Service: Instant Issue Card • Service: Manage Cardholder
Release Strategy	Updated schedule diagrams for the 2014 release. Added schedule diagrams for the 2015 release. Removed schedule diagrams for 2013 releases.
Changes Made on October 25, 2013:	
Release Strategy	Updated schedule diagrams for the 2013 release. Added schedule diagrams for the 2014 release. Removed schedule diagrams for 2012 releases.
Changes Made on February 1, 2013:	
Available Services	Updated sections: • Service: Manage Buyer • Service: Manage Cardholder
Connectivity Options	Added descriptions for each of the web services connectivity options.

Section	Change Description
Changes Made on August 31, 2012:	
Available Services	Added new sections: • Search for Account Holder • Service: Manage Account Holder • Service: Large Card Order Updated sections: • Service: Search • Service: Manage Buyer • Service: Manage Mail Order Card • Service: Inventory Control • Service: Fraud
Release Strategy	Updated schedule diagrams for the 2012 and 2013 releases. Removed schedule diagrams for 2011 releases.
Changes Made on March 12, 2012:	
Available Services	Added new Service: Fraud section
Release Strategy	Updated schedule diagrams for the 2012 releases.
Changes Made on September 9, 2011:	
Release Strategy	Updated with schedules and recommendations.

1 Web Services: A Case Study

What are Web Services?

To help explain the web service concept, this manual begins with an example of how you might integrate Visa's prepaid web services product into your existing service environment.

Consider the following scenario. Your house is equipped with a large number of electrical outlets, many of which you use to supply the power to your appliances, entertainment systems, computers, and so forth.

You leave some of these appliances plugged in at all times. Others you plug in only when you need them. The excess capacity is something you take for granted. You also take for granted the fact that the power coming into your house is compatible with all current and future appliances.

So it is with web services. Web services are compatible with all your existing business functions and processes. Moreover, they provide excess capacity in the form of additional functions you can elect to use in the future, but can ignore until such time as you need them.

Now suppose that you work for the hypothetical Mile High National Bank. The tellers who work in your organization use a web application called the Customer Relationship Profile system (Figure 1) to manage customer transactions, deposits, withdrawals, loan payments, and so forth.

Upper management has decided that it wants to begin selling prepaid cards in addition to the other services it offers its current banking customers.

In shopping this requirement around, you discover that what the majority of the vendors have to offer is a full service application designed to manage your prepaid card business, handling everything from fulfillment to funding, from reports to chargebacks.

The drawback of such a solution, however, is that you would have to run it alongside your other applications, imposing additional training and administrative requirements on your current staff.

If there were some way of adding a prepaid card function to the Customer Relationship Profile System, your tellers could use the same interface as they currently use, and you could integrate the necessary back-end functions with the rest of your business.

As it turns out, this is exactly what prepaid web services allows you to do. As in the analogy with the electrical outlet, you would be able to "plug" your Customer Relationship Profile System into the prepaid web services "interface," and thereby add new prepaid functions to your current teller platform.

For example, in adding an instant issue prepaid card function to your Customer Relationship Profile System, you might start with a link to the function on the main screen:

Prepaid Instant Issue function →

To utilize the prepaid web services instant issue functionality, the first thing you would need to do is locate the name of the customer in your own teller system who is making the prepaid card purchase, and pass this data to the collect buyer information function.

With this information collected, you might then prompt the teller to enter the number of the card to be issued and the amount:

Type in the card number and enter the amount of the card.

Once you submit this data, you would then collect information pertaining to the cardholder.

Enter buyer information in accordance with web services data collection requirements.

Information confirming the entered data is displayed by the interface, at which point you can either close out the sale or process additional cards:

The first card sold is shown in the Card List.

To sell this same customer another card, select the desired card type and click Add a Card.

To complete the sale, you ask the customer how they wish to pay for the card, and debit their account accordingly.

Clicking Process the Order sends message via web services to the prepaid platform.

By electing to Process the Order, this sends a message via web services to the prepaid platform, which in turn creates the cardholder and buyer records and records the sale.

Once the cardholder record has been established, you can use other web service calls to view transaction data, run reports, and perform other functions as necessary to monitor balance information. The only step remaining is to give the customer their card and print a receipt.

2   Prepaid Web Services Overview

Visa Prepaid is the name for a collection of functions and infrastructure elements used to create and manage prepaid card programs. Until recently, issuers accessed these functions using Visa's *Program Administration System (PAS)* or *Prepaid Administration Tool (PAT)*, providing necessary consumer access through an issuer-branded website or VRU.

Web services provide an alternate means of accessing these functions by allowing issuers to program their own interfaces, thereby making it possible to integrate basic prepaid program functions (such as the ability to issue prepaid cards) into their own systems and applications.

In its most basic form, a web service is an *Extensible Markup Language (XML)*-based software application accessed over an Internet connection. XML messages transmitted across this connection are formatted using the *Simple Object Access Protocol (SOAP)* specification, which consists of a *message* sent to the service followed by whatever parameters are needed to complete the request.

When a message is sent by the service consumer, it is called a *request*. Visa, the service provider, sends back a *response*. Requests sent by an issuer generally accomplish one of two things—they either query the Visa database or update the database. It is on this simple framework that web services are based. This configuration is illustrated below.

Figure 1:   Requests and Responses

| Service Provider (Visa) | SOAP ←→ | Service Consumer |
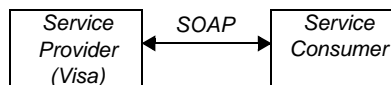
Prepaid Architecture

Prepaid consists of the following functional components:

- A Tandem back end processor, responsible for housing the database and all configuration parameters, and for processing updates

- An SQL (replication) server, responsible for serving all data requests and queries (note that due to replication, users might experience a slight delay between the time that an update is received and processed by the back end processor and availability of the data on the SQL server)

- A web services tier, on which the web services reside—all update operations and queries are routed through the services tier, regardless of the interface

- A presentation tier, which houses the application interfaces PAS, PAT, and the Consumer Websites

- An external client presentation tier, used by the consumer of Web Services

The following diagram shows these components in simplified form.

D-0273-07
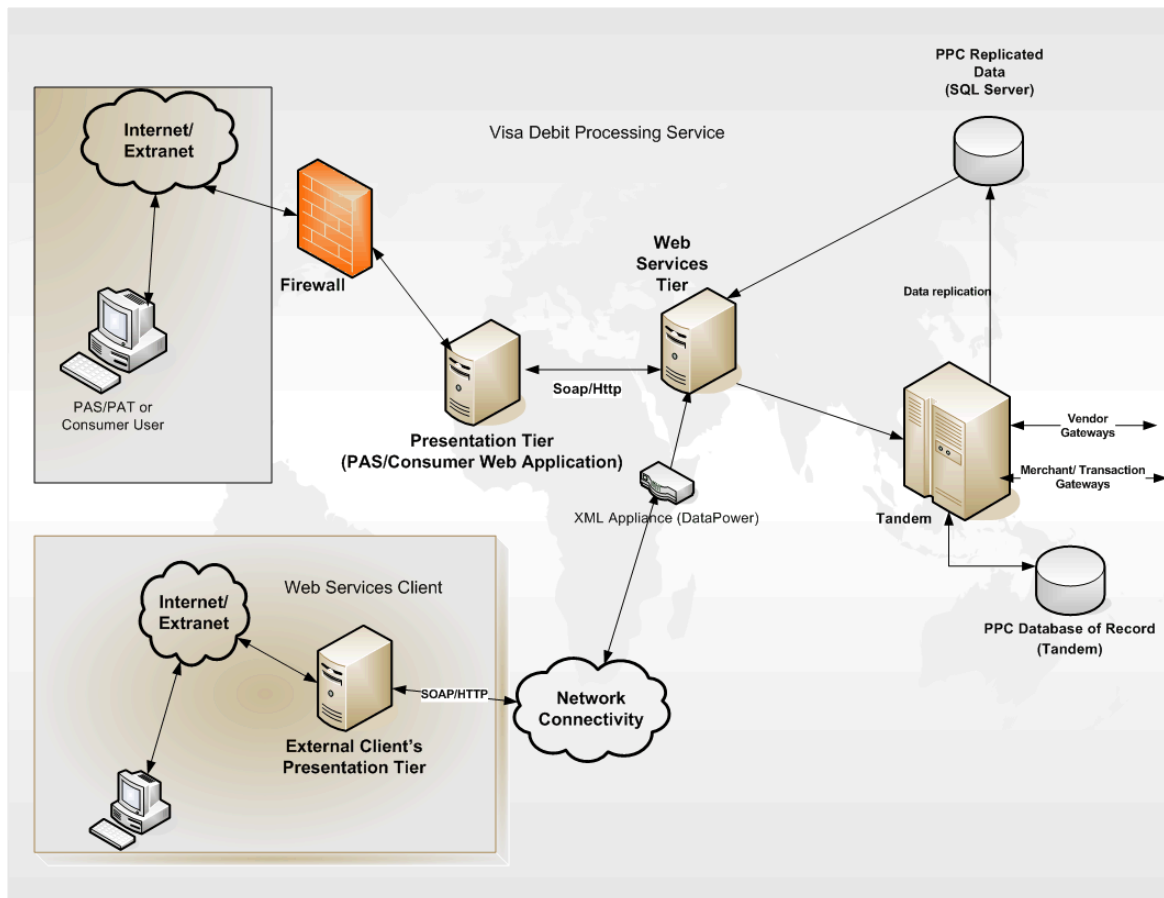
Regardless of the interface you use to access the services tier, the back end processor does not change. Thus, you can run web services in conjunction with the presentation tier without conflict or adverse effect.

Service Architecture

Prepaid SOAP messages adhere to a predictable pattern consisting of one or two operations per business service, with individual operations aggregated within the service. For example, the InstantIssueCard service includes both a Get operation (GetInstantIssueMessage) and an Update operation (UpdateInstantIssueMessage). Because SOAP over HTTPS can create significant per operation overhead, the ability to group operations helps to reduce this overhead.

Grouped within each Get or Update operation are one or more requests, which differ according to the specific data or entities you intend to access in the database. FeeDetailsRequest is an example of a request within this service.

Get operations are used to get configuration information or entity tables while Update operations are used to change entity states in the back end processor. An *entity* is defined as any combination of related fields or rows in the database.

Operation names adhere to the following naming conventions:

Table 1:   **Operation Naming Conventions**

Format	Example
Get<*BusinessFunction*>Message	GetInstantIssueMessage
Update<*Business Function*>Message	UpdateInstantIssueMessage

Available Services

The following services are available in the Visa Prepaid web services model.

Service: Instant Issue Card

Operations defined within this service allow the application designer to:

- Issue cards to a cardholder in a pending issuance status
- Issue replacement cards
- Request fee information
- Standardize address information
- Verify a card number

- Add funds to a card

- Authorize a load against a load source, company account, or on-us transaction

- Update location card inventory

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Instant Issue Card Messages

The Instant Issue Card service consists of two operations, one to get information needed for card issuance, and the other to place the card order (update). To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Search

Operations defined within this service allow the application designer to:

- Initiate a search for account holder information

- Initiate a search for buyer information

- Initiate a search for cardholder information

- Initiate a search for location information

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Search Messages

All Search service operations are get operations.

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Search for Account Holders

You can search for a list of account holders using one of the following elements:

- Account Holder Name

- Address

- Funding Card Number

- Funding Account Number

- Card Number (of Teen Cardholder)

The following account holder information is returned as part of the response:

- Card Program

- Alias ID

- Name

- Account Holder Identifier

- Balance For All Teen Cardholders

- Number of Cards

- Address

Search for Buyers

You can search for a list of buyers using one of the following elements:

- Prepaid Card Number

- Order Confirmation Number

- Buyer Name

- Company Name

- Funding Card Number

- Funding Account

- Bulk Gift Company Name

The following buyer information is returned as part of the response:

- Card Program

- Primary Funding Account

- Buyer Alias Id

- Buyer Type

- Buyer Status

- Is Fraud Blocked

- Company

- Name

- Address

- Government Id

- Phone Number

- Number of Outstanding Cards

- Last Purchase Date Time

- Last Purchase Volume

Search for Cardholders

You can search for a list of cardholders using one of the following elements:

- Buyer Name

- Cardholder Name

- Company Name

- Order Confirmation Number

- PAN (card number)

- Employee Id

- Account Number

- Phone Number

- Proxy Id

The following cardholder information is returned as part of the response:

- Cardholder Identifier

- Cardholder Demographics

- Card Program

- Cardholder Special Instructions

- Card Summary

- Location

Search for Locations

You can search for a list of locations using one of the following elements:

- Contact Name

- Contact Phone

- Location Address

- Location Name

- Location Setup

The following location information is returned as part of the response:

- Address

- Contact Phone

- Location Contact

- Location Identifier

Service: Manage Account Holder

Operations defined within this service allow the application designer to update account holder information.

Use the Manage Account Holder service to perform the following functions:

- Manage an account holder profile

- Manage an account holder funding account

- Enroll an account holder

- Manage statements

- Manage alerts

- Get list of teen cardholders

- View fees

- Get funding transaction history

- Manage account notes

- View account history

- View list of fraud cases

- Manage limit overrides

- Close an account

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Manage Buyer

Operations defined within this service allow the application designer to update buyer information.

Use the Manage Buyer service to perform the following functions:

- Update a buyer profile

- Update a buyer or funding account

- Get account and buyer history

- Get a list of associated fees or limits

- Standardize an address

- Bulk activate all cards in an order

- Activate specific batches within an order

- Create merchant adjustments or merchant adjustment reversals

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Manage Buyer Messages

The Manage Buyer service consists of two operations, one to request (get) information needed for buyer maintenance, and the other to update buyer information. To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Mail Order Card

Operations defined within this service allow the application designer to:

- Request mail order fee and funding account information

- Open a secondary cardholder account

- Enter a mail order for prepaid cards (Commercial, Gift, Payroll, Reloadable, or Teen)

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Mail Order Card Messages

The Mail Order Card service consists of two operations, one to request (get) information, and the other for placing the order. To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Manage Cardholder

Operations defined within this service allow the application designer to:

- View account and transaction history

- Get fees

- Get funding account information

- Activate a prepaid card

- Close a primary or secondary cardholder account

- Create a load request

- Manage PIN creation

- Set a default PIN for a card before activation

- Manage statement options

- Issue emergency card replacements

- Update returned card information

- Manage convenience checks

- Manually reissue a prepaid card

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Manage Cardholder Messages

The Manage Cardholder service consists of two operations, one to request (get) information, and the other for placing the order. To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Manage Location

This service allows the application designer to view location-specific information.

Service: Diagnostics

The Diagnostics service verifies connectivity with the server by performing a simple test designed to get the name of the card program running on the remote server. This service gives programmers a means to verify connectivity with the server prior to sending a message.

Service: Inventory Control

Operations defined within this service give the application designer access to:

- Create a new stock order

- Sub-divide a card order into multiple batches

- Maintain contact information

- Maintain contact notification information

- Maintain "default" order settings

- Get historical stock order information

- Search for completed orders

- Search for queued orders

- Transfer order between sub clients

- Transfer cards and update card statuses using CSV Bulk Uploads

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Inventory Control Messages

The Inventory Control service consists of two operations, one to request (get) information needed for Inventory Control maintenance, and the other to update Inventory Control information. To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Reference Data

Operations defined within this service replace the previous enumerated list functions with operations that allow programmers to request and manipulate reference data.

Use the Reference Data service to perform the following functions:

- Get a list of all available reference data and its corresponding information

- Get the list of possible values for a specified set of reference data

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Large Card Order

Operations defined within this service allow the application designer to:

- Create large card orders for personalized and non-personalized cards

- Get fees

- Get limits

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Service: Fraud

Operations defined within this service allow the application designer to:

- Open a fraud case on an individual account for a cardholder, buyer, or gift giver.

- Manage the Negative File

To review the definition of the request and the associated response types, start with the message request and response type for the operation defined in the WSDL.

Standardize Cardholder Address

Many of the services described in the previous pages include a standardized address function. This function is used to compare a supplied or entered address with the postal service standardized address, and then to return the standardized address in the response.

Get Configuration Information

Use the get configuration information function to query the data stored for a given card program or value in the database. Most of the services described in the previous pages use a get configuration function to capture information needed to complete the particular operation.

Assumptions

Use of these services assumes the following:

1. The issuer has coordinated with Implementations to set up the required card programs, and that all issue options have been fully configured in the database.

NOTE: *Before commencing web service operations, Visa recommends that issuers do a configuration request and create a table of values for their card programs for reference purposes.*

2. Connectivity with Visa is in place and has been verified.

3. All required Service Account IDs and passwords have been provisioned.

4. The User ID (`UserIdentity`) sent in each request must correspond to a unique user ID belonging to an actual user on the client-end system.

Connectivity Options

To issue a request using the Prepaid Web Services, either a direct (leased) line, a secure VPN connection, or an Internet connection is required.

When determining which connectivity method to utilize for Prepaid Web Services, clients should consider their:

• Security requirements

• Network infrastructure and capacity

• Projected bandwidth for web service message traffic

Internet Connection

Access to Prepaid Web Services via the Internet requires a valid Visa Online (VOL) account and password. Installation of security certificates are required to allow for authentication with Visa DPS. Certificates expire and need to be reissued every three years.

Figure 2:   **Internet Connection Diagram**
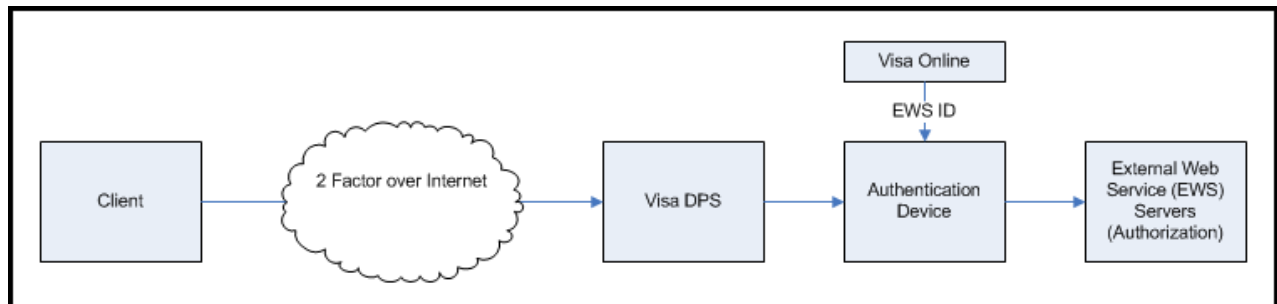
Advantages:

• Self-service support for dynamic password changes through Visa Online

• No hardware requirements, resulting in faster set up time

• Ability to originate web service calls from multiple locations, including Disaster Recovery/Business Continuity sites

Considerations:

• Client must manage and install certificates

• Coding is required to digitally sign web service messages

• Connectivity is dependent upon Visa Online (VOL) access

Visa DPS Managed Connection (VPN or Direct Line)

Access to Prepaid Web Services can be established using either a direct (leased) line or secure VPN connection. Clients need to allow time for the procurement and installation of routers at Visa DPS and client locations. Firewall changes are required to accommodate the connections being established between the client and Visa DPS.

Figure 3:　　VPN Connection Diagram
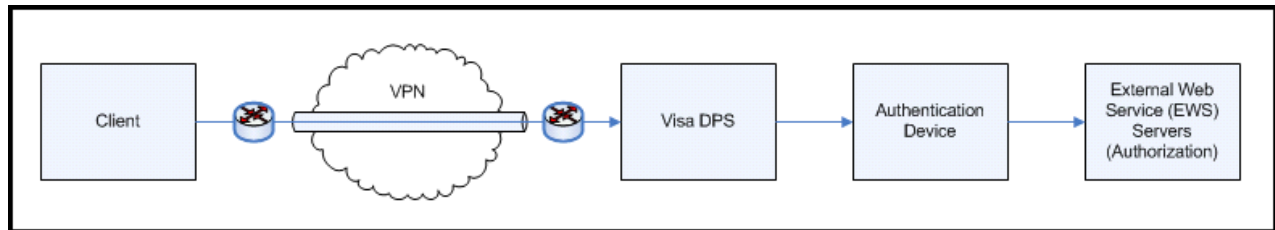
Figure 4:　　Direct Line (T1) Connection Diagram
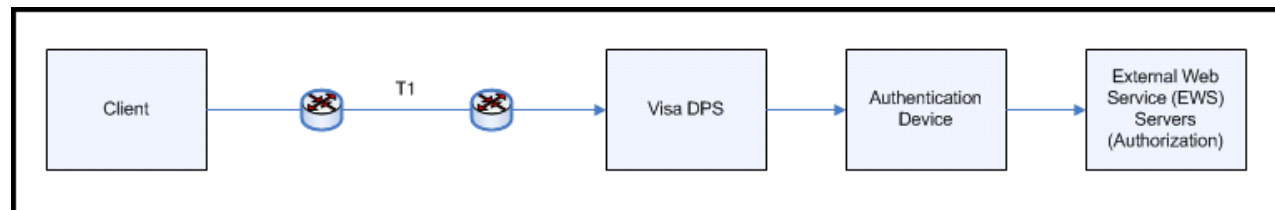
Advantages:

- Certificate management is not required
- Direct connection (VOL access not required)

Considerations:

- A separate connection is required to enable web services from multiple locations (including Disaster Recovery/Business Continuity sites)
- Support for dynamic password changes is provided by your Visa DPS Prepaid Account Manager
- Allow for 6 to 8 weeks lead time to procure VPN routers
- Allow for 8 to 10 weeks lead time to procure Direct Line (T1) routers
- Allow for 3 weeks to configure firewall changes

Security

Private line access, secure VPN, or Internet access must be established before connecting to Visa's Prepaid Web Services. Connections are routed through an XML Appliance (a web service gateway device) that validates, authenticates, and then passes the message on to the prepaid application server. The following sections deal with accessing the Prepaid Web Services via a private line or Secure VPN. For information regarding Internet access, please refer to the *WSI Service Developers Guide*.

Release Strategy

Updates to External Web Services are released twice per year, in conjunction with the two regular prepaid releases that occur every year in April and October.

If a release enhancement can be accessed through Web Services (not all enhancements in a release necessarily affect Web Services; for example, cosmetic improvements in the PAS interface), this will be noted in the Release Guide for the particular release. Web Services clients wishing to take advantage of these enhancements must upgrade to the new release. This represents one of the more compelling reasons to keep your Web Services application in sync with the current release.

Version Numbers. To identify specific versions of External Web Services, DPS assigns each release a version number:

Format:

yy_mm

where *yy* is the last two digits of the release year (e.g., 15 for 2015) and *mm* is the two-digit month of the release (e.g., 04 for April). Thus, the version number for the April 2015 release is written as:

14_05

Versions were previously assigned a single digit version number (e.g., Version 9), and you still may encounter references to these earlier versions in DPS documents.

Versions Supported

At any given point in time, two versions of External Web Services are always available—meaning that two versions of External Web Services are always available in the Certification (CERT) environment and in the Production (PROD) environment. Moreover, the versions available in CERT and PROD will differ at certain points in the following schedules due to staggered installation dates. When the October release is installed into CERT or PROD,

the previous April release continues to be available. And when the April release is installed into CERT or PROD, the previous October release continues to be available.

However, once a new version is installed in an environment (April or October), the availability of the release from the same month of the previous year ceases (i.e., the environment needed to support the release has been uninstalled). This process is illustrated below.
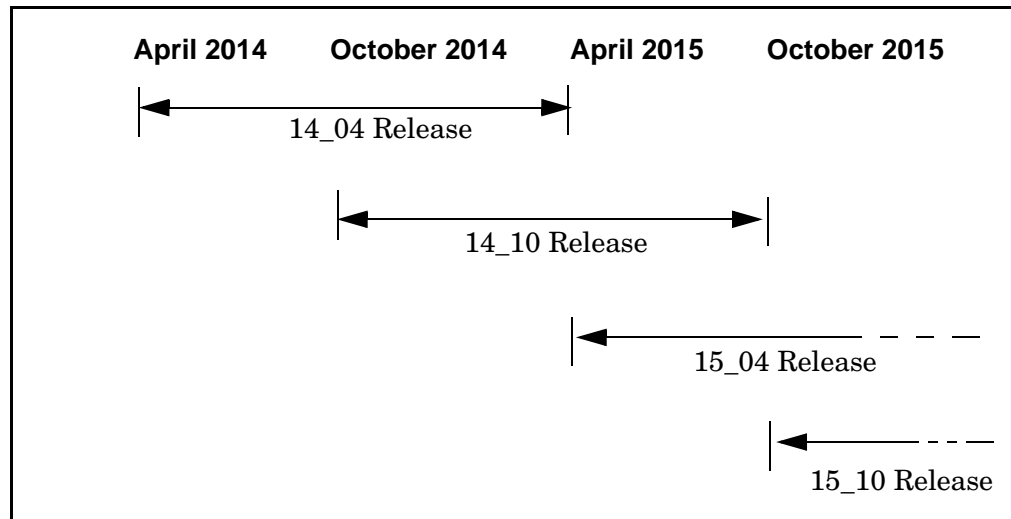
As shown above, the release in April remains available for a full year, as does the release in October. However, every time that a new release is installed, the availability of the same month release from the previous year terminates.

The longest that any client can use a given release of External Web Services is one year.

WSDL Release Dates

Client development activities cannot commence on a particular release until the underlying WSDL has been posted to Visa DPS Online. WSDL releases dates are noted in the timelines that follow, but to summarize:

Version 14_04: WSDL Release Date March 10, 2014

Version 14_10: WSDL Release Date September 8, 2014

Version 15_04: WSDL Release Date March 23, 2015 (estimated)

Version 15_10: WSDL Release Date September 28, 2014 (estimated)

In the unlikely event that a change becomes necessary to a given WSDL following the release date, clients using the WSDL will be directly informed of the update so that they can make appropriate adjustments in their timelines. Again, changes after the release date would be the exception, not the rule.

Upgrade Strategy Options

To better prepare clients for the eventual obsolescence of the version they are using in production, DPS has developed a series of timelines designed to help clients select the update schedule best suited to their own environments and schedules. Each timeline takes into account such factors as:

- Maintenance release dates

- WSDL publication dates

- External Web Services rollover dates

- Periods of potential instability in the certification testing environment as well as the production environment

- Visa control periods

The upgrade strategies diagrammed in the following pages are comprised of two complete sets of upgrade schedules, one for 2014 and the other for 2015. *However, it is important to note that variations of these strategies can also be adopted, depending on client needs and other considerations.* Before selecting a schedule, clients should consider:

- Environmental constraints

- Availability of in-house development and testing resources

- Internal client dates and/or milestones that could impact testing or implementation

The options are as follows:

- Update once per year in conjunction with April MR (see [Annual Update (during April 2015 MR) on page 29](#) and [Annual Update (during April 2014 MR) on page 24](#)). This model requires the client to code and test all updates incorporated into the platform throughout the previous year, including all updates in the April release.

- Update once per year in conjunction with October MR (see [Annual Update (during October 2015 MR) on page 30](#) and [Annual Update (during October 2014 MR) on page 25](#)). This model requires the client to code and test all updates incorporated into the platform throughout the previous year, including all updates in the October release.

- Update twice per year (see [Biannual Update (2015) on page 31](#) and [Biannual Update (2014) on page 26](#)) near the April and October MRs. This model requires the client to upgrade along with the relevant MR release.

- Update twice per year, near the April MR (see [Hybrid Update (during April 2015 MR) on page 32](#) and [Hybrid Update (during April 2014 MR) on page 27](#)). This model enables the client to consolidate all development and testing into two back-to-back phases, one that addresses the previous

release and one that address the current release. If the enhancements from the previous release would have had little or no impact on the client's application, this option may represent an advantage over the twice per year option.

- Update twice per year, near the October MR (see [Hybrid Update (during October 2015 MR) on page 33](#) and [Hybrid Update (during October 2014 MR) on page 28](#)). This model enables the client to consolidate all development and testing into two back-to-back phases, one that addresses the previous release and one that address the current release. If the enhancements from the previous release would have had little or no impact on the client's application, this option may represent an advantage over the twice per year option.

Annual Update (during April 2014 MR)

The following update schedule incorporates one External Web Services update per year, scheduled in conjunction with the April 2014 MR.

Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development and testing. Hard lines (e.g., the Release timeline) represent mandatory end dates—*clients must move the release into production on the date indicated.*

Advantages: Only one update required per year – uses fewer testing and development resources

Disadvantages: Delays adoption of previous release enhancements; shortened coding and testing windows; *mandatory release date* of 4/19/14; no fallback; post-release date issues must be fixed in production

Figure 5:   Annual Update (during April 2014 MR)

Visa DPS Prepaid External Web Services (EWS)
2014 Upgrade Strategy Options (Annual MR Apr)

D-0273-07

Annual Update (during October 2014 MR)

The following update schedule incorporates one External Web Services update per year, scheduled in conjunction with the October 2014 MR.
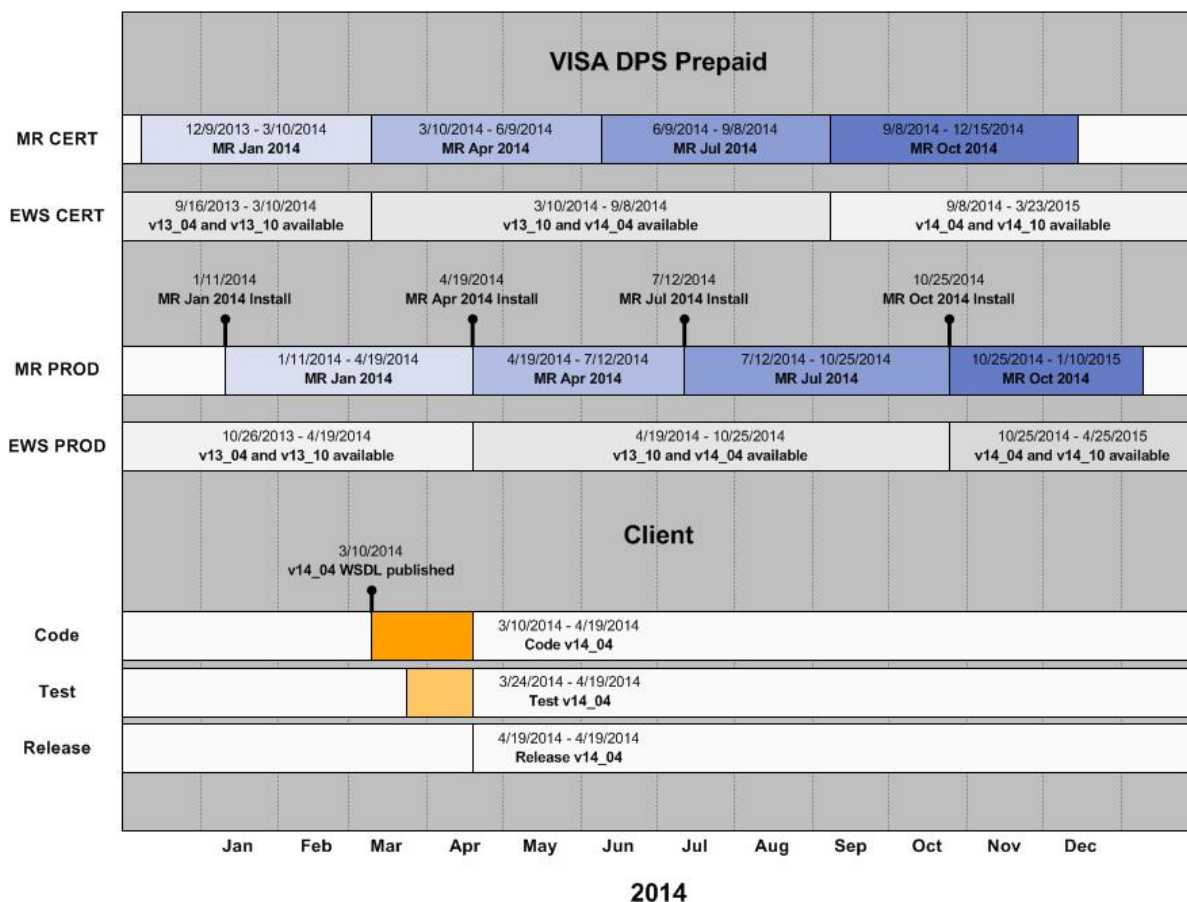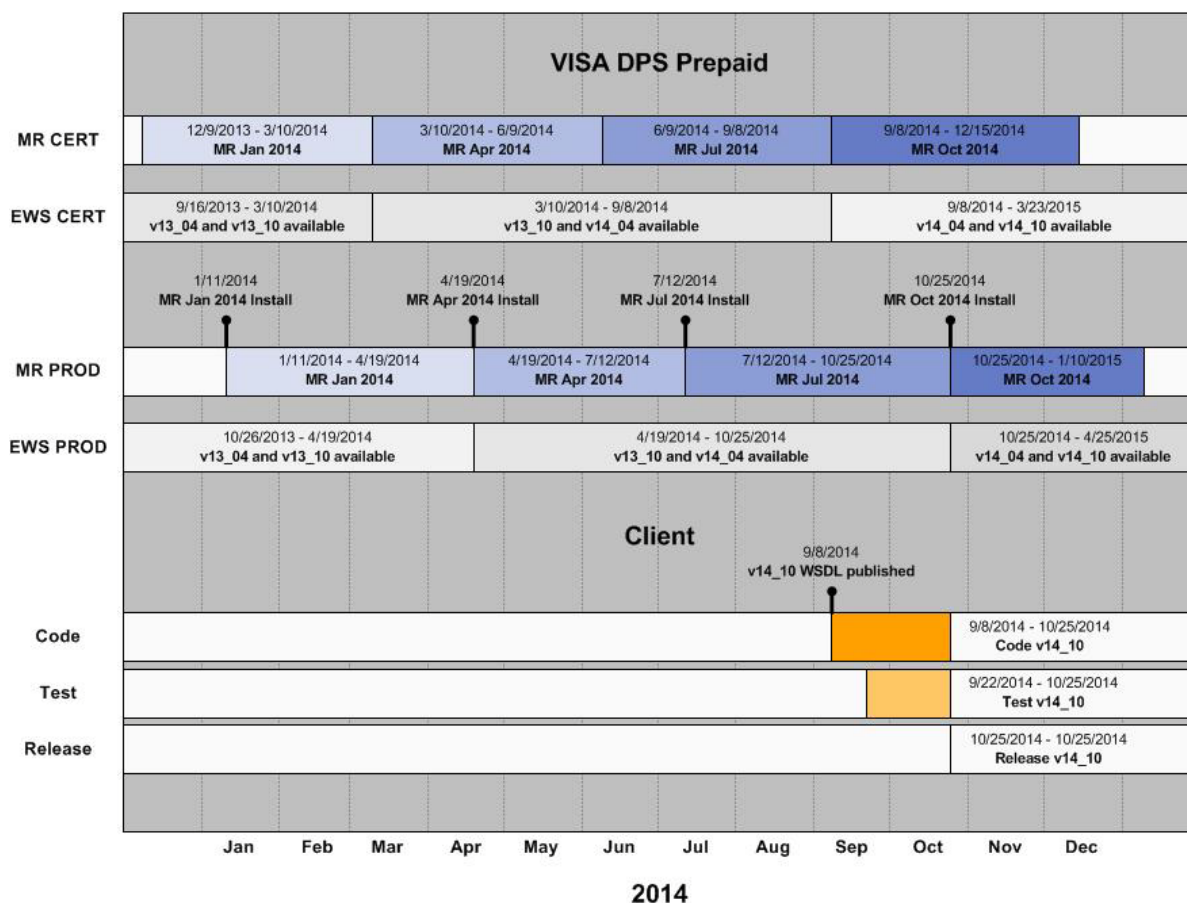
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development and testing. Hard lines (e.g., the Release timeline) represent mandatory end dates—*clients must move the release into production on the date indicated*.

Advantages: Only one update required per year – uses fewer testing and development resources

Disadvantages: Delays adoption of previous release enhancements; shortened coding and testing windows; *mandatory release date* of 10/25/14; no fallback; post-release date issues must be fixed in production

Figure 6:   Annual Update (during October 2014 MR)

Visa DPS Prepaid External Web Services (EWS) 2014 Upgrade Strategy Options (Annual MR Oct)
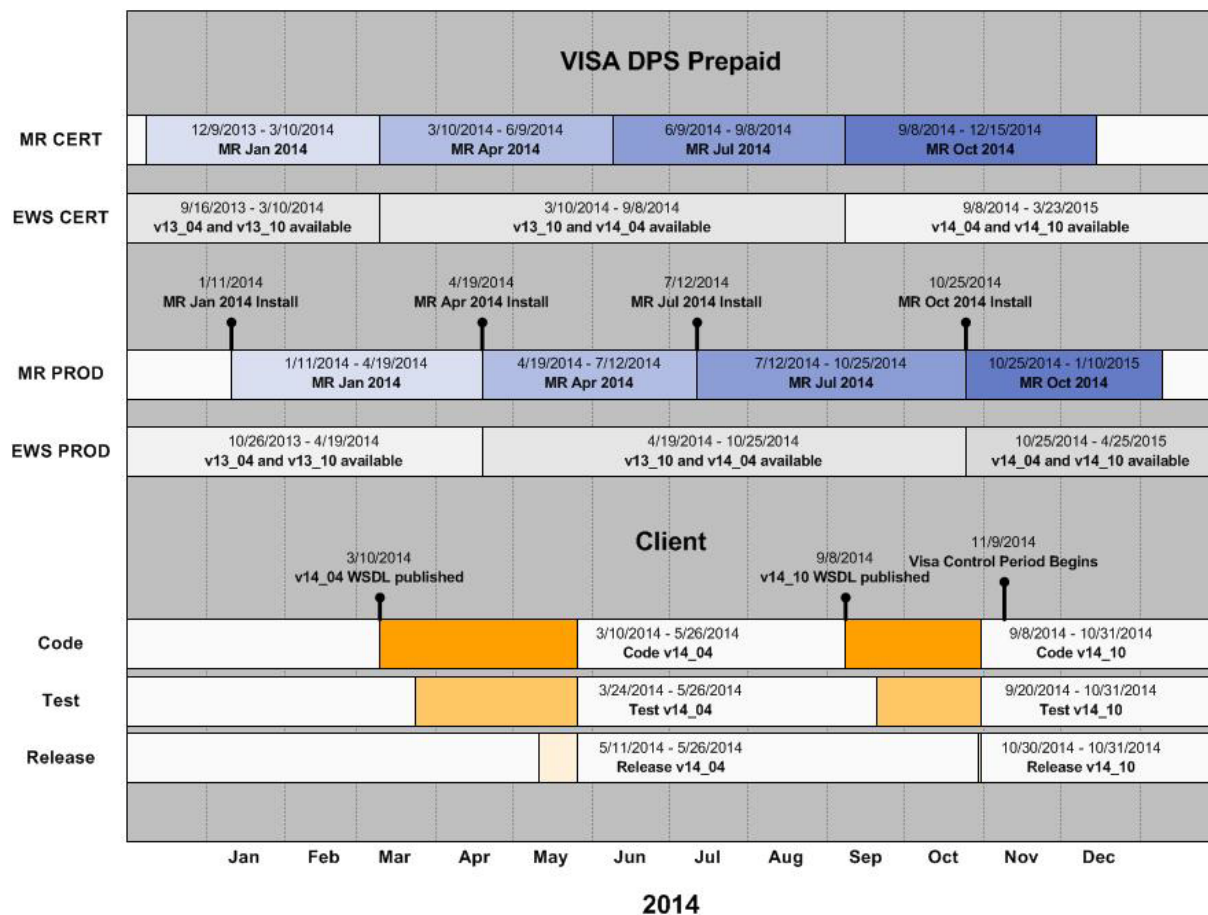
Biannual Update (2014)

The following update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the April 2014 MR and the October 2014 MR.

Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Version of EWS always current with production release; longer coding and testing windows than annual timelines; greater flexibility in release dates; lower risk – previous version still supported

Disadvantages: More development and testing resources required

Figure 7:   Biannual Update (2014)

Hybrid Update (during April 2014 MR)

The following hybrid update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the April 2014 MR.

Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Consolidates development and testing efforts into two back-to-back phases; uses fewer testing and development resources; greater release date flexibility; lower overall risk

Disadvantages: Slightly increased development/testing effort compared to annual; CERT and PROD are on different MR releases

Figure 8: **Hybrid Update (during April 2014 MR)**

Hybrid Update (during October 2014 MR)

The following hybrid update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the October 2014 MR.
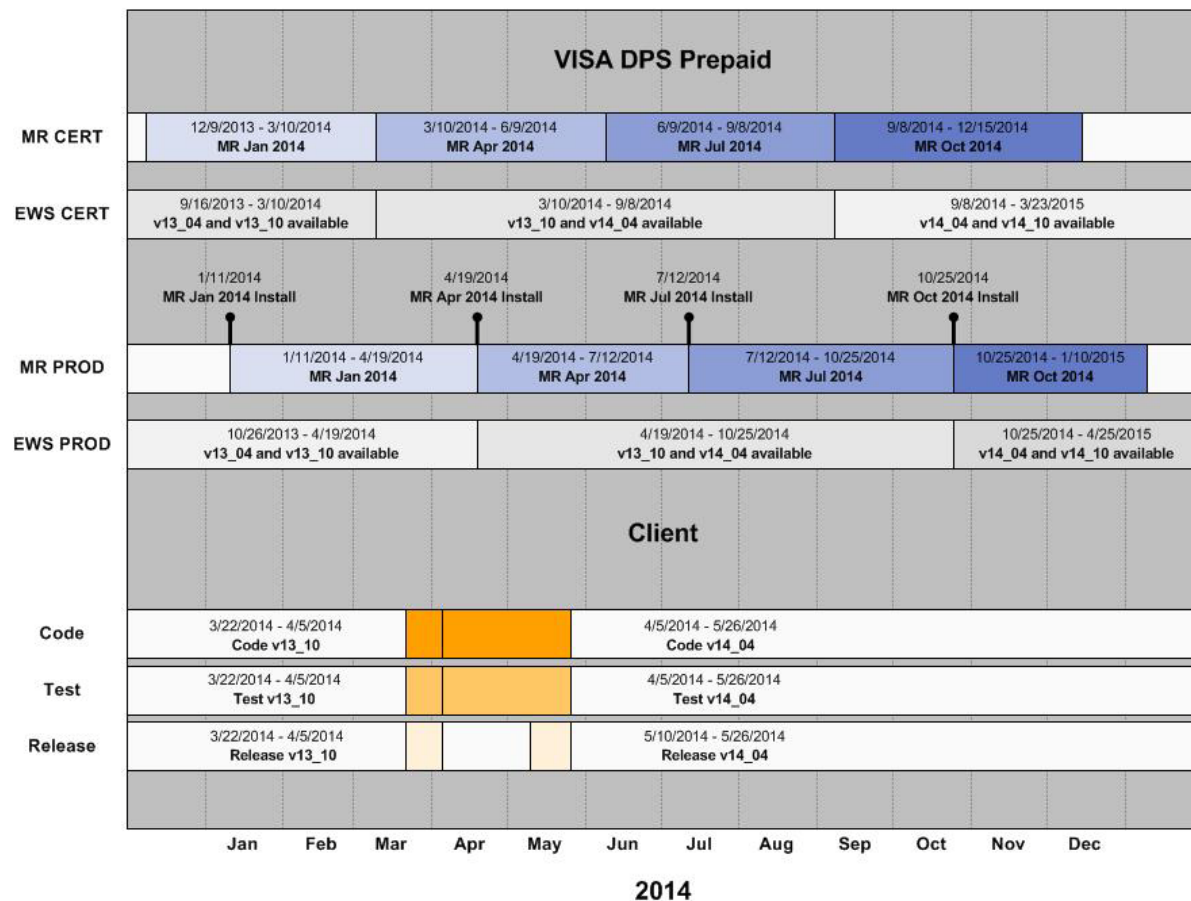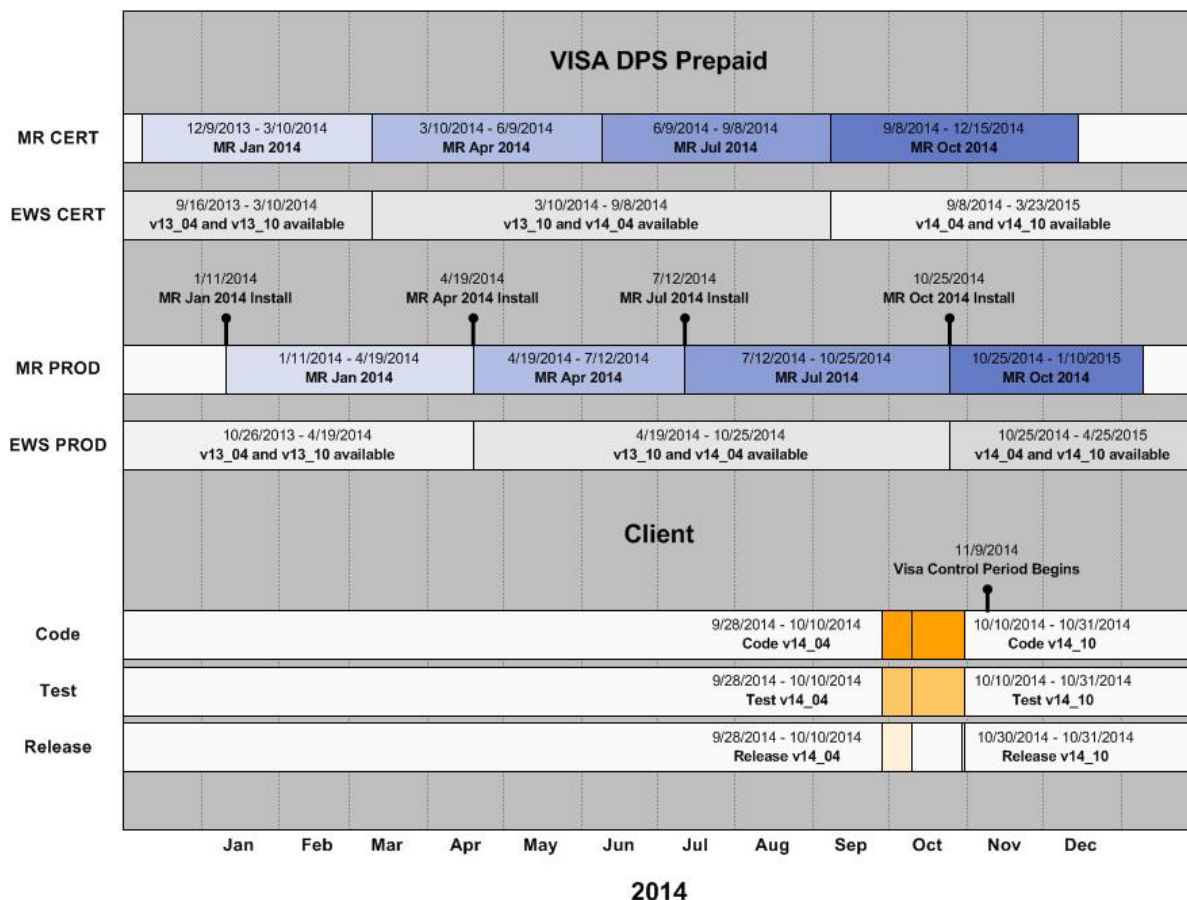
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Consolidates development and testing efforts into two back-to-back phases; uses fewer testing and development resources; greater release date flexibility; lower overall risk

Disadvantages: Slightly increased development/testing effort compared to annual; shortened coding, testing, and release windows due to Visa control period; CERT and PROD are on different MR releases

Figure 9:   **Hybrid Update (during October 2014 MR)**

Visa DPS Prepaid External Web Services (EWS)
2014 Upgrade Strategy Options (Annual Hybrid MR Oct)

Annual Update (during April 2015 MR)

The following update schedule incorporates one External Web Services update per year, scheduled in conjunction with the April 2015 MR.

Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development and testing. Hard lines (e.g., the Release timeline) represent mandatory end dates—*clients must move the release into production on the date indicated.*

Advantages: Only one update required per year – uses fewer testing and development resources

Disadvantages: Delays adoption of previous release enhancements; shortened coding and testing windows; *mandatory release date* of 4/25/15; no fallback; post-release date issues must be fixed in production

Figure 10:   **Annual Update (during April 2015 MR)**

Annual Update (during October 2015 MR)

The following update schedule incorporates one External Web Services update per year, scheduled in conjunction with the October 2015 MR.
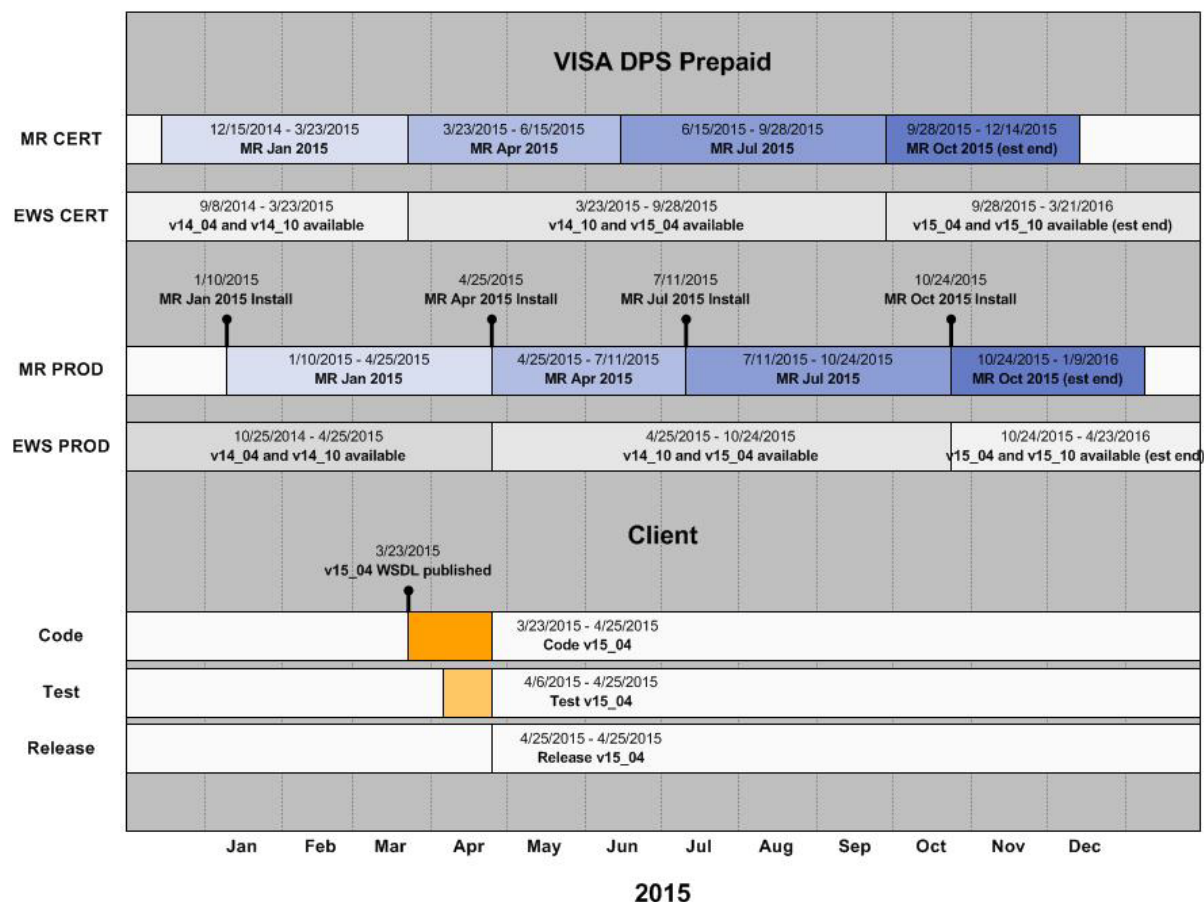
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development and testing. Hard lines (e.g., the Release timeline) represent mandatory end dates—*clients must move the release into production on the date indicated*.

Advantages: Only one update required per year – uses fewer testing and development resources

Disadvantages: Delays adoption of previous release enhancements; shortened coding and testing windows; *mandatory release date* of 10/24/15; no fallback; post-release date issues must be fixed in production

Figure 11:   Annual Update (during October 2015 MR)

Visa DPS Prepaid External Web Services (EWS) 2015 Upgrade Strategy Options (Annual MR Oct)

D-0273-07

Biannual Update (2015)

The following update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the April 2015 MR and the October 2015 MR.
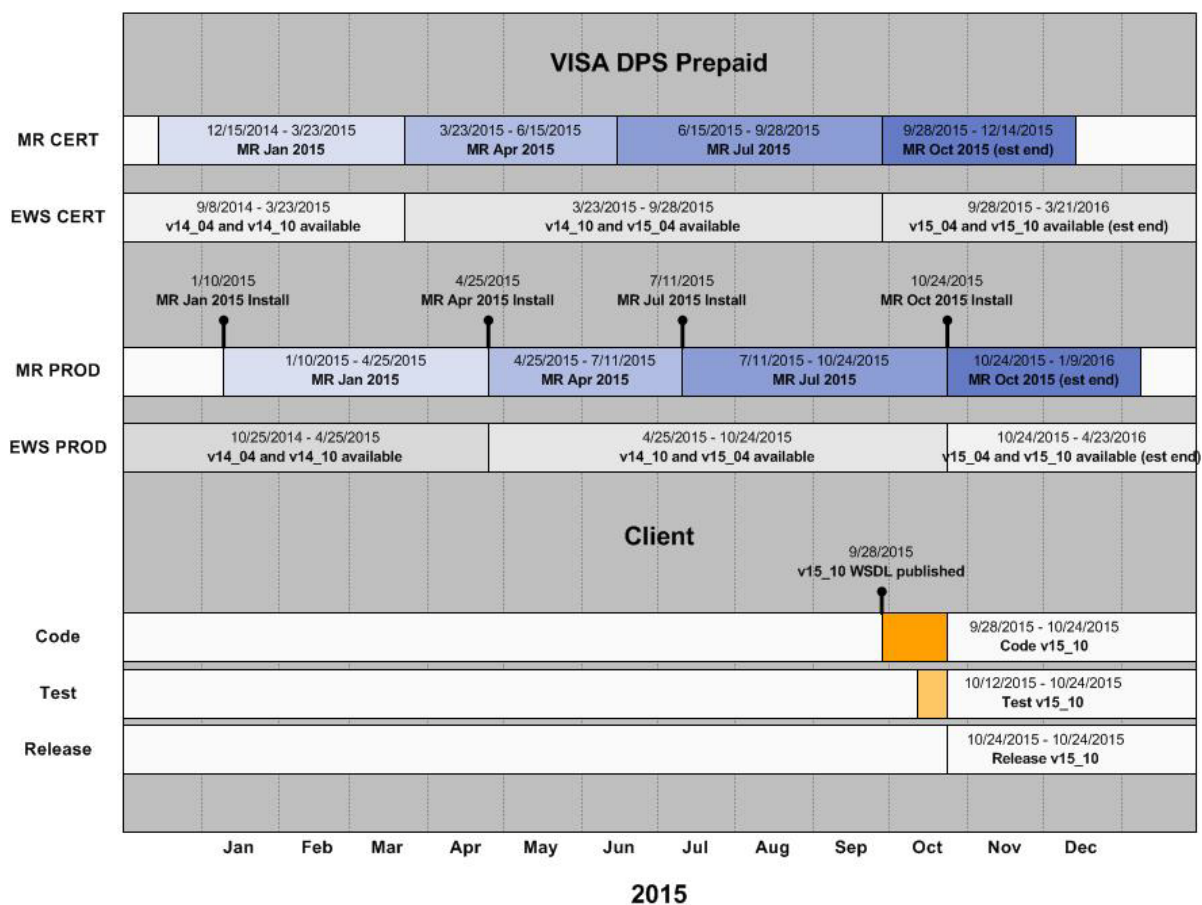
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Version of EWS always current with production release; longer coding and testing windows than annual timelines; greater flexibility in release dates; lower risk – previous version still supported

Disadvantages: More development and testing resources required

Figure 12:   **Biannual Update (2015)**

Visa DPS Prepaid External Web Services (EWS)
2015 Upgrade Strategy Options (Biannual)

Hybrid Update (during April 2015 MR)

The following hybrid update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the April 2015 MR.
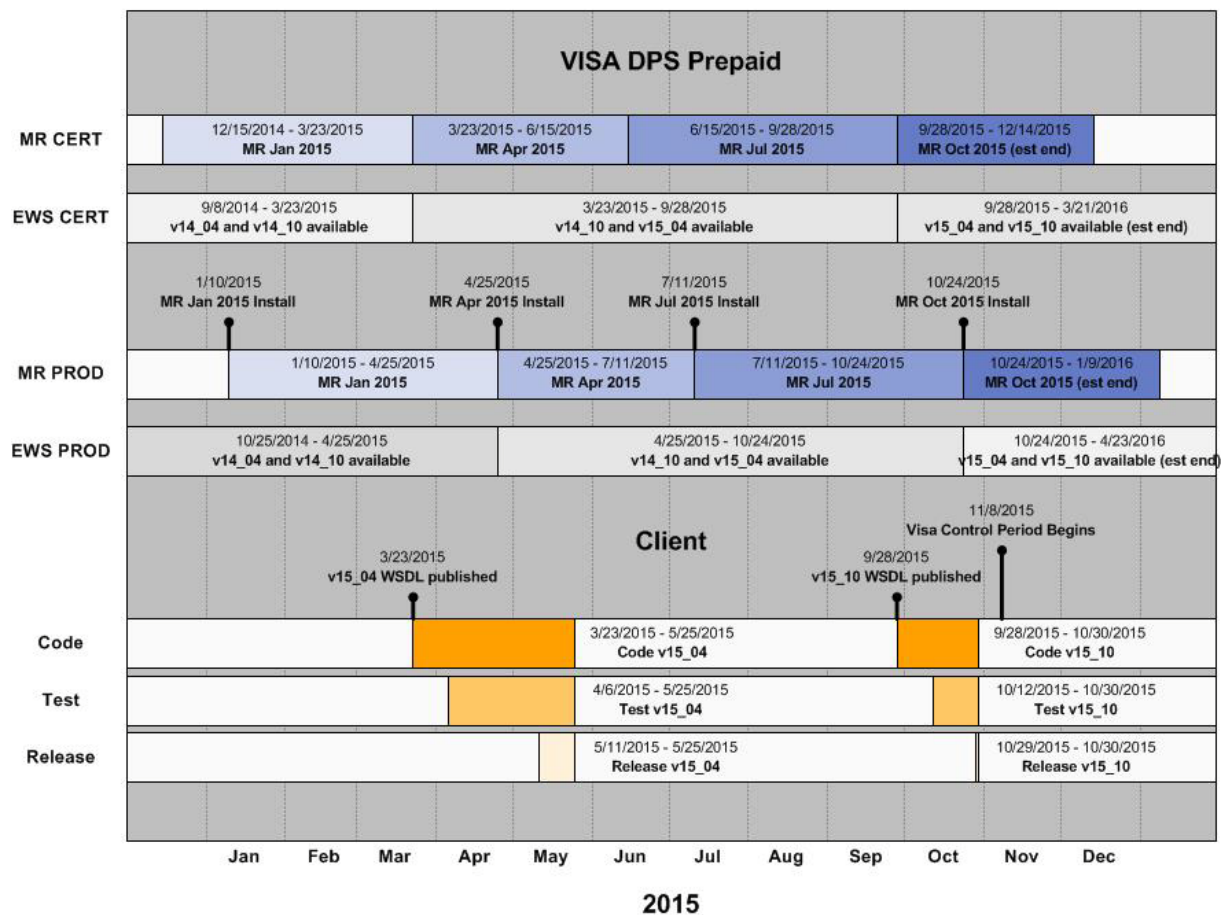
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Consolidates development and testing efforts into two back-to-back phases; uses fewer testing and development resources; greater release date flexibility; lower overall risk

Disadvantages: Slightly increased development/testing effort compared to annual; CERT and PROD are on different MR releases

Figure 13:   Hybrid Update (during April 2015 MR)

Hybrid Update (during October 2015 MR)

The following hybrid update schedule incorporates two External Web Services updates per year, scheduled in conjunction with the October 2015 MR.
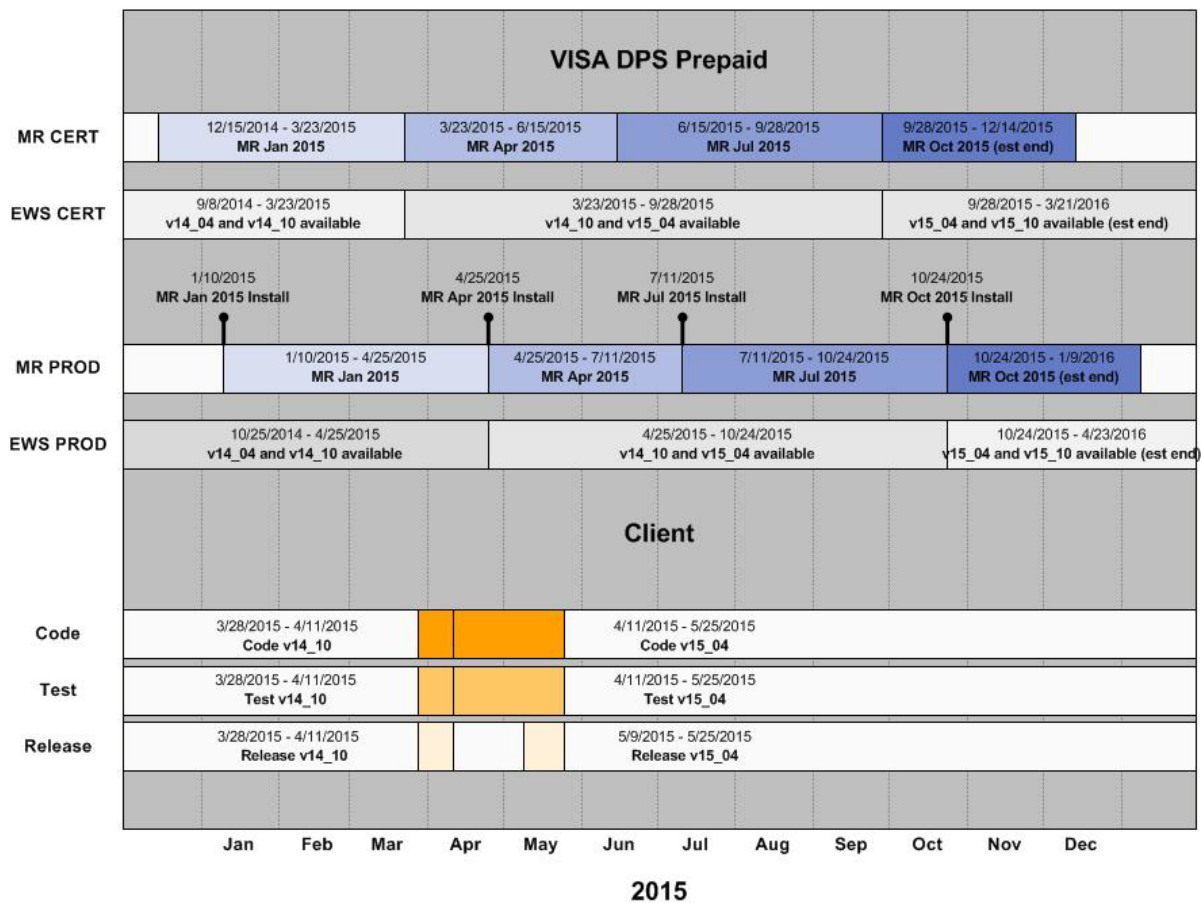
Refer to the **Client** section of the timeline, below, to determine specific coding, testing, and release dates. Shaded boxes represent recommended windows for development, testing, and production release. Ideally, clients would release into production on any date that falls within the Release window.

Advantages: Consolidates development and testing efforts into two back-to-back phases; uses fewer testing and development resources; greater release date flexibility; lower overall risk

Disadvantages: Slightly increased development/testing effort compared to annual; shortened coding, testing, and release windows due to Visa control period; CERT and PROD are on different MR releases

Figure 14:   Hybrid Update (during October 2015 MR)

NOTICE: This information is Confidential and must be used exclusively by the Visa customer in connection with Visa programs. The information contained in this media may be copied and distributed for internal use by the Visa customer in connection with Visa programs. It may not otherwise be duplicated, published, or disclosed without written permission from Visa.
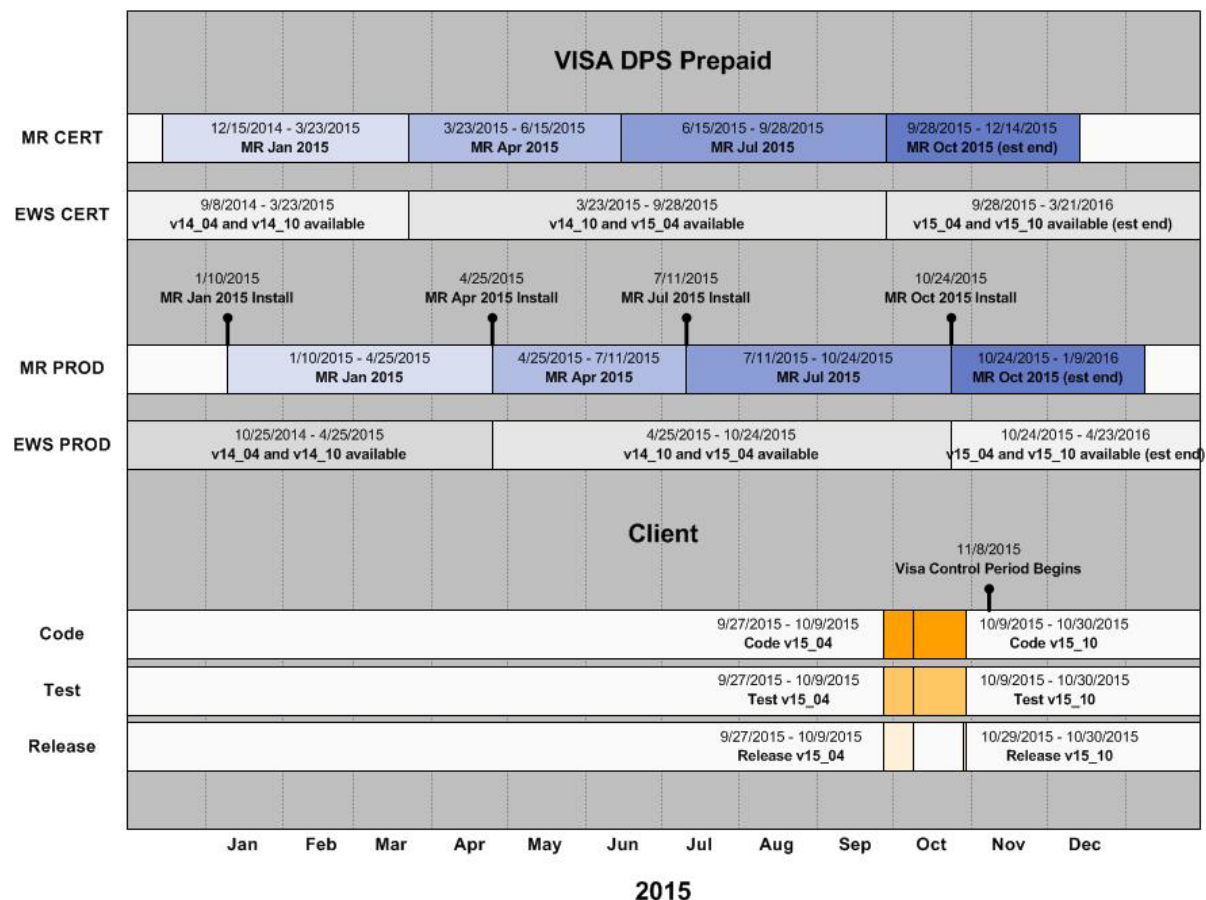
Factors To Be Considered When Developing a Custom Schedule

As previously stated, the above models are only suggestions, and do not represent all of the possible development and test timelines a client could implement. However, before creating a different schedule, or deciding to work outside the windows in the models provided, consider the following:

- Has the version you're upgrading to been installed into CERT? You will not be able to test your changes without the right version in CERT.

- Is the testing environment stable; that is, was the version in CERT installed recently, or has it been in the environment for some period of time?

- Should you encounter problems after you move your code into PROD, is there an environment available that still supports your previous release?

Reviewing these variables with DPS before making your decision will help prevent problems down the road.

External Web Services Reports

As an aid in assessing the impact of new External Web Services releases and in obtaining information on what new messages are available, the prepaid platform provides two reports designed to assist with the assessment process:

- EWS WSDL Version Comparison Report – This report returns a list of differences between a base WSDL version selected by the client (generally the version that the client is currently using) and a target version also selected by the client (the version that the client is considering upgrading to). Clients can also choose to include all EWS methods that have changed, or only those methods actually used by the client (based on the client's EWS call history in the given environment).

  Note that to run a comparison between a version that the client is currently using and a target release version, both versions *must be resident (be installed) in the environment*.

- Web Service Call Report – This report lists all Web Service calls made by the client or sub-client to the PPC in the environment.

Refer to the *Visa DPS Prepaid Reports Reference Guide* for additional information and samples of these reports.

Glossary

The following list defines a number of terms commonly used in the context of explaining web services and also includes several Prepaid concepts.

Table 2:   Glossary

Term	Description
Alias ID	A unique prepaid entity key.
Buyer	The purchaser of a prepaid card. May be an individual buyer or a company.
Card Class	A prepaid card class allows for a grouping of a set of properties for a card. Typical card classes include Personalized, Non-Personalized, and ATM-only cards.
Card Program	Card programs are groups of configuration details that form a logical grouping. A card program often translates conceptually into a marketed product by a banking institution.
Cardholder	The owner of a prepaid card.
Client	The issuer of a prepaid card.
Entity	Any combination of related fields or rows in the database.
Foreign	A foreign transaction or fee is one that involves an account not owned by the initiating financial institution.
Limit	Limits are specific values associated with particular Prepaid activities (for example, the maximum number of cards allowed in a single purchase).
Message	Information sent between a source and a receiver in the form of an XML file, formatted in accordance with SOAP specifications.
Namespace	A string of characters that uniquely identifies a particular service within a web services schema. Although the namespace string may look like a web address, its purpose is different.
Nonce	A value used only once by a computer security system.
On-Us	An on-us transaction or fee is one that involves only accounts owned by the initiating financial institution.
Operation	In Prepaid web services, operations are divided into Get and Update operations. Get operations are used to request information from the Prepaid database; Update operations update or change information stored in the database.
PAN	The credit card number embossed on the card. PAN is an acronym for Primary Account Number.

Table 2:   Glossary

Term	Description
PAS	Program Administration System. PAS is the name of the Visa interface used by prepaid clients that do not use web services to access the prepaid system.
PAT	Prepaid Administration Tool. PAT is the internet version of PAS.
Provision	The step or steps used to enable a function in the Prepaid processing environment.
Relative Service URL	The URL string of a specific web service, relative to a machine or environment.
Request	A Get message sent by service consumer to a web service; for example, `GetFeeDetailsRequest`.
Response	A message returned by a web service via the service provider in response to a client Get or Update request.
Schema	A document that defines the structure, content, and required syntax of an XML document or request.
SOAP	SOAP is a protocol used for exchanging XML-based messages over a computer network, normally using HTTP.
SSL	SSL, or Secure Sockets Layer, is a secure protocol used for transmitting documents via the Internet.
Stored Value Account (SVA)	Unlike other prepaid card programs, the stored value account concept allows Visa to retain cardholder and buyer information as a separate entity from the card number itself. Consequently, if a card is lost or stolen, or if the primary cardholder desires to add other secondary cardholders, replacement or additional cards may be issued without having to rebuild the buyer/cardholder record.
Sub-client	A member or subordinate of a client's responsibility structure. A sub client is often a banking institution's branch location or it may be a company with a special trusted relationship with a banking institution.
Threshold	Thresholds alert the Issuer and Visa when a program parameter has been reached or exceeded. Limits often map to a corresponding threshold that further refines the system's response to potentially fraudulent conditions. The intent of the threshold is to find transactions that are below a limit but are still suspect.
Type	The name of an element contained in the schema that defines the content of a message.
WSDL	Web Services Description Language. WSDL is an XML-based language that provides a model for describing Web services.